

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**PEDRO HENRIQUE COL CARVALHO**

**CONSTRUÇÃO, IDENTIFICAÇÃO, CONTROLE E ANÁLISE DE SISTEMAS  
DE ESTABILIDADE E POSICIONAMENTO DE UM QUADRIROTOR COM  
UM GRAU DE LIBERDADE: ESTUDO DE CASO**

**CAMPO MOURÃO**

**2024**

**PEDRO HENRIQUE COL CARVALHO**

**CONSTRUÇÃO, IDENTIFICAÇÃO, CONTROLE E ANÁLISE DE SISTEMAS DE ESTABILIDADE E POSICIONAMENTO DE UM QUADRIROTOR COM UM GRAU DE LIBERDADE: ESTUDO DE CASO**

*Construction, identification, control and analysis of stability and positioning systems of a quadrotor with a degree of freedom: case study*

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Leandro Castilho Brolin.

Coorientador(a): Flávio Luiz Rossini.

**CAMPO MOURÃO**

**2024**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**PEDRO HENRIQUE COL CARVALHO**

**CONSTRUÇÃO, IDENTIFICAÇÃO, CONTROLE E ANÁLISE DE SISTEMAS DE  
ESTABILIDADE E POSICIONAMENTO DE UM QUADRIROTOR COM UM GRAU  
DE LIBERDADE: ESTUDO DE CASO**

Trabalho de Conclusão de Curso de Graduação apresentado  
como requisito para obtenção do título de Bacharel em  
Engenharia Eletrônica da Universidade Tecnológica Federal  
do Paraná (UTFPR).

Data de aprovação: 12/fevereiro/2025

---

Leandro Castilho Brolin  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Flávio Luiz Rossini  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Marcelo Nanni  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Eduardo Giometti Bertogna  
Doutor  
Universidade Tecnológica Federal do Paraná

**CAMPO MOURÃO**

**2024**

## **AGRADECIMENTOS**

Agradeço a minha família pelo apoio durante a graduação.

Ao meu orientador Prof. Dr. Leandro Castilho Brolin e ao meu coorientador Prof. Dr. Flávio Luiz Rossini pela instrução e conhecimento transmitido.

E a todos os professores que contribuíram direta ou indiretamente para a construção desse trabalho.

## RESUMO

A busca por maior eficiência e competitividade impulsiona o desenvolvimento de novas tecnologias em diversas áreas. Nesse contexto, Veículo Aéreo Não Tripulado (VANT) tem sido amplamente utilizado. Este trabalho aborda a construção, identificação do sistema e controle de um VANT com um grau de liberdade, especificamente um quadricóptero. Inicialmente, construiu-se a estrutura do protótipo, composta por uma barra horizontal com um eixo móvel em seu centro de massa e motores Brushless nas extremidades. Realizou-se o controle da planta por um sistema embarcado. Projetou-se o controlador como um servossistema do tipo 1 quando a planta possui integrador. A identificação dos parâmetros do sistema segue a abordagem de caixa cinza, a qual aplicou-se para plantas com parâmetros desconhecidos. Para determinar esses parâmetros, utilizou-se o Método dos Mínimos Quadrados (MMQ), que possibilitou a descrição do comportamento dinâmico do protótipo. Com o modelo dinâmico identificado, o trabalho propõe-se a implementação de um controle moderno, especificamente em espaço de estados. Após a validação do modelo, discretizou-se e implementou-se o modelo no protótipo construído. Os resultados mostraram que o modelo matemático utilizado apresentou uma resposta apropriada às perturbações externas, portanto, assegurando a estabilidade e o retorno rápido a condição de operação. Por fim, concluiu-se que o sistema de controle em espaço de estados é um método eficiente para modelos instáveis.

Palavras-chave: quadricóptero; identificação de sistemas; servossistema do tipo 1; controle em espaço de estados.

## ABSTRACT

The pursuit of greater efficiency and competitiveness drives the development of new technologies across various fields. In this context, Unmanned Aerial Vehicles (UAVs) have been widely utilized. This work addresses the construction, system identification, and control of a UAV with one degree of freedom, specifically a quadrotor. Initially, the prototype structure was developed, consisting of a horizontal bar with a movable axis at its center of mass and Brushless motors at the ends. The plant was controlled by an embedded system. The controller was designed as a Type 1 servosystem when the plant includes an integrator. The system parameters were identified using the gray-box approach, which is applied to plants with unknown parameters. To determine these parameters, the Least Squares Method (LSM) was employed, enabling the description of the prototype's dynamic behavior. With the identified dynamic model, this study proposes the implementation of a modern control strategy, specifically in state-space representation. After model validation, the system was discretized and implemented in the constructed prototype. The results demonstrated that the mathematical model provided an appropriate response to external disturbances, thereby ensuring stability and a rapid return to the operating condition. Finally, it is concluded that the state-space control system is an effective method for unstable models.

Keywords: quadrotor; system identification; type 1 servosystem; state-space control.

## LISTA DE ILUSTRAÇÕES

|  |           |
|--|-----------|
| <b>Figura 1: Sistema de coordenadas de um quadricóptero .....</b>    | <b>14</b> |
| <b>Figura 2: Descrição de rotação e movimento .....</b>              | <b>15</b> |
| <b>Figura 3: Referencial inercial.....</b>                           | <b>17</b> |
| <b>Figura 4: Servossistema do tipo 1 em malha fechada .....</b>      | <b>27</b> |
| <b>Figura 11: Aferição do empuxo gerado pelos motores.....</b>       | <b>35</b> |
| <b>Figura 12: Curva dos Motores .....</b>                            | <b>38</b> |
| <b>Figura 13: Linha de ajuste dos Motores.....</b>                   | <b>39</b> |
| <b>Figura 14: Resposta em Malha Aberta.....</b>                      | <b>40</b> |
| <b>Figura 15: Resposta em Malha Fechada .....</b>                    | <b>43</b> |
| <b>Figura 16: Resposta em Malha Fechada do Sistema Discreto.....</b> | <b>44</b> |
| <b>Figura 17: Resposta do Protótipo a Distúrbios Manuais.....</b>    | <b>46</b> |
| <b>Figura 18: Sinal <math>r</math> .....</b>                         | <b>46</b> |
| <b>Figura 19: Resposta ao Sinal <math>r</math> .....</b>             | <b>47</b> |
| <b>Figura 20: Resposta Experimental do Protótipo ao Degrau .....</b> | <b>47</b> |

## LISTA DE QUADROS

|  |           |
|--|-----------|
| <b>Quadro 1: Principais informações do ATmega328p .....</b>                | <b>29</b> |
| <b>Quadro 2: Principais informações do MUP6050 .....</b>                   | <b>30</b> |
| <b>Quadro 3: Conexão da Alimentação do motor Brushless.....</b>            | <b>32</b> |
| <b>Quadro 4: Conexão do Sinal de controle do motor Brushless.....</b>      | <b>32</b> |
| <b>Quadro 5: Relação do empuxo com sinal de controle do Motor 4 .....</b>  | <b>37</b> |
| <b>Quadro 6: Relação do empuxo com sinal de controle do Motor 2 .....</b>  | <b>38</b> |
| <b>Quadro 7: Comparação das Características da Resposta ao Degrau.....</b> | <b>49</b> |

## SUMÁRIO

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUÇÃO</b>  | <b>10</b> |
| 1.1      | Contextualização do tema                                     | 10        |
| 1.2      | Revisão bibliográfica  | 11        |
| 1.3      | Definição do problema  | 12        |
| 1.4      | Objetivos  | 12        |
| 1.4.1    | Objetivo geral   | 12        |
| 1.4.2    | Objetivos específicos  | 12        |
| 1.5      | Justificativa  | 13        |
| <b>2</b> | <b>FUNDAMENTAÇÃO TEÓRICA</b>                                 | <b>14</b> |
| 2.1      | Quadratores  | 14        |
| 2.1.1    | Graus de liberdade   | 14        |
| 2.1.2    | Dinâmica do quadricotol                                      | 15        |
| 2.2      | Modelo Dinâmico do Quadratores                               | 16        |
| 2.2.1    | Sistema de coordenadas do Quadratores                        | 16        |
| 2.2.2    | Cinemática do Quadratores                                    | 17        |
| 2.2.3    | Dinâmica de Corpo Rígido                                     | 18        |
| 2.2.4    | Análise de Forças, Torques e Momentos                        | 21        |
| 2.2.5    | Modelo   | 23        |
| 2.3      | Método dos Mínimos Quadrados                                 | 24        |
| 2.4      | Controle em Espaço de Estados                                | 25        |
| 2.4.1    | Alocação de polos  | 26        |
| 2.4.2    | Espaço de estados discreto                                   | 27        |
| <b>3</b> | <b>METODOLOGIA</b>   | <b>28</b> |
| 3.1      | Estrutura do Protótipo                                       | 28        |
| 3.1.1    | Lista de materiais   | 29        |
| 3.1.2    | Arquitetura do protótipo                                     | 31        |
| 3.2      | Desenvolvimento Firmware para o Controle do Microcontrolador | 33        |
| 3.3      | Identificação do Sistema                                     | 34        |
| 3.3.1    | Torque   | 34        |
| 3.3.2    | Modelo em Espaço de Estados                                  | 35        |
| <b>4</b> | <b>RESULTADOS</b>  | <b>37</b> |
| 4.1      | Identificação do Sistema                                     | 37        |
| 4.1.1    | Momento de Inércia   | 39        |
| 4.2      | Resposta em Malha Aberta                                     | 40        |
| 4.3      | Escolha dos Polos Desejados                                  | 41        |
| 4.4      | Alocação de Polos  | 41        |
| 4.5      | Resposta em Malha Fechada                                    | 42        |
| 4.6      | Espaço de Estados Discreto                                   | 43        |
| 4.7      | Lei de Controle  | 44        |

|            |   |           |
|------------|---|-----------|
| <b>4.8</b> | <b>Comparação entre o Modelo Teórico e os Dados Experimentais .....</b> | <b>45</b> |
| <b>4.9</b> | <b>Análise Experimental do Protótipo ao Degrau .....</b>                | <b>48</b> |
| <b>5</b>   | <b>CONCLUSÃO.....</b>   | <b>50</b> |
|            | <b>REFERÊNCIAS .....</b>  | <b>51</b> |
|            | <b>APÊNDICE A - Script do MATLAB.....</b>                               | <b>53</b> |
|            | <b>APÊNDICE B - Código do Arduino .....</b>                             | <b>62</b> |

## 1 INTRODUÇÃO

Neste capítulo são abordadas a introdução, os objetivos e a justificativa do trabalho.

### 1.1 Contextualização do tema

Desde o século XVIII a humanidade desenvolve de modo relevante técnicas de controle. Entre os avanços significativos, é importante ressaltar os estudos desenvolvidos por Minorsky e Nyquist na análise da estabilidade dos sistemas. Assim como, as regras criadas por Ziegler e Nichols para a sintonia dos controladores Proporcional Integral Derivativo (PID) e o método do lugar das raízes de Evans. Até o final da década de 1950 foram desenvolvidos métodos de controle no domínio da frequência, hoje conhecido como controle clássico, mas com o desenvolvimento dos computadores foi possível analisar sistemas diretamente no domínio do tempo, o que resultou no surgimento de técnicas de controle moderno (OGATA, 2010).

Após a invenção do estabilizador giroscópio criado por Elmer Sperry em 1991, que permite controlar a direção de navios e manter em linha reta e nivelada de forma autônoma, a Marinha dos Estados Unidos usou este mecanismo para transformar a aeronave de treinamento Curtiss N-9 no primeiro Veículo Aéreo não Tripulado (VANT) controlado de forma remota (BOUABDALLAH, 2007).

No roteiro *Unmanned Aircraft Systems Roadmap, 2005-2030* criado pelo Departamento de Defesa dos Estados Unidos, os VANTs são definidos como veículos aéreos que fazem o uso de forças aerodinâmicas para fornecer sustentação. Os VANTs podem atuar de forma autônoma, ou serem pilotados de maneira remota (*Office of the Secretary of Defence, 2008*).

O tema abordado neste trabalho está relacionado aos quadrirotores, que consistem em uma classe específica dos VANTs. Segundo o conceito de Samir Bouabdallah (2007), quadrirotores são VANTs que possuem quatro hélices em uma configuração cruzada, sendo que os dois pares de hélices giram em direções opostas para evitar o momento angular, diferente dos helicópteros que necessitam de um motor na cauda para compensar tal efeito. Ao alterar a velocidade angular dos pares dos rotores é possível mudar a sua posição. Apesar dos quatro atuadores, o quadrirotor continua a ser um sistema dinamicamente instável (BOUABDALLAH, 2007).

A utilização de quadrirotores ganha cada vez mais espaço em diversas áreas, filmagens, fotografias, engenharia civil, inspeção, monitoramento, agricultura e outros. As principais utilizações na agricultura são: análise e demarcação da plantação, acompanhamento do desenvolvimento da safra e pastagem, pulverização, identificação de danos e acompanhamento e monitoramento de animais (FUNDAP, 2019).

Pesquisas na área da agricultura demonstram que a utilização do quadrirotor pode aumentar em 15% a 20% a produtividade e reduzir consideravelmente, a quantidade de insumos (GONÇALVES, 2023).

Devido ao crescimento da utilização de VANTs, a produção científica relacionada aos VANTs no meio militar aumentou significativamente. Entre os anos de 2000 e 2023, foram publicadas 860 pesquisas, sendo que, nos últimos quatro anos, foram 442, o que corresponde a 51,39% do total. (MELLO *et al.*, 2024).

## 1.2 Revisão bibliográfica

A evolução de sistemas de controle aplicadas a VANTs tem sido marcada por aprimoramentos na forma de abordar a estabilidade e desempenho do sistema. Dentre os trabalhos que se destacam nessa área, podem ser mencionados:

- Sá (2012), apresentou o projeto e a construção da estrutura de quadrirotor com três graus de liberdade, modelou a dinâmica do sistema e projetou um controlador PID;
- Terzian; Silva; Demarch (2018), desenvolveu e aplicou a técnica de controle ótimo em um VANT com um grau de liberdade, dessa forma demonstrou-se suas vantagens em relação ao controle PID;
- Resende; Silva; Lima; Sena (2018), propôs um controlador híbrido FUZZI-PID para a estabilização de um VANT com dois graus de liberdade. Além disso, desenvolveram um supervisor capaz de monitorar ângulos e permitir a sintonização do controlador em tempo real.

O presente estudo propõe-se a análise do mesmo problema sob uma nova perspectiva, a partir do controle em espaço de estados como alternativa aos métodos anteriormente citados.

### 1.3 Definição do problema

Os modelos dos quadrirotores são complexos, apresentam natureza instável que é uma característica do desempenho dos motores. Para a estabilização do quadrirotor, faz-se necessário o desenvolvimento de um sistema de controle. (BEARD, 2008; SÁ, 2012).

Segundo Randal W. Beard (2008) para estudar e compreender o modelo dos quadrirotores é realizado uma simplificação no modelo, no qual deve-se linearizar e retirar os termos que não terão ganhos significativos (BEARD, 2008).

Outro desafio encontrado está relacionado à identificação da planta do sistema, devido às diversas variáveis de controle dos modelos matemáticos que descrevem a rotação de um corpo rígido. Para isso, adotou-se o método da caixa cinza, que permitiu a modelagem da planta e a determinação experimental dos parâmetros.

### 1.4 Objetivos

Nesta seção são apresentados o objetivo geral e os objetivos específicos do trabalho desenvolvido.

#### 1.4.1 Objetivo geral

Desenvolver um protótipo de um quadrirotor com um grau de liberdade, identificar o modelo dinâmico que descreve seu comportamento e implementar um sistema de controle baseado em espaço de estados.

#### 1.4.2 Objetivos específicos

- Desenvolver um protótipo de um Quadrirotor com um grau de liberdade;
- Implementar o controle dos motores com uma plataforma embarcada;
- Aplicar o método dos mínimos quadrados para determinar os valores dos parâmetros das equações do protótipo no espaço de estados;
- Validar o modelo matemático do protótipo através de simulação computacionais com utilização do programa MATLAB;
- Construir a síntese de um controlador com o método de alocação de polos;

- Simular sistema de controle em malha fechada no programa MATLAB;
- Implementar o sistema de controle em malha fechada na plataforma embarcada;
- Comparar as respostas teóricas e práticas do protótipo.

### **1.5 Justificativa**

O principal problema dos sistemas de controle relacionados a área de quadrirotores está na complexidade do modelo matemático que descreve o comportamento dinâmico desses sistemas. O estudo realizado favorece na compreensão e aplicação de uma técnica moderna de controle (espaço de estados), em sistema instáveis como o quadrirotor. Além disso, o protótipo desenvolvido pode servir como base para estudos futuros, facilitando a aplicação de novas estratégias de controle.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada a fundamentação teórica que embasa o processo de modelagem e controle do protótipo.

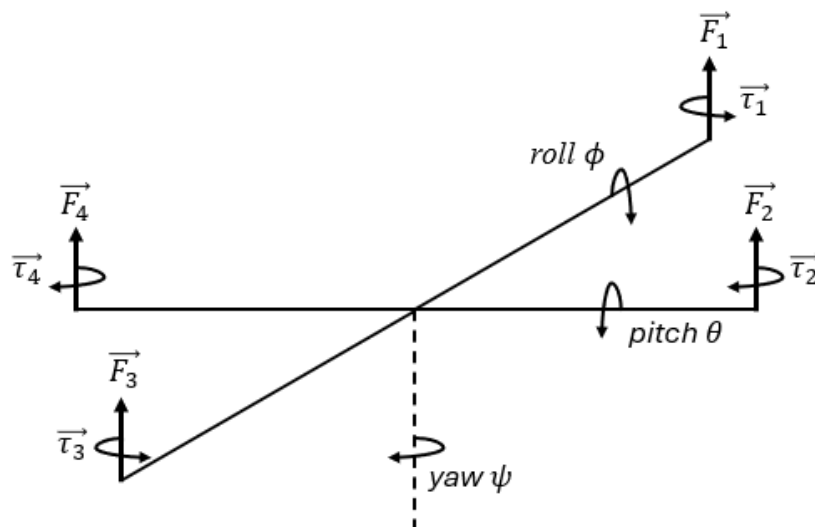
### 2.1 Quadrirotores

O quadrirotor utiliza quatro motores para locomover-se, a estrutura pode ser construída no formato de  $\times$  ou  $+$  com um motor em cada extremidade. Cada motor tem uma hélice e o arranjo é feito de forma em que dois motores giram no sentido horário e dois no sentido anti-horário, o que reduz o torque gerado pela velocidade angular (SÁ, 2012).

#### 2.1.1 Graus de liberdade

Um quadrirotor é composto por seis graus de liberdade, três são responsáveis pelas coordenadas de posição ( $x$ ,  $y$ ,  $z$ ) e três pelos ângulos de orientação ( $\theta$ ,  $\phi$ ,  $\psi$ ). Os ângulos de orientação são: *pitch*, *roll* e *yaw*, também conhecidos como ângulos de Euler (SÁ, 2012). Na Figura 1, apresenta-se o sistema de coordenadas.

Figura 1: Sistema de coordenadas de um quadrirotor

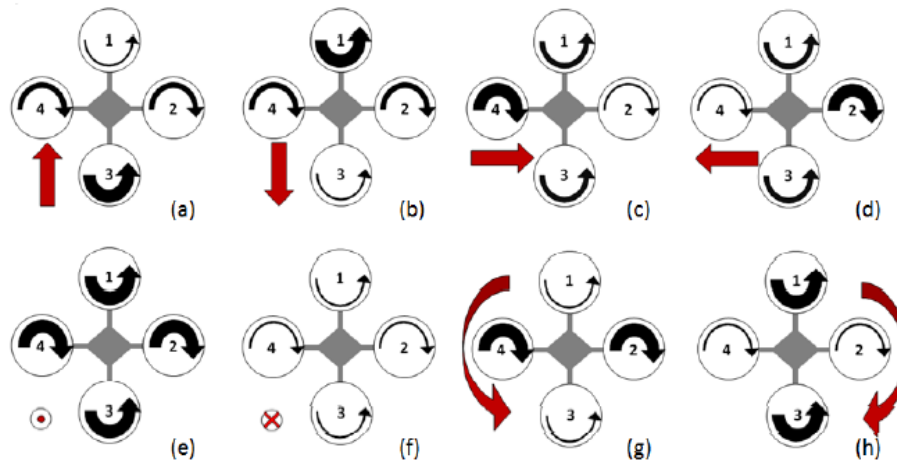


Fonte: adaptado de BEARD (2008)

### 2.1.2 Dinâmica do quadricóptero

Na Figura 2, apresenta-se uma visão superior da rotação dos motores e o resultado de cada movimento. As setas em preto representam a rotação e as em vermelho o movimento, a largura da seta é proporcional à velocidade (SÁ, 2012).

Figura 2: Descrição de rotação e movimento



Fonte: adaptado de SÁ (2012)

Para realizar o movimento *pitch*, a velocidade dos Motores 2 e 4 são fixadas e a velocidade do Motor 1 é variada em relação ao Motor 3, o que move o quadricóptero para frente (a) ou para trás (b). Para o movimento *roll*, a velocidade dos Motores 1 e 3 são fixadas e a velocidade dos Motores 2 e 4 são variadas, que resulta no deslocamento do quadricóptero para os lados (c e d) (SÁ, 2012).

Durante o movimento de subida (e) ocorre o aumento simultâneo da velocidade dos quatro motores, já o movimento de descida (f), ocorre com a redução da velocidade dos quatro motores (SÁ, 2012).

O movimento *yaw*, decorre com a variação da velocidade dos motores presentes no mesmo eixo, o que gera um torque em torno do eixo. Assim, o aumento da velocidade dos Motores 2 e 4 resulta em uma velocidade angular no sentido anti-horário (g) e o aumento da velocidade dos Motores 1 e 3 resulta em uma velocidade angular no sentido horário (h) (SÁ, 2012).

## 2.2 Modelo Dinâmico do Quadrirotor

O conhecimento do modelo matemático do sistema traz agilidade durante a sintonia no controlador além de entender a dinâmica e estabilidade do sistema. Em um sistema dinâmico o modelo matemático é definido como um conjunto de equações que descrevem seus processos de forma adequada, conforme o projeto (OGATA, 2010).

O desenvolvimento das equações contidas nas Subseções (2.2.2) à (2.2.5) tem como referência o artigo *Quadrotor dynamics and control*, de BEARD, publicado por *Brigham Young University* em 2008 (BEARD, 2008).

### 2.2.1 Sistema de coordenadas do Quadrirotor

Conforme ilustrado na Figura 1, a matriz  $R$  descreve a rotação das coordenadas de um objeto em um espaço tridimensional em torno de um eixo específico (MAGNUSSEN; SKJONHAUG, 2011).

$$R = R(\theta)R(\phi)R(\psi) \quad (2.1)$$

sendo,  $R(\theta)$ ,  $R(\phi)$  e  $R(\psi)$  as matrizes de rotação correspondentes aos ângulos *pitch*, *roll* e *yaw*, respectivamente. Essas matrizes são definidas por,

$$R(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

$$R(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.3)$$

$$R(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

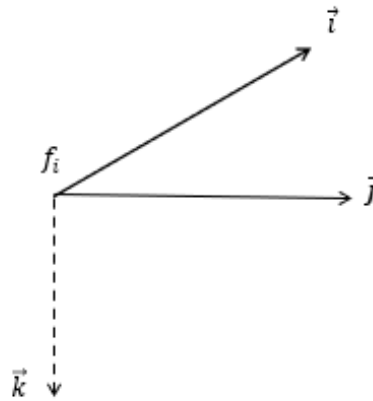
Para converter o referencial do quadrirotor  $f_b$  para o referencial inercial  $f_i$  é necessário transpor a matriz  $R$ , Equação (2.1) (MAGNUSSEN; SKJONHAUG, 2011). Com isso, tem-se,

$$R^T = \begin{bmatrix} \cos \theta \cos \psi & \sin \theta \sin \phi \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \theta \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \theta \cos \phi \end{bmatrix} \quad (2.5)$$

### 2.2.2 Cinemática do Quadrirotor

O referencial inercial  $f_i$  tem sua origem localizada no centro de massa do quadrirotor. O vetor unitário  $\vec{i}$  aponta para o Motor 1,  $\vec{j}$  apontado para o Motor 2 e  $\vec{k}$  apontado para o centro da terra. Na Figura 3, ilustra-se o referencial inercial.

Figura 3: Referencial inercial



Fonte: adaptado de BEARD (2008)

A relação entre a posição inercial e a velocidade no referencial do corpo do quadrirotor é dada por,

$$\frac{d}{dt} \begin{bmatrix} p_i \\ p_j \\ -h \end{bmatrix} = R^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.6)$$

sendo,  $p_i$  e  $p_j$  as posições inerciais do quadrirotor ao longo do referencial inercial,  $\vec{i}$  e  $\vec{j}$ , respectivamente;  $h$  a altitude do quadrirotor ao longo do referencial inercial  $\vec{k}$ ; e  $u$ ,  $v$  e  $w$  as velocidades no referencial do quadrirotor ao longo dos referenciais inerciais  $\vec{i}$ ,  $\vec{j}$  e  $\vec{k}$ , respectivamente.

Para relacionar os ângulos de Euler  $\theta$ ,  $\phi$  e  $\psi$  às taxas de variação angular no referencial do quadricóptero, substituí-se as coordenadas do objeto na Equação (2.6),

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \quad (2.7)$$

sendo,  $p$ ,  $q$  e  $r$  as velocidades angulares medidas ao longo de  $\vec{i}$ ,  $\vec{j}$  e  $\vec{k}$ , respectivamente, no referencial do corpo do quadricóptero; e  $\dot{\theta}$ ,  $\dot{\phi}$  e  $\dot{\psi}$  as velocidades referentes inerciais.

Ao reorganizar obtém-se,

$$\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.8)$$

Ao considerar que ângulos  $\theta$  e  $\phi$  são pequenos, a Equação (2.8) é simplificada como,

$$\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.9)$$

### 2.2.3 Dinâmica de Corpo Rígido

Seja  $\vec{v}$  o vetor velocidade do quadricóptero, a lei de Newton aplicada ao movimento de translação é,

$$m \frac{d\vec{v}}{dt_i} = \vec{F} \quad (2.10)$$

sendo,  $m$  a massa do quadricóptero,  $\frac{d}{dt_i}$  a derivada temporal no referencial inercial e  $\vec{F}$  a força total.

Da Equação de Coriolis tem-se,

$$m \frac{d\vec{v}}{dt_i} = m \left( \frac{d\vec{v}}{dt_b} + w_b \times \vec{v} \right) = \vec{F} \quad (2.11)$$

sendo,  $w_b$  a velocidade angular em relação ao referencial inercial e  $\frac{d}{dt_b}$  a derivada temporal no referencial do quadrirotor.

Para expressar as grandezas em coordenadas do quadrirotor e ao reorganizar a Equação (2.11), obtém-se,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} \vec{F}_x \\ \vec{F}_y \\ \vec{F}_z \end{bmatrix} \quad (2.12)$$

sendo,  $\dot{u}$ ,  $\dot{v}$  e  $\dot{w}$  as acelerações no referencial do quadrirotor; e  $\vec{F}_x$ ,  $\vec{F}_y$  e  $\vec{F}_z$  as forças aplicadas nas coordenadas  $x$ ,  $y$  e  $z$ , respectivamente.

Para o movimento rotacional, a segunda lei de Newton afirma que,

$$m \frac{d\vec{h}}{dt_i} = \vec{\tau} \quad (2.13)$$

sendo,  $\vec{h}$  o momento angular e  $\vec{\tau}$  o torque aplicado.

A Equação de Coriolis para o momento angular é dado por,

$$m \frac{d\vec{h}}{dt_i} = \frac{d\vec{h}}{dt_b} + w_b \times \vec{h} = \vec{\tau} \quad (2.14)$$

Novamente, a equação é mais facilmente resolvida em coordenadas corporais, onde,  $\vec{h} = Jw_b$  e  $J$  é a matriz de inercia,

$$J = \begin{bmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{bmatrix} \quad (2.15)$$

O quadricóptero é essencialmente simétrico em relação aos três eixos, portanto,

$$J = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \quad (2.16)$$

Para converter  $J$  do referencial do quadricóptero para o referencial inercial, se faz necessário a Equação a seguir,

$$J^T = \begin{bmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix} \quad (2.17)$$

Da Equação (2.14) pode-se reescrever a equação de Coriolis nas coordenadas do quadricóptero, assim,

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix} \left( \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \vec{\tau}(\phi) \\ \vec{\tau}(\theta) \\ \vec{\tau}(\psi) \end{bmatrix} \right) \quad (2.18)$$

sendo,  $\dot{p}$ ,  $\dot{q}$  e  $\dot{r}$  as acelerações angulares; e  $\vec{\tau}(\phi)$ ,  $\vec{\tau}(\theta)$  e  $\vec{\tau}(\psi)$  os torques aplicados nos ângulos  $\phi$ ,  $\theta$  e  $\psi$ , respectivamente. Ao realizar as operações matemáticas na Equação (2.18), obtém-se,

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_x - J_z}{J_y} pr \\ \frac{J_x - J_y}{z} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} \vec{\tau}(\phi) \\ \frac{1}{J_y} \vec{\tau}(\theta) \\ \frac{1}{z} \vec{\tau}(\psi) \end{bmatrix} \quad (2.19)$$

Para simplificar o modelo assume-se que os termos de Coriolis  $qr$ ,  $pr$  e  $pq$  são pequenos e ao substituir a Equação (2.9) em (2.19), obtém-se,

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{J_x} \vec{\tau}(\phi) \\ \frac{1}{J_y} \vec{\tau}(\theta) \\ \frac{1}{z} \vec{\tau}(\psi) \end{bmatrix} \quad (2.20)$$

sendo,  $\ddot{\phi}$ ,  $\ddot{\theta}$  e  $\ddot{\psi}$  as acelerações angulares.

#### 2.2.4 Análise de Forças, Torques e Momentos

No quadrirotor as principais forças do sistema são devido a gravidade, sendo possível assumir que as forças de empuxo e momentos aerodinâmicos são insignificantes. Como pode-se ver na Figura 1 cada motor produz uma força  $\vec{F}$  e um torque  $\vec{\tau}$ . Portanto, a força total que atua no quadrirotor é dada por,

$$\vec{F}_T = \vec{F}_1 + \vec{F}_2 + \vec{F}_3 + \vec{F}_4 \quad (2.21)$$

sendo,  $\vec{F}_T$  a força total aplicada ao quadrirotor e  $\vec{F}_1$ ,  $\vec{F}_2$ ,  $\vec{F}_3$  e  $\vec{F}_4$  as forças produzidas pelos Motores 1, 2, 3 e 4, respectivamente.

O torque  $\vec{\tau}(\phi)$ , responsável pelo ângulo *roll*, é produzido pela força dos Motores 2 e 4, aplicadas a uma distância  $l$  do eixo de rotação, portanto,

$$\vec{\tau}(\phi) = l(\vec{F}_4 - \vec{F}_2) \quad (2.22)$$

Da mesma forma, o torque *pitch*  $\tau(\theta)$ , produzido pelas forças dos Motores 1 e 3 é,

$$\vec{\tau}(\theta) = l(\vec{F}_1 - \vec{F}_3) \quad (2.23)$$

Segundo à terceira lei de Newton, o arrasto das hélices produz um torque *yaw*  $\tau(\psi)$  no corpo do quadricóptero. A direção desse torque é oposta à direção do movimento da hélice. Portanto, o torque total de *yaw* é dado por,

$$\vec{\tau}(\psi) = \vec{\tau}_2 + \vec{\tau}_4 - \vec{\tau}_1 - \vec{\tau}_3 \quad (2.24)$$

A sustentação e o arrasto produzidos pelas hélices são proporcionais ao quadrado da velocidade angular. Assumiu-se que a velocidade angular é diretamente proporcional ao sinal enviado ao motor. Portanto, a força e o torque de cada motor podem ser expressos como,

$$\vec{F}_i = k_1 \delta_i \quad (2.25)$$

$$\vec{\tau}_i = lk_2 \delta_i \quad (2.26)$$

sendo  $k_1$  e  $k_2$  são constantes que precisam ser determinadas experimentalmente,  $\delta_i$  é o sinal de comando do motor e  $i \in \{1, 2, 3, 4\}$  representam índices dos Motores 1, 2, 3 e 4.

Portanto, as forças e torques no quadricóptero podem ser escritos como,

$$\begin{bmatrix} \vec{F}_T \\ \vec{\tau}(\phi) \\ \vec{\tau}(\theta) \\ \vec{\tau}(\psi) \end{bmatrix} = \begin{bmatrix} k_1 & k_1 & k_1 & k_1 \\ 0 & -lk_2 & 0 & lk_2 \\ lk_2 & 0 & -lk_2 & 0 \\ -k_2 & k_2 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \quad (2.27)$$

Além da força exercida pelos motores, a gravidade também exerce uma força sobre o quadricóptero. A força gravitacional que atua no centro de massa é dada por,

$$\vec{F}_g = \begin{bmatrix} 0 \\ 0 \\ m'g \end{bmatrix} \quad (2.28)$$

sendo,  $\vec{F}_g$  a força gravitacional,  $m'$  a massa total do quadricóptero e  $g$  a gravidade.

Os momentos de inércia do quadricóptero são calculados a partir de um centro esférico de raio  $R$  e massa  $m$ . Para representar os motores são consideradas massas pontuais  $M$  a uma distância  $l$  do centro. Como apresentado na Figura 1 o quadricóptero é simétrico, portanto, a inércia para cada uma das posições para uma esfera sólida mais as massas pontuais são dadas por,

$$J_x = \frac{2mR^2}{5} + 2l^2M \quad (2.29)$$

$$J_y = \frac{2mR^2}{5} + 2l^2M \quad (2.30)$$

$$J_z = \frac{2mR^2}{5} + 2l^2M \quad (2.31)$$

sendo,  $J_x$ ,  $J_y$  e  $J_z$  os momentos de inércia em relação as posições  $x$ ,  $y$  e  $z$ , respectivamente.

### 2.2.5 Modelo

O objetivo do modelo é ser utilizado para a construção de um sistema capaz de controlar a posição, a altitude e a direção do quadricóptero com base em medições em relação a uma posição alvo. A dinâmica do sistema é analisada no próprio referencial do quadricóptero. As equações apresentadas a seguir representam os seis graus de liberdade do sistema,

$$\ddot{x} = -\cos\phi \sin\theta \frac{\vec{F}_T}{m} \quad (2.32)$$

$$\ddot{y} = \sin\phi \frac{\vec{F}_T}{m} \quad (2.33)$$

$$\ddot{z} = g - \cos \phi \cos \theta \frac{\overrightarrow{F_T}}{m} \quad (2.34)$$

$$\ddot{\phi} = \frac{1}{J_x} \tau(\phi) \quad (2.35)$$

$$\ddot{\theta} = \frac{1}{J_y} \tau(\theta) \quad (2.36)$$

$$\ddot{\psi} = \frac{1}{J_z} \tau(\psi) \quad (2.37)$$

sendo,  $\ddot{x}$ ,  $\ddot{y}$  e  $\ddot{z}$  as acelerações lineares ao longo dos eixos  $x$ ,  $y$  e  $z$ , respectivamente, e  $\ddot{\phi}$ ,  $\ddot{\theta}$  e  $\ddot{\psi}$  as acelerações angulares ao longo desses mesmos ângulos.

As Equações (2.35) a (2.37) são derivadas a partir da Equação (2.20). Neste projeto, o controle de um quadricóptero com um grau de liberdade utiliza exclusivamente a Equação (2.35).

### 2.3 Método dos Mínimos Quadrados

O Método dos Mínimos Quadrados tem como objetivo realizar a estimação de certos parâmetros a partir do sinal de entrada do sistema, cujo dados estão sujeitos a ruídos de medições (HELENE, 2006).

Sabe-se que a equação da reta é conhecida como  $y = a_0 + b_0x$ , sendo  $y$  a saída do sistema,  $x$  o sinal de entrada e  $a_0$ ,  $b_0$  os parâmetros da reta. A técnica dos Mínimos quadrados pode ser aplicada com objetivo de ajustar os parâmetros  $a_0$  e  $b_0$  de forma a reduzir a soma das diferenças quadráticas entre os pontos da reta e do sinal de entrada, dividido pela respectiva variância. Esse método leva em conta que o desvio padrão  $\sigma$  e a variância  $\sigma^2$  são conhecidos (HELENE, 2006).

A expressão generalizada do Método dos Mínimos Quadrados é conhecida como,

$$\begin{bmatrix} \tilde{a} \\ \tilde{b} \end{bmatrix} = \begin{bmatrix} \sum \frac{1}{\sigma_i^2} & \sum \frac{x_i}{\sigma_i^2} \\ \sum \frac{x_i}{\sigma_i^2} & \sum \frac{x_i^2}{\sigma_i^2} \end{bmatrix}^{-1} \begin{bmatrix} \sum \frac{y_i}{\sigma_i^2} \\ \sum \frac{x_i y_i}{\sigma_i^2} \end{bmatrix} \quad (2.38)$$

sendo,  $\tilde{a}$  e  $\tilde{b}$  os valores ajustados de  $a_0$  e  $b_0$ ,  $y_i$  a saída e  $x_i$  a entrada para os parâmetros  $i = 1, 2, \dots, n$ .

## 2.4 Controle em Espaço de Estados

O modelo em espaço de estados, também conhecido como abordagem moderna ou no domínio do tempo, aborda sistemas não lineares, variantes no tempo e com condições iniciais não nulas, torna-se uma opção para analisar e projetar uma grande variedade de sistemas (NISE, 2017).

Diferente do controle clássico que é fundamentado na relação entrada-saída, o controle moderno utiliza notação vetorial-matricial com o intuito de simplificar o sistema de equações, esse sistema é descrito em  $n$ -ésimas equações diferenciais de primeira ordem (OGATA, 2010).

O modelo no domínio do espaço de estados é representado como,

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.39)$$

$$y(t) = Cx(t) \quad (2.40)$$

sendo,  $x(t)$  o vetor de estados,  $\dot{x}(t)$  a derivada temporal do vetor de estados,  $y(t)$  o vetor de saída,  $u(t)$  o vetor de entrada,  $A$  é a matriz de transição do sistema,  $B$  é a matriz de entrada e  $C$  é a matriz de saída.

É necessário ressaltar que os autovalores da matriz  $A$  também são conhecidos como raízes características e traçam um paralelo com os pólos da função de transferência, portanto é possível saber o comportamento do sistema através da análise dos autovalores de  $A$  (OGATA, 2010).

### 2.4.1 Alocação de polos

Neste trabalho, projetou-se um servossistema do tipo 1 quando a planta possui um integrador. Esse sistema tem como objetivo fazer com que a saída rastreie uma referência desejada, mesmo na presença de distúrbios. Quando um distúrbio é aplicado, a saída desviará da referência, e o sistema atua para minimizar o erro (OGATA, 2010).

Para realizar a alocação de polos o sistema deve ser controlável e ter as variáveis de estados disponíveis para a medição, dessa forma torna-se possível alocar os polos de malha fechada através de uma realimentação de estados (OGATA, 2010).

Com base no modelo representado na Equação (2.20), pode-se definir o sinal de controle como,

$$u(t) = -K x(t) + k_1 r \quad (2.41)$$

sendo,  $u(t)$  o sinal de controle,  $K$  a matriz de ganho,  $k_1$  o primeiro elemento de  $K$  e  $r$  o sinal de referência. A Equação (2.41) é conhecida como realimentação de estados.

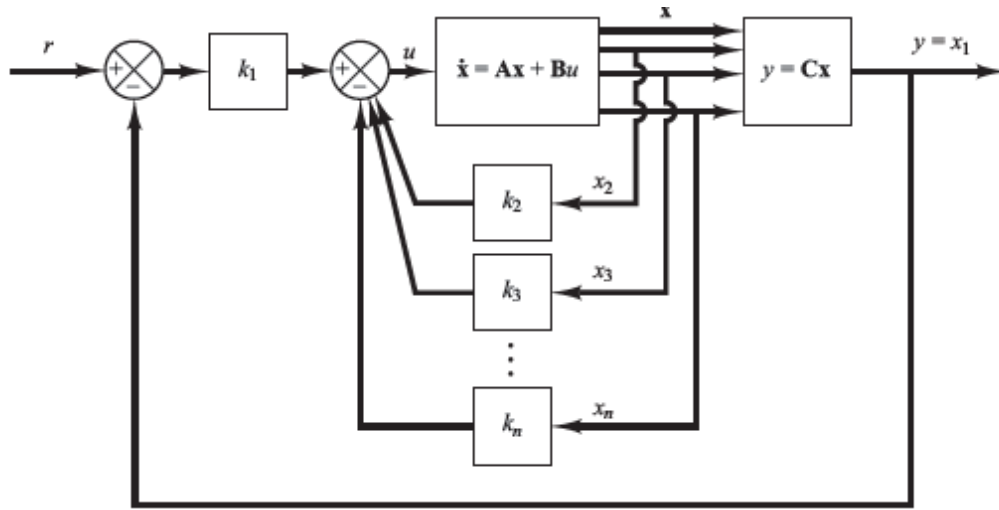
Ao substituir a Equação (2.41) na Equação (2.39), obtém-se,

$$\dot{x}(t) = (A - BK) x(t) + Bk_1 r \quad (2.42)$$

Como os autovalores da matriz  $A$  são os pólos do sistema, dessa forma ajustar a matriz  $K$  corretamente fará com que a matriz  $(A - BK)$  torne o sistema assintoticamente estável, fazendo que a saída se torne igual a referência  $r$  (OGATA, 2010).

Na Figura 4, ilustra-se o diagrama do servossistema do tipo 1 quando a planta possui um integrador,

Figura 4: Servossistema do tipo 1 em malha fechada



Fonte: adaptado de OGATA (2010)

#### 2.4.2 Espaço de estados discreto

Para o controle de sistemas reais, é necessário considerar que a medição dos sinais é realizada de forma discreta, tornando essencial o uso de modelos discretos (ROSSINI, 2020).

Assumiui-se que a matriz  $B$  e o vetor de entrada  $u(t)$  permanecem constantes entre os instantes  $t_{d-1}$  e  $t_d$ , com  $t_{d-1} < t < t_d$ , tem-se o modelo de discretização conhecida como Segurador de Ordem Zero, do inglês *Zero-Order Hold* (ZOH). Nessas condições, o instante  $t_d$  pode ser determinado a partir do instante anterior,  $t_{d-1}$ . Assim, as Equações (2.39) e (2.40), são representadas por,

$$\dot{x}(t_{d+1}) = A_D x(t_d) + B_D u(t_d) \quad (2.43)$$

$$y(t_d) = C_D x(t_d) \quad (2.44)$$

sendo,  $t_d$  o tempo discreto e  $A_D$ ,  $B_D$  e  $C_D$  são versões discretizadas das matrizes  $A$ ,  $B$  e  $C$ , respectivamente (ROSSINI, 2020).

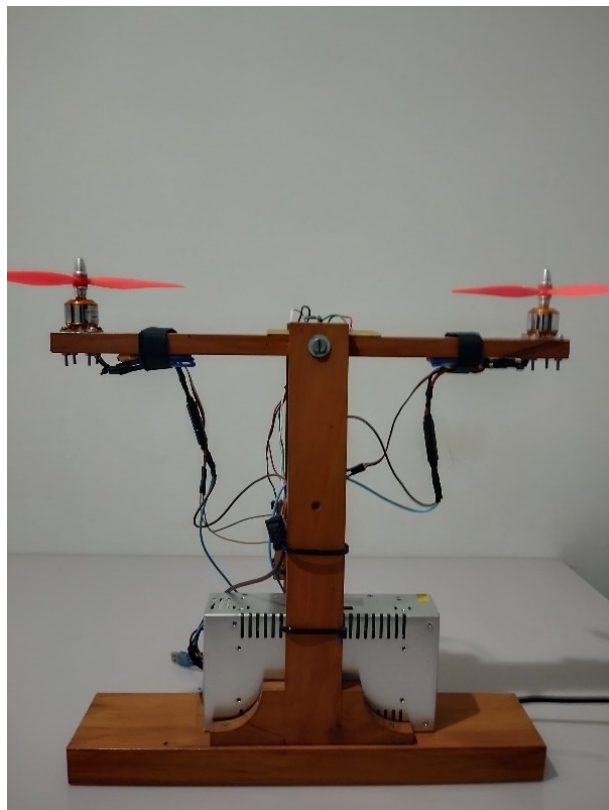
### 3. METODOLOGIA

Neste capítulo, é apresentada a lista de materiais, os procedimentos adotados na construção do protótipo, a metodologia utilizada para o desenvolvimento, modelagem e controle do sistema de estabilidade e posicionamento de um Quadrirotor com um grau de liberdade.

#### 3.1 Estrutura do Protótipo

Para o primeiro passo do projeto construiu-se o protótipo. A parte principal da estrutura é composta por duas peças de madeira: a base, com  $420\text{ mm}$  de altura, e a barra principal, com  $470\text{ mm}$  de comprimento. A barra é fixada no topo da base, e permite sua mobilidade no eixo horizontal, sua estrutura é construída de forma simétrica. O comprimento do centro de massa até o centro do motor é de  $l = 210\text{ mm}$ . Os motores Brushless, com massa  $M = 67\text{ g}$  cada, são posicionados em cada extremidade da barra, que possui uma massa  $m = 352\text{ g}$  e um raio  $R = 235\text{ mm}$ . Na Figura 5, apresenta-se uma foto do protótipo.

Figura 5: Protótipo do Quadrirotor com um grau de liberdade



Fonte: Autoria própria (2024)

Com a finalização da construção do protótipo, foi possível realizar a análise do sistema de posicionamento, seguida pela identificação e implementação do controle da planta.

### 3.1.1 Lista de materiais

Para a implementação do controle, utilizou-se o sistema embarcado Arduino Uno que contém o microcontrolador ATmega328p, sua escolha deve-se ao fato ao suporte as bibliotecas específicas utilizadas para o projeto e seu custo acessível. As principais informações estão no Quadro 1.

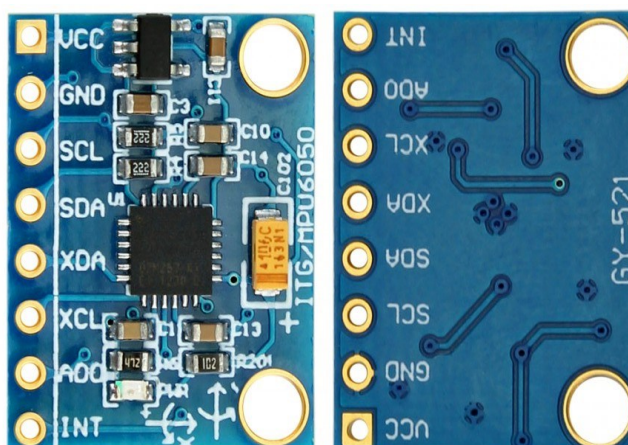
**Quadro 1: Principais informações do ATmega328p**

|                                |                       |
|--------------------------------|-----------------------|
| Núcleo                         | AVR                   |
| Pinos                          | 28 pinos DIP          |
| Largura do barramento de dados | 8 bits                |
| Tensão de operação             | 1,8 - 5 V             |
| Frequência máxima de operação  | 20 MHz                |
| Memória Flash                  | 32 Kb                 |
| SRAM                           | 2 Kb                  |
| EEPROM                         | 1 Kb                  |
| Tipo de interface              | TWI (I2C), SPI, USART |
| Número de I/O programáveis     | 23                    |

Fonte: adaptado de Microchip (2024)

Utilizou-se o módulo MPU6050 para medições do ângulo. Este módulo é composto por um acelerômetro e um giroscópio. Baseou-se a escolha na facilidade de integrá-lo ao sistema embarcado e por ser bastante utilizada no projeto de quadrirotores. Na Figura 6, apresenta-se o módulo MPU6050.

Figura 6: MPU6050



Fonte: adaptado de USINAINFO (2024)

Os pinos do MPU6050 são: VCC, pino da alimentação, GND, pino terra, SCL, pino de sinal de dados e SDA, pino de sinal de *clock*. Os demais pinos não foram utilizados no projeto.

As principais informações sobre o MPU6050 estão apresentadas no Quadro 2.

Quadro 2: Principais informações do MUP6050

|                       |                                |
|-----------------------|--------------------------------|
| Tensão de operação    | 3 - 5 V                        |
| Conversor AD          | 16 bits                        |
| Comunicação           | I2C                            |
| Faixa do giroscópio   | $\pm 250, 500, 1000, 2000$ °/s |
| Faixa do acelerômetro | $\pm 2, \pm 4, \pm 8, \pm 16g$ |

Fonte: adaptado de USINAINFO (2024)

O motor escolhido para o projeto é o *Brushless* A2212/10T 1400Kv. Este motor de corrente contínua possui três terminais de alimentação, e por não possuir escovas apresenta uma melhor eficiência energética. Sua principal aplicação são os aeromodelos, portanto, torna-se uma excelente escolha para os quadrirotores. Na Figura 7, apresenta-se o motor *Brushless* A2212/10T.

**Figura 7: Brushless A2212/10T**

**Fonte: adaptado de RBO Classificados (2024)**

O controle de velocidade é realizado pela placa ESC-30A, escolhida devido à sua compatibilidade com o motor escolhido para o projeto. A ESC-30A possui um microcontrolador interno responsável por criar a sequência de acionamento das bobinas do motor. O controle é efetuado através de um sinal de Modulação por Largura de Pulso (PWM) enviado pelo ATmega328p. Na Figura 8, apresenta-se a ESC-30A.

**Figura 8: ESC-30A**

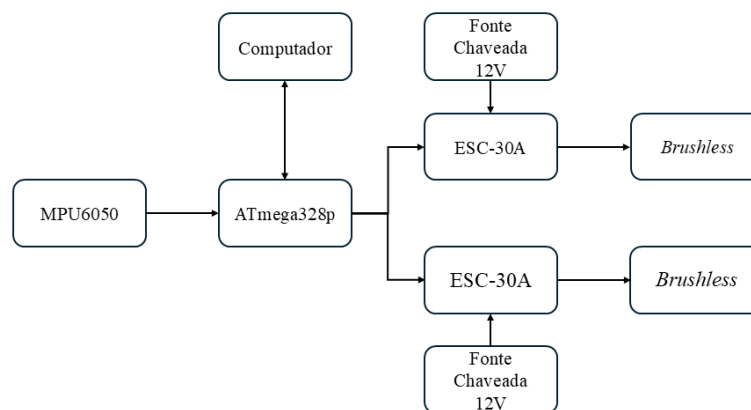
**Fonte: adaptado de Maker Hero (2024)**

### 3.1.2 Arquitetura do protótipo

O circuito desenvolvido para o projeto tem como objetivo ler o valor do ângulo e a velocidade angular do sistema. Cabe ao microcontrolador processar essas informações e gerar o sinal de controle para cada motor.

Na Figura 9, apresenta-se o diagrama de blocos do sistema, com os componentes do sistema.

**Figura 9: Diagrama de blocos do protótipo**



Fonte: Autoria própria (2024)

A conexão entre o motor *Brushless* e a ESC-30A deve ser utilizada com as informações no Quadro 3.

**Quadro 3: Conexão da Alimentação do motor Brushless**

| ESC-30A | Motor <i>Brushless</i> |
|---------|------------------------|
| A       | Fio Preto              |
| B       | Fio Amarelo            |
| C       | Fio Vermelho           |

Fonte: Autoria própria (2024)

E para conectar a ESC-30A ao microcontrolador, utiliza-se um cabo *flat* composto por 3 fios. No Quadro 4, ilustra-se a identificação da ligação dos dois componentes,

**Quadro 4: Conexão do Sinal de controle do motor Brushless**

| ATmega328p | Motor <i>Brushless</i> |
|------------|------------------------|
| GND        | Fio Preto              |
| NC         | Fio Vermelho           |
| SINAL      | Fio Branco             |

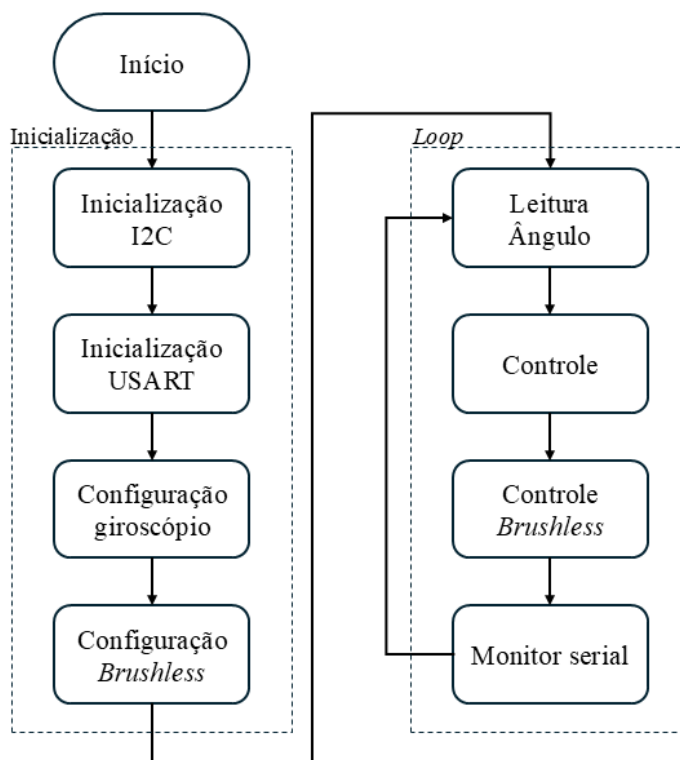
Fonte: Autoria própria (2024)

No protótipo, o pino D3 do ATmega328p foi utilizado para o sinal de controle do Motor 4, e pino D5 para o Motor 2.

### 3.2 Desenvolvimento Firmware para o Controle do Microcontrolador

Após a inicialização, o *firmware* realiza como primeiro passo a identificação do ângulo atual do protótipo, por meio da comunicação entre o módulo MPU6050 e o ATmega328p via protocolo I2C. Em seguida, o *firmware* executa a ação de controle e transmite os sinais correspondentes para as ESC-30A por meio dos pinos digitais do microcontrolador. Para possibilitar o monitoramento da ação de controle e do ângulo atual, os dados são enviados para um computador pelo protocolo USART. Esses passos são executados em *Loop*, com período de amostragem  $T_s$  de 0,01 segundos, o que limita tanto a comunicação quanto a frequência do controle. Na Figura 10, apresenta-se o fluxograma do *firmware*.

Figura 10: Fluxograma *Firmware*



Fonte: Autoria própria (2024)

### 3.3 Identificação do Sistema

Nessa seção, aborda-se os métodos para identificar os parâmetros do sistema do quadricóptero.

#### 3.3.1 Torque

Com base na Equação (2.22), pode-se determinar o torque produzido pelo quadricóptero. Portanto,

$$\vec{\tau}(\phi) = \vec{\tau}_4 - \vec{\tau}_2 \quad (3.1)$$

sendo,  $\vec{\tau}_4$  e  $\vec{\tau}_2$  os torques produzidos pelos Motores 4 e 2, respectivamente. Assim o torque pode ser descrito como,

$$\vec{\tau}(\phi) = l(k'\delta_4 - k''\delta_2) \quad (3.2)$$

onde,  $k'$  e  $k''$  são as constantes a serem determinadas para os Motores 4 e 2, respectivamente.

Para determinar as constantes considera-se a seguinte equação,

$$m_{empuxo}g = k\delta_i \quad (3.3)$$

onde,  $m_{empuxo}g$  representa o empuxo produzido pelo motor devido ao sinal de controle  $\delta_i$  e  $k$  a constante a ser determinada.

Para determinar a relação  $k$  entre o empuxo e o sinal de controle, utilizou-se uma balança digital para aferir o empuxo referente ao sinal de controle aplicado, sendo que para cada motor foi levantado 11 pontos.

O motor foi fixado a um peso, que por sua vez foi fixado à balança, utilizou-se fita dupla face para as fixações. O objetivo de utilizar o peso era para compensar o empuxo gerado pelo motor. Na Figura 11, ilustra-se o processo de aferição do empuxo gerado pelos motores.

**Figura 5: Aferição do empuxo gerado pelos motores**



**Fonte: Autoria própria (2024)**

Dando continuidade à análise, por meio do sistema embarcado, ajustou-se gradativamente o sinal de controle: aumentou-se para o Motor 4 e diminuiu-se para o Motor 2. Dessa forma, coletaram-se 11 pontos de cada motor. Para aproximar os resultados do sistema real, coletaram-se 30 amostras para cada ponto, e calculou-se a média a partir delas. Esses sinais de controle são inversamente proporcionais entre si, de modo que, ao ocorrer um distúrbio no protótipo, um dos motores reduz a rotação enquanto o outro aumenta, permitindo a correção do ângulo.

### 3.3.2 Modelo em Espaço de Estados

O protótipo construído para o presente trabalho avalia apenas o ângulo  $\phi$ . Considerando que a aceleração angular  $\ddot{\phi} = \frac{1}{J_x} \vec{\tau}(\phi)$  relaciona os sinais de controle com a velocidade angular, substituindo a Equação (3.2) na Equação (2.35), obtém-se,

$$\ddot{\phi} = \frac{1}{J_x} l(k'\delta_4 - k''\delta_2) \quad (3.4)$$

Assim, as variáveis de estados são definidas como,

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \ddot{\phi} \end{cases} \quad (3.5)$$

sendo,  $x_1$  a variável de estados que representa o ângulo de rotação  $\phi$  e  $x_2$  referente a velocidade angular  $\dot{\phi}$ .

Dessa forma, na Equação (3.4), tem-se

$$\dot{x}_2 = \frac{1}{J_x} l(k' \delta_4 - k'' \delta_2) \quad (3.6)$$

Ao reorganizar, obtém-se

$$\dot{x}_2 = \frac{1}{J_x} l k' \delta_4 - \frac{1}{J_x} l k'' \delta_2 \quad (3.7)$$

Assim, o modelo no espaço de estados é representado por,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{l k'}{J_x} & -\frac{l k''}{J_x} \end{bmatrix} \begin{bmatrix} \delta_4 \\ \delta_2 \end{bmatrix} \quad (3.8)$$

A saída do sistema para o ângulo  $\phi$ , é dada por,

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.9)$$

É importante destacar que o sistema de um quadrirotor é intrinsecamente instável e não linear, com ambos os polos situados na origem. No entanto, para o presente projeto, simplificou-se o modelo para fins de estudo, sendo linearizado em torno do ponto de equilíbrio, correspondente ao ângulo zero.

## 4 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos a partir das simulações computacionais e experimentos práticos realizados para validar o modelo e o controlador do quadricóptero.

### 4.1 Identificação do Sistema

Para medir o empuxo de cada motor, realizou-se a rotina de teste apresentada na Subseção 3.3.1. No caso do Motor 4, aumentou-se o sinal de controle gradativamente, enquanto o empuxo correspondente era aferido. Para o Motor 2, reduziu-se o sinal de controle, seguindo o mesmo procedimento de medição. Para cada motor, coletaram-se 30 amostras para cada um dos 11 pontos, e a média dessas amostras foi calculada. No Quadro 5, apresenta-se relação do empuxo com o sinal de controle para o Motor 4,

**Quadro 5: Relação do empuxo com sinal de controle do Motor 4**

| $m_{empuxo}g$ (N) | $\delta_4$ |
|-------------------|------------|
| 451,260           | 1300       |
| 485,595           | 1310       |
| 524,835           | 1320       |
| 559,170           | 1330       |
| 598,410           | 1340       |
| 647,460           | 1350       |
| 676,890           | 1360       |
| 721,035           | 1370       |
| 765,180           | 1380       |
| 809,325           | 1390       |
| 853,470           | 1400       |

Fonte: autor próprio (2024)

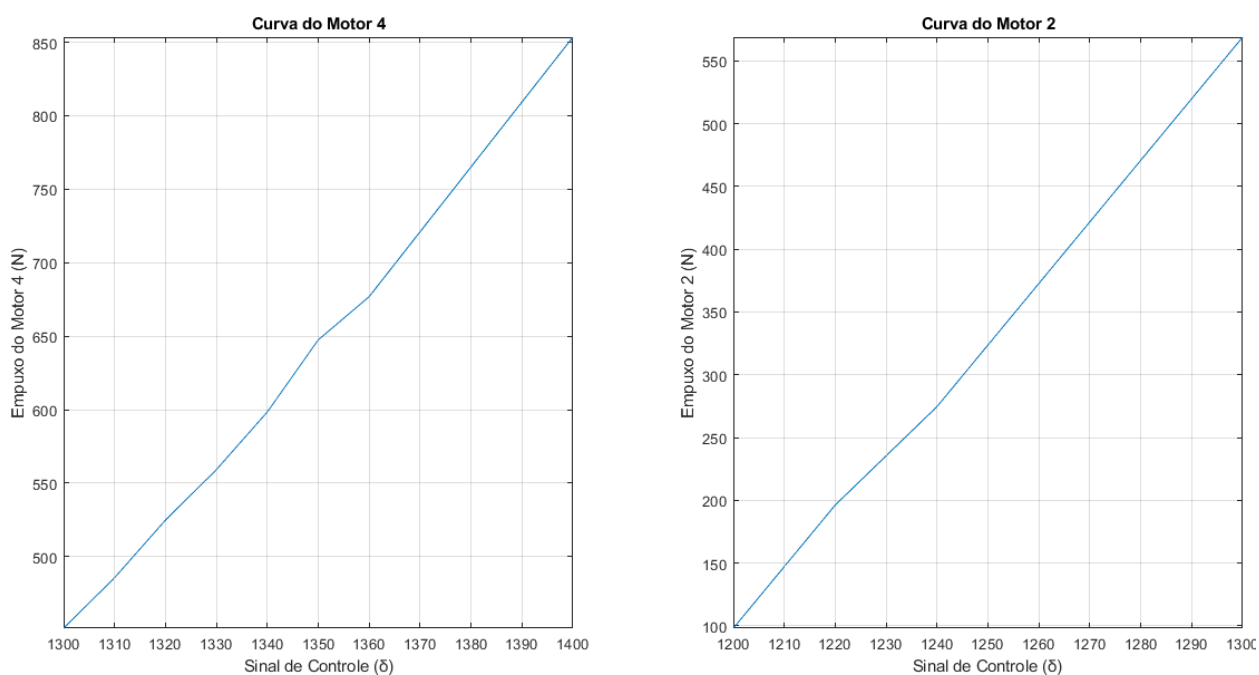
No Quadro 6, apresenta-se a relação do empuxo com o sinal de controle para o Motor 2,

**Quadro 6: Relação do empuxo com sinal de controle do Motor 2**

| $m_{empuxo}g$ (N) | $\delta_2$ |
|-------------------|------------|
| 568,980           | 1300       |
| 519,930           | 1290       |
| 470,880           | 1280       |
| 421,830           | 1270       |
| 372,780           | 1260       |
| 323,730           | 1250       |
| 274,680           | 1240       |
| 235,440           | 1230       |
| 196,200           | 1220       |
| 147,150           | 1210       |
| 098,100           | 1200       |

Fonte: autor próprio (2024)

Na Figura 12, apresenta-se a curva dos Motores 4 e 2, com base nas relações de sinal de controle e empuxo dos Quadros 3 e 4,

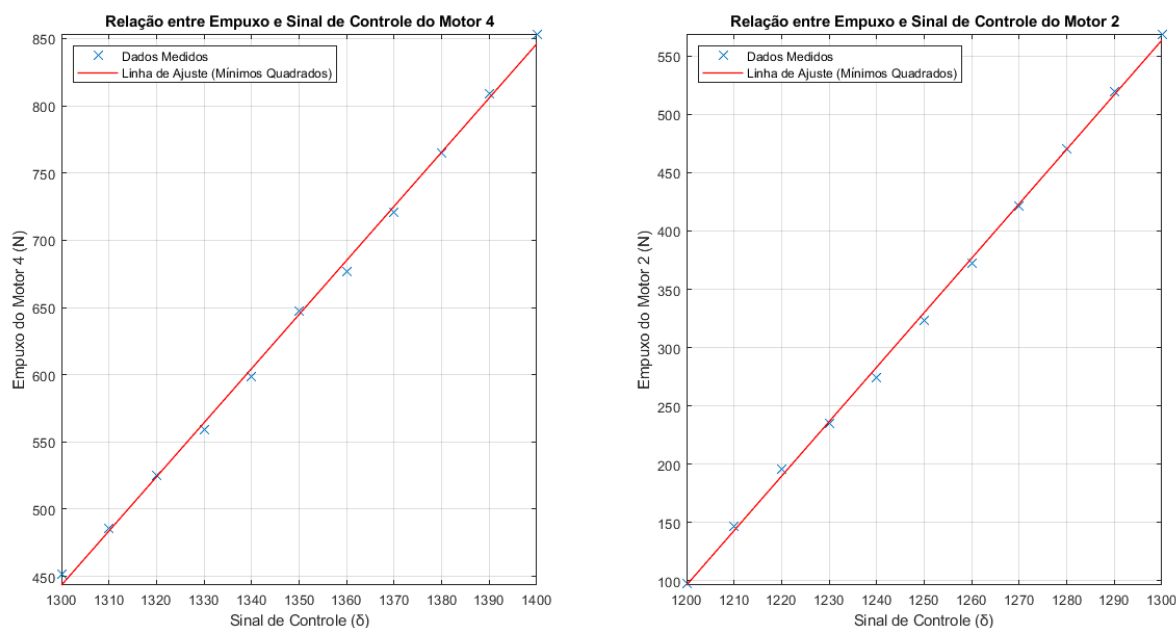
**Figura 6: Curva dos Motores**

Fonte: autor próprio (2024)

Conforme observa-se na Figura 12, o sinal de controle do Motor 4 é inversamente proporcional ao Motor 2, tendo o sinal de controle 1300 como base.

Para determinar a linha de ajuste e os coeficientes de cada sinal, utilizou-se a função *polyfit()* do MATLAB. Na Figura 13, apresenta-se as linhas de ajuste.

**Figura 7: Linha de ajuste dos Motores**



**Fonte: autor próprio (2024)**

Ao calcular o coeficiente angular das retas de ajuste, obtém-se que  $k' = 4,0266$  para o Motor 4 e  $k'' = 4,6731$  para o Motor 2. Portanto, determinaram-se as constantes que relacionam o sinal de controle aplicado e o empuxo de cada motor. Destaca-se que foram desprezadas as dinâmicas dos motores uma vez que a dinâmica dos motores é significativamente mais rápida ao comparar com o sistema completo.

#### 4.1.1 Momento de Inércia

Devido a simetria da estrutura do protótipo, o momento de inércia é o mesmo para os três eixos,  $J_x = J_y = J_z$ . Ao aplicar os dados da estrutura do protótipo na Equações (2.14), obtém-se,

$$J_x = 0,0137 \text{ Kg.m}^2$$

## 4.2 Resposta em Malha Aberta

Ao substituir os parâmetros calculados, referentes ao torque e momento de inércia, na Equação (3.2), obtém-se o seguinte modelo em espaço de estados para o quadricóptero,

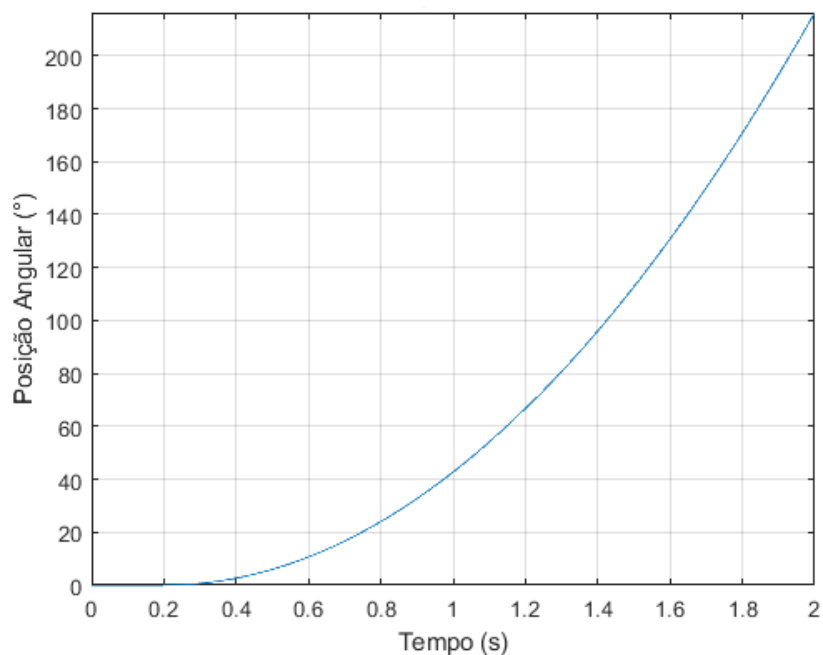
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 61,79 & -71,71 \end{bmatrix} \begin{bmatrix} \delta_4 \\ \delta_2 \end{bmatrix} \quad (4.1)$$

Com a saída do sistema para o ângulo  $\phi$ ,

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.2)$$

Os polos do quadricóptero estão localizados na origem do plano complexo, que indica o comportamento marginalmente estável do sistema. Isto significa que o sistema não tem a capacidade inerente de retornar ao equilíbrio após uma perturbação. Na Figura 14, ilustra-se esse comportamento,

**Figura 8: Resposta em Malha Aberta**



Fonte: autor próprio (2024)

Para gerar a resposta apresentada, aplicou-se um degrau iniciando 0,2 segundos aos sinais dos motores. Como mencionado na Subseção (3.3.2), os polos do sistema formam um par conjugado localizado na origem. Portanto, como pode ser visto na Figura 14, a resposta no tempo é uma rampa, caracterizando um sistema que não é BIBO (*Bounded Input, Bounded Output*), ou seja, um sistema no qual uma entrada limitada não resulta em uma saída igualmente limitada, apresentando tendência a não retornar ao equilíbrio.

### 4.3 Escolha dos Polos Desejados

Para a sintonia do controlador, o primeiro passo é determinar os polos desejados. Para estabilizar rapidamente definiu-se um tempo de estabelecimento  $T_S$  de 0,3 segundos e um sobressinal  $M_P$  de 1% para a determinação dos polos desejados.

A relação entre o sobressinal e o coeficiente de amortecimento  $\zeta$  é dada por:  $M_P = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$ . Além disso, o tempo de estabelecimento pode ser aproximado por:  $T_S \approx \frac{4}{\zeta\omega_n}$  (OGATA, 2010).

A partir dos parâmetros definidos e com um sistema de segunda ordem subamortecido, ao calcular o coeficiente de amortecimento obtém-se,  $\zeta = 0,8261$  e para a frequência natural  $\omega_n = 16,1404 \text{ rad/s}$ .

Ao aplicar os parâmetros obtidos os polos desejados são,

$$p_1 = -13,3333 + 9,0958j$$

$$p_2 = -13,3333 - 9,0958j$$

sendo,  $p_1$  e  $p_2$  um par conjugado complexo, e  $j$  representa a unidade imaginária.

Os polos encontrados indicam que o novo modelo terá uma resposta oscilatória amortecida para uma entrada em degrau, assim projetou-se o sistema para ter uma resposta rápida com o retorno estável após perturbações.

### 4.4 Alocação de Polos

Com os polos desejados calculados é possível aplicar a técnica de alocação de polos. Para determinar a matriz de ganho  $K$  responsável pela realimentação de estados, utilizou-se a função

*place()* o que permite posicionar os polos de forma a ajustar o comportamento do sistema conforme desejado. A matriz  $K$  é apresentada na Equação (4.3),

$$K = \begin{bmatrix} 1,7965 & 0,1839 \\ -2,0849 & -0,2134 \end{bmatrix} \quad (4.3)$$

Após a alocação de polos, com a Equação (2.42) é possível determinar a nova matriz de estados, com isso o novo modelo em espaço de estados é representado como,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -260,5 & -26,67 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 260,5 \end{bmatrix} r \quad (4.4)$$

Com a saída do sistema para o ângulo  $\phi$ ,

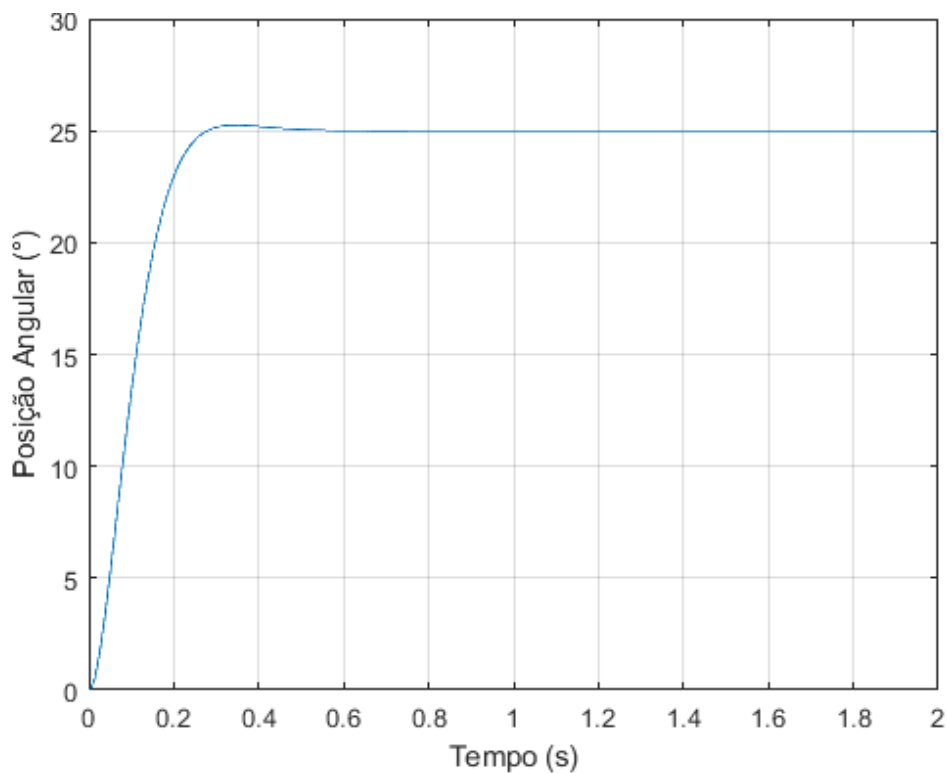
$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.5)$$

É importante destacar que, no projeto, as duas variáveis de estados são observáveis e controláveis.

#### 4.5 Resposta em Malha Fechada

Com a realimentação de estados, ao aplicar a mesma entrada aplicada no teste em malha aberta, ajustou-se o comportamento do sistema de malha fechada para atender as especificações definidas. Na Figura 15, apresenta-se o comportamento do sistema,

**Figura 9: Resposta em Malha Fechada**



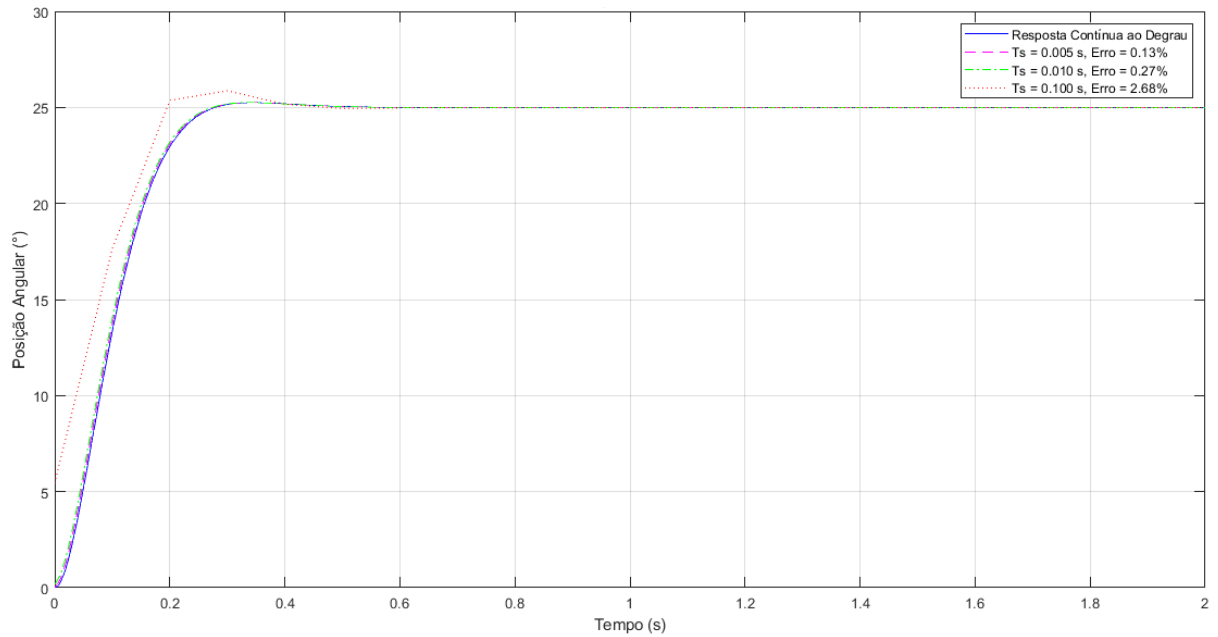
Fonte: autor próprio (2024)

Os polos encontrados indicam que o sistema apresenta uma resposta adequada a uma entrada em degrau, o que garante a estabilidade e um retorno rápido à condição de operação. Ao utilizar a função *stepinfo()*, obtém-se um sobressinal de 1% e um tempo de estabelecimento de 0,2461 segundos.

#### 4.6 Espaço de Estados Discreto

Para discretizar o sistema, utiliza-se a função *c2d()*, com o método de Tustin, aplicou-se esse processo para diferentes períodos de amostragem  $T_s$ : 0,005, 0,01 e 0,1 segundos. Na Figura 16, ilustra-se a resposta em malha fechada do sistema discreto,

**Figura 10: Resposta em Malha Fechada do Sistema Discreto**



**Fonte: autor próprio (2024)**

Após a comparação dos resultados para diferentes valores de  $T_s$ , optou-se por implementar o modelo no sistema embarcado com  $T_s = 0,01$  segundos. Essa escolha garante que as características do modelo contínuo sejam preservadas de forma adequada, sem sobrecarregar o processamento do sistema embarcado. Portanto, o modelo discreto é dado por,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0,9886 & 0,008773 \\ -2,286 & 0,7546 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0,01143 \\ 2,286 \end{bmatrix} r \quad (4.6)$$

E a saída do sistema é descrita por,

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.7)$$

#### 4.7 Lei de Controle

Para determinar a lei de controle, utiliza-se a Equação (2.41). Substituíram-se as variáveis e obteve-se o modelo em tempo discreto,

$$\begin{bmatrix} \delta_4(t_d) \\ \delta_2(t_d) \end{bmatrix} = -K \begin{bmatrix} \phi(t_d) \\ \dot{\phi}(t_d) \end{bmatrix} + k_1 r \quad (4.8)$$

sendo,  $\delta_4(t_d)$  e  $\delta_2(t_d)$  os sinais de controle no tempo discreto aplicados aos Motores 4 e 2, respectivamente,  $\phi(t_d)$  o ângulo de rotação no tempo discreto e  $\dot{\phi}(t_d)$  a velocidade angular no tempo discreto.

Ao subsistir a matriz  $K$  na Equação (4.8),

$$\begin{bmatrix} \delta_4(t_d) \\ \delta_2(t_d) \end{bmatrix} = - \begin{bmatrix} 1,7965 & 0,1839 \\ -2,0849 & -0,2134 \end{bmatrix} \begin{bmatrix} \phi(t_d) \\ \dot{\phi}(t_d) \end{bmatrix} + \begin{bmatrix} 1,7965 \\ -2,0849 \end{bmatrix} r \quad (4.9)$$

Assim, o sinal de controle correspondente ao Motor 4 é dado por,

$$\delta_4(t_d) = -1,7965\phi(t_d) - 0,1839\dot{\phi}(t_d) + 1,7965r \quad (4.10)$$

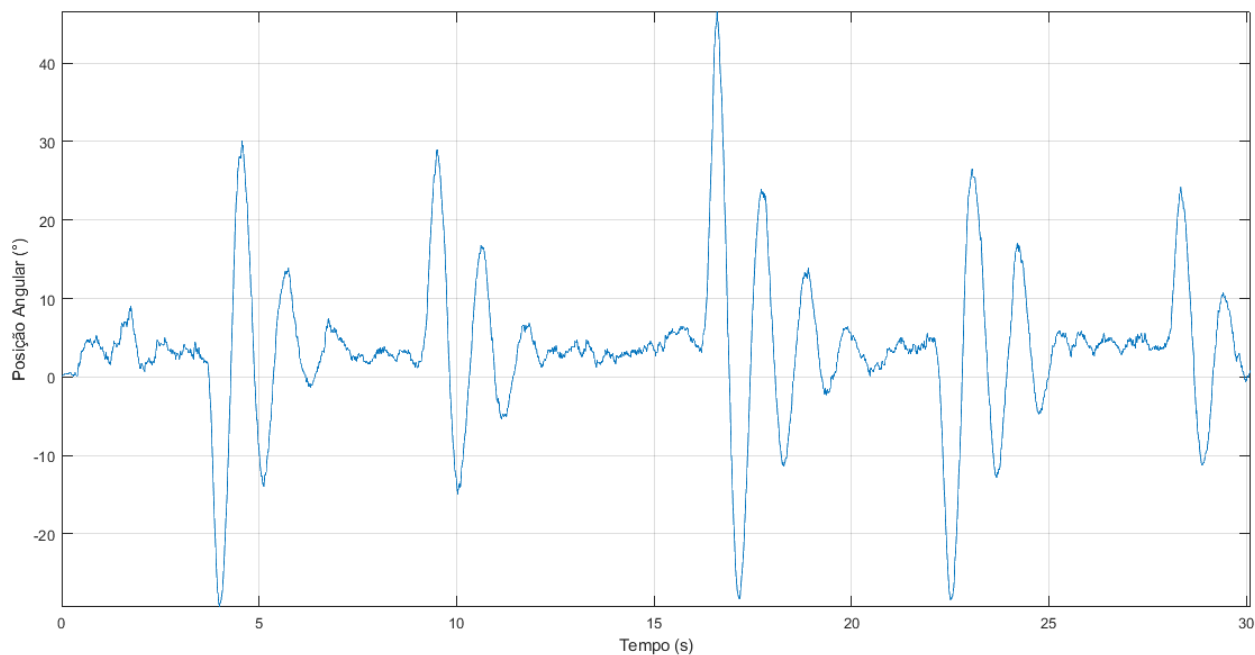
E para o Motor 2 o sinal de controle é,

$$\delta_2(t_d) = 2,0849\phi(t_d) + 0,2134\dot{\phi}(t_d) - 2,0849r \quad (4.11)$$

Para evitar que os sinais de controle saturarem, os valores de  $\delta_4(t_d)$  e  $\delta_2(t_d)$  são limitados ao intervalo de 1200 a 1400. Assim, quando o sinal de controle estiver fora desse intervalo, o sistema embarcado ajustará o valor para o limite superior ou inferior do intervalo, conforme necessário.

#### 4.8 Comparação entre o Modelo Teórico e os Dados Experimentais

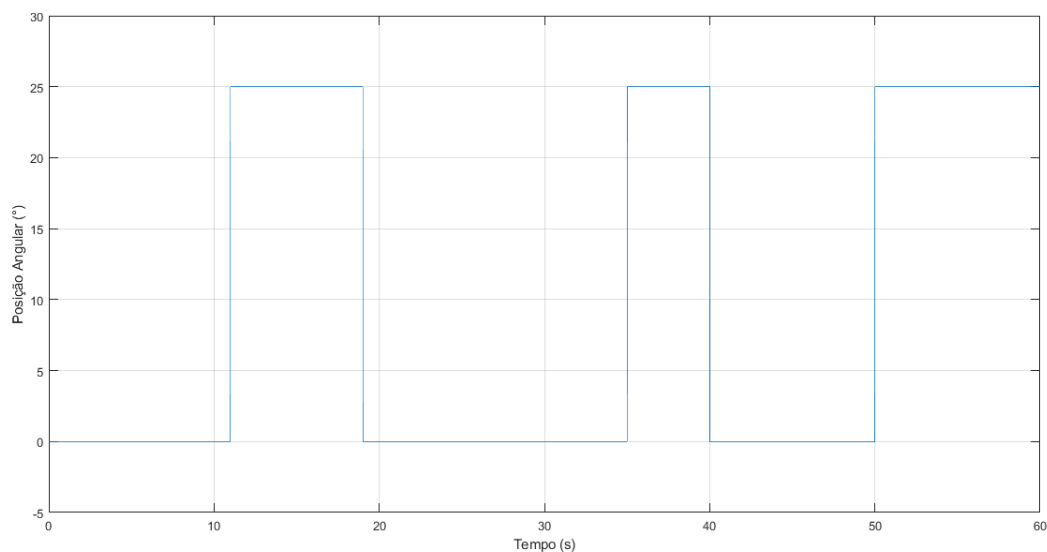
Com a lei de controle implementada no sistema embarcado, aplicaram-se distúrbios manuais ao protótipo para verificar sua capacidade de reestabelecer a condição de equilíbrio. Na Figura 17, ilustra-se a tal comportamento.

**Figura 11: Resposta do Protótipo a Distúrbios Manuais**

Fonte: autor próprio (2024)

Como pode ser observado na Figura 17, apesar do sobressinal superior ao previsto, o sistema manteve a estabilidade e controla a saída em torno do ângulo de referência zero.

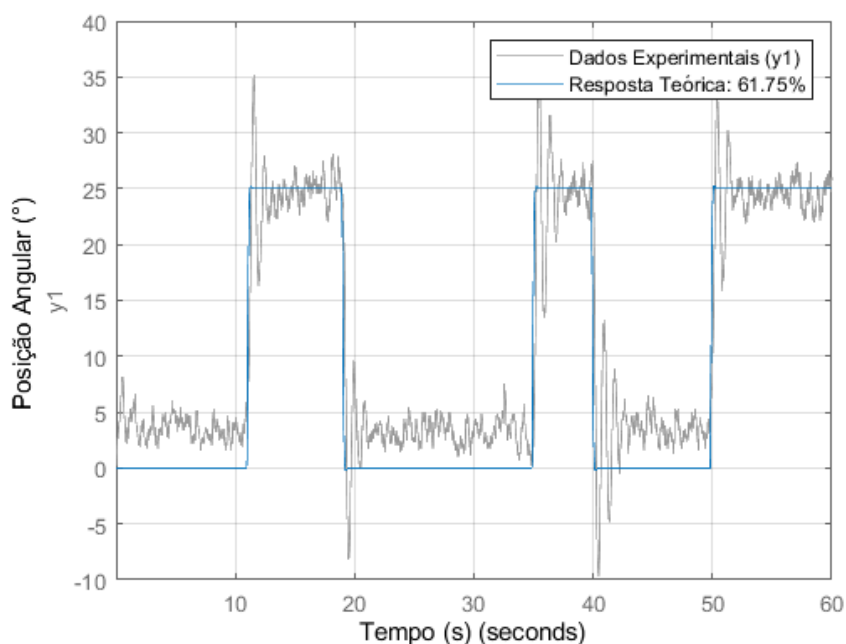
Para a validação do modelo, aplicou-se um sinal  $r$  pseudoaleatório, tendo uma amplitude de 25°. Na Figura 18, ilustra-se o sinal  $r$  aplicado ao modelo teórico e ao protótipo.

**Figura 12: Sinal  $r$** 

Fonte: autor próprio (2024)

Ao utilizar a função *compare()*, compara-se o modelo em espaço de estados discreto com os dados obtidos experimentalmente, e calcula-se o erro quadrático médio normalizado (NRMSE) entre as saídas previstas pelo modelo e os valores medidos. Na Figura 19, ilustra-se a comparação entre a resposta simulada e os dados experimentais,

**Figura 13: Resposta ao Sinal  $r$**



**Fonte: autor próprio (2024)**

Os resultados indicam que, com base na análise da função *compare()*, o protótipo apresentou uma precisão de 61,75%. Apesar do tempo de estabelecimento estar conforme projetado, o sistema apresenta um sobressinal significativamente superior ao esperado, o que impacta negativamente na precisão. Além disso, o sinal de ângulo apresentou ruído e um deslocamento próximo a 5 graus em relação ao ângulo zero, o que também afetou o desempenho do sistema.

Uma possível causa da diferença entre o modelo teórico e comportamento experimental está na natureza não linear do sistema. Durante a identificação do sistema, foi adotado um modelo linear com dois polos na origem, e mesmo com os dois integradores na planta, o protótipo apresentou um erro em regime permanente.

O comportamento indesejado no regime permanente foi o deslocamento em 5 graus quando o referencial é configurado em zero grau. Esse fenômeno não ocorre quando a referência está com ângulos positivos. Uma possível explicação para esse comportamento é a

não simetria perfeita da estrutura ou a falta de não simetria na lei de controle, conforme descrita na Equação (4.9).

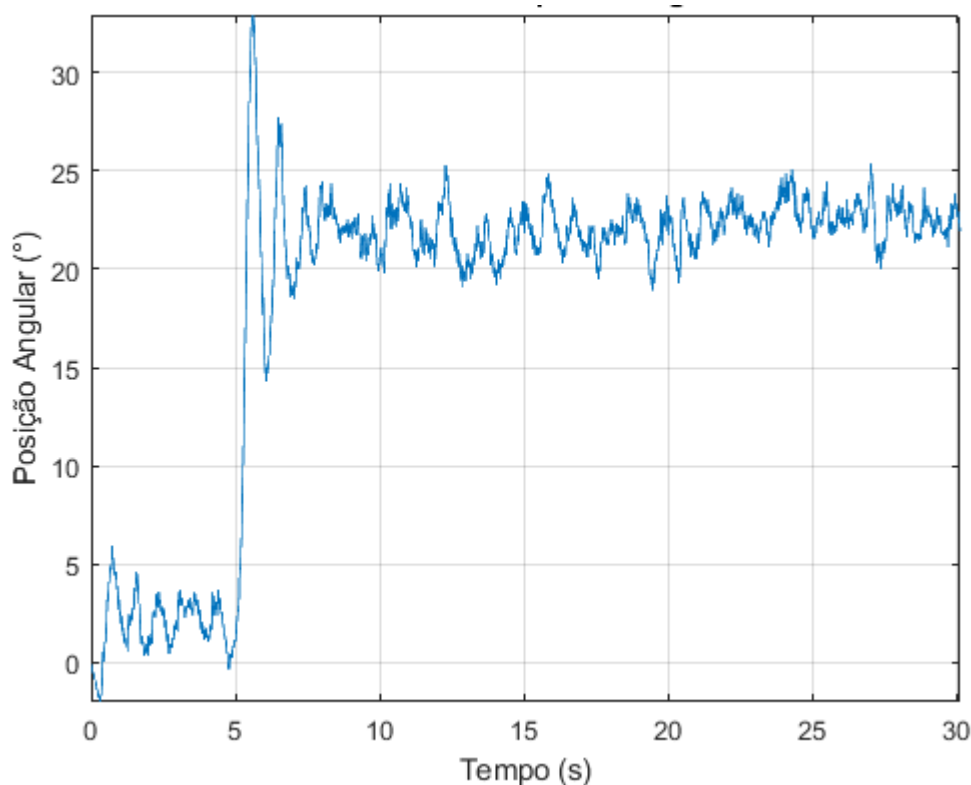
Outro fator foi a simplificação do modelo, com a remoção dos termos de Coriolis, conforme descrito na Subseção (2.2.3). Além disso, a dinâmica do motor foi desprezada, embora seja significativamente superior ao do sistema, podendo ser uma das causas da divergência obtida.

Para uma primeira aproximação do modelo, os resultados obtidos foram satisfatórios em termos práticos. Mesmo ao considerar as limitações encontradas, o desempenho do protótipo indicou ser eficaz para o controle do sistema, demonstrando a viabilidade da abordagem proposta.

#### 4.9 Análise Experimental do Protótipo ao Degrau

Com objetivo de comparar o desempenho do sistema de controle teórico e prático, aplicou-se um sinal degrau no protótipo e analisaram-se os dados obtidos. Na Figura 20, ilustra-se a resposta do protótipo ao degrau.

Figura 20: Resposta Experimental do Protótipo ao Degrau



Fonte: autor próprio (2024)

Para determinar o sobressinal e o tempo de pico, utilizou-se a função *stepinfo()*. Devido ao ruído presente no sistema real, adotou-se uma aproximação para o tempo de estabelecimento por meio da constante de tempo. Esse parâmetro foi estimado observando-se o tempo para o sistema atingir 63,33% do valor em regime permanente. Esse valor foi multiplicado por quatro, resultando em uma aproximação do tempo de estabelecimento. No Quadro 7, apresenta-se os resultados obtidos.

**Quadro 7: Comparação das Características da Resposta ao Degrau**

|                                     | <b>Resposta Teórica</b>                      | <b>Resposta Prática</b> |
|-------------------------------------|--|-------------------------|
| <b>Polos do sistema</b>             | $-13,3333 + 9,0958j$<br>$-13,3333 - 9,0958j$ | -                       |
| <b>Tempo de estabelecimento (s)</b> | 0,2461                                       | 1,36                    |
| <b>Tempo de Pico (s)</b>            | 0,3454                                       | 0,6100                  |
| <b>Sobressinal (%)</b>              | 1  | 50,0456                 |

Fonte: autor próprio (2024)

Observa-se que na resposta experimental apresenta um sobressinal significativamente superior a resposta teórica. Além disso, os tempos de resposta do sistema, como tempo de estabelecimento e o tempo de pico, também foram superiores ao projetado. No entanto, para uma primeira aproximação do modelo, o sistema prático apresentou uma resposta satisfatória, corrigindo o ângulo de acordo com o referencial.

## 5 CONCLUSÃO

Este trabalho teve como objetivo a identificação, análise e controle de um quadricóptero com um grau de liberdade, para o qual foi desenvolvido um protótipo. Os componentes selecionados para a construção atenderam aos requisitos estabelecidos no escopo do projeto, as quais permitiram a identificação e modelagem do sistema, o que possibilitou a aplicação da técnica de controle e a avaliação de modelo.

A modelagem do protótipo revelou-se o maior desafio do trabalho, devido à dificuldade em representar o sistema de modo satisfatório. Acredita-se que, sem as simplificações aplicadas ao modelo, o sistema poderia alcançar uma precisão maior. Contudo, o controle em espaço de estados proposto demonstrou bons resultados na estabilização e no posicionamento do ângulo *roll*, com sua eficácia sendo verificada nos testes realizados.

A implementação do sistema de controle ocorreu sem dificuldades, o que permitiu a observação do comportamento esperado nos testes, com um resultado de 61,57% de precisão na validação do modelo, e no levantamento dos parâmetros restantes por meio do método dos mínimos quadrados.

O *script* desenvolvido no MATLAB permitiu a simulação do sistema em malha aberta, Equações (4.1) e (4.2), e em malha fechada, Equações (4.5) e (4.6), além de possibilitar a síntese da alocação de polos, Equação (4.1) e determinar a lei de controle, Equação (4.7). Também foi possível a validação do método teórico com o prático.

Como direcionamento para trabalhos futuros, busca-se explorar outras técnicas de controle, a fim de comparar os resultados obtidos e identificar a técnica que apresente o melhor desempenho. Além disso, recomenda-se projetar um filtro para a redução do ruído no sinal do ângulo, com o objetivo de melhorar o desempenho do sistema. Como avaliação futura, realizará um ajuste fino para reduzir o sobressinal e avaliar as dinâmicas desprezadas no modelo. Outra proposta é a construção de um sistema com dois graus de liberdade, o que permitirá a implementação completa de um sistema de estabilidade e posicionamento do quadricóptero.

## REFERÊNCIAS

- AGUIRRE, L. A. Introdução à identificação de sistemas: técnicas lineares e não lineares aplicadas a sistemas reais. Belo Horizonte: Editora UFMG, 2004.
- BEARD, R. Quadrotor dynamics and control. Provo: Brigham Young University, 2008.
- BOUABDALLAH, S. Design and control of quadrotors with application to autonomous flying. 2007. 155 f. Tese (Doutorado em Ciências) – Escola Politécnica Federal de Lausanne, Lausanne, 2007.
- DORF, R. C.; BISHOP, R. H. Sistemas de controle moderno. São Paulo: Addison Wesley Longman, 1998.
- FUNDAP. Saiba mais sobre o uso de drones na pecuária. 10 maio 2019. Disponível em: <https://www.fundapba.com/informativo/33/saiba-mais-sobre-o-uso-de-drones-na-pecuaria>. Acesso em: 14 jan. 2025.
- GONÇALVES, J. V. F. Impacto do uso do drone na agricultura e pecuária: revisão bibliográfica. 2023.
- HELENE, O. Método dos mínimos quadrados: com formalismo matricial. São Paulo: Editora Livraria Física, 2006.
- MAGNUSSEN, Ø.; SKJØNHAUG, K. E. Modeling, design and experimental study for a quadcopter system construction. 2011.
- MAKER HERO. ESC 30A Brushless com BEC interno 2A 5V para aeromodelos. Disponível em: <https://www.makehero.com/produto/esc-30a-brushless-com-bec-interno-2a5v-aeromodelos/?srsltid=AfmBOoqhsWscR1AsUBbfGi4tCm2Z5T3tlcUTRmND3tMDZaIw-pwgjTs0>. Acesso em: 14 jan. 2025.
- MICROCHIP. ATmega328P. Disponível em: <https://www.microchip.com/en-us/product/atmega328p>. Acesso em: 14 jan. 2025.
- MELLO, A. A. R.; ELOI, J. C. R.; GUENTA JUNIOR, J. K.; TELLES, R. Drones militares: visão geral das investigações de 2000 a 2023. *Revista Observatorio de la Economía Latinoamericana*, v. 22, n. 10, 2023.
- NISE, N. S. Control systems engineering. 7. ed. Hoboken: John Wiley and Sons, 2015.
- OFFICE OF THE SECRETARY OF DEFENSE. Unmanned aircraft systems roadmap, 2005-2030. Washington, D.C.: Department of Defense, 2008.
- OGATA, K. Engenharia de controle moderno. 5. ed. São Paulo: Pearson Hall, 2010.
- RBO CLASSIFICADOS. Motor Brushless A2212/10T 1400KV. Disponível em: <https://rboclassificados.com.br/produto/motor-brushless-a221210t-1400kv>. Acesso em: 14 jan. 2025.

RESENDE, D. Q. M.; SILVA, A. S.; LIMA, É. F. S.; SENA, A. P. C. Desenvolvimento de um controlador híbrido Fuzzy-PID e supervisor para estabilização de drone quadricóptero. In: II CONGRESSO INTERNACIONAL DE GESTÃO E TECNOLOGIAS COINTER - PDVGT, 2018. Apresentação: Comunicação Oral.

ROSSINI, F. L. Métodos de filtragem, estimação e controle adaptativo indireto aplicados a sistemas de teleoperação bilateral. 2020. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2020.

SÁ, R. C. Construção, modelagem e controle PID para estabilidade de um veículo aéreo não tripulado do tipo quadricóptero. 2012. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Ceará, Fortaleza, 2012.

TERZIAN, E. F.; SILVA, F. G. A.; DEMARCH, L. N. Aplicação de controle ótimo na estabilização de veículo aéreo não tripulado quadricóptero limitado a um grau de liberdade. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2018.

USINAINFO. Acelerômetro e giroscópio MPU-6050 módulo GY521 3 eixos 6 DOF. Disponível em: <https://www.usinainfo.com.br/acelerometro-arduino/acelerometro-e-giroscopio-mpu-6050-modulo-gy521-3-eixos-6-dof-3895.html>. Acesso em: 14 jan. 2025.

## **APÊNDICE A - Script do MATLAB**

## Script do MATLAB

```

close all
clear
clc

%% DEFINIÇÃO DOS PARAMETROS

m = 0.352; % Massa da estrutura (kg)
R = 0.235; % Raio da estrutura (m)
M = 0.067; % Massa dos motores (kg)
l = 0.21; % Distância do centro de massa ao centro
do motor (m)

%% CÁLCULO DO MOMENTO DE INÉRCIA

Jx = (2/5 * m * R^2) + (2 * l^2 * M);

%% IDENTIFICAÇÃO DO TORQUE

empuxo4 = [451.2600, 485.5950, 524.8350,
559.1700, 598.4100, 647.4600, 676.8900, 721.0350,
765.1800, 809.3250, 853.4700];
empuxo2 = [568.9800, 519.9300, 470.8800,
421.8300, 372.7800, 323.7300, 274.6800, 235.4400,
196.2000, 147.1500, 98.1000];
sinalMotor4 = [1300, 1310, 1320, 1330, 1340, 1350,
1360, 1370, 1380, 1390, 1400];
sinalMotor2 = [1300, 1290, 1280, 1270, 1260, 1250,
1240, 1230, 1220, 1210, 1200];

% Plotagem da Curva dos Motores
figure
subplot(1, 2, 1)
plot(sinalMotor4, empuxo4)
title('Curva do Motor 4')

xlabel('Sinal de Controle ( $\delta$ )')
ylabel('Empuxo do Motor 4 (N)')
grid on
axis tight

subplot(1, 2, 2)
plot(sinalMotor2, empuxo2)
title('Curva do Motor 2')
xlabel('Sinal de Controle ( $\delta$ )')
ylabel('Empuxo do Motor 2 (N)')
grid on
axis tight
set(gcf, 'Color', 'white');

% Ajuste linear MOTOR 4
p1 = polyfit(sinalMotor4, empuxo4, 1);
K1 = p1(1);

% Ajuste linear MOTOR 2
p2 = polyfit(sinalMotor2, empuxo2, 1);
K2 = p2(1);

% Plotagem MMQ
figure
subplot(1, 2, 1)
plot(sinalMotor4, empuxo4, 'x', 'DisplayName',
'Dados Medidos', 'MarkerSize', 8)
hold on
plot(sinalMotor4, polyval(p1, sinalMotor4), '-r',
'LineWidth', 1, 'DisplayName', 'Linha de Ajuste
(Mínimos Quadrados)')
title('Relação entre Empuxo e Sinal de Controle do
Motor 4')
xlabel('Sinal de Controle ( $\delta$ )')
ylabel('Empuxo do Motor 4 (N)')

```

```

legend('Location', 'NorthWest')
grid on
axis tight
subplot(1, 2, 2)
plot(sinalMotor2, empuxo2, 'x', 'DisplayName',
'Dados Medidos', 'MarkerSize', 8)
hold on
plot(sinalMotor2, polyval(p2, sinalMotor2), '-r',
'LineWidth', 1, 'DisplayName', 'Linha de Ajuste
(Mínimos Quadrados)')
title('Relação entre Empuxo e Sinal de Controle do
Motor 2')
xlabel('Sinal de Controle ( $\delta$ )')
ylabel('Empuxo do Motor 2 (N)')
legend('Location', 'NorthWest')
grid on
axis tight
set(gcf, 'Color', 'white');

%% MODELO EM ESPAÇO DE ESTADOS

A = [0, 1; 0, 0];
B = [0, 0; 1*K1/Jx, -1*K2/Jx];
C = [1, 0];
D = 0;

% Criar modelo em espaço de estados
sys = ss(A, B, C, D);

% Análise de Estabilidade
autovalores = eig(A);
figure
pzmap(sys)
title('Polos e Zeros do Sistema em Malha Aberta')
grid on
set(gcf, 'Color', 'white');

% Simulação da Resposta em Malha Aberta
% Resposta ao Impulso
t = 0:0.001:2;

r_imp = zeros(length(t), 2);
r_imp(201, 1) = 1;
r_imp(201, 2) = -1;

[y_imp, t_imp] = lsim(sys, r_imp, t);
figure
plot(t_imp, y_imp)
title('Resposta ao Impulso em Malha Aberta')
xlabel('Tempo (s)')
ylabel('Posição Angular (°)')
grid on
axis tight
set(gcf, 'Color', 'white');

% Resposta ao Degrau
r_step(t >= 0.2, 1) = 1;
r_step(t >= 0.2, 2) = -1;

[y_deg, t_deg] = lsim(sys, r_step, t);
figure
plot(t_deg, y_deg)
title('Resposta ao Degrau em Malha Aberta')
xlabel('Tempo (s)')
ylabel('Posição Angular (°)')
grid on
axis tight
set(gcf, 'Color', 'white');

% Matriz de controlabilidade
controlabilidade = ctrb(A, B);
posto_ctrl = rank(controlabilidade);

%% ESCOLHA DOS POLOS DESEJADOS
% Especificações de desempenho
Ts = 0.3; % Tempo de assentamento desejado (s)
Mp = 0.01; % Sobressinal desejado (%)

% Cálculo do coeficiente de amortecimento (zeta) e
da frequência natural (omega_n)
zeta = -log(Mp) / sqrt(pi^2 + log(Mp)^2);

```

```

omega_n = 4 / (Ts * zeta);

% Cálculo dos polos desejados
parte_real = -zeta * omega_n;
parte_imaginaria = omega_n * sqrt(1 - zeta^2);
p1 = parte_real + 1i * parte_imaginaria;
p2 = parte_real - 1i * parte_imaginaria;
polos_desejados = [p1 p2];

%% PROJETO DE SERVOSISTEMAS DO TIPO
1 QUANDO A PLANTA POSSUI INTEGRADOR

K = place(A, B, polos_desejados);
A_cl = A - B * K;
k1 = [K(1, 1); K(2, 1)];

sys_cl = ss(A_cl, B*k1, C, D);
stepinfo(sys_cl)

% Análise de Estabilidade
novosAutovalores = eig(A_cl);
figure
pzmap(sys_cl)
title('Polos e Zeros do Sistema em Malha Fechada')
grid on
set(gcf, 'Color', 'white');

% Simulação da Resposta em malha fechada
% Resposta ao Impulso
r_imp2 = zeros(1, length(t));
r_imp2(201) = 25;

[y_imp2, t_imp2] = lsim(sys_cl, r_imp2, t);
figure
plot(t_imp2, y_imp2)
title('Resposta ao Impulso em Malha Fechada')
xlabel('Tempo (s)')
ylabel('Posição Angular (°)')
grid on
set(gcf, 'Color', 'white');

% Resposta ao Degrau
for n = 1:length(t)
    r_step2(n) = 25;
end

[y_deg2, t_deg2] = lsim(sys_cl, r_step2, t);
figure
plot(t_deg2, y_deg2)
title('Resposta ao Degrau em Malha Fechada')
xlabel('Tempo (s)')
ylabel('Posição Angular (°)')
grid on
set(gcf, 'Color', 'white');

%% ESPAÇO DE ESTADOS DISCRETO
% Tempo de amostragem para o sistema discreto
Ts_d = 0.005;
Ts_d1 = 0.01;
Ts_d2 = 0.1;
T_fim = 2;

% Criação dos vetores de tempo para cada Ts
t_d = 0:Ts_d:T_fim;
t_d1 = 0:Ts_d1:T_fim;
t_d2 = 0:Ts_d2:T_fim;

% Discretização usando diferentes Ts
sys_zoh = c2d(sys_cl, Ts_d, 'tustin');
sys_zoh1 = c2d(sys_cl, Ts_d1, 'tustin');
sys_zoh2 = c2d(sys_cl, Ts_d2, 'tustin');

% Sinal de impulso discretizado para cada Ts
for n = 1:length(t_d)
    r_step_d(n) = 25;
end
for n = 1:length(t_d1)
    r_step_d1(n) = 25;
end
for n = 1:length(t_d2)

```

```

    r_step_d2(n) = 25;
end

% Simulação para diferentes tempos de amostragem
[y_step_zoh, t_step_zoh] = lsim(sys_zoh, r_step_d,
t_d);
[y_step_zoh1, t_step_zoh1] = lsim(sys_zoh1,
r_step_d1, t_d1);
[y_step_zoh2, t_step_zoh2] = lsim(sys_zoh2,
r_step_d2, t_d2);

y_deg2_interp = interp1(t_deg2, y_deg2,
t_step_zoh, 'linear', 'extrap');
y_deg2_interp1 = interp1(t_deg2, y_deg2,
t_step_zoh1, 'linear', 'extrap');
y_deg2_interp2 = interp1(t_deg2, y_deg2,
t_step_zoh2, 'linear', 'extrap');

% Calcular o erro médio absoluto (EMA) e erro
percentual médio (EPM)
Erro_zoh = mean(abs(y_deg2_interp - y_step_zoh));
Erro_zoh1 = mean(abs(y_deg2_interp1 -
y_step_zoh1));
Erro_zoh2 = mean(abs(y_deg2_interp2 -
y_step_zoh2));

Erro_percentual_zoh = (Erro_zoh /
mean(abs(y_deg2_interp))) * 100;
Erro_percentual_zoh1 = (Erro_zoh1 /
mean(abs(y_deg2_interp1))) * 100;
Erro_percentual_zoh2 = (Erro_zoh2 /
mean(abs(y_deg2_interp2))) * 100;

% Plot
figure
plot(t_deg2, y_deg2, 'b-', 'DisplayName', 'Resposta
Continua ao Degrau');
hold on

```

```

plot(t_step_zoh, y_step_zoh, 'm--', 'DisplayName',
sprintf('Ts = %.3f s, Erro = %.2f%%', Ts_d,
Erro_percentual_zoh));
plot(t_step_zoh1, y_step_zoh1, 'g-', 'DisplayName',
sprintf('Ts = %.3f s, Erro = %.2f%%', Ts_d1,
Erro_percentual_zoh1));
plot(t_step_zoh2, y_step_zoh2, 'r:', 'DisplayName',
sprintf('Ts = %.3f s, Erro = %.2f%%', Ts_d2,
Erro_percentual_zoh2));
title('Resposta ao Degrau em Malha Fechada')
xlabel('Tempo (s)')
ylabel('Posição Angular (°)')
legend('Location', 'northeast')
grid on
set(gcf, 'Color', 'white');

[AA, BB, CC, DD] = ssdata(sys_zoh1);
KK = place(AA, BB, polos_desejados);
kk1 = [K(1, 1); K(2, 1)];

%% Dados Teoricos

filename = 'impulso.txt';
data = readmatrix(filename);

tempo = data(:,1);
tempo = tempo - 5;
angulo = data(:,4);
imp_4 = data(:,2);
imp_4 = imp_4 - 1300;
imp_2 = data(:,3);
imp_2 = imp_2 - 1300;
U = [imp_2 imp_4];

% Plot dos dados - impulso
figure;
plot(tempo, angulo);
grid on;
title('Gráfico de Tempo vs. Ângulo');
xlabel('Tempo (s)');

```

```

ylabel('Posição Angular (°)');
axis tight
set(gcf, 'Color', 'white');

data_imp = iddata(angulo, U, Ts_d1);
figure
compare(data_imp, sys_zoh1)
title('Comparação: Dados Experimentais vs.
Resposta Teórica');
legend('Dados Experimentais', 'Resposta Teórica');
xlabel('Tempo (s)');
ylabel('Posição Angular (°)');
grid on;
set(gcf, 'Color', 'white');

filename = 'degrau.txt';
data = readmatrix(filename);

tempo = data(:,1);
tempo = tempo - 5;
r = data(:,2);
deg_4 = data(:,3);
deg_4 = deg_4 - 1300;
deg_2 = data(:,4);
deg_2 = deg_2 - 1300;
angulo_deg = data(:,5);
U2 = [deg_2 deg_4];

stepinfo(angulo_deg, tempo)

% Plot dos dados - sinal r
figure;
plot(tempo, r);
grid on;
title('Gráfico de Tempo vs. Ângulo');
xlabel('Tempo (s)');
ylabel('Posição Angular (°)');
ylim([-5 30]);
xlim([0 60]);
set(gcf, 'Color', 'white');

% Plot dos dados - degrau
figure;
plot(tempo, angulo_deg);
grid on;
title('Gráfico de Tempo vs. Ângulo');
xlabel('Tempo (s)');
ylabel('Posição Angular (°)');
axis tight;
set(gcf, 'Color', 'white');

tempo = 0:0.01:60;
resposta_teorica = lsim(sys_zoh1, r, tempo);

figure;
plot(tempo, angulo_deg, 'DisplayName', 'Dados
Experimentais');
hold on;
plot(tempo, resposta_teorica, 'DisplayName',
'Resposta Teórica');
grid on;
title('Comparação: Dados Experimentais vs.
Resposta Teórica');
xlabel('Tempo (s)');
ylabel('Posição Angular (°)');
legend('show');
axis tight;
set(gcf, 'Color', 'white');

% Criar objeto iddata
data_experimental = iddata(angulo_deg, [],
tempo(2) - tempo(1));
modelo_id = iddata(resposta_teorica, [], tempo(2) -
tempo(1));

% Comparar modelo teórico com os dados reais
figure;
compare(data_experimental, modelo_id);
title('Comparação: Dados Experimentais vs.
Resposta Teórica');

```

```
legend('Dados Experimentais', 'Resposta Teórica');  
xlabel('Tempo (s)');  
ylabel('Posição Angular (°)');  
grid on;  
set(gcf, 'Color', 'white');
```

## APÊNDICE B - Código do Arduino

## Código do Arduino

```

#include <Servo.h>
#include <Wire.h>

Servo brushless_4;
Servo brushless_2;

int16_t Acc_rawX, Acc_rawY, Acc_rawZ,
Gyr_rawX, Gyr_rawY, Gyr_rawZ;

float Acceleration_angle[2];
float Gyro_angle[2];
float Total_angle[2];

float time, elapsedTime, timePrev;
float rad_to_deg = 180/3.141592654;

/*float PID, PWM_4, PWM_2, erro, erro_anterior;
float pid_p = 0;
float pid_i = 0;
float pid_d = 0;

double kp = 3.5;//3.55
double ki = 0.001;//0.003
double kd = 2.3;//2.05*/

float PWM_4, PWM_2;
float PWM_base = 1300;
double PWM = 1000;

int r = 0;

void monitor_serial()
{
    float timeInSeconds = time / 1000.0;

    Serial.print(timeInSeconds);

    Serial.print("\t");
    Serial.print(r);
    Serial.print("\t");
    Serial.print(PWM_4);
    Serial.print("\t");
    Serial.print(PWM_2);
    Serial.print("\t");
    Serial.print(Gyro_angle[1]);
    Serial.print("\t");
    Serial.println(Total_angle[1]);
}

/*void controlador_PID()
{
    erro = Total_angle[1] - setpoint;

    pid_p = kp * erro;
    pid_i = pid_i + (ki * erro);
    pid_d = kd * ((erro - erro_anterior) / elapsedTime);

    erro_anterior = erro;

    PID = pid_p + pid_i + pid_d;

    PWM_E = PWM - PID;
    PWM_D = PWM + PID;

    if(PWM_E < 1200) PWM_E = 1200;
    if(PWM_E > 2000) PWM_E = 2000;

    if(PWM_D < 1200) PWM_D = 1200;
    if(PWM_D > 2000) PWM_D = 2000;
}*/

void sinal()
{

```

```

float timeInSeconds = time / 1000.0;

if(0.00 <= timeInSeconds & timeInSeconds <=
07.00) r = 0;
if(07.00 < timeInSeconds & timeInSeconds <=
15.00) r = 25;
if(15.00 < timeInSeconds & timeInSeconds <=
25.00) r = 0;
if(25.00 < timeInSeconds & timeInSeconds <=
30.00) r = 25;
if(30.00 < timeInSeconds & timeInSeconds <=
40.00) r = 0;
if(40.00 < timeInSeconds & timeInSeconds <=
45.00) r = 25;
if(45.00 < timeInSeconds & timeInSeconds <=
55.00) r = 0;
if(55.00 < timeInSeconds & timeInSeconds <=
60.00) r = 25;

}

void espacio_de_estados()
{
  sinal();
  //r = 0;
  //PWM_4 = PWM_base + (-1.7965 *
Total_angle[1] - 0.1839 * Gyro_angle[1]) + 1.7965
* r; // Ts = 0.3;
  //PWM_2 = PWM_base + ( 2.0849 *
Total_angle[1] + 0.2134 * Gyro_angle[1]) - 2.0849
* r; // Mp = 0.01;

  //PWM_4 = PWM_base + (-3.5081 *
Total_angle[1] - 0.1839 * Gyro_angle[1]) + 3.5081
* r; // Ts = 0.3;
  //PWM_2 = PWM_base + ( 4.0714 *
Total_angle[1] + 0.2134 * Gyro_angle[1]) - 4.0714
* r; // Mp = 0.1;

  //PWM_4 = PWM_base + (-1.2629 *
Total_angle[1] - 0.1103 * Gyro_angle[1]) + 1.2629
* r; // Ts = 0.5;
  //PWM_2 = PWM_base + ( 1.0178 *
Total_angle[1] + 0.1281 * Gyro_angle[1]) - 1.0178
* r; // Mp = 0.1;

  //PWM_4 = PWM_base + (-0.8770 *
Total_angle[1] - 0.0919 * Gyro_angle[1]) + 0.8770
* r; // Ts = 0.6;
  //PWM_2 = PWM_base + ( 1.0178 *
Total_angle[1] + 0.1281 * Gyro_angle[1]) - 1.0178
* r; // Mp = 0.1;

  PWM_4 = PWM_base + (-0.6467 * Total_angle[1]
- 0.1103 * Gyro_angle[1]) + 0.6467 * r; // Ts = 0.5;
  PWM_2 = PWM_base + ( 0.7506 * Total_angle[1]
+ 0.7506 * Gyro_angle[1]) - 1.4657 * r; // Mp = 0.01;

  //PWM_4 = PWM_base + (-0.3157 *
Total_angle[1] - 0.0552 * Gyro_angle[1]) + 0.3157
* r; // Ts = 1;
  //PWM_2 = PWM_base + ( 0.3664 *
Total_angle[1] + 0.0640 * Gyro_angle[1]) - 1.4657
* r; // Mp = 0.01;

  if (PWM_4 < 1200) PWM_4 = 1200;
  if (PWM_4 > 1400) PWM_4 = 1400;

  if (PWM_2 < 1200) PWM_2 = 1200;
  if (PWM_2 > 1400) PWM_2 = 1400;
}

void angulo()
{
  Wire.beginTransmission(0x68);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(0x68, 6, true);
}

```

```

Acc_rawX=Wire.read()<<8|Wire.read();
Acc_rawY=Wire.read()<<8|Wire.read();
Acc_rawZ=Wire.read()<<8|Wire.read();

Acceleration_angle[0] = atan((Acc_rawY/16384.0)/sqrt(pow((Acc_rawX/16384.0),2) + pow((Acc_rawZ/16384.0),2)))*rad_to_deg;
Acceleration_angle[1] = atan(-1*(Acc_rawX/16384.0)/sqrt(pow((Acc_rawY/16384.0),2) + pow((Acc_rawZ/16384.0),2)))*rad_to_deg;

Wire.beginTransmission(0x68);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(0x68,4,true);

Gyr_rawX=Wire.read()<<8|Wire.read();
Gyr_rawY=Wire.read()<<8|Wire.read();

Gyro_angle[0] = Gyr_rawX/131.0;
Gyro_angle[1] = Gyr_rawY/131.0;

timePrev = time;
time = millis();
elapsedTime = (time - timePrev) / 1000;

Total_angle[0] = 0.98 *(Total_angle[0] + Gyro_angle[0]*elapsedTime) + 0.02*Acceleration_angle[0];
Total_angle[1] = 0.98 *(Total_angle[1] + Gyro_angle[1]*elapsedTime) + 0.02*Acceleration_angle[1];
}

void setup()
{
Wire.begin();
Wire.beginTransmission(0x68);
Wire.write(0x6B);
Wire.write(0);
Wire.endTransmission(true);

Serial.begin(250000);
brushless_4.attach(3);
brushless_2.attach(5);
brushless_4.write(PWM);
brushless_2.write(PWM);

delay(5000);
time = millis();
}

unsigned long previousMillis = 0;
float Ts = 10;

void loop()
{
unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= Ts)
{
previousMillis = currentMillis;

angulo();
espaco_de_estados();
monitor_serial();
brushless_4.write(PWM_4);
brushless_2.write(PWM_2);
}
}

void setup()
{
Wire.begin();
Wire.beginTransmission(0x68);

```