

Mateus Soares Marques

**Projeto de um sistema de controle de atitude e  
estimação de orientação para quadrotoros com  
abordagem não-linear no  $SO(3)$**

Campina Grande, Paraíba

Junho de 2024

Mateus Soares Marques

**Projeto de um sistema de controle de atitude e estimação  
de orientação para quadrotoros com abordagem não-linear  
no  $SO(3)$**

Trabalho de Conclusão de Curso submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharelado em Ciências no Domínio da Engenharia Elétrica.

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Departamento de Engenharia Elétrica – DEE

Orientador: Antonio Marcus Nogueira Lima, D. Sc.

Campina Grande, Paraíba

Junho de 2024

Mateus Soares Marques

**Projeto de um sistema de controle de atitude e estimação  
de orientação para quadrotoros com abordagem não-linear  
no  $SO(3)$**

Trabalho de Conclusão de Curso submetido  
à Coordenação de Graduação em Engenharia  
Elétrica da Universidade Federal de Campina  
Grande, Campus Campina Grande, como  
parte dos requisitos necessários para a ob-  
tenção do grau de Bacharelado em Ciências  
no Domínio da Engenharia Elétrica.

Trabalho aprovado. Campina Grande, Paraíba, 07 de Junho de 2024:

---

**Antonio Marcus Nogueira Lima, D. Sc.**  
Orientador

---

**Marcos Ricardo Alcântara Morais, D.  
Sc.**  
Convidado

Campina Grande, Paraíba  
Junho de 2024

*Dedico este trabalho aos meus pais, Marcondes e Adriana.*

# Agradecimentos

Agradeço os meus pais pela paciência e pelo apoio moral e financeiro, sem os quais não estaria concluindo a minha graduação. Ao meu irmão Gabriel pela companhia e pelas risadas de todo dia.

Agradeço a minha namorada, Bibi, e sua família, Seu Carlos, Bia e Isadora, pelo acolhimento e pelos almoços e conversas.

Agradeço os meus amigos do NSC, pela fé que sempre tiveram no meu potencial.

Agradeço os meus amigos de curso Icaro e William pelas ideias trocadas e o incentivo mútuo.

Agradeço os meus amigos do 2015.1 Augusto e Pedro Saulo, com os quais passei ótimos momentos e enfrentei os desafios dessa graduação.

Agradeço os meus professores do departamento, em especial o professor Antonio Marcus pela oportunidade e pela orientação concedida.

*“A simplicidade é a sofisticação suprema.”*  
*(Leonardo da Vinci)*

# Resumo

Este trabalho apresenta a aplicação de técnicas não-lineares ao problema de controle de rotação e estimação de orientação de quadrópteros, com foco na representação da rotação por elementos do  $SO(3)$  (Grupo Ortogonal Especial de Terceira Ordem) e formulação do problema usando o Teorema de Lyapunov para a estabilidade de sistemas não-lineares aplicado à problemas de controle geométrico. Um modelo combinado de estimador, controlador de atitude e controlador de posição é proposto e implementado no ambiente de simulação computacional Simulink, onde os resultados foram produzidos e analisados. Os erros de estimação e de rastreamento de trajetória observados no sistema implementado mostram que o uso de técnicas baseadas na geometria da rotação proporcionam robustez e rápida convergência do sistema ao equilíbrio.

**Palavras-chave:** Quadrotores, Controle geométrico, Observadores Não-lineares de Estado, Modelagem de Sistemas Mecânicos, Simulação Computacional.

# Abstract

This work presents the application of nonlinear techniques to the problem of rotational control and orientation estimation in quadcopters, focusing on the representation of rotation using elements from  $SO(3)$  (Special Orthogonal Group of Third Order) and formulating the problem using Lyapunov's Theorem for the stability of nonlinear systems applied to geometric control problems. A combined model of estimator, attitude controller, and position controller is proposed and implemented in the Simulink computational simulation environment, where results were produced and analyzed. The estimation errors and trajectory tracking observed in the implemented system show that the use of techniques based on the geometry of rotation provide robustness and rapid convergence of the system to equilibrium.

**Keywords:** Quadrotors, Geometric Control, Non-linear State Observers, Mechanical Systems Modeling, Computational Simulation.

# Lista de ilustrações

Figura 1 – Sistemas de coordenadas inercial e do quadrotor em voo. . . . .	20
Figura 2 – <i>Rotor flapping</i> . . . . .	26
Figura 3 – Modelo de projeção central. . . . .	30
Figura 4 – Modelo de projeção central com plano de imagem e pixels discretos. . .	32
Figura 5 – Diagrama de blocos do estimador de atitude no Simulink. . . . .	44
Figura 6 – Estimativa da orientação frente aos estados. . . . .	44
Figura 7 – Estimativa da velocidade angular frente aos estados. . . . .	45
Figura 8 – Erros de estimativa . . . . .	45
Figura 9 – Diagrama de blocos do controlador de atitude. . . . .	47
Figura 10 – Orientação do drone com o controlador projetado. . . . .	48
Figura 11 – Velocidade angular do drone com o controlador projetado. . . . .	48
Figura 12 – Erros de orientação e velocidade . . . . .	49
Figura 13 – Sinal de controle produzido pelo controlador. . . . .	49
Figura 14 – Empuxo comandado por propulsor. . . . .	50
Figura 15 – Sinal de comando para os motores. . . . .	50
Figura 16 – Diagrama de blocos do controle de posição. . . . .	52
Figura 17 – Controlador de pose completo. . . . .	53
Figura 18 – Trajetória de referência. . . . .	55
Figura 19 – Diagrama de blocos do planejamento de trajetória. . . . .	55
Figura 20 – Diagrama de blocos da função geradora de caminho. . . . .	56
Figura 21 – Posição do quadrotor no tempo. . . . .	56
Figura 22 – Comandos de orientação gerados. . . . .	57
Figura 23 – Erros de pose. . . . .	57
Figura 24 – Posição do quadrotor no espaço. . . . .	58

# Lista de abreviaturas e siglas

VANT	<i>Veículo Aéreo Não Tripulado</i>
ESC	<i>Electronic Speed Controller</i>
IMU	<i>Inertial Measurement Unit</i>
GPS	<i>Global Positioning System</i>

# Lista de símbolos

$\{B\}$	Sistemas de coordenadas do quadrotor
$\vec{i}_B$	Base ortogonal na direção $x$ de $\{B\}$
$\vec{j}_B$	Base ortogonal na direção $y$ de $\{B\}$
$\vec{k}_B$	Base ortogonal na direção $z$ de $\{B\}$
$\{I\}$	Sistemas de coordenadas do inercial
$\vec{i}_I$	Base ortogonal na direção $x$ de $\{B\}$
$\vec{j}_I$	Base ortogonal na direção $y$ de $\{B\}$
$\vec{k}_I$	Base ortogonal na direção $z$ de $\{B\}$
$p$	Velocidade angular do quadrotor em torno de $\vec{i}_B$
$q$	Velocidade angular do quadrotor em torno de $\vec{j}_B$
$r$	Velocidade angular do quadrotor em torno de $\vec{k}_B$
$\omega$	Velocidade angular do quadrotor expressa no referencial $\{B\}$
$\vec{F}_i$	Força gerada pelo propulsor $i$
$M_i$	Torque reativo do propulsor $i$
$J$	Matriz de inércia do quadrotor
$m$	Massa do quadrotor
$\vec{g}$	Aceleração da gravidade expressa no referencial $\{I\}$
$R$	Matriz de rotação do referencial $\{B\}$ para o referencial $\{I\}$
$SO(3)$	Grupo Ortogonal Especial de Terceira Ordem
$\mathbf{p}$	Posição do centro de massa do quadrotor no referencial $\{I\}$
$\mathbb{R}^m$	Espaço cartesiano de ordem $m$
$\mathbf{v}$	Velocidade linear do drone no referencial $\{I\}$
$SE(3)$	Grupo Euclidiano Especial de Terceira Ordem

$\xi$	Pose do referencial $\{B\}$ em relação ao referencial $\{I\}$
$\dot{\xi}$	Derivada da pose $\xi$
$\dot{R}$	Derivada da rotação $R$
$\dot{\mathbf{p}}$	Derivada da posição $\mathbf{p}$
$V_I$	Velocidade total do quadrotor representada em $\{I\}$
$V_B$	Velocidade total do quadrotor representada em $\{B\}$
$[\cdot]_{\times}$	Mapeamento anti-simétrico de um vetor
$(\cdot)^{\vee}$	Mapeamento de uma matriz anti-simétrica em um vetor
$\omega_I$	Velocidade angular do quadrotor representada no referencial $\{I\}$
$\mathbf{v}_B$	Velocidade linear do drone representada no referencial $\{B\}$
$\mathfrak{g}(n)$	Álgebra de Lie de ordem $n$
$G(n)$	Grupo de Lie de ordem $n$
$\mathfrak{so}(3)$	Álgebra de Lie do $SO(3)$
$\mathfrak{se}(3)$	Álgebra de Lie do $SE(3)$
$Ad_{\xi}$	Transformação adjunta na pose $\xi$
$\vec{F}$	Força total aplicada sobre o drone representada em $\{B\}$
$M$	Torque total aplicado sobre o drone representado no referencial $\{B\}$
$\mathbf{r}_i^B$	Posição do propulsor $i$ em relação ao referencial $\{B\}$
$\dot{\mathbf{v}}_B$	Aceleração linear do drone expressa no referencial $\{B\}$
$\dot{\mathbf{v}}_I$	Aceleração linear do drone expressa no referencial $\{B\}$
$\dot{V}_B$	Aceleração total do drone expressa no referencial $\{B\}$
$\dot{V}_I$	Aceleração total do drone expressa no referencial $\{I\}$
$\mathbf{v}$	Velocidade linear do drone representada no referencial $\{I\}$
$\tau_{i_B}$	Torque aplicado no quadrotor em torno do eixo $\vec{i}_B$
$\tau_{j_B}$	Torque aplicado no quadrotor em torno do eixo $\vec{j}_B$
$\tau_{k_B}$	Torque aplicado no quadrotor em torno do eixo $\vec{k}_B$

$F$	Norma da força total aplicada sobre o drone
$C_T$	Coefficiente de empuxo da hélice
$\rho$	Densidade do ar
$A_{rp}$	Área de cobertura da hélice
$r_p$	Raio da hélice
$\bar{\omega}_i$	Velocidade angular da hélice $i$
$c_T$	Constante de empuxo generalizada
$c_M$	Constante de torque reativo generalizada
$\mathbf{u}$	Entrada de controle do sistema
$\Gamma$	Mapeamento das velocidades de hélice para $\mathbf{u}$
$l$	Distância do propulsor ao centro de massa do quadrotor
$R_d$	Referência de rotação
$G$	Matriz de pesos do erro de orientação
$e_R$	Vetor erro de orientação
$e_\omega$	Vetor erro de velocidade angular
$\omega_d$	Referência de velocidade angular
$c$	Peso do erro de configuração total
$k_R$	Ganho de orientação do controlador
$k_\omega$	Ganho de velocidade do controlador
$k_J$	Ganho do estimador de inércia
$\delta$	Amplitude da perturbação de torque
$\epsilon$	Ganho de configuração do controle robusto
$\sigma$	Ganho de estimação da inércia do controlador robusto
$\eta$	Ruído de medição dos sensores da IMU
$\omega_y$	Velocidade angular medida pela IMU
$\mathbf{b}$	Viés constante da IMU

$\mathbf{a}$	Aceleração linear medida pela IMU
$R_{\vec{i}}$	Rotação pura em torno de $\vec{i}_I$
$R_{\vec{j}}$	Rotação pura em torno de $\vec{j}_I$
$R_{\vec{k}}$	Rotação pura em torno de $\vec{k}_I$
$\mathbf{a}_{\vec{i}}$	Aceleração medida na direção de $\vec{i}_B$
$\mathbf{a}_{\vec{j}}$	Aceleração medida na direção de $\vec{j}_B$
$\mathbf{a}_{\vec{k}}$	Aceleração medida na direção de $\vec{k}_B$
$R_y$	Rotação reconstruída à partir das leituras da IMU
$\hat{R}$	Rotação estimada
$\hat{\omega}$	Velocidade angular estimada
$\hat{\mathbf{b}}$	Viés estimado
$k_P$	Ganho proporcional do estimador
$k_I$	Ganho integral do estimador
$\omega_c$	Termo corretivo do estimador
$\tilde{R}$	Erro de rotação do estimador
$\Phi$	Métrica da distância entre duas matrizes de rotação
$e_p$	Erro de posição do drone
$e_v$	Erro de velocidade linear do drone
$\mathbf{p}_d$	Posição desejada no referencial $\{I\}$
$\mathbf{v}_d$	Velocidade desejada no referencial $\{I\}$
$\vec{F}_d$	Força desejada no referencial $\{I\}$
$\kappa_p$	Ganho proporcional do controlador de posição
$\kappa_i$	Ganho integral do controlador de posição
$\kappa_r$	Ganho do planejador de atitude
$R_p$	Rotação planejada

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>Objetivos</b>	<b>17</b>
1.1.1	Objetivos Específicos	17
1.1.2	Estrutura do trabalho	17
<b>2</b>	<b>MODELAGEM DO QUADROTOR</b>	<b>19</b>
<b>2.1</b>	<b>Modelo dinâmico</b>	<b>19</b>
2.1.1	Cinemática de corpo rígido	20
2.1.2	Dinâmica de corpo rígido	22
2.1.3	Dinâmica do quadrotor	23
<b>2.2</b>	<b>Atuadores</b>	<b>24</b>
2.2.1	Aerodinâmica dominante	24
2.2.2	Efeitos aerodinâmicos adicionais	25
<b>2.3</b>	<b>Sensores</b>	<b>27</b>
2.3.1	IMU - <i>Inertial Measurement Unit</i>	28
2.3.1.1	Giroscópio	28
2.3.1.2	Acelerômetro	29
2.3.2	Medição de Altitude	29
2.3.2.1	Barômetro	29
2.3.2.2	Sonar	29
2.3.3	Câmera e odometria visual	29
2.3.3.1	Modelo de Imagem de Projeção Central	30
2.3.3.2	Cálculo do fluxo óptico	33
2.3.3.3	Relação entre fluxo óptico e velocidades da câmera	34
<b>3</b>	<b>CONTROLE DE ATITUDE</b>	<b>37</b>
<b>3.1</b>	<b>Controle adaptativo</b>	<b>38</b>
<b>3.2</b>	<b>Controle Robusto</b>	<b>39</b>
<b>4</b>	<b>OBSERVADOR DE ATITUDE</b>	<b>40</b>
<b>4.1</b>	<b>Medidas</b>	<b>40</b>
4.1.1	Giroscópio	40
4.1.2	Acelerômetro	40
<b>4.2</b>	<b>Filtro complementar passivo com correção de viés</b>	<b>41</b>
4.2.1	Análise da estabilidade	42

<b>5</b>	<b>IMPLEMENTAÇÃO E SIMULAÇÃO</b>	<b>43</b>
<b>5.1</b>	<b>Observador</b>	<b>43</b>
<b>5.2</b>	<b>Controle</b>	<b>46</b>
<b>5.3</b>	<b>Seguimento de caminho</b>	<b>51</b>
<b>5.4</b>	<b>Análise dos resultados</b>	<b>58</b>
5.4.1	Observador	58
5.4.2	Controle	59
5.4.3	Seguimento de caminho	59
<b>6</b>	<b>CONCLUSÃO</b>	<b>60</b>
<b>6.1</b>	<b>Trabalhos futuros</b>	<b>61</b>
<b>A</b>	<b>ANÁLISE DA ESTABILIDADE</b>	<b>62</b>
<b>A.1</b>	<b>Controle de atitude</b>	<b>62</b>
A.1.1	Erro de configuração	62
A.1.2	Estabilidade do controle adaptativo	64
A.1.2.1	Função candidata de Lyapunov	64
A.1.2.2	Análise da estabilidade	67
	<b>REFERÊNCIAS</b>	<b>78</b>

# 1 Introdução

Nos últimos anos, os quadrotoros têm emergido como uma tecnologia promissora com aplicações em diversas áreas, incluindo vigilância, mapeamento de terrenos, entrega de pacotes e entretenimento. Esses veículos aéreos não tripulados (VANTs) apresentam uma agilidade e versatilidade únicas, o que os torna especialmente adequados para tarefas que exigem manobras complexas e acesso a áreas de difícil alcance. No entanto, a realização eficaz dessas tarefas depende crucialmente do desenvolvimento de sistemas de controle de voo robustos e precisos.

O controle de voo de um quadrotor apresenta desafios significativos devido à sua natureza não linear, além das limitações impostas por sensores e atuadores disponíveis. A estabilidade e a capacidade de resposta do drone são influenciadas por uma série de fatores, como ventos turbulentos, variações de carga, efeitos aerodinâmicos não lineares e incertezas no modelo do sistema. Portanto, projetar um controlador eficiente que possa lidar com essas complexidades é fundamental para garantir o desempenho adequado do quadrotor em diversas condições operacionais.

Este trabalho propõe um sistema de controle de voo para drones, com foco na abordagem estritamente não-linear na solução. Foram considerados aspectos teóricos e práticos, incluindo modelagem dinâmica, controle de atitude e estimação de orientação. Além disso foram realizadas simulações para ilustrar a eficácia das abordagens propostas e suas aplicações em um cenário de seguimento de trajetória.

Ao compreender os desafios inerentes ao controle de voo de quadrotoros e as estratégias disponíveis para enfrentá-los, espera-se que este trabalho possa contribuir para o avanço contínuo desta tecnologia, ampliando suas possibilidades de aplicação e melhorando sua segurança e desempenho.

## 1.1 Objetivos

Os objetivos deste trabalho são o estudo do controle geométrico e observadores não-lineares no  $SO(3)$  aplicados ao contexto de controle de vôo de quadrotores e a implementação dessas técnicas em ambiente de simulação.

### 1.1.1 Objetivos Específicos

Os objetivos específicos definidos foram:

1. Revisão da literatura escolhida sobre controladores e observadores não lineares no contexto de quadrotores;
2. Modelagem e simulação computacional do sistema dinâmico do quadrotor;
3. Projeto do controlador de voo;
4. Análise da estabilidade do controlador;
5. Projeto do observador de estados;
6. Análise da estabilidade do observador;
7. Implementação do controlador e do observador em ambiente de simulação;
8. Implementação do controlador de posição;
9. Simulação do sistema para referências de orientação e de posição;
10. Análise da convergência do sistema na estimação e no rastreamento de trajetória.

### 1.1.2 Estrutura do trabalho

O trabalho está dividido em 6 capítulos. O Capítulo 2 apresenta a mecânica de corpo rígido do quadrotor representada no  $SO(3)$  através de matrizes de rotação. Ainda no Capítulo 2 discute-se a modelagem dos efeitos aerodinâmicos presentes no vôo de quadrotores bem como o modelo usado para simular os sensores embarcados no drone, em específico os sensores presentes no Parrot Mambo, cujo o pacote de simulação para Simulink ([MathWorks, 2023](#)) foi usado para simular as técnicas de controle e estimação de estados desenvolvidas nesse trabalho.

Nos Capítulos 3 e 4 são apresentados o controlador e o observador estudados. Nesses capítulos os principais resultados são comentados e a estrutura geral da abordagem não-linear no  $SO(3)$  é trazida sob forma de equações.

No Capítulo 5 é explicada a implementação do controle e do estimador no ambiente de simulação computacional Simulink, onde usou-se o modelo de simulação do Parrot

Mambo para Simulink para testar as leis formuladas nos capítulos 3 e 4. Ainda no Capítulo 5 implementa-se um controle de posição para testar o funcionamento do controlador e do observador no contexto de seguimento de uma curva no espaço. Ao final do capítulo os resultados obtidos das simulações são analisados.

O Capítulo 6 conclui o trabalho com uma análise geral do conteúdo apresentado nos demais capítulos e sugestões de melhorias e estudos futuros são apresentadas.

No Apêndice A constam as deduções das equações e os resultados matemáticos importantes para compreensão do controle geométrico de sistemas mecânicos.

## 2 Modelagem do quadrotor

O quadrotor é um robô móvel aéreo, com quatro propulsores dispostos em X que permitem estabilizar e movimentar o drone no ar. Do ponto de vista de controle, o drone quadrotor é um sistema sub-atuado, uma vez que possui seis graus de liberdade mas apenas quatro atuadores, de tal maneira que o controle de posição e orientação são acoplados. Assim, para controlar a trajetória do drone no espaço, controla-se sua orientação a fim de direcionar a propulsão total produzida e o torque total aplicado sobre sua estrutura.

O sistema de propulsão do drone é feito por quatro pares motores + hélice. A dinâmica do drone tem por entrada os torques e forças gerados por cada um de seus propulsores através dos efeitos aerodinâmicos associados a rotação das hélices, de modo que as entradas do sistema são produzidas pelo controle da velocidade angular individual de cada propulsor.

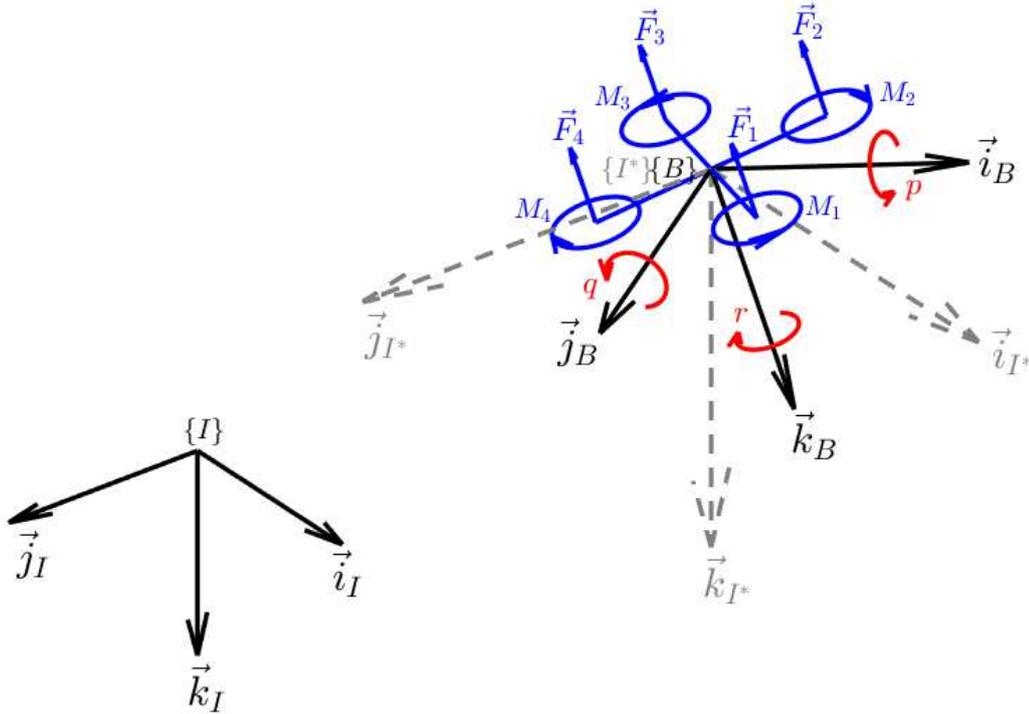
Dadas as considerações dos dois últimos parágrafos, pode-se abstrair o sistema dinâmico do quadrotor como um corpo rígido com seis graus de liberdade cuja a pose é controlada pelo torque e pela força total aplicada. Uma vez que as leis de controle para as entradas força e torque tenham sido encontradas, por meio do entendimento da aerodinâmica do propulsor pode-se obter as velocidades de cada propulsor que conjuntamente permitem a realização do controle de pose.

Nas seções seguintes os modelos analíticos das plantas considerados no projeto do controle de voo serão apresentados, tendo por referência o drone Parrot Mambo e o pacote de suporte à drones Parrot para Simulink® ([MathWorks, 2023](#)), onde o projeto foi desenvolvido e testado seguindo o detalhamento trazido por ([OLIVEIRA, 2020](#)) para uso e otimização de projetos baseados em modelo no contexto da toolbox mencionada.

### 2.1 Modelo dinâmico

A Figura 1 ilustra o quadrotor no espaço. O sistema de coordenadas fixado ao quadrotor é  $\{B\}$  de bases ortogonais  $\vec{i}_B, \vec{j}_B$  e  $\vec{k}_B$ . O sistema de coordenadas inercial é  $\{I\}$  de bases ortogonais  $\vec{i}_I, \vec{j}_I$  e  $\vec{k}_I$ . O sistema de coordenadas  $\{I^*\}$  (de bases  $\vec{i}_{I^*}, \vec{j}_{I^*}$  e  $\vec{k}_{I^*}$ ) representa a translação de  $\{I\}$  para a posição do drone no sistema em coordenadas inercial.  $p, q$  e  $r$  são as velocidades angulares do quadrotor no referencial  $\{B\}$ , sendo o vetor da velocidade angular dado por  $\omega = [p \ q \ r]^T \in \mathbb{R}^3$ . Os vetores  $\vec{F}_i \in \mathbb{R}^3, 1 \leq i \leq 4$  representam os empuxos gerados por cada um dos quatro propulsores do veículo (em coordenadas de  $\{B\}$ ); já as grandezas  $M_i \in \mathbb{R}^3, 1 \leq i \leq 4$  representam os torques gerados por cada propulsor sobre a estrutura do drone (em coordenadas de  $\{B\}$ ).

Figura 1 – Sistemas de coordenadas inercial e do quadrotor em voo.



Fonte: Autoria própria.

Das variáveis não representadas na Figura 1 temos a inércia total do quadrotor em  $\{B\}$ , representada por  $J$ , a massa total do quadrotor representada por  $m$ , a aceleração da gravidade dada por  $\vec{g} = [0 \ 0 \ g]^T$  no sistema de coordenadas  $\{I\}$ , a matriz de rotação  $R \in SO(3)$  de  $\{I^*\}$  para  $\{B\}$ , a posição  $\mathbf{p} \in \mathbb{R}^3$  de  $\{B\}$  em coordenadas de  $\{I\}$ , bem como a velocidade linear  $\mathbf{v} \in \mathbb{R}^3$  de  $\{B\}$  em relação a  $\{I\}$ .

### 2.1.1 Cinemática de corpo rígido

A pose  $\{B\}$  com relação a  $\{I\}$  pode ser dada por  $\xi \in SE(3)$ :

$$\xi = \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2.1)$$

Seja  $\dot{\xi}$  a derivada da pose no tempo. Podemos escrever este termo como (MURRAY; LI; SASTRY, 1994)

$$\dot{\xi} = [V_I]_{\times} \xi = \xi [V_B]_{\times} = \begin{bmatrix} \dot{R} & \dot{\mathbf{p}} \\ 0_{1 \times 3} & 0 \end{bmatrix} \quad (2.2)$$

onde  $V_I, V_B \in \mathbb{R}^6$  representam as velocidades do drone em  $\{I\}$  e  $\{B\}$ , respectivamente

$$V_I = \begin{bmatrix} \mathbf{v}_I \\ \omega_I \end{bmatrix} = \begin{bmatrix} -\dot{R}R^T \mathbf{p} + \dot{\mathbf{p}} \\ (\dot{R}R^T)^\vee \end{bmatrix} \quad (2.3)$$

$$V_B = \begin{bmatrix} \mathbf{v}_B \\ \omega \end{bmatrix} = \begin{bmatrix} R^T \dot{\mathbf{p}} \\ (R^T \dot{R})^\vee \end{bmatrix} \quad (2.4)$$

onde  $\dot{\mathbf{p}}$  é a derivada no tempo do vetor posição  $\mathbf{p}$ .

Os operadores  $[\cdot]_\times$  e  $(\cdot)^\vee$  representam os mapas  $\text{hat} : \mathbb{R}^m \rightarrow \mathfrak{g}(n)$  e  $\text{vex} : \mathfrak{g}(n) \rightarrow \mathbb{R}^m$ , respectivamente, onde  $\mathfrak{g}(n)$  é a álgebra de Lie associada ao grupo  $G(n)$ . Os grupos de Lie para os quais se explicitará os mapas mencionados são o grupo ortogonal especial de terceira ordem ( $SO(3)$ , de álgebra  $\mathfrak{so}(3)$ ), que engloba todas as matrizes de rotação no espaço tridimensional e o grupo euclidiano especial de terceira ordem ( $SE(3)$  de álgebra  $\mathfrak{se}(3)$ ), que representa todas as poses em três dimensões. Para  $a, b \in \mathbb{R}^3$

$$\text{hat}(a) = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_\times \triangleq \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} : \mathbb{R}^3 \rightarrow \mathfrak{so}(3) \quad (2.5)$$

$$\text{hat} \left( \begin{bmatrix} b \\ a \end{bmatrix} \right) = \begin{bmatrix} b \\ a \end{bmatrix}_\times \triangleq \begin{bmatrix} [a]_\times & b \\ 0_{1 \times 3} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -a_z & a_y & b_x \\ a_z & 0 & -a_x & b_y \\ -a_y & a_x & 0 & b_z \\ 0 & 0 & 0 & 0 \end{bmatrix} : \mathbb{R}^6 \rightarrow \mathfrak{se}(3) \quad (2.6)$$

Para  $c \in \mathfrak{so}(3)$  e  $d \in \mathfrak{se}(3)$

$$\text{vex}(c) = \begin{bmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{bmatrix}^\vee \triangleq \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} : \mathfrak{so}(3) \rightarrow \mathbb{R}^3 \quad (2.7)$$

$$\text{vex}(d) = \begin{bmatrix} 0 & -d_z & d_y & d_{xx} \\ d_z & 0 & -d_x & d_{yy} \\ -d_y & d_x & 0 & d_{zz} \\ 0 & 0 & 0 & 0 \end{bmatrix}^\vee \triangleq \begin{bmatrix} d_{xx} \\ d_{yy} \\ d_{zz} \\ d_x \\ d_y \\ d_z \end{bmatrix} : \mathfrak{se}(3) \rightarrow \mathbb{R}^6 \quad (2.8)$$

Podemos representar  $V_I$  em termos de  $V_B$  (e vice-versa) através da transformação adjunta  $Ad_\xi$  (MURRAY; LI; SASTRY, 1994)

$$V_I = Ad_\xi V_B \implies V_B = Ad_\xi^{-1} V_I \quad (2.9)$$

onde

$$Ad_\xi = \begin{bmatrix} R & [\mathbf{p}]_\times R \\ 0_{3 \times 3} & R \end{bmatrix} \quad \text{e} \quad Ad_\xi^{-1} = \begin{bmatrix} R^T & -R^T[\mathbf{p}]_\times \\ 0_{3 \times 3} & R^T \end{bmatrix} \quad (2.10)$$

Antes de prosseguir com o desenvolvimento matemático, é importante notar que a componente  $\mathbf{v}_I$  da equação 2.3 não representa a velocidade do centro de massa do drone em relação a origem de  $\{I\}$ , mas sim a velocidade de um ponto (imaginário) fixado no sistema de coordenadas de  $\{B\}$  que passa pela origem de  $\{I\}$  no instante em que a velocidade  $V_I$  é medida. Essa informação é importante para entender que a velocidade do centro de massa é dada por  $\dot{\mathbf{p}}$ , não por  $\mathbf{v}_I$ .

É conveniente separar a cinemática em suas componentes translacionais e rotacionais. Tomando 2.2 em coordenadas de  $\{B\}$  e aplicando a definição 2.6:

$$\begin{bmatrix} \dot{R} & \dot{\mathbf{p}} \\ 0_{1 \times 3} & 0 \end{bmatrix} = \xi[V_B]_\times = \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} [\omega]_\times & \mathbf{v}_B \\ 0_{1 \times 3} & 0 \end{bmatrix} \quad (2.11)$$

$$\implies \begin{cases} \dot{R} = R[\omega]_\times \\ \dot{\mathbf{p}} = R\mathbf{v}_B \end{cases}$$

### 2.1.2 Dinâmica de corpo rígido

Sejam  $\vec{F}$  e  $M$  a força e o momento totais aplicados sobre o drone e representados em  $\{B\}$ , computados da seguinte maneira

$$\vec{F} = \sum_{i=1}^4 \vec{F}_i \quad \text{e} \quad M = \sum_{i=1}^4 (\mathbf{r}_i^B \times \vec{F}_i + M_i) \quad (2.12)$$

onde  $\vec{F}_i$  é o empuxo gerado pelo propulsor  $i$ ,  $\mathbf{r}_i^B$  é a posição do propulsor  $i$  em relação ao centro de massa do drone e  $M_i$  é o torque reativo aplicado pelo propulsor na estrutura do quadrotor.

Podemos representar a dinâmica de corpo rígido do drone no espaço através das equações de Newton-Euler (MURRAY; LI; SASTRY, 1994) em coordenadas de  $\{B\}$

$$\begin{bmatrix} m\mathbf{I}_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & J \end{bmatrix} \dot{V}_B + \begin{bmatrix} [\omega]_\times & 0_{3 \times 3} \\ 0_{3 \times 3} & [\omega]_\times \end{bmatrix} \begin{bmatrix} m\mathbf{I}_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & J \end{bmatrix} V_B = \begin{bmatrix} \vec{F} + R^T m\vec{g} \\ M \end{bmatrix} \quad (2.13)$$

onde  $\vec{g} = g\vec{k}_I$  é a aceleração da gravidade expressa em  $\{I\}$ .

Separando as dinâmicas translacionais e as rotacionais e isolando as velocidades, temos o sistema

$$\begin{cases} \dot{\mathbf{v}}_B = m^{-1}\vec{F} + R^T\vec{g} - [\omega]_\times \mathbf{v}_B \\ \dot{\omega} = J^{-1}(M - [\omega]_\times J\omega) \end{cases} \quad (2.14)$$

O termo  $[\omega]_{\times} \mathbf{v}_B$  some se representarmos a dinâmica translacional com relação a  $\{I\}$ , o que nos dá:

$$\begin{cases} \dot{\mathbf{v}} = m^{-1} R \vec{F} + \vec{g} \\ \dot{\omega} = J^{-1} (M - [\omega]_{\times} J \omega) \end{cases} \quad (2.15)$$

onde  $\mathbf{v} = R \mathbf{v}_B$ .

### 2.1.3 Dinâmica do quadrotor

Combinando as equações 2.11 e 2.15, pode-se expressar a mecânica do quadrotor no sistema

$$\begin{cases} \dot{R} = R[\omega]_{\times} \\ \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{v}} = m^{-1} R \vec{F} + \vec{g} \\ \dot{\omega} = J^{-1} (M - [\omega]_{\times} J \omega) \end{cases} \quad (2.16)$$

O sistema 2.16 servirá de base para a formulação tanto do algoritmo de controle quando de estimação dos estados que serão apresentados nos capítulos subsequentes. No sistema apresentado, se considera que força  $\vec{F}$ , expressa em coordenadas de  $\{B\}$ , só contém componentes na direção  $\vec{k}_B$ . Essa hipótese parte da suposição de que o sistema de propulsão gera empuxo apenas na direção axial dos atuadores. Como explicado na seção seguinte, essa hipótese não condiz com um situação real de voo. Apesar disso, tomar essa suposição por verdadeira simplifica o desenvolvimento analítico do controlador. Podemos então reescrever o sistema 2.16 de forma a explicitar a direção da força  $\vec{F}$  de magnitude  $F$ . Buscando evidenciar as entradas de controle, pode-se expandir o termo  $M$  em  $M = [\tau_{i_B} \ \tau_{j_B} \ \tau_{k_B}]^T$

$$\begin{cases} \dot{R} = R[\omega]_{\times} \\ \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{v}} = \vec{g} - m^{-1} R F \vec{k}_B \\ \dot{\omega} = -J^{-1} [\omega]_{\times} J \omega + J^{-1} \begin{bmatrix} \tau_{i_B} \\ \tau_{j_B} \\ \tau_{k_B} \end{bmatrix} \end{cases} \quad (2.17)$$

Dizemos então que a entrada de controle do sistema 2.17 é

$$\mathbf{u} = \begin{bmatrix} F \\ \tau_{i_B} \\ \tau_{j_B} \\ \tau_{k_B} \end{bmatrix} \quad (2.18)$$

## 2.2 Atuadores

O modelo apresentado na seção anterior usa os torques e força total como entradas. No entanto estas entradas resultam dos efeitos aerodinâmicos associados a rotação das hélices do sistema de propulsão, e estão sujeitas a dinâmicas adicionais como a eletromecânica do motor utilizado, o descarregamento da bateria e a dinâmica elétrica do controlador eletrônico de velocidade (ESC). Nesta seção são apresentados os modelos simplificados da aerodinâmica da hélice na geração dos torques e forças e a geração de velocidade angular do rotor do motor (MAHONY; KUMAR; CORKE, 2012).

### 2.2.1 Aerodinâmica dominante

Em regime permanente, o empuxo gerado por um propulsor em condições de *hovering* é dado por

$$\vec{F}_i := -C_T \rho A_{r_p} r_p^2 \bar{\omega}_i^2 \vec{k}_B \quad (2.19)$$

onde  $A_{r_p}$  é a área dentro da qual a hélice rotaciona,  $r_p$  é o comprimento de uma das pás,  $\bar{\omega}_i$  é a velocidade angular da hélice em torno do eixo do propulsor,  $C_T$  é o coeficiente de empuxo da hélice, que depende de sua geometria e  $\rho$  é a densidade do ar. Na prática, considera-se o modelo simplificado

$$\vec{F}_i = -c_T \bar{\omega}_i^2 \vec{k}_B \quad (2.20)$$

A constante  $c_T > 0$  pode ser determinada por testes de empuxo estático.

O torque reativo (produzido pelo arrasto da hélice) que age sobre a estrutura do quadrotor em condição de *hovering* pode ser modelado por

$$M_i := \pm c_M \bar{\omega}_i^2 \vec{k}_B \quad (2.21)$$

onde  $c_M$  também depende de  $A_{r_p}$ ,  $r_p$  e  $\rho$  e pode ser determinado por testes de empuxo estático. O sinal da expressão acima depende do sentido de rotação da hélice em questão. Se considerarmos a Figura 1

$$M_i = \begin{cases} -c_M \bar{\omega}_i^2 \vec{k}_B, & \text{para } i \in \{1, 3\} \\ c_M \bar{\omega}_i^2 \vec{k}_B, & \text{para } i \in \{2, 4\} \end{cases} \quad (2.22)$$

Considerando 2.18, busca-se explicitar a relação entre as entradas do sistema dinâmico que descreve a mecânica do quadrotor e as velocidades de cada propulsor, possibilitando a alocação das velocidades (que são as variáveis que o drone pode impor através da ESC) para os sinais de controle dados em termos de força e torque. Dada a

equação 2.12 e tomando a Figura 1 por referência, com as devidas substituições chega-se a

$$\mathbf{u} = \begin{bmatrix} F \\ \tau_{i_B} \\ \tau_{j_B} \\ \tau_{k_B} \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -\frac{l}{\sqrt{2}}c_T & \frac{l}{\sqrt{2}}c_T & \frac{l}{\sqrt{2}}c_T & -\frac{l}{\sqrt{2}}c_T \\ \frac{l}{\sqrt{2}}c_T & \frac{l}{\sqrt{2}}c_T & -\frac{l}{\sqrt{2}}c_T & -\frac{l}{\sqrt{2}}c_T \\ -c_M & c_M & -c_M & c_M \end{bmatrix} \begin{bmatrix} \bar{\omega}_1 \\ \bar{\omega}_2 \\ \bar{\omega}_3 \\ \bar{\omega}_4 \end{bmatrix} = \Gamma \begin{bmatrix} \bar{\omega}_1 \\ \bar{\omega}_2 \\ \bar{\omega}_3 \\ \bar{\omega}_4 \end{bmatrix} \quad (2.23)$$

onde  $l$  é a distância de qualquer propulsor ao centro de massa do VANT.

No contexto da implementação,  $\Gamma$  pode variar à depender da maneira como se enumerou os propulsores. Em todo caso, as velocidades de rotação de cada hélice para uma dada entrada de controle  $\mathbf{u}$  podem ser calculadas por

$$\begin{bmatrix} \bar{\omega}_1 \\ \bar{\omega}_2 \\ \bar{\omega}_3 \\ \bar{\omega}_4 \end{bmatrix} = \Gamma^{-1}\mathbf{u} \quad (2.24)$$

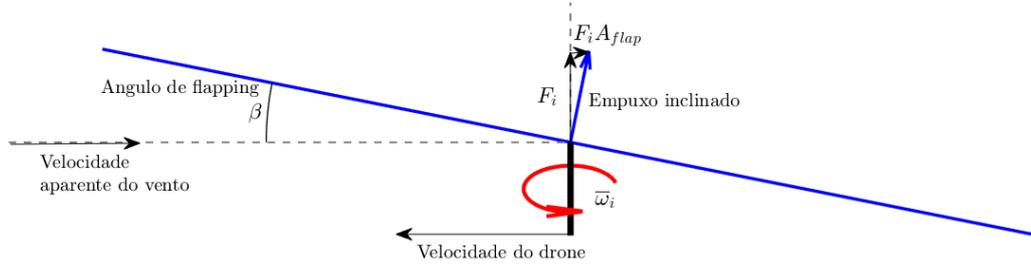
## 2.2.2 Efeitos aerodinâmicos adicionais

No modelo da dinâmica do VANT 2.17, considerou-se por entrada do sistema forças geradas apenas na direção de  $\vec{k}_B$ , o que torna o controle do drone nas direções remanescentes sub-atuado. Dado isso, dois efeitos aerodinâmicos têm influência significativa na dinâmica do sistema como um todo (por gerarem forças no plano  $\vec{i}_B \times \vec{j}_B$ ), sendo eles o de batimento das pás, (comumente chamado na literatura de *rotor flapping*) e o arrasto induzido.

Quando um quadricóptero se desloca no plano horizontal  $\vec{i}_I \times \vec{j}_I$ , a velocidade na ponta da hélice que está avançando é superior à da hélice que está recuando. Essa diferença de velocidade desbalanceia o empuxo gerado por cada hélice. A alta velocidade do rotor amplia os efeitos giroscópicos, que por sua vez produzem um torque atuando perpendicularmente à direção do vento percebido. Como resultado, a força de empuxo gerada desvia-se dos eixos do motor, inclinando-se para trás em relação à direção de movimento do quadricóptero.

Esse fenômeno também leva ao surgimento de um arrasto adicional, denominado arrasto induzido, causado pelas diferentes velocidades relativas do ar entre as hélices que avançam e as que recuam. A hélice que avança, ao se mover mais rapidamente, gera mais sustentação, mas também maior arrasto em comparação com a hélice que recua.

Adicionalmente, (MAHONY; KUMAR; CORKE, 2012) mostra que é possível modelar esses efeitos utilizando um modelo simplificado, onde uma única expressão é suficiente para representar ambos os fenômenos. Uma expressão detalhada para o *rotor flapping* envolve o modelo mecânico da deformação da hélice sujeita a forças aerodinâmicas e centrípeta à medida que ele realiza uma rotação completa. As equações de movimento

Figura 2 – Rotor *flapping*.

Fonte: Autoria própria.

resultante são um sistema dinâmico não-linear de segunda ordem com uma resposta oscilatória altamente amortecida na frequência forçada correspondente à velocidade do rotor. Para uma hélice típica, a dinâmica converge para o regime permanente após um ciclo completo de rotação. Para o modelo usado neste trabalho, considerou-se apenas o regime permanente do *rotor flapping*, presente no pacote do Parrot para Simulink.

Considera-se que o centro de massa do drone desloca-se com velocidade linear  $\dot{\mathbf{p}} = [v \ 0 \ 0]^T$  no referencial  $\{I\}$ , tal como ilustrado na Figura 2.  $F_i$  é a força gerada pelo rotor  $i$  na direção axial do propulsor,  $\bar{\omega}_i$  é a velocidade angular da hélice do propulsor  $i$  e  $\beta$  é o ângulo de *flapping* em regime permanente. Seja  $r_p$  o raio da hélice (distância do eixo do propulsor à extremidade de uma das pás) e  $\mu = |v|/(\bar{\omega}_i r_p)$  a razão de avanço, isto é, a razão da magnitude da velocidade horizontal do propulsor pela velocidade linear da ponta de uma pá, temos (MAHONY; KUMAR; CORKE, 2012)

$$\beta := -\frac{\mu A_{1c}}{(1 - \frac{1}{2}\mu^2)} \quad \beta^\perp := -\frac{\mu A_{1s}}{(1 + \frac{1}{2}\mu^2)} \quad (2.25)$$

onde  $\beta^\perp$  é a inclinação ortogonal ao o vento aparente e as constantes  $A_{1c}$  e  $A_{1s}$  abstraem o defasamento adicional entre as componentes cossenoidal e senoidal dos ângulos de *flapping* que advém do modelo detalhado da deformação das pás.

O *rotor flapping* faz com que o empuxo total gerado por um dos propulsores não seja coaxial ao rotor, mas perpendicular ao plano da hélice. Na prática, a hélice é rígida e opõe-se a força aerodinâmica que levanta a pá que avança contra o vento de forma que seu empuxo maximizado em razão da velocidade da ponta não seja totalmente neutralizado por um ângulo de ataque menor e um menor coeficiente de sustentação. Já para a pá que desloca-se à favor do vento aparente, o empuxo é reduzido.

Para um aerofólio que gera sustentação, como é o caso das pás da hélice, a inclinação da força aerodinâmica leva à produção de um arrasto induzido proporcional à sustentação gerada pelo aerofólio o que, em condições normais de *hovering*, traduz-se em forças geradas em todas as direções em torno da área da hélice e é responsável pelo torque  $M_i$  de cada propulsor. No entanto, quando há *flapping* da hélice, a pá que avança contra o vento produz mais arrasto induzido do que a que movimenta-se à favor do vento. O resultado é

um arrasto induzido que se opõe à direção aparente do vento do ponto de vista da hélice e que é proporcional à velocidade aparente do vento. Este efeito é insignificante para o caso de veículos grandes, mas pode ser significativo para modelos menores e mais leves. A maior consequência dos efeitos combinados de *flapping* e arrasto induzido são um notável arrasto percebido pelo quadrotor, mesmo quando este viaja a uma baixa velocidade horizontal.

Partindo do pressuposto que todos os propulsores são iguais e estão girando à velocidades angulares similares ( $\bar{\omega}_i = \bar{\omega}$ ,  $1 \leq i \leq 4$ ), pode-se afirmar que as quatro hélices encontram-se sob as mesmas condições de *flapping* e aerodinâmicas desbalanceadas. Sob tais condições, como pares de propulsores consecutivos giram em sentidos opostos, seus torques reativos sobre a estrutura do quadrotor se cancelam. Considerando o movimento geral do quadrotor à uma velocidade  $\mathbf{v}_B = [v_i \ v_j \ 0]^T$  expressa em  $\{B\}$

$$\mu = \frac{\sqrt{v_i^2 + v_j^2}}{\bar{\omega} r_p} \quad (2.26)$$

Define-se (MAHONY; KUMAR; CORKE, 2012)

$$A_{\text{flap}} = \frac{1}{\bar{\omega} r_p} \begin{bmatrix} A_{1c} & -A_{1s} & 0 \\ A_{1s} & A_{1c} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.27)$$

A matriz acima representa a sensibilidade do ângulo de *flapping* com relação ao vento aparente em coordenadas de  $\{B\}$ . Em coordenadas de  $\{B\}$  o arrasto induzido é

$$D_{\text{ind}} \mathbf{v}_B \approx \begin{bmatrix} d_i & 0 & 0 \\ 0 & d_j & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{v}_B \quad (2.28)$$

onde  $d_i = d_j$  é o coeficiente de arrasto induzido. A força total aplicada sobre o drone pode ser modelada por

$$\vec{F}_{\text{total}} := \vec{F} - F \left( A_{\text{flap}} + \begin{bmatrix} d_i & 0 & 0 \\ 0 & d_j & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \mathbf{v}_B \quad (2.29)$$

O modelo apresentado tem por finalidade simular com maior precisão a situação real de voo. Sua importância reside na modelagem de fenômenos percebidos pelo quadrotor que podem ser considerados no desenvolvimento do estimador de estados e análise da robustez do controlador projetado.

## 2.3 Sensores

O quadrotor dispõe de um conjunto de sensores para medição de seus estados (velocidades lineares e angulares, posição e orientação) direta ou indiretamente. Num

arranjo mínimo de sensores, os drones geralmente possuem uma IMU (*Inertial Measurement Unit*), que combina acelerômetro e giroscópio para medir as acelerações e velocidades angulares em três eixos; e um sonar empregado na medição da altitude do VANT, que também pode ser combinado com um sensor de pressão barométrica. Esse arranjo mínimo produz dados suficientes para estabilizar o drone e manobrá-lo no espaço, no entanto pode não ser ideal para a estimação precisa da posição do drone.

Em arranjos mais robustos, o sensor combina aos sensores já mencionados outros recursos como câmeras para odometria visual, estimação de velocidade translacional e atitude, medição da posição relativa de objetos, mapeamento, detecção de colisão, etc.; magnetômetros para estimação da orientação em torno do eixo vertical e compensação do desvio do giroscópio; GPS para medição de posição global, altitude; entre outros.

Nesse capítulo abordaremos o arranjo considerado nesse trabalho, baseado no quadrotor de pequeno porte Parrot Mambo (cujos detalhes de hardware e software são tratados na próxima seção), que possui uma IMU, um sensor de pressão barométrica, um sonar para medição de altitude e uma câmera posicionada na parte inferior do corpo do drone, que é empregada na estimação da velocidade do VANT no plano paralelo ao solo.

### 2.3.1 IMU - *Inertial Measurement Unit*

O sensor IMU combina dois tipos de sensores diferentes, o acelerômetro e o giroscópio, para obter as acelerações e velocidades angulares em três eixos. Esse dispositivo é amplamente utilizado na indústria aeroespacial e é crucial para o controle de atitude do quadrotor. O maior desafio no emprego desse tipo de sensor está no ruído e no desvio apresentado nas suas medições, sendo este último costumeiramente de baixa frequência e modelado como um processo *random walk*, o que dificulta sua filtragem. À seguir são apresentados os modelos estocásticos dos dois sensores da IMU (FIGUEIRÔA; MOUTINHO; AZINHEIRA, 2014).

#### 2.3.1.1 Giroscópio

O giroscópio mede a velocidade angular de  $\{B\}$  em relação a  $\{I\}$ , expressa em  $\{B\}$ .

O modelo de erro usado é

$$\bar{\omega} = \omega + \sigma_{\omega} + \mu_{\omega} \quad (2.30)$$

com  $\omega \in \mathbb{R}^3$  denotando o valor verdadeiro da taxa angular,  $\sigma_{\omega}$  denotando ruído branco aditivo e  $\mu_{\omega}$  correspondendo a um viés constante (ou lentamente variável no tempo).

### 2.3.1.2 Acelerômetro

Deixe o vetor de aceleração gravitacional constante ser  $\vec{g} \in \mathbb{R}^3$  e  $\dot{\mathbf{v}}_B$  denote a aceleração linear de  $\{B\}$  em relação a  $\{I\}$  expressa em  $\{B\}$ . O acelerômetro mede a aceleração instantânea de  $\{B\}$ ,  $\dot{\mathbf{v}}_B$ , menos o efeito da aceleração gravitacional expressa em  $\{B\}$ :

$$\dot{\hat{\mathbf{v}}}_B = \dot{\mathbf{v}}_B - R^T \vec{g} + \sigma_{\dot{\mathbf{v}}_B} + \mu_{\dot{\mathbf{v}}_B} \quad (2.31)$$

Novamente, a medição  $\dot{\hat{\mathbf{v}}}_B$  é corrompida por um termo de ruído branco  $\sigma_{\dot{\mathbf{v}}_B}$  e um termo de viés  $\mu_{\dot{\mathbf{v}}_B}$ .

## 2.3.2 Medição de Altitude

Como mencionado anteriormente, a medição de altitude no Mambo emprega dois sensores: o barômetro e o sonar. A seguir, o modelo matemático utilizado para descrever o comportamento das leituras desses sensores.

### 2.3.2.1 Barômetro

O quadricóptero é equipado com um sensor de pressão digital que mede a pressão do ar ambiente para obter uma estimativa de posição vertical baseada no gradiente de pressão atmosférica com o seguinte modelo de erro:

$$\bar{z}_{\text{baro}} = z + \sigma_{\text{baro}} \quad (2.32)$$

### 2.3.2.2 Sonar

Utilizado para medições de altitude de alta precisão, ao contrário do GPS e do sensor barométrico, o sonar mede uma altura relativa e não uma altura absoluta, e tipicamente tem um alcance limitado (aproximadamente 5m).

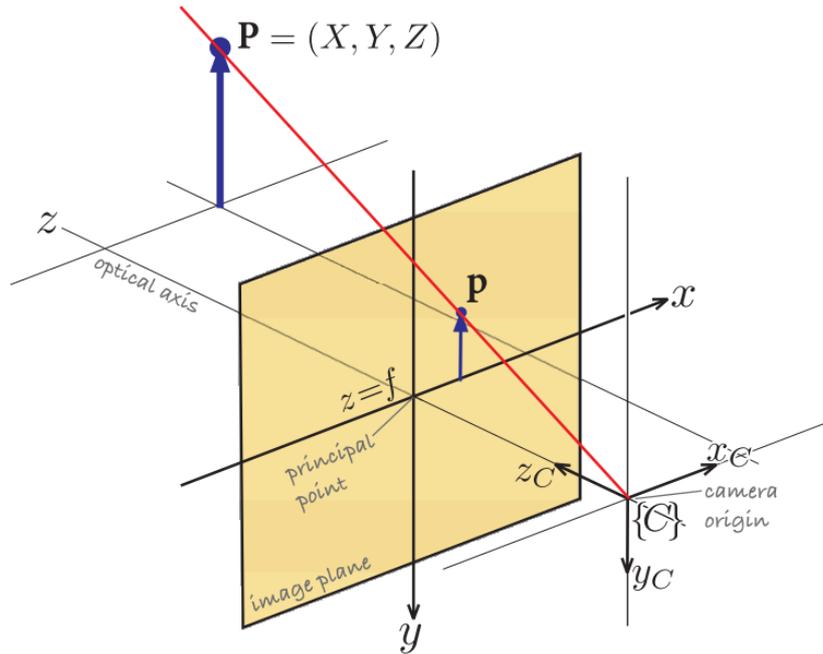
$$\bar{z}_{\text{sonar}} = z + \sigma_{\text{sonar}} \quad (2.33)$$

## 2.3.3 Câmera e odometria visual

O Parrot Mambo possui uma câmera acoplada ao seu corpo na parte inferior que é usada para medição do fluxo óptico, à partir do qual se obtém estimativas das velocidades translacionais do drone.

O fluxo óptico é a distribuição de velocidades aparentes de movimento de padrões de brilho em uma imagem. O fluxo óptico pode surgir do movimento relativo de objetos e do observador. Consequentemente, o fluxo óptico pode fornecer informações importantes sobre o arranjo espacial dos objetos visualizados e a taxa de mudança desse arranjo (HORN; SCHUNCK, 1981).

Figura 3 – Modelo de projeção central.



Fonte: (CORKE, 2017).

### 2.3.3.1 Modelo de Imagem de Projeção Central

Em visão computacional, é comum usar o modelo de imagem de perspectiva central mostrado na Figura 3. Os raios convergem na origem do quadro da câmera  $\{C\}$  e uma imagem não invertida é projetada no plano de imagem localizado em  $z = f$ . O eixo  $z$  intercepta o plano de imagem no ponto principal, que é a origem do quadro de coordenadas de imagem 2D. Usando triângulos semelhantes, podemos mostrar que um ponto em coordenadas de  $\{C\}$   $P = (X, Y, Z)$  é projetado no ponto de imagem  $p = (x, y)$  por (CORKE, 2017)

$$x = \frac{fX}{Z} \quad y = \frac{fY}{Z} \quad (2.34)$$

o que é uma transformação projetiva, ou mais especificamente, uma projeção perspectiva.

Podemos escrever as coordenadas dos pontos no plano da imagem em forma homogênea como  $\bar{p} = (\bar{x}, \bar{y}, \bar{z})$ , onde  $\bar{x} = fX$ ,  $\bar{y} = fY$  e  $\bar{z} = Z$ .

Ou em forma de matriz compacta como

$$\bar{p} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.35)$$

onde as coordenadas da imagem não homogêneas são

$$x = \frac{\bar{x}}{\bar{z}}, \quad y = \frac{\bar{y}}{\bar{z}} \quad (2.36)$$

Essas são frequentemente referidas como coordenadas do plano da imagem retinal. Para o caso em que  $f = 1$ , as coordenadas são referidas como normalizadas, retinal ou canônicas coordenadas do plano da imagem.

Se escrevermos  $P$  em forma homogênea também como  $\bar{P} = (X, Y, Z, 1)^T$ , então a projeção perspectiva pode ser escrita em forma linear como

$$\bar{p} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bar{P} = C\bar{P} \quad (2.37)$$

onde  $C$  é uma matriz  $3 \times 4$  conhecida como a matriz da câmera. Observe que escrevemos  $\bar{P}$  para destacar que esta é a coordenada do ponto com respeito ao referencial da câmera  $\{C\}$ . A barra indica quantidades homogêneas. A matriz da câmera pode ser fatorada como

$$\bar{p} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bar{P} \quad (2.38)$$

onde a segunda matriz é a matriz de projeção.

Em geral, a câmera terá uma pose arbitrária  $\xi_c$  com relação ao sistema de coordenadas  $\{C\}$  conforme mostrado na Fig. 11.5. A posição do ponto com relação à câmera é

$$\mathbf{C}_P = (\xi_c)^{-1}\mathbf{P} \quad (2.39)$$

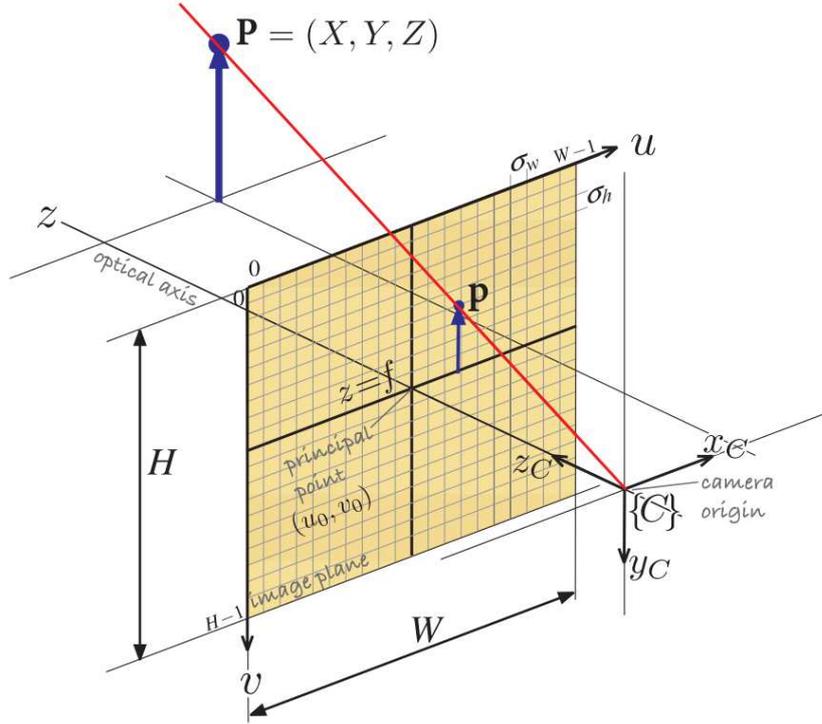
Em uma câmera digital, o plano da imagem é uma grade  $W \times H$  de elementos sensíveis à luz chamados fotossítios, que correspondem diretamente aos elementos da imagem (ou pixels) como mostrado na Figura 4. As coordenadas dos pixels são um vetor de 2 componentes  $(u, v)$  de inteiros não negativos e, por convenção, a origem está no canto superior esquerdo do plano da imagem. No MATLAB<sup>®</sup>, o pixel superior esquerdo é  $(1, 1)$ . Os pixels são uniformes em tamanho e centralizados em uma grade regular, portanto, a coordenada do plano da imagem é relacionada à coordenada do plano da imagem por

$$u = \frac{x}{\rho_w} + u_0, \quad v = \frac{y}{\rho_h} + v_0 \quad (2.40)$$

onde  $\rho_w$  e  $\rho_h$  são a largura e altura de cada pixel, respectivamente, e  $(u_0, v_0)$  é o ponto principal - a coordenada do pixel do ponto onde o eixo óptico intersecta o plano da imagem com respeito à nova origem. Podemos escrever a Eq. 11.4 em termos de coordenadas de pixel, precedendo uma matriz de parâmetros da câmera  $K$

$$\bar{p} = K \begin{bmatrix} \frac{1}{\rho_w} & 0 & u_0 \\ 0 & \frac{1}{\rho_h} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \bar{P} \quad (2.41)$$

Figura 4 – Modelo de projeção central com plano de imagem e pixels discretos.



Fonte: (CORKE, 2017).

onde  $\bar{p} = (\bar{u}, \bar{v}, \bar{w})$  é a coordenada homogênea do ponto  $\{C\}$   $P$  em coordenadas de pixel. As coordenadas de pixel não homogêneas da imagem são

$$u = \frac{\bar{u}}{\bar{w}}, \quad v = \frac{\bar{v}}{\bar{w}} \quad (2.42)$$

Combinando a 2.39 e 2.41 podemos escrever a projeção da câmera na forma geral como

$$\begin{aligned} \bar{p} &= \begin{bmatrix} \frac{f}{\rho_w} & 0 & u_0 \\ 0 & \frac{f}{\rho_h} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^{-1} \bar{P} = \\ &= KP_0 T_c^{-1} \bar{P} = C \bar{P} \end{aligned} \quad (2.43)$$

onde todos os termos são agrupados na matriz da câmera  $C$ . Esta é uma transformação homogênea  $3 \times 4$  que realiza escalonamento, translação e projeção perspectiva. Frequentemente também é referida como a matriz de projeção ou a matriz de calibração da câmera.

A projeção também pode ser escrita em forma funcional como

$$p = \mathcal{P}(P, K, \xi_c) \quad (2.44)$$

onde  $P$  é o vetor de coordenadas do ponto no referencial  $\{C\}$ ,  $K$  é a matriz de parâmetros da câmera e compreende os parâmetros intrínsecos que são as características inatas da

câmera e do sensor, como  $f$ ,  $\rho_w$ ,  $\rho_h$ ,  $u_0$  e  $v_0$ .  $\xi_c$  é a pose da câmera e compreende um mínimo de seis parâmetros - os parâmetros extrínsecos - que descrevem a translação e orientação da câmera em  $SE(3)$ .

### 2.3.3.2 Cálculo do fluxo óptico

O algoritmo considerado no cálculo do fluxo óptico é o apresentado em (HORN; SCHUNCK, 1981). O algoritmo Horn-Schunck assume suavidade no fluxo óptico em toda a imagem. Assim, busca minimizar distorções no fluxo e prefere soluções que apresentem mais suavidade.

O fluxo é formulado como uma função de energia global que se busca minimizar. Esta função é dada para fluxos de imagem bidimensionais como:

$$E = \iint (I_x \dot{u} + I_y \dot{v} + I_t)^2 + \alpha^2 (\|\nabla \dot{u}\|^2 + \|\nabla \dot{v}\|^2) dx dy \quad (2.45)$$

onde  $I_x, I_y$ , e  $I_t$  são as derivadas dos valores de intensidade da imagem ao longo das dimensões  $x$ ,  $y$  e tempo, respectivamente,  $\bar{\mathbf{V}} = [\dot{u}(x, y), \dot{v}(x, y)]^T$  é o vetor de fluxo óptico a ser resolvido, e o parâmetro  $\alpha$  é uma constante de regularização. Valores maiores de  $\alpha$  levam a um fluxo mais suave. Esta funcional pode ser minimizada resolvendo as equações de Euler-Lagrange associadas:

$$\begin{aligned} \frac{\partial L}{\partial \dot{u}} - \frac{\partial}{\partial x} \frac{\partial L}{\partial \dot{u}_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial \dot{u}_y} &= 0 \\ \frac{\partial L}{\partial \dot{v}} - \frac{\partial}{\partial x} \frac{\partial L}{\partial \dot{v}_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial \dot{v}_y} &= 0 \end{aligned} \quad (2.46)$$

onde  $L$  é o integrando da expressão de energia, resultando em:

$$\begin{aligned} I_x(I_x \dot{u} + I_y \dot{v} + I_t) - \alpha^2 \Delta \dot{u} &= 0 \\ I_y(I_x \dot{u} + I_y \dot{v} + I_t) - \alpha^2 \Delta \dot{v} &= 0 \end{aligned} \quad (2.47)$$

onde os subscritos indicam diferenciação parcial e  $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  denota o operador Laplaciano. Na prática, o Laplaciano é aproximado numericamente usando diferenças finitas e pode ser escrito como  $\Delta \dot{u}(x, y) = \mu_{\dot{u}}(x, y) - \dot{u}(x, y)$ , onde  $\mu_{\dot{u}}(x, y)$  é uma média ponderada de  $\dot{u}$  calculada na vizinhança do pixel  $(x, y)$ . Usando essa notação, o sistema de equações acima pode ser escrito como:

$$\begin{aligned} (I_t^2 + \alpha^2) \dot{u} + I_x I_y \dot{v} &= \alpha^2 \mu_{\dot{u}} - I_x I_t \\ I_x I_y \dot{u} + (I_t^2 + \alpha^2) \dot{v} &= \alpha^2 \mu_{\dot{v}} - I_y I_t \end{aligned} \quad (2.48)$$

que é linear em  $\dot{u}$  e  $\dot{v}$  e pode ser resolvido para cada pixel na imagem. No entanto, como a solução depende dos valores vizinhos do campo de fluxo, deve-se repetir uma vez que os

vizinhos tenham sido atualizados. O seguinte esquema iterativo é derivado usando a regra de Cramer:

$$\begin{aligned} \dot{u}^{k+1} &= \mu_{\dot{u}}^k - \frac{I_x(I_x\mu_{\dot{u}}^k + I_y\mu_{\dot{v}}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \\ \dot{v}^{k+1} &= \mu_{\dot{v}}^k - \frac{I_y(I_x\mu_{\dot{u}}^k + I_y\mu_{\dot{v}}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \end{aligned} \quad (2.49)$$

onde o superíndice  $k + 1$  denota a próxima iteração, que deve ser calculada, e  $k$  é o resultado da última iteração calculada. Isso é, essencialmente, um método de divisão de matriz, semelhante ao método de Jacobi, aplicado ao grande sistema de equações que surge ao resolver para todos os pixels simultaneamente.

### 2.3.3.3 Relação entre fluxo óptico e velocidades da câmera

Derivando a equação 2.39 no tempo

$$\dot{p} = J_p(P, K, \xi_c)\nu \quad (2.50)$$

onde  $\nu = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T \in \mathbb{R}^6$  é a velocidade da câmera no referencial  $\{C\}$ ,  $J_p$  é chamada de jacobiana da imagem.

Considere uma câmera movendo-se com uma velocidade de corpo  $\nu = [v, \omega]^T$  no referencial  $\{C\}$  e observando um ponto  $P$  em  $\{C\}$  com coordenadas relativas à câmera  $P = (X, Y, Z)$ . A velocidade do ponto relativa ao referencial da câmera é

$$\dot{P} = -\omega \times P - v$$

que podemos escrever na forma escalar como

$$\begin{aligned} \dot{X} &= \omega_y Z - \omega_z Y - v_x \\ \dot{Y} &= \omega_z X - \omega_x Z - v_y \\ \dot{Z} &= \omega_x Y - \omega_y X - v_z \end{aligned} \quad (2.51)$$

A projeção perspectiva para coordenadas de plano de imagem normalizadas é

$$x = \frac{X}{Z}, \quad y = \frac{Y}{Z}$$

e a derivada temporal, usando a regra do quociente, é

$$\dot{x} = \frac{X\dot{Z} - Z\dot{X}}{Z^2}, \quad \dot{y} = \frac{Y\dot{Z} - Z\dot{Y}}{Z^2}$$

Substituindo em 2.51,  $X = xZ$  e  $Y = yZ$ , podemos escrever isso na forma matricial como

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.52)$$

As coordenadas de plano de imagem normalizadas estão relacionadas às coordenadas de pixel pela equação 2.40 que pode ser rearranjada como

$$x = \frac{\rho_u}{f} \bar{u}, \quad y = \frac{\rho_v}{f} \bar{v}$$

onde  $\bar{u} = u - u_0$  e  $\bar{v} = v - v_0$ . No caso típico onde  $\rho_u = \rho_v = \rho$ , pode-se expressar a distância focal em pixels  $f' = f/\rho$ . Substituindo  $f'$  e derivando a expressão anterior, tem-se

$$\dot{x} = \frac{\dot{\bar{u}}}{f'}, \quad \dot{y} = \frac{\dot{\bar{v}}}{f'} \quad (2.53)$$

Substituindo 2.53 em 2.52

$$\begin{bmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{f'}{Z} & 0 & \frac{\bar{u}}{Z} & \frac{\bar{u}\bar{v}}{f'} & -\frac{f'^2+\bar{u}^2}{f'} & \bar{v} \\ 0 & -\frac{f'}{Z} & \frac{\bar{v}}{Z} & \frac{f'^2+\bar{v}^2}{f'} & -\frac{\bar{u}\bar{v}}{f'} & -\bar{u} \end{bmatrix}}_{J_p(p,Z)} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.54)$$

em termos de coordenadas de pixel com respeito ao ponto principal. De forma mais compacta

$$\dot{p} = J_p(p, Z)\nu \quad (2.55)$$

Pode-se considerar o movimento de dois ou mais pontos empilhando suas jacobianas. Para três pontos:

$$\begin{bmatrix} \dot{\bar{u}}_1 \\ \dot{\bar{v}}_1 \\ \dot{\bar{u}}_2 \\ \dot{\bar{v}}_2 \\ \dot{\bar{u}}_3 \\ \dot{\bar{v}}_3 \end{bmatrix} = \begin{bmatrix} J_p(p_1, Z_1) \\ J_p(p_2, Z_2) \\ J_p(p_3, Z_3) \end{bmatrix} \nu$$

a matriz será de posto completo, não singular, contanto que os pontos não sejam coincidentes ou colineares. Dito isso, obtém-se a velocidade da câmera em coordenadas de  $\{C\}$  por

$$\nu = \begin{bmatrix} J_p(p_1, Z_1) \\ J_p(p_2, Z_2) \\ J_p(p_3, Z_3) \end{bmatrix}^{-1} \begin{bmatrix} \dot{\tilde{u}}_1 \\ \dot{\tilde{v}}_1 \\ \dot{\tilde{u}}_2 \\ \dot{\tilde{v}}_2 \\ \dot{\tilde{u}}_3 \\ \dot{\tilde{v}}_3 \end{bmatrix}$$

sendo o vetor das velocidades dos pontos na imagem calculado através do algoritmo Horn-Schunck apresentado anteriormente. Se considerarmos todos os pontos da imagem (o fluxo óptico completo) a equação acima será:

$$\nu = \begin{bmatrix} J_p(p_1, Z_1) \\ \vdots \\ J_p(p_N, Z_N) \end{bmatrix}^+ \begin{bmatrix} [\dot{\tilde{u}}_1 \ \dot{\tilde{v}}_1]^T \\ \vdots \\ [\dot{\tilde{u}}_1 \ \dot{\tilde{v}}_1]^T \end{bmatrix} \quad (2.56)$$

onde o superescrito  $+$  representa a matriz pseudo-inversa.

### 3 Controle de atitude

O controle de atitude implementado neste trabalho foi baseado na formulação proposta por (LEE, 2011) e (FERNANDO et al., 2011), onde um controlador robusto adaptativo é derivado diretamente da geometria da mecânica rotacional do drone.

Isolando da equação 2.16 a dinâmica rotacional temos

$$\begin{cases} \dot{R} = R[\omega]_{\times} \\ \dot{\omega} = J^{-1}(M - [\omega]_{\times}J\omega) \end{cases} \quad (3.1)$$

Seja a  $R_d(t)$  é a trajetória desejada para a rotação, cuja dinâmica é dada por

$$\dot{R}_d = R_d[\omega_d]_{\times} \quad (3.2)$$

Define-se os erros de configuração

$$\Phi(R, R_d) = \frac{1}{2} \text{tr}[G(\mathbf{I}_3 - R_d^T R)] \quad (3.3)$$

$$e_R = \left( \frac{1}{2} (GR_d^T R - R^T R_d G) \right)^{\vee} \quad (3.4)$$

$$e_{\omega} = \omega - R^T R_d \omega_d \quad (3.5)$$

onde  $\Phi(R, R_d) : SO(3) \times SO(3) \rightarrow \mathbb{R}^1$  é uma métrica bi-invariante não negativa que expressa a distância entre as configurações atual  $R$  e a desejada  $R_d$  onde  $G = \text{diag}(g_1, g_2, g_3)$ ,  $e_R$  é o vetor erro de configuração no espaço tangente ao  $SO(3)$  e  $e_{\omega}$  é o vetor erro de velocidade. As seguintes afirmações são verdadeiras

$$\frac{d}{dt} \Phi(R, R_d) = \langle e_R, e_{\omega} \rangle \quad (3.6)$$

$$\dot{e}_R = E(R, R_d) e_{\omega} \quad (3.7)$$

$$\dot{e}_{\omega} = J^{-1}(M - [\omega]_{\times}J\omega) - \alpha_d \quad (3.8)$$

onde

$$E(R, R_d) = \frac{1}{2} (\text{tr}(R^T R_d G) \mathbf{I}_3 - R^T R_d G) \quad (3.9)$$

$$\alpha_d = -[\omega]_{\times} R^T R_d \omega_d + R^T R_d \dot{\omega}_d \quad (3.10)$$

sendo  $\dot{\omega}_d$  a aceleração angular desejada.

Dadas as definições acima, diz-se que a função potencial

$$V = \frac{1}{2} \langle e_{\omega}, J e_{\omega} \rangle + k_R \Phi(R, R_d) + c \langle J e_{\omega}, e_R \rangle, \quad k_R, c > 0 \quad (3.11)$$

representa a energia do sistema em termos dos erros de configuração e para a lei de controle

$$M = -k_R e_R - k_\omega e_\omega + [\omega]_\times J \omega + J \alpha_d, \quad k_\omega > 0 \quad (3.12)$$

a derivada temporal de  $V$  é

$$\dot{V} = -k_\omega \|e_\omega\|^2 - ck_R \|e_R\|^2 - ck_\omega \langle e_R, e_\omega \rangle + c \langle J e_\omega, E(R, R_d) e_\omega \rangle \quad (3.13)$$

A lei de controle  $M$  estabiliza assintoticamente o sistema em torno de  $R_d$ , fazendo  $V$  positiva semi-definida e  $\dot{V}$  negativa semi-definida para a condição

$$c < \min \left\{ \sqrt{\frac{2b_1 k_R \lambda_m}{\lambda_M^2}}, \frac{\sqrt{2} k_\omega}{\lambda_M \text{tr}(G)}, \frac{4k_R k_\omega}{k_\omega^2 + \frac{1}{\sqrt{2}} k_R \lambda_M \text{tr}(G)} \right\} \quad (3.14)$$

onde  $\lambda_m$  e  $\lambda_M$  são os autovalores mínimo e máximo, respectivamente, da matriz de inércia  $J$ . Os ganhos  $k_R$ ,  $k_\omega$  e  $c$  são parâmetros de sintonia do controlador apresentado e define-se

$$b_1 = \frac{h_1}{h_2 + h_3} \quad (3.15)$$

onde

$$\begin{aligned} h_1 &= \min\{g_1 + g_2, g_2 + g_3, g_3 + g_1\} \\ h_2 &= \max\{(g_1 - g_2)^2, (g_2 - g_3)^2, (g_3 - g_1)^2\} \\ h_3 &= \max\{(g_1 + g_2)^2, (g_2 + g_3)^2, (g_3 + g_1)^2\} \end{aligned} \quad (3.16)$$

A prova da estabilidade bem como a dedução dos termos apresentados nesse capítulo seguem no Apêndice A.

Como apontado por (FERNANDO et al., 2011), o controle geométrico definido em termos de uma matriz de inércia  $J$  constante demanda o conhecimento prévio da inércia do corpo rígido controlado (que no contexto desse trabalho é o quadrotor). Embora essa informação seja disponível para o Parrot Mambo, a estimativa é ideal e imprecisa. (FERNANDO et al., 2011) também expõe que a aplicação direta de uma lei de controle adaptativo nesse caso sem considerar possíveis perturbações no sistema podem levar à instabilidade.

### 3.1 Controle adaptativo

Como mencionado, o conhecimento da inércia real do quadrotor nem sempre é possível. Portanto

$$\tilde{J} = J - \bar{J} \quad (3.17)$$

onde  $\tilde{J}$  é o erro de estimação da inércia  $J$  e  $\bar{J}$  é a inércia estimada. Assumi-se que  $\tilde{J}$ ,  $\bar{J}$  e  $J$  são matrizes simétricas. A função potencial

$$V = \frac{1}{2} \langle e_\omega, J e_\omega \rangle + k_R \Phi(R, R_d) + c \langle J e_\omega, e_R \rangle + \frac{1}{2k_J} \|\tilde{J}\|_F^2, \quad k_J > 0 \quad (3.18)$$

representa a energia do sistema em termos dos erros de configuração e estimação da inércia e para a lei de controle

$$M = -k_R e_R - k_\omega e_\omega + [\omega]_\times \bar{J}\omega + \bar{J}\alpha_d \quad (3.19)$$

e a dinâmica de  $\dot{\bar{J}}$

$$\dot{\bar{J}} = \frac{k_J}{2} (-\alpha_d e_A^T - e_A \alpha_d^T + \omega \omega^T [e_A]_\times - [e_A]_\times \omega \omega^T) \quad (3.20)$$

$$e_A = e_\omega + c e_R$$

a derivada temporal de  $V$  é

$$\dot{V} = -k_\omega \|e_\omega\|^2 - c k_R \|e_R\|^2 - c k_\omega \langle e_R, e_\omega \rangle + c \langle J e_\omega, E(R, R_d) e_\omega \rangle \quad (3.21)$$

Para a condição 3.14 o termo  $V$  é positivo semi-definido e  $\dot{V}$  é negativo semi-definido, fazendo o sistema assintoticamente estável. Neste caso, considera-se um chute inicial de  $J$  na obtenção de  $\lambda_m$  e  $\lambda_M$ . No caso do Mambo, uma aproximação ideal de  $J$  é disponibilizada no pacote para Simulink ([MathWorks, 2023](#)).

## 3.2 Controle Robusto

A lei de controle desenvolvida na seção anterior considera que não há perturbações na dinâmica de atitude. Sistemas de controle robusto adaptativo modificam a estrutura do controle adaptativo para fazê-los robustos frente à dinâmicas não modeladas e perturbações limitadas ([LEE, 2011](#)). Seja

$$\begin{cases} \dot{R} = R[\omega]_\times \\ \dot{\omega} = J^{-1}(M + \Delta - [\omega]_\times J\omega) \end{cases} \quad (3.22)$$

o modelo da dinâmica rotacional levando em consideração a presença de uma perturbação  $\Delta$  limitada por uma constante  $\delta$  conhecida ( $\|\Delta\| \leq \delta$ ). Define-se a seguinte lei de controle e de atualização da estimativa  $\bar{J}$  ([FERNANDO et al., 2011](#))

$$\begin{cases} M = -k_R e_R - k_\omega e_\omega + [\omega]_\times \bar{J}\omega + \bar{J}\alpha_d + v \\ v = -\frac{\delta^2 e_A}{\delta \|e_A\| + \epsilon} \\ \dot{\bar{J}} = \frac{k_J}{2} (-\alpha_d e_A^T - e_A \alpha_d^T + \omega \omega^T [e_A]_\times - [e_A]_\times \omega \omega^T - 2\sigma \bar{J}) \end{cases} \quad (3.23)$$

onde  $e_A = e_\omega + c e_R$  e  $k_R, k_\omega, c, k_J, \epsilon$  e  $\sigma$  são constantes positivas. Uma análise similar a apresentada para o caso do controle adaptativo pode ser realizada levando em conta os novos termos que aparecem acima. Essa análise não foi feita para este trabalho. Assumiu-se que uma escolha suficientemente pequena dos valores de  $\epsilon$  e  $\sigma$  levam à uma mudança igualmente pequena nas fronteiras de  $V$  e  $\dot{V}$  calculadas no Apêndice A, não trazendo mudanças significativas à dinâmica considerada na seção anterior.

## 4 Observador de atitude

Uma vez que o problema de controle foi formulado usando a geometria do problema no  $SO(3)$ , por coerência implementa-se um observador de atitude não-linear que forneça uma estimativa de orientação como uma matriz de rotação. O observador implementado foi o filtro complementar passivo com estimação de viés (MAHONY; HAMEL; PFLIMLIN, 2008).

### 4.1 Medidas

As medidas disponíveis da IMU do Mambo são as de um acelerômetro e de um giroscópio, ambos de 3 eixos. A origem da IMU é o referencial  $\{B\}$ .

#### 4.1.1 Giroscópio

O giroscópio mede a velocidade angular  $\omega$  de  $\{B\}$  expressa em  $\{B\}$ . O modelo de erro usado neste trabalho é  $\omega_y = \omega + \mathbf{b} + \eta \in \mathbb{R}^3$ , onde  $\omega_y$  é a medição,  $\omega$  denota o valor verdadeiro,  $\eta$  denota ruído de medição e  $\mathbf{b}$  denota um viés constante (ou variante no tempo) de giroscópio.

#### 4.1.2 Acelerômetro

Denota-se a aceleração linear instantânea de  $\{B\}$  relativa a  $\{I\}$ , expressa em  $\{I\}$ , por  $\mathbf{a}$ . Um quadro ideal de acelerômetro, fixado em  $\{B\}$ , mede a aceleração linear instantânea de  $\{B\}$  mais a aceleração gravitacional  $\vec{g}$  em  $\{I\}$ . Na prática, a saída de um acelerômetro possui viés e ruído,

$$\mathbf{a} = R^T(\dot{\mathbf{v}} + \vec{g}) + \mathbf{b}_a + \eta_a.$$

onde  $\mathbf{b}_a$  é o viés do acelerômetro e  $\eta_a$  é o ruído limitado. No entanto, para o filtro implementado considerou-se a medição  $\mathbf{a}$  como uma representação direta de  $\vec{g}$  em  $\{B\}$ , uma vez que o viés do acelerômetro no Mambo pode ser estimado durante um processo de calibração dos sensores antes da missão e que, viajando sob baixas velocidades, a influência de  $\dot{\mathbf{v}}$  é muito pequena em relação a  $\vec{g}$ . Assim, subtraiu-se o viés obtido e durante a missão o ruído de alta frequência é filtrado usando um filtro passa baixa. Para reconstruir a

rotação medida  $R_y$  à partir de  $\mathbf{a}$ , calculou-se

$$\theta_{\vec{k}} = \int_0^t \omega_{\vec{k}} d\tau \quad (4.1)$$

$$\theta_{\vec{j}} = \arcsin\left(\frac{\mathbf{a}_{\vec{j}}}{g}\right) \quad (4.2)$$

$$\theta_{\vec{i}} = \arctan\left(\frac{\mathbf{a}_{\vec{j}}}{\mathbf{a}_{\vec{k}}}\right) \quad (4.3)$$

$$R_y = R_{\vec{k}}(\theta_{\vec{k}}) R_{\vec{j}}(\theta_{\vec{j}}) R_{\vec{i}}(\theta_{\vec{i}}) \quad (4.4)$$

onde  $\omega_{\vec{k}}$  é a velocidade angular em torno de  $\vec{k}_B$ ,  $\mathbf{a}_{\vec{i}}$ ,  $\mathbf{a}_{\vec{j}}$  e  $\mathbf{a}_{\vec{k}}$  são as leituras do acelerômetro nos seus 3 eixos, correspondentes aos do referencial  $\{B\}$ ,  $g$  é a norma de  $\vec{g}$  ( $g \approx 9.81\text{m/s}$ ) e

$$R_{\vec{k}}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$$R_{\vec{j}}(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (4.6)$$

$$R_{\vec{i}}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (4.7)$$

são matrizes de rotação pura em torno de  $\vec{k}_B$ ,  $\vec{j}_B$  e  $\vec{i}_B$ , respectivamente.

## 4.2 Filtro complementar passivo com correção de viés

Seja o modelo das medições dado por

$$R_y \approx R, \quad \text{para baixas frequências} \quad (4.8)$$

$$\omega_y \approx \omega + \mathbf{b}, \quad \text{para um viés } \mathbf{b} \text{ constante} \quad (4.9)$$

o filtro complementar passivo no  $SO(3)$  consiste em

$$\begin{cases} \dot{\hat{R}} = \hat{R}[\hat{\omega}]_{\times}, & \hat{R}(0) = \mathbf{I}_3 \\ \dot{\hat{\omega}} = \omega_y + k_P \omega_c - \hat{\mathbf{b}} \\ \dot{\hat{\mathbf{b}}} = -k_I \omega_c, & \hat{\mathbf{b}}(0) = \hat{\mathbf{b}}_0 \\ \omega_c = P_a(\tilde{R})^\vee, & \tilde{R} = \hat{R}^T R_y \end{cases} \quad (4.10)$$

onde  $k_P$  e  $k_I$  são ganhos positivos que podem ser sintonizados.  $\hat{R}(0)$  e  $\hat{\mathbf{b}}(0)$  são os valores de inicialização do filtro, onde  $\hat{\mathbf{b}}_0$  pode ser obtido durante a calibração dos sensores.  $\omega_c$  é um termo de correção que compensa a dinâmica de alta frequência associada aos ruídos do giroscópio enquanto define-se a dinâmica do viés estimado de forma a capturar a dinâmica em baixa frequência associada ao viés do sensor.

### 4.2.1 Análise da estabilidade

A análise de estabilidade não linear é baseada na ideia de uma estimativa adaptativa para o valor desconhecido do viés. Definimos uma função candidata de Lyapunov por

$$V = \frac{1}{2}\text{tr}(I - \tilde{R}) + \frac{1}{2k_I}\|\tilde{\mathbf{b}}\|^2 \quad (4.11)$$

Diferenciando  $\tilde{R}$  e usando o modelo proposto das medições, em um processo muito similar ao feito para o controle quando se diferenciou  $\Phi$ , se chega a

$$\begin{aligned} \dot{\tilde{R}} &= \tilde{R}[\omega]_{\times} + [\omega]_{\times}\tilde{R}^T - k_P[\omega_c]_{\times}\tilde{R} - [\tilde{\mathbf{b}}]_{\times}\tilde{R} = \\ &= [(\text{tr}(\tilde{R}^T)\mathbf{I}_3 - \tilde{R}^T)\omega]_{\times} - k_P[\omega_c]_{\times}\tilde{R} - [\tilde{\mathbf{b}}]_{\times}\tilde{R} \end{aligned}$$

Como  $\tilde{\mathbf{b}} = \mathbf{b} - \hat{\mathbf{b}}$ , têm-se  $\dot{\tilde{\mathbf{b}}} = -\dot{\hat{\mathbf{b}}}$ . Diferenciando  $V$  e fazendo essas substituições é possível se chegar a

$$\begin{aligned} \dot{V} &= -\frac{1}{2}\text{tr}(\dot{\tilde{R}}) - \frac{1}{k_I}\langle\tilde{\mathbf{b}}, \dot{\hat{\mathbf{b}}}\rangle = \\ &= -\frac{1}{2}\text{tr}\left([\text{tr}(\tilde{R}^T)\mathbf{I}_3 - \tilde{R}^T\right)\omega]_{\times} - k_P[\omega_c]_{\times}\tilde{R} - [\tilde{\mathbf{b}}]_{\times}\tilde{R} - \frac{1}{k_I}\langle\tilde{\mathbf{b}}, \dot{\hat{\mathbf{b}}}\rangle = \\ &\quad \frac{1}{2}\text{tr}(k_P[\omega_c]_{\times}\tilde{R}) + \frac{1}{2}\text{tr}([\tilde{\mathbf{b}}]_{\times}\tilde{R}) - \frac{1}{k_I}\langle\tilde{\mathbf{b}}, \dot{\hat{\mathbf{b}}}\rangle = \\ &= -k_P\langle\omega_c, P_a(\tilde{R})^\vee\rangle - \langle\tilde{\mathbf{b}}, P_a(\tilde{R})^\vee\rangle - \frac{1}{k_I}\langle\tilde{\mathbf{b}}, \dot{\hat{\mathbf{b}}}\rangle \end{aligned} \quad (4.12)$$

Substituindo na equação acima  $\omega_c = P_a(\tilde{R})^\vee$  e  $\dot{\hat{\mathbf{b}}} = -k_I\omega_c = -k_IP_a(\tilde{R})^\vee$

$$\begin{aligned} \dot{V} &= -k_P\|P_a(\tilde{R})^\vee\|^2 - \langle\tilde{\mathbf{b}}, P_a(\tilde{R})^\vee\rangle + \langle\tilde{\mathbf{b}}, P_a(\tilde{R})^\vee\rangle = \\ &= -k_P\|P_a(\tilde{R})^\vee\|^2 \end{aligned} \quad (4.13)$$

provando a estabilidade do observador projetado.

## 5 Implementação e Simulação

Este capítulo trata da implementação e simulação dos algoritmos desenvolvidos nos Capítulos 3 e 4. O controle de sistemas mecânicos por controle geométrico apresenta como maior desafio a sintonia, tendo em vista que as provas matemáticas fornecidas permitem analisar a estabilidade frente à escolha dos ganhos mas não oferecem meios para avaliar a dinâmica que uma dada escolha possa proporcionar. Uma vez que o pacote de simulação para drones Parrot no Simulink permite configurar ruídos dos sensores baseados em dados de calibração reais e modela a aerodinâmica discutida no Capítulo 2, usou-se o próprio ambiente para sintonizar o controle e o estimador por tentativa e erro. O ambiente foi mantido nas configurações padrão para o pacote do Parrot.

### 5.1 Observador

Implementou-se o observador de estados proposto no Capítulo 4 no modelo de simulação do Parrot no Simulink. O estimador proposto está representado em tempo contínuo. No entanto o ciclo de atuação do controle de vôo do Mambo é de tempo discreto com período de amostragem  $T_s = 5 \cdot 10^{-3}$  s. Portanto é necessário adaptar o processo de integração da orientação estimada  $\hat{R}$  e do viés  $\hat{\mathbf{b}}$ . Para o viés

$$\hat{\mathbf{b}}_{k+1} = \hat{\mathbf{b}}_k + T_s \dot{\hat{\mathbf{b}}}_k \quad (5.1)$$

onde  $\dot{\hat{\mathbf{b}}}_k$  é dado na equação 4.10.

Para  $\hat{R}$  uma integração direta não é possível, pois isso levaria  $\hat{R}_{k+1}$  à valores não pertencentes ao  $SO(3)$ , deixando de representar uma rotação. Uma alternativa retirada de (CORKE, 2017) é tratar a integração como a realização de rotações incrementais

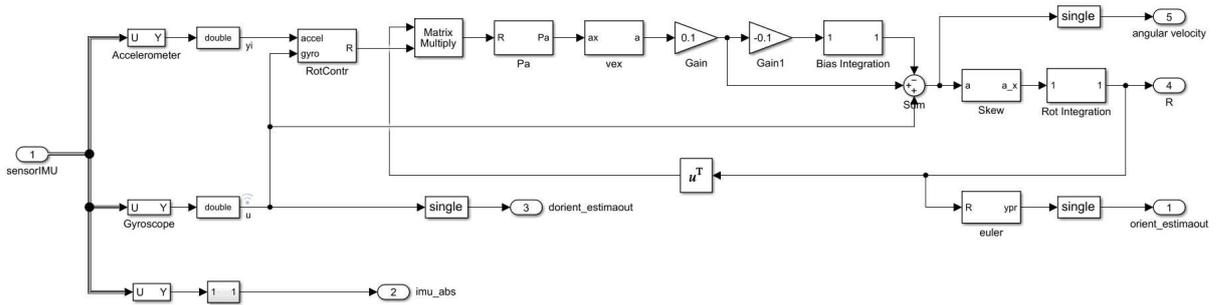
$$\hat{R}_{k+1} = \hat{R}_k \exp([\hat{\omega}_k]_{\times} \cdot T_s) \quad (5.2)$$

onde  $\hat{\omega}_k$  é dado em 4.10. Já a exponencial matricial foi implementada usando a equação de Rodrigues

$$\exp([x]_{\times}) = \begin{cases} \mathbf{I}_3, & x = 0 \\ \mathbf{I}_3 + \frac{\sin \|x\|}{\|x\|} [x]_{\times} + \frac{1 - \cos \|x\|}{\|x\|^2} [x]_{\times}^2, & x \neq 0 \end{cases} \quad (5.3)$$

A implementação realizada segue na Figura 5. Os ganhos  $k_P = 0.1$  e  $k_I = 0.1$  do observador foram sintonizados por tentativa e erro, partindo-se de um valor muito pequeno de ganhos e aumentando gradualmente as escolhas até a atitude estimada acompanhasse a evolução real dos ângulos de saída do modelo dinâmico do quadrotor. Para tal implementou-se as leis de controle 3.23 (detalhes da implementação seguem na próxima seção) e fez-se o

Figura 5 – Diagrama de blocos do estimador de atitude no Simulink.



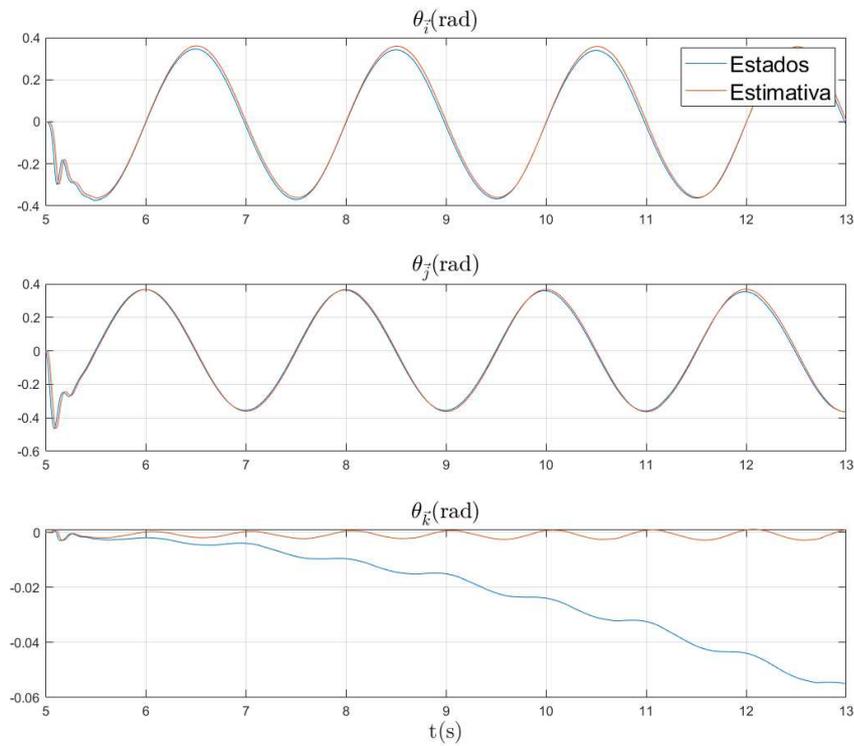
Fonte: Autoria própria.

drone acompanhar uma referência de atitude dada por

$$\begin{bmatrix} \theta_i \\ \theta_j \\ \theta_k \end{bmatrix} = \begin{bmatrix} \frac{\pi}{9} \sin(\pi t) \\ \frac{\pi}{9} \cos(\pi t) \\ 0 \end{bmatrix} \quad (5.4)$$

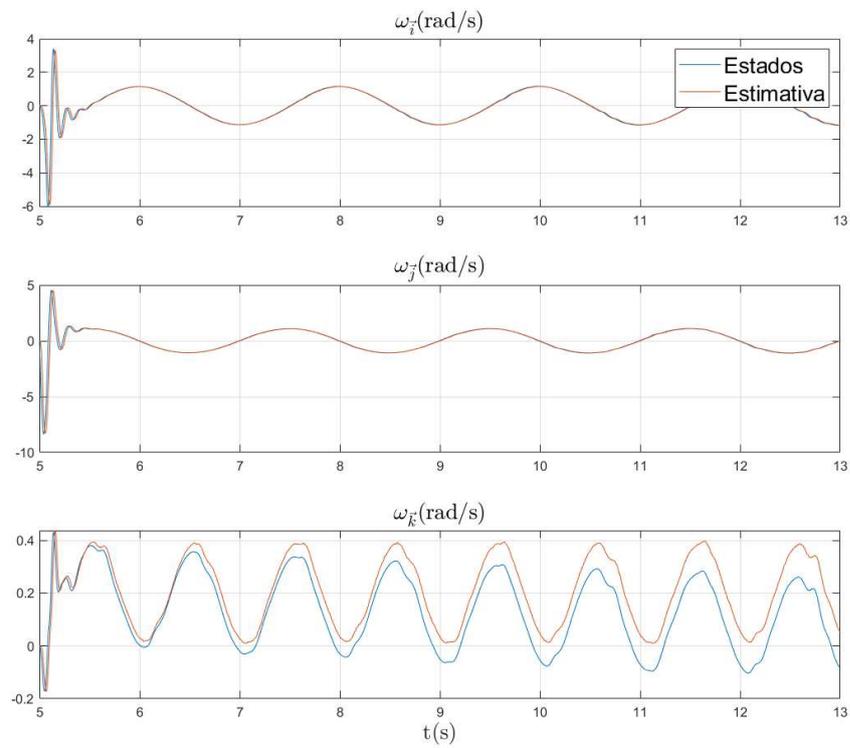
O resultado para os ganhos do observador da Figura 5 segue nas Figuras 6, 7 e 8.

Figura 6 – Estimativa da orientação frente aos estados.



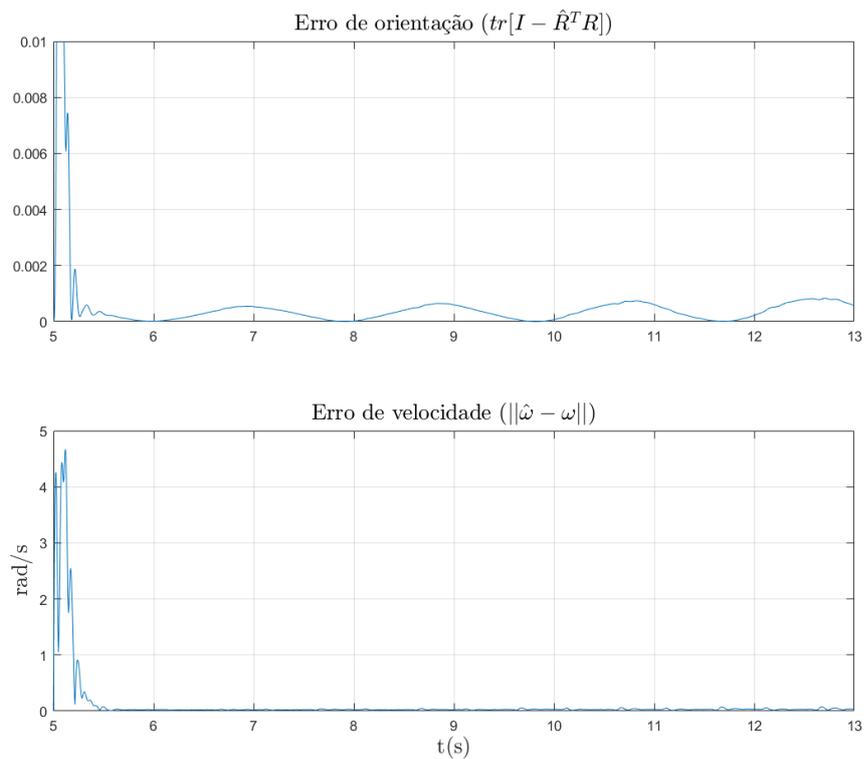
Fonte Autoria própria

Figura 7 – Estimativa da velocidade angular frente aos estados.



Fonte Autoria própria

Figura 8 – Erros de estimativa



Fonte Autoria própria

## 5.2 Controle

O controlador foi implementado em substituição ao controle de orientação original do pacote do Mambo. O bloco inserido no modelo segue na Figura 9.

O código contido no bloco MATLAB Function implementa as equações 3.23

```
function [M,dJ] = control(Rd,wd,dwd,R,w,J)

G = diag([1 1 1]);
kR = 0.05;
kW = 0.0025;
kJ = 0.001;
c = 1.5;
d = 0.001;
s = 0.001;
e = 0.1;

RT = transpose(R); RdT = transpose(Rd);
wT = transpose(w); wx = skew(w);

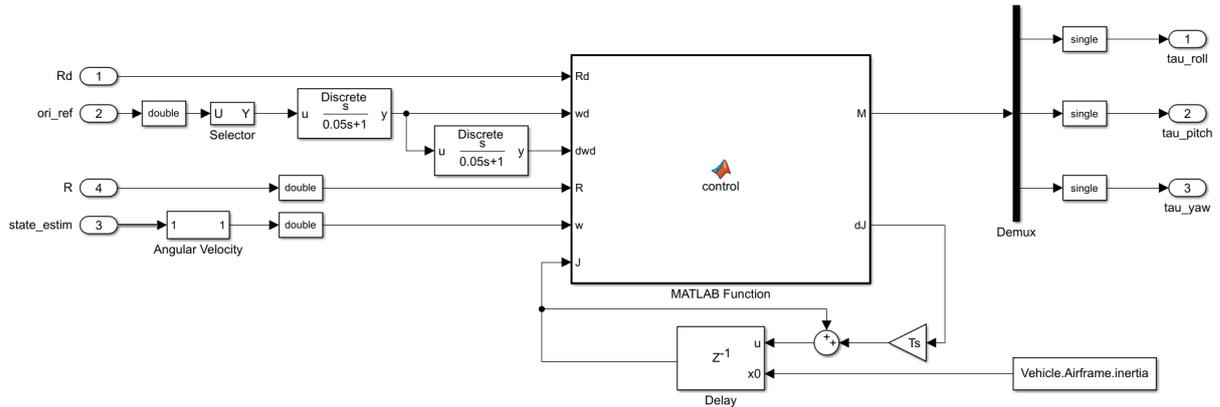
ad = -wx*RT*Rd*wd + RT*Rd*dwd;
ew = w - RT*Rd*wd;
eR = 0.5*vex(G*RdT*R - RT*Rd*G);
eA = ew + c*eR;

eAx = skew(eA);
adT = transpose(ad); eAT = transpose(eA);

v = -d^2*eA/(d*norm(eA) + e);
M = -kR*eR - kW*ew + wx*J*w + J*ad + v;
dJ = (kJ/2)*(-ad*eAT - eA*adT + w*wT*eAx - eAx*w*wT - 2*s*J);
```

Como mostrado na Figura 9, a integração discreta de  $\dot{J}$  foi feita da mesma maneira que para  $\hat{\mathbf{b}}$  no observador. Já as velocidades e acelerações angulares desejadas foram calculadas derivando a referência de atitude com uma derivada filtrada. A razão para tal é que a referência de atitude pode conter mudanças abruptas à depender da missão. O sinal  $R_d$  é obtido em uma etapa externa ao controlador, onde usando a função `eul2rotm` do MATLAB se converte as referências de ângulos `ori_ref` na representação de Euler para uma matriz de rotação.

Figura 9 – Diagrama de blocos do controlador de atitude.

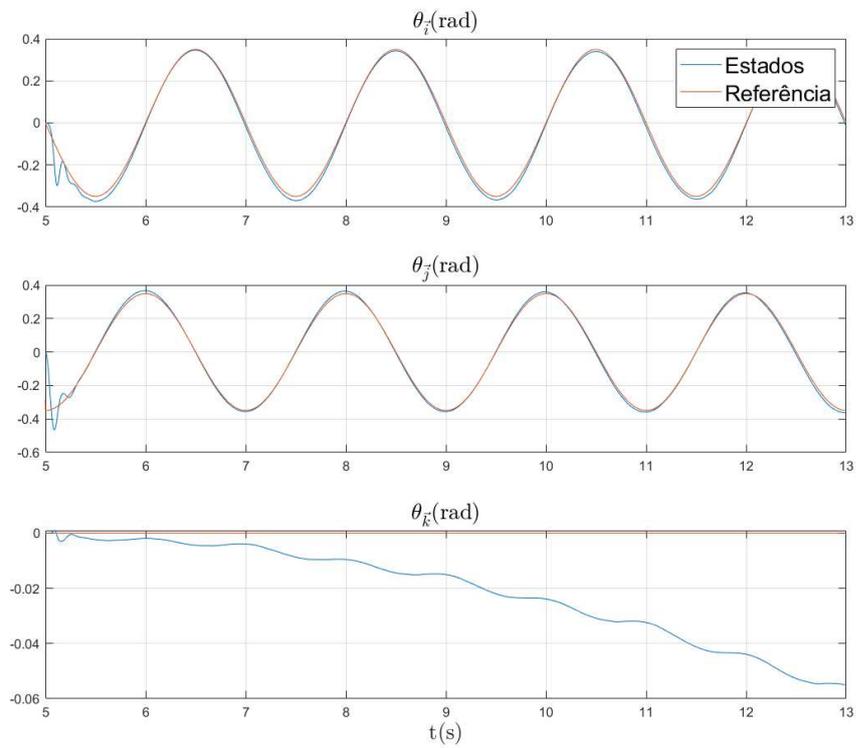


Fonte: Autoria própria

A sintonia dos ganhos do controlador foi feita por etapas. Primeiro alimentou-se o bloco do controlador diretamente com os estados do modelo da mecânica de corpo rígido do Mambo. Nesse caso, fixou-se  $c = 1.5$  e sintonizou-se os ganhos  $k_R$  e  $k_\omega$  para o controle geométrico puro, sem os termos associados ao controle adaptativo e robusto. A etapa adaptativa do controle não foi sintonizada apropriadamente pois a inércia simulada no Parrot é exatamente a disponibilizada no Pacote. Portanto, manteve-se  $k_J$  e  $\sigma$  pequenos como forma de compensar pelas perturbações sem levar a inércia estimada à valores muito distantes da inércia do modelo.

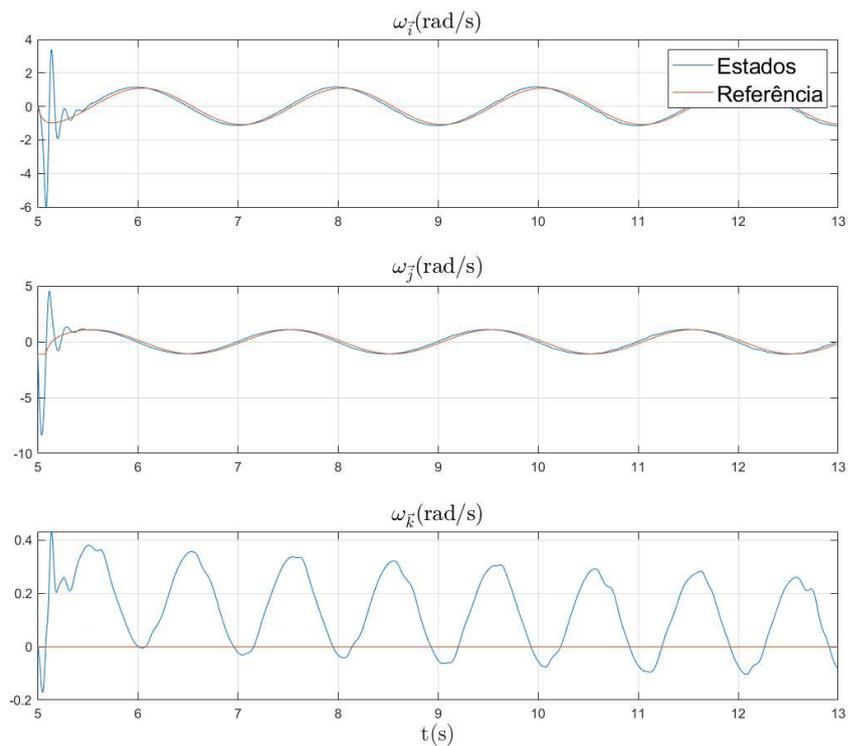
A sintonia da etapa robusta do controle foi feita uma vez que o observador implementado foi adicionado ao modelo e sintonizado. Partindo de valores da ordem de  $10^{-5}$ , aumentou-se  $\delta$  e  $\epsilon$  até que o controle fosse menos sensível aos ruídos que o observador não conseguiu filtrar. O sinal de referência usado foi o mesmo da seção anterior, dado por 5.4. Os resultados finais seguem nas Figuras 10 - 15.

Figura 10 – Orientação do drone com o controlador projetado.



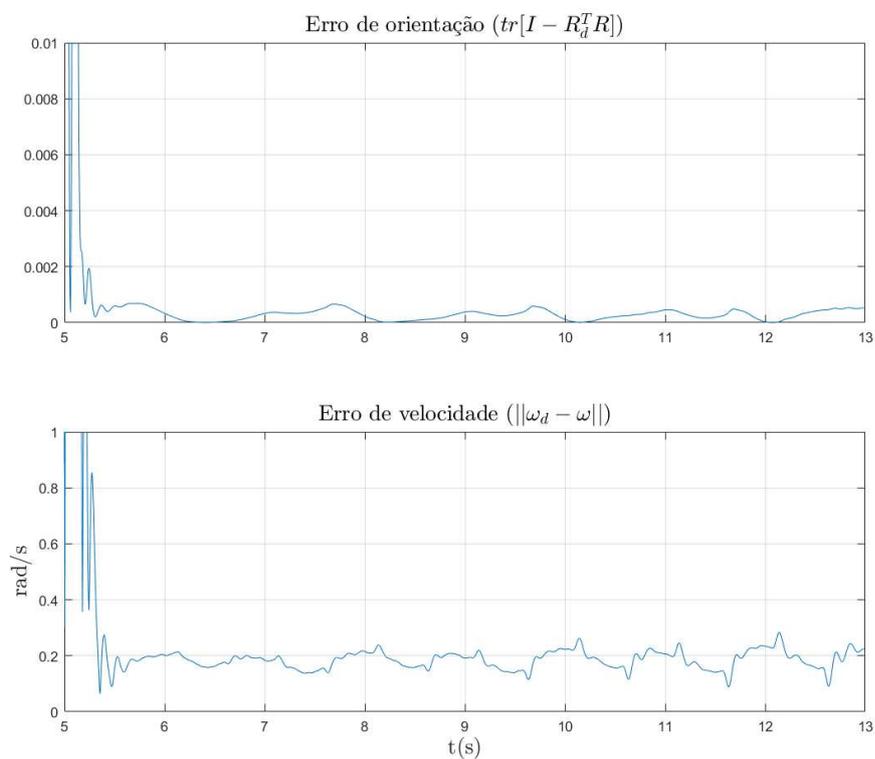
Fonte Autoria própria

Figura 11 – Velocidade angular do drone com o controlador projetado.



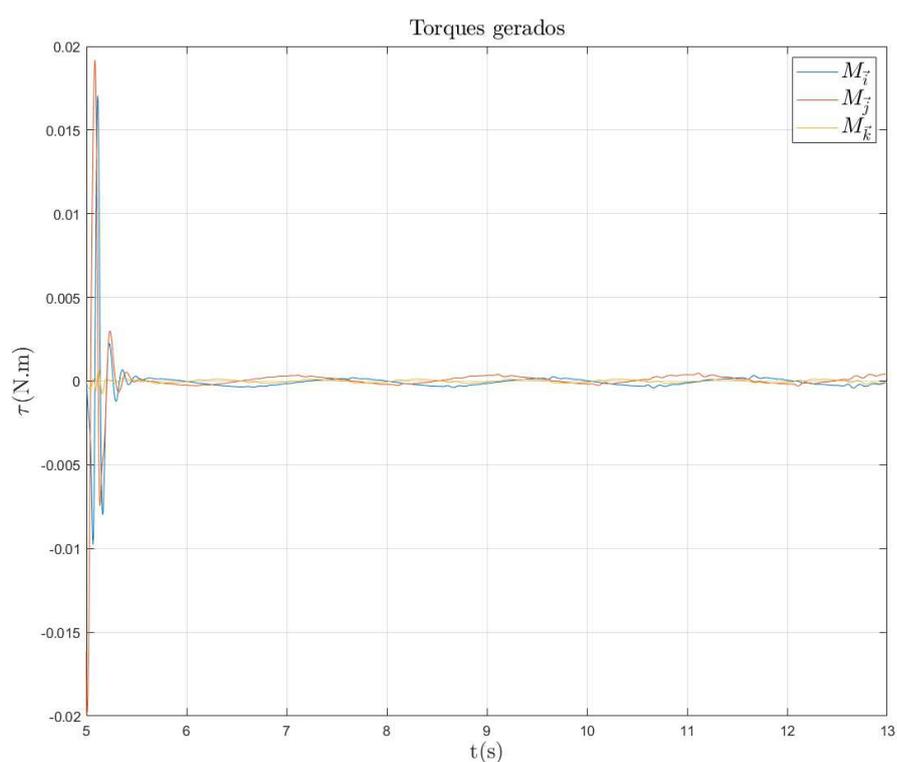
Fonte Autoria própria

Figura 12 – Erros de orientação e velocidade



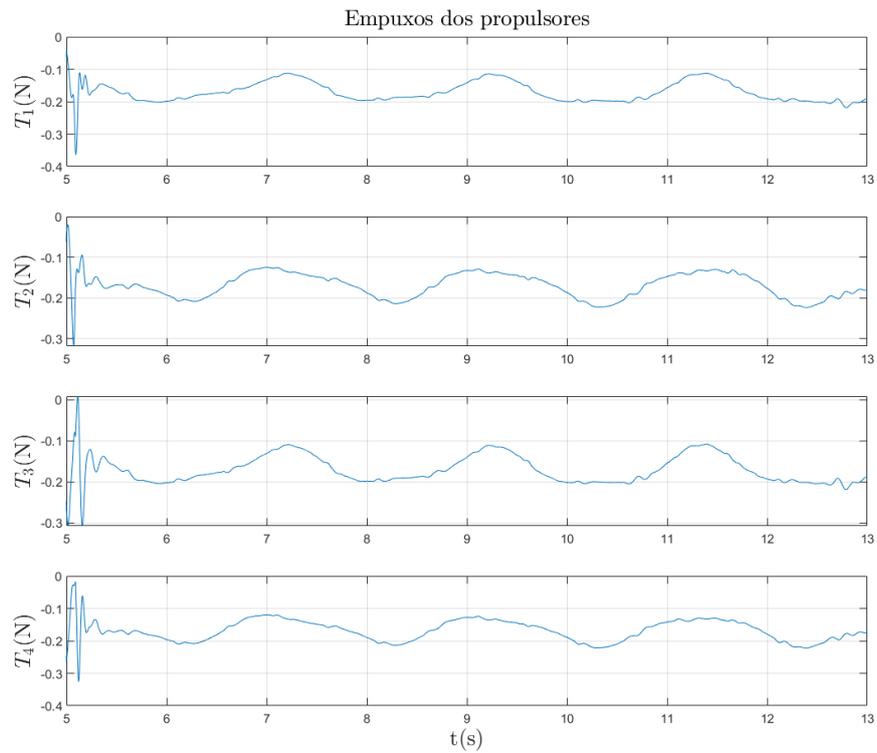
Fonte Autoria própria

Figura 13 – Sinal de controle produzido pelo controlador.



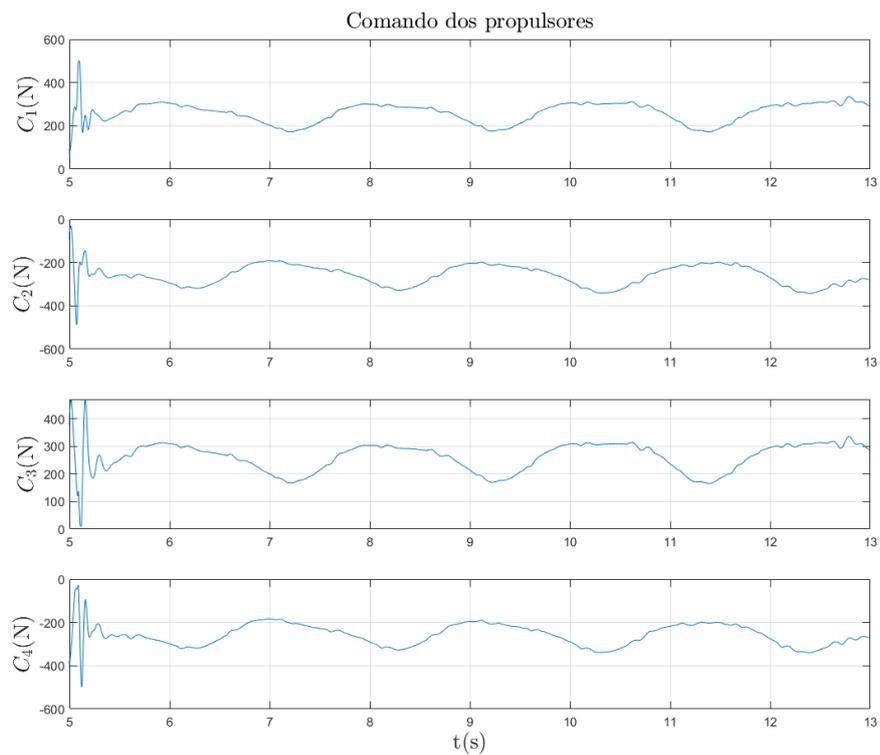
Fonte Autoria própria

Figura 14 – Empuxo comandado por propulsor.



Fonte Autoria própria

Figura 15 – Sinal de comando para os motores.



Fonte Autoria própria

### 5.3 Seguimento de caminho

Para testar o sistema com o observador e o controlador combinados em uma situação mais próxima de uma aplicação real, projetou-se uma etapa de controle de posição para gerar comandos de orientação e possibilitar que o quadrotor seguisse um caminho de referência. Sejam

$$e_{\mathbf{p}} = \mathbf{p}_d - \mathbf{p} \quad (5.5)$$

$$e_{\mathbf{v}} = \mathbf{v}_d - \mathbf{v} \quad (5.6)$$

os erros de posição e velocidade, onde  $\mathbf{p}_d$  e  $\mathbf{v}_d$  são a posição e a velocidade desejadas, respectivamente. Dada a dinâmica translacional 2.16, projeta-se a força desejada  $-\vec{F}_d = F_d R_d \vec{k}_B$  que leva os erros  $(e_{\mathbf{p}}, e_{\mathbf{v}})$  à zero, conforme proposto em (LEE et al., 2013)

$$\vec{F}_d = -m\mathbf{p}_d + mD^2 e_{\mathbf{p}} + m\vec{g} - \kappa_p r_{\mathbf{p}} - \kappa_i \int_0^t r_{\mathbf{p}} d\tau \quad (5.7)$$

onde

$$r_{\mathbf{p}} = e_{\mathbf{v}} + D e_{\mathbf{p}} \quad (5.8)$$

é chamado de erro de modo deslizante.  $D$ ,  $\kappa_p$  e  $\kappa_i$  são ganhos positivos a serem sintonizados sendo  $D$  uma matriz diagonal positiva e  $\kappa_p$  e  $\kappa_i$  escalares. Pode-se obter  $R_d$  de  $\vec{F}_d$

$$\vec{k}_{Bd} = \frac{\vec{F}_d}{\|\vec{F}_d\|}$$

$$\vec{j}_{Bd} = \frac{\vec{k}_{Bd} \times \vec{i}_{Bd}^*}{\|\vec{k}_{Bd} \times \vec{i}_{Bd}^*\|}$$

$$R_d = \begin{bmatrix} \vec{j}_{Bd} \times \vec{k}_{Bd} & \vec{j}_{Bd} & \vec{k}_{Bd} \end{bmatrix} \quad (5.9)$$

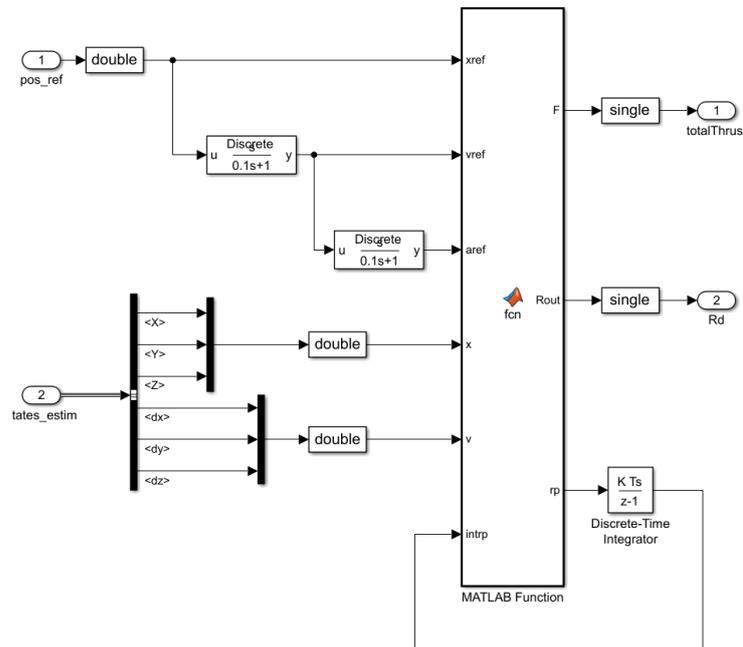
onde  $\vec{i}_{Bd}^*$  é a direção desejada da projeção de  $\vec{i}_B$  no plano  $xy$  do referencial  $\{I\}$ , ou seja, a direção para onde deve apontar a frente do drone.

Para suavizar o sinal  $R_d$  produzido, fez-se o planejamento de atitude numa versão simplificada do proposto por (INVERNIZZI; LOVERA; ZACCARIAN, 2020), acrescentando um estado a mais entre a atitude produzida pelas equações acima e a atitude enviada ao controlador de orientação.

$$\dot{R}_p = R_p[\kappa_r(R_p^T R_d - R_d^T R_p)^\vee]_\times, \quad R_p(0) = \mathbf{I}_3 \quad (5.10)$$

para um ganho  $\kappa_r$  positivo. A implementação desse controle de posição com planejamento de atitude segue na Figura 16.

Figura 16 – Diagrama de blocos do controle de posição.



Fonte Autoria própria

O código implementado no bloco MATLAB Function foi

```
function [F,Rout,rp] = fcn(xref,vref,aref,x,v,intrap)

persistent R w
if isempty(R)
    w = [0;0;0];
    R = eye(3);
end

m = 0.0630;
g = 9.8100;
D1 = diag([0.1 0.1 0.6]);

ex = xref - x;
ev = vref - v;

rp = ev + D1*ex;
Fd = -m*aref + m*D1^2*ex + m*[0;0;g] - .17*rp - .01*intrap;

F = -sqrt(sum(Fd.^2));

b3d = Fd/(-F);
```

```

b2d = cross(b3d, [1;0;0]);
b2d = b2d/sqrt(sum(b2d.^2));

Rd = [cross(b2d,b3d),b2d,b3d];
RdT = transpose(Rd);
RT = transpose(R);

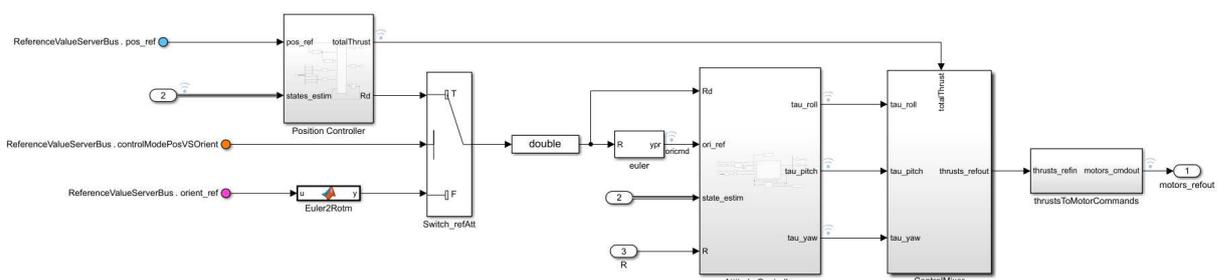
w = - 0.01*vex(RdT*R - RT*Rd);
if(norm(w) == 0)
    expw = eye(3);
else
    expw = eye(3) + sin(norm(w))/norm(w)*skew(w) + ...
        (1 - cos(norm(w)))/norm(w)^2*skew(w)^2;
end
R = R*expw;
Rout = R;

```

O bloco completo do controle de pose é apresentado na Figura 17. Os sinais de posição e velocidade linear usados para o controle de posição são fornecidos pelo estimador de posição já implementado pelo pacote do Mambo, que usa o fluxo óptico para estimar a posição e velocidade no plano  $xy$  e as leituras do sonar e do barômetro para estimar a altitude e velocidade no eixo  $\vec{k}_I$ . Essa estimação é feita por um par de filtros de Kalman projetados à partir do modelo linearizado de translação e de dados estatísticos dos ruídos dos sensores que são coletados durante a calibração.

A referência escolhida foi uma trajetória no plano  $z_I = -1.5$  (o eixo  $\vec{k}_I$  aponta para o solo) em forma de um quadrado com bordas arredondadas, como apresentado na Figura 18, que deve ser percorrida no sentido anti-horário com velocidade linear constante de 0.25 m/s. Para isso implementou-se uma função para gerar essa referência dinamicamente dentro do bloco Path Planning, chamada RoundedSquare. Detalhes desses dois blocos seguem nas Figuras 19 e 20. Dentro do RoundedSquare implementou-se o seguinte código

Figura 17 – Controlador de pose completo.



Fonte: Autoria própria.

```
function [x, y] = next_square_point(straightLength, arcRadius, speed)
persistent pos vel dt

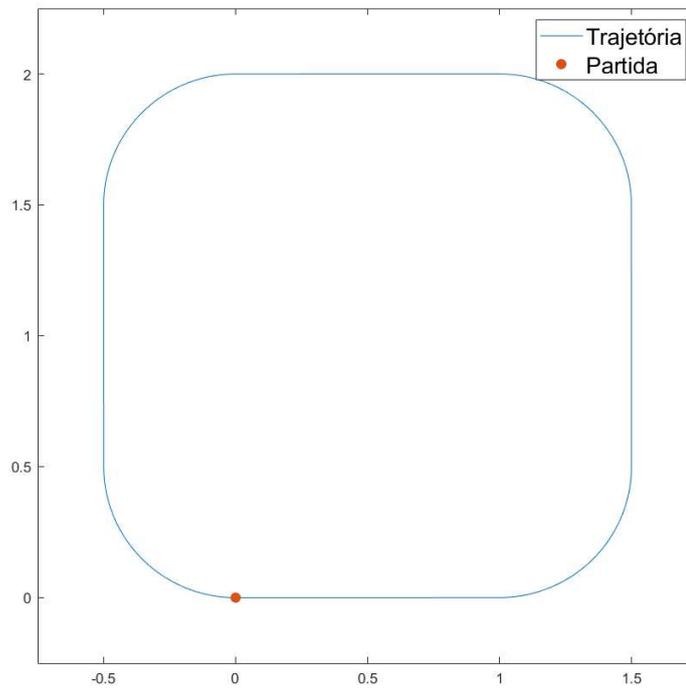
if isempty(pos)
    dt = 0.005;
    pos = [0;0];
    vel = speed*[-1;0];
end

if(pos(1) > 0 && pos(1) < straightLength && pos(2) < 0)
    vel = speed*[-1;0];
elseif(pos(1) > 0 && pos(1) < straightLength && ...
    pos(2) > straightLength + 2*arcRadius)
    vel = speed*[1;0];
elseif(pos(1) < -arcRadius && pos(2) > arcRadius && ...
    pos(2) < straightLength + arcRadius)
    vel = speed*[0;1];
elseif(pos(1) > straightLength + arcRadius && pos(2) > arcRadius && ...
    pos(2) < straightLength + arcRadius)
    vel = speed*[0;-1];
elseif((pos(1) < 0 && (pos(2) < arcRadius || ...
    pos(2) > straightLength + arcRadius)) || ...
    pos(1) > straightLength && (pos(2) < arcRadius || ...
    pos(2) > straightLength + arcRadius))
    vel = rot2(-speed/arcRadius*dt)*vel;
end

pos = pos + vel*dt;

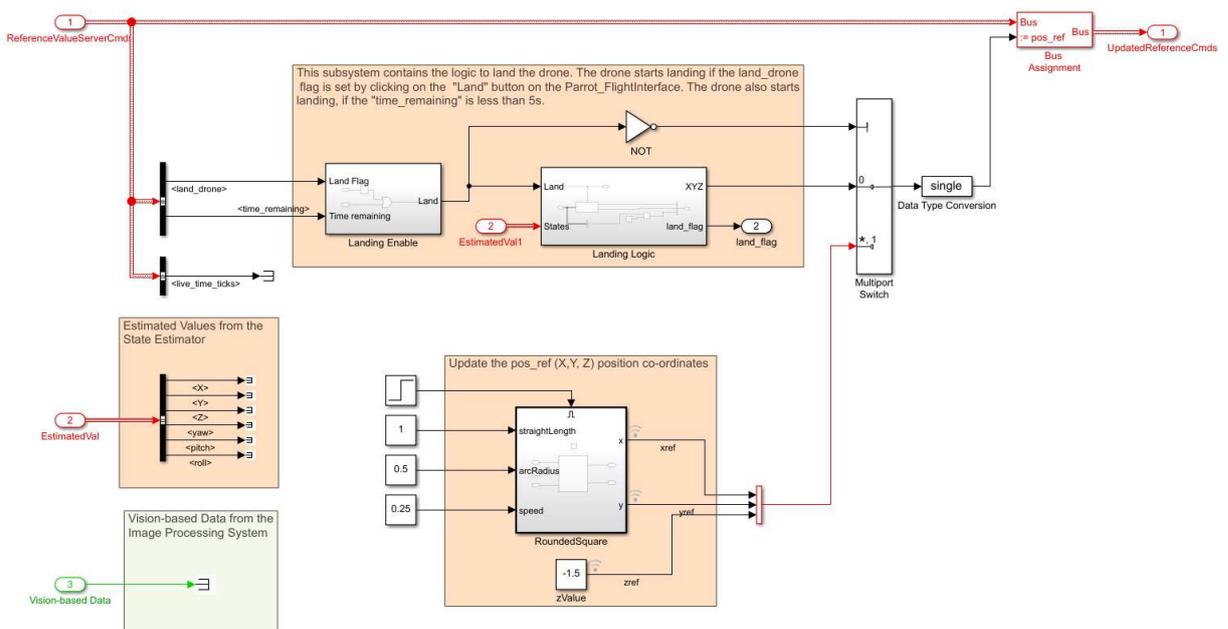
x = pos(1); y = pos(2);
```

Figura 18 – Trajetória de referência.



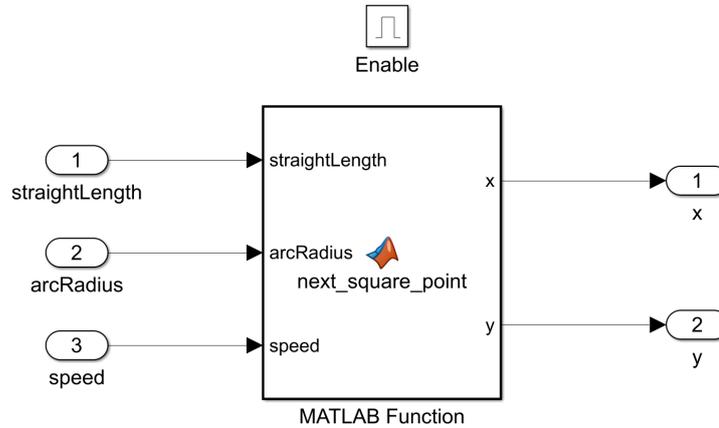
Fonte: Autoria própria.

Figura 19 – Diagrama de blocos do planejamento de trajetória.



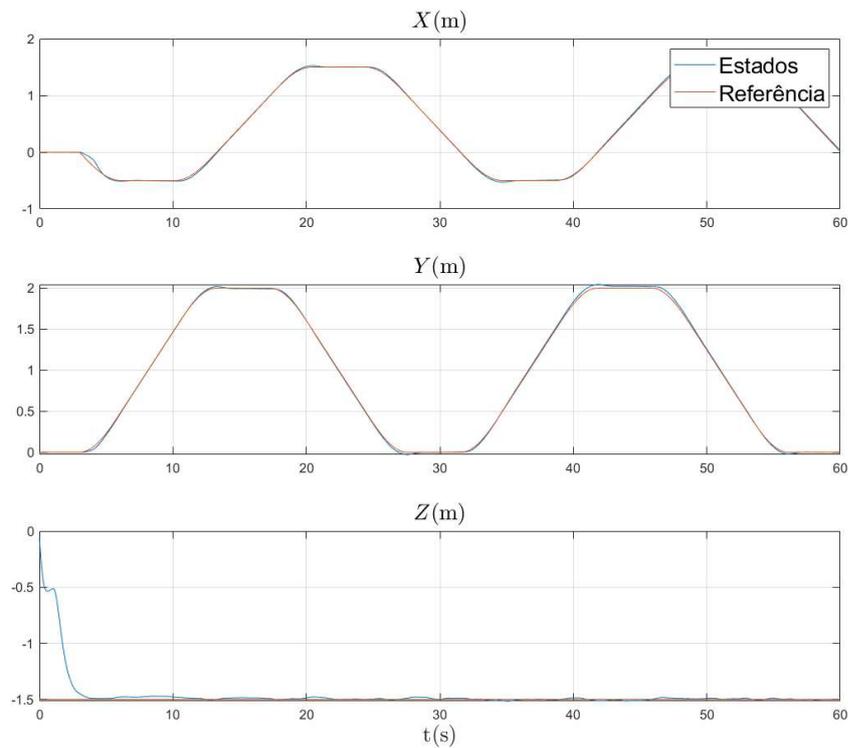
Fonte: Autoria própria.

Figura 20 – Diagrama de blocos da função geradora de caminho.



Fonte: Autoria própria.

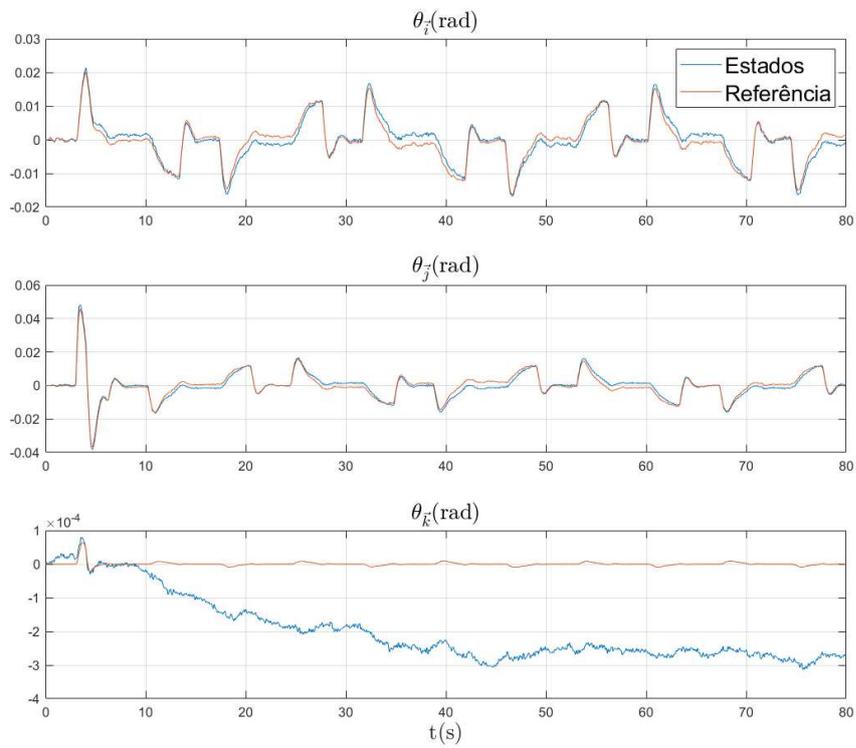
Figura 21 – Posição do quadrotor no tempo.



Fonte: Autoria própria.

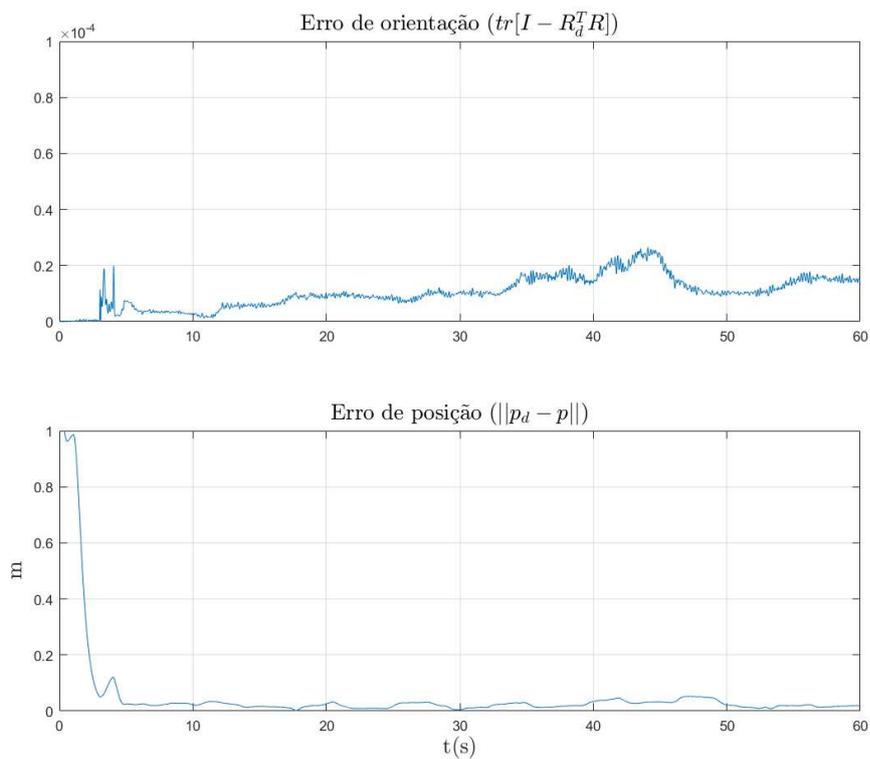
O quadrotor deve primeiro decolar e, depois de atingida a altura de 1.5 m, começa a mover-se seguindo a referência. Os resultados para esse cenário são apresentados nas Figuras 21 - 24.

Figura 22 – Comandos de orientação gerados.



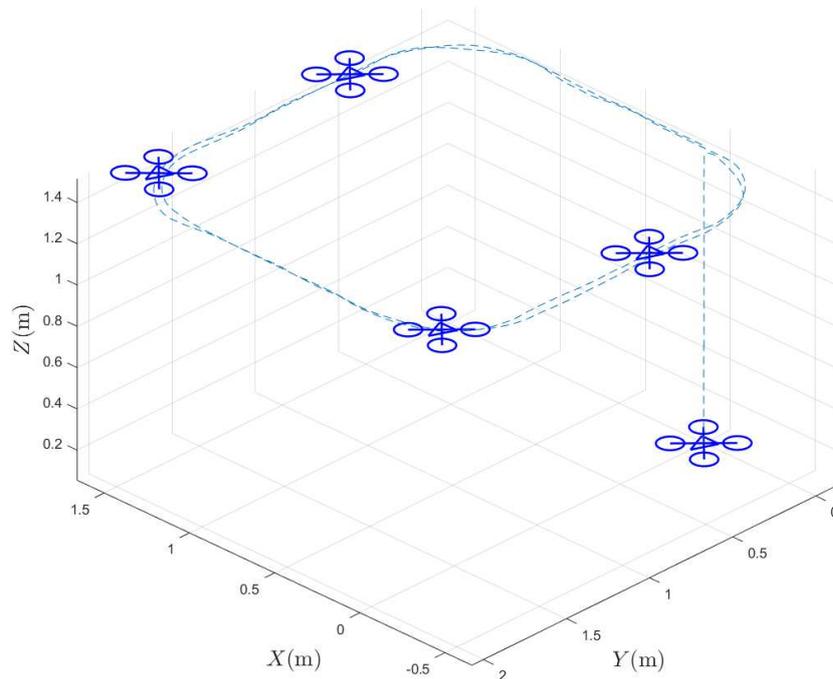
Fonte: Autoria própria.

Figura 23 – Erros de pose.



Fonte: Autoria própria.

Figura 24 – Posição do quadrotor no espaço.



Fonte: Autoria própria.

## 5.4 Análise dos resultados

### 5.4.1 Observador

Como é possível observar nas Figuras 6 e 7, o estimador projetado conseguiu rastrear com precisão os ângulos e as velocidades em torno dos eixos  $\vec{i}$  e  $\vec{j}$  mas não convergiu para o eixo  $\vec{k}$ . Isso ocorre porque com apenas uma medida vetorial de direção (a gravidade medida pelo acelerômetro), fez-se necessário integrar diretamente a leitura da velocidade angular em torno de  $\vec{k}$  fornecida pelo giroscópio, levando à um erro crescente na medição de  $\theta_{\vec{k}}$  e, por consequência, um distanciamento crescente entre a estimativa de  $\theta_{\vec{k}}$  e o valor real. Esse problema é abordado em (MAHONY; HAMEL; PFLIMLIN, 2008) e a solução proposta é a de um filtro complementar explícito com a inclusão de uma segunda medida vetorial que pode ser fornecida por um segundo acelerômetro ou um magnetômetro. Soluções que dispensam o uso de uma medida adicional também são possíveis. (MARTIN; SARRAS, 2018) propõe um método onde, dada a existência de uma excitação angular suficiente, é possível reconstruir a orientação em todos os eixos.

Como o problema do rastreamento do ângulo  $\theta_{\vec{k}}$  é inerente à formulação 4.10 quando aplicada ao caso de uma única medida vetorial de direção, os gráficos 8 e 12 representam os erros apenas em relação aos eixos  $\vec{i}$  e  $\vec{j}$ . Como visto no caso do rastreamento de trajetória da

seção anterior, quando a excitação angular da referência não é persistente (como era para uma referência senoidal) o desvio da estimativa de  $\vec{\theta}_k$  cresce mais lentamente.

### 5.4.2 Controle

O controle implementado trouxe resultados satisfatórios no rastreo da referência de orientação e velocidade angular, como pode ser visto na Figura 12. Um dos problemas observados é a sensibilidade do controlador as variações de aceleração angular, o que implicou na necessidade do uso de derivadas filtradas para manter a estabilidade do sistema. Como o observador não consegue estimar corretamente o viés em torno de  $\vec{k}$ , é possível observar na Figura 10 que  $\vec{\theta}_k$  progressivamente se distancia da referência. Esse problema não é observado quando se alimenta o controle diretamente com os estados gerados pelo modelo da mecânica de corpo rígido do Mambo.

Na Figura 12 é possível ver que o erro de rastreo de velocidade angular estabiliza em 0,2 rad/s. Isso é consequência da filtragem da aceleração, que impõe um atraso no rastreo da velocidade, como exposto na Figura 11, provocando o erro observado.

Dos gráficos 13, 14 e 15 é possível concluir que nem o transiente inicial nem o rastreo da referência de atitude provocam variações abruptas nos sinais de controle. Do ponto de vista do uso do sistema implementado em uma situação real de controle de pose de um drone, esse comportamento é positivo pois minimiza as chances de que um erro de configuração grande provoque movimentos bruscos o suficiente para levar o quadrotor às singularidades (mais detalhes no Apêndice).

### 5.4.3 Seguimento de caminho

O seguimento de caminho foi implementado como uma forma de demonstrar o controle e o estimador implementados numa situação mais próxima da realidade, onde o drone executa uma missão. A Figura 22 mostra que a combinação de um controle geométrico de posição com o controle geométrico de orientação permitem o seguimento de caminho com erros mínimos sem geração de comandos de orientação com variações bruscas. A eficiência dessa estratégia fica evidente no gráfico 21 e na variação dos erros no tempo exposta na Figura 23. A Figura 24 traz uma representação tridimensional da trajetória realizada pelo quadrotor no espaço como uma forma de melhor visualizar a missão realizada.

## 6 Conclusão

Neste trabalho de conclusão de curso foi discutido a aplicação de técnicas não-lineares ao problema de estimação e controle de atitude de um quadricóptero, em especial o controle geométrico de sistemas mecânicos. Modelou-se a dinâmica do quadrotor representando a dinâmica rotacional por matrizes de rotação e as velocidades por mapeamentos anti-simétricos dos vetores. Derivou-se dessa modelagem as equações de controle e de estimação pela aplicação do teorema de Lyapunov para estabilidade de sistemas não-lineares e definiu-se as condições para convergência.

Também foi discutido nesse trabalho a modelagem dos sensores, em específico os sensores disponíveis no minidrone Parrot Mambo. A aerodinâmica das hélices foi modelada e dela foi possível definir como os sinais de controle são mapeados em velocidades de hélice.

O observador de estados e o controlado foram implementados e testados no ambiente de simulação computacional Simulink, onde os recursos de visualização de sinais e formulação de cenários foram usados para sintonizar os sistemas projetados.

Por fim, testou-se o sistema completo para uma missão de seguimento de caminho no espaço por meio da implementação de um controlador de posição, que foi colocado entre as referências de posição e o controlador de orientação desenvolvido, num arranjo em cascata. Os dados coletados da simulação desse cenário apontaram para um bom desempenho do sistema projetado no rastreamento da trajetória planejada e na suavidade do movimento.

O método de sintonia empregado foi baseado tão somente nas condições de estabilidade e nos recursos de simulação, que permitiram uma sintonia por tentativa e erro. Essa abordagem não é ideal e expõe uma das principais desvantagens da aplicação do controle geométrico, que é a dificuldade de aplicar uma sintonia baseada em controle clássico, à partir dos tempos de resposta, sobresinal, tempo de subida, etc. Para a sintonia de controladores geométricos existe na literatura a aplicação de diversas técnicas, que vão desde o uso de aprendizagem por reforço (LEE; BANG, 2019) à extensão do problema para considerar a dinâmica de ganhos variáveis, num processo de sintonia online dos ganhos (VANG; TRON, 2022) (INVERNIZZI; PANZA; LOVERA, 2020).

## 6.1 Trabalhos futuros

Como sugestão de trabalhos e melhorias futuras nessa linha de pesquisa é possível citar

- Estudo mais aprofundado dos formalismos do controle geométrico de sistemas mecânicos;
- Implementação do controle projetado numa situação real de voo;
- Estudo e implementação de técnicas de sintonia de controladores geométricos, sejam elas técnicas offline e/ou online.

# A Análise da estabilidade

## A.1 Controle de atitude

### A.1.1 Erro de configuração

A função erro de configuração  $\Phi : SO(3) \times SO(3) \rightarrow \mathbb{R}^1$  é uma conhecida métrica sobre elementos do  $SO(3)$  que é bi-invariante, positiva semi-definida e diferenciável (BULLO; LEWIS, 2005), dada por

$$\Phi(R, R_d) = \frac{1}{2} \text{tr}[G(\mathbf{I}_3 - R_d^T R)] \quad (\text{A.1})$$

onde  $\text{tr}(A)$  é o traço da matriz  $A$  e  $G \in \mathbb{R}^{3 \times 3}$  é uma matriz diagonal positiva. Essa função pode ser interpretada como uma forma de medir em valor absoluto a distância entre as rotações  $R$  e  $R_d$ , ou seja, entre o estado e a referência de atitude. Deseja-se definir vetores de erro de configuração e velocidade no espaço tangente de  $SO(3)$  usando as derivadas de  $\Phi(R, R_d)$  à partir dos quais as entradas de controle são projetadas através de uma análise de Lyapunov no  $SO(3)$ . Assim sendo

$$\begin{aligned} \frac{d}{dt} \Phi(R, R_d) &= -\frac{1}{2} \text{tr}[G(\frac{d}{dt}(R_d^T R))] = \\ &= -\frac{1}{2} \text{tr}[G(\frac{d}{dt} R_d^T R + R_d^T \frac{d}{dt} R)] = \\ &= -\frac{1}{2} \text{tr}[G(-[\omega_d]_{\times} R_d^T R + R_d^T R[\omega]_{\times})] \end{aligned} \quad (\text{A.2})$$

usando do fato que  $\frac{d}{dt} R_d^T = (\frac{d}{dt} R_d)^T = (R_d[\omega_d]_{\times})^T = [\omega_d]_{\times}^T R_d^T$ . Observando a definição do mapa  $[\cdot]_{\times}$ , é possível concluir que  $[a]_{\times}^T = -[a]_{\times}$ . Portanto  $\frac{d}{dt} R_d^T = -[\omega_d]_{\times} R_d^T$ . Retomando

$$\begin{aligned} \frac{d}{dt} \Phi(R, R_d) &= -\frac{1}{2} \text{tr}[G(R_d^T R[\omega]_{\times} - [\omega_d]_{\times} R_d^T R)] = \\ &= -\frac{1}{2} \text{tr}[G R_d^T R([\omega]_{\times} - R^T R_d[\omega_d]_{\times} R_d^T R)] \end{aligned} \quad (\text{A.3})$$

Usando a propriedade da álgebra  $\mathfrak{so}(3)$   $[Aa]_{\times} = A[a]_{\times} A^T$ , onde  $A \in SO(3)$  e  $a \in \mathfrak{so}(3)$

$$\begin{aligned} \frac{d}{dt} \Phi(R, R_d) &= -\frac{1}{2} \text{tr}[G R_d^T R([\omega]_{\times} - [R^T R_d \omega_d]_{\times})] = \\ &= -\frac{1}{2} \text{tr}[G R_d^T R[\omega - R^T R_d \omega_d]_{\times}] \end{aligned} \quad (\text{A.4})$$

onde valeu-se do fato que  $[a]_{\times} + [b]_{\times} = [a + b]_{\times}$ .

A expressão  $-\frac{1}{2} \text{tr}([a]_{\times} [b]_{\times})$  é equivalente ao produto interno de  $a$  por  $b$ , para  $a, b \in \mathbb{R}^3$

$$-\frac{1}{2} \text{tr}([a]_{\times} [b]_{\times}) = \langle a, b \rangle = \langle b, a \rangle = a^T b = b^T a$$

Toda matriz quadrada pode ser decomposta em uma parte simétrica e uma outra anti-simétrica dadas respectivamente por  $P_s(A) = \frac{1}{2}(A + A^T)$  e  $P_a(A) = \frac{1}{2}(A - A^T)$ . Essas parcelas são ortogonais quanto ao produto interno. Aplicando estas propriedades, podemos substituir  $GR_d^T R$  por sua projeção anti-simétrica  $P_a(GR_d^T R)$ :

$$\begin{aligned} \frac{d}{dt}\Phi(R, R_d) &= -\frac{1}{2}\text{tr} \left[ \frac{1}{2}(GR_d^T R - R^T R_d G)[\omega - R^T R_d \omega_d]_{\times} \right] = \\ &= \left\langle \left( \frac{1}{2}(GR_d^T R - R^T R_d G) \right)^{\vee}, (\omega - R^T R_d \omega_d) \right\rangle = \langle e_R, e_{\omega} \rangle \end{aligned} \quad (\text{A.5})$$

sendo

$$e_{\omega} = \omega - R^T R_d \omega_d \quad (\text{A.6})$$

e

$$e_R = \left( \frac{1}{2}(GR_d^T R - R^T R_d G) \right)^{\vee} \quad (\text{A.7})$$

os erros de rastreamento de rotação e velocidade angular, respectivamente.  $e_R$  foi construído valendo-se do fato que  $G$  é uma matriz diagonal.

Antes de definir partir para a análise de Lyapunov, é importante determinar as derivadas temporais dos termos  $e_R$  e  $e_{\omega}$  obtidos.

$$\begin{aligned} \dot{e}_R &= \frac{d}{dt} \left( \frac{1}{2}(GR_d^T R - R^T R_d G) \right)^{\vee} = \\ &= \left( \frac{1}{2} \left( G \frac{d}{dt}(R_d^T R) - \frac{d}{dt}(R^T R_d) G \right) \right)^{\vee} \end{aligned} \quad (\text{A.8})$$

O termo  $\frac{d}{dt}(R_d^T R)$  já foi calculado anteriormente no processo de obtenção de  $\frac{d}{dt}\Phi(R, R_d)$ :  $\frac{d}{dt}(R_d^T R) = R_d^T R[\omega - R^T R_d \omega_d]_{\times} = R_d^T R[e_{\omega}]_{\times}$ . Assim  $\frac{d}{dt}(R^T R_d) = -[e_{\omega}]_{\times} R^T R_d$ . Substituindo esses termos:

$$\begin{aligned} \dot{e}_R &= \left( \frac{1}{2}(GR_d^T R[e_{\omega}]_{\times} - (-[e_{\omega}]_{\times} R^T R_d)G) \right)^{\vee} = \\ &= \left( \frac{1}{2}(GR_d^T R[e_{\omega}]_{\times} + [e_{\omega}]_{\times} R^T R_d G) \right)^{\vee} \end{aligned} \quad (\text{A.9})$$

Usando da propriedade  $[a]_{\times} A + A^T [a]_{\times} = [(\text{tr}(A)\mathbf{I}_3 - A)a]_{\times}$ , para  $a \in \mathfrak{so}(3)$  e  $A \in \mathbb{R}^{3 \times 3}$

$$\dot{e}_R = \left( \frac{1}{2}[(\text{tr}(R^T R_d G)\mathbf{I}_3 - R^T R_d G)e_{\omega}]_{\times} \right)^{\vee} \quad (\text{A.10})$$

como  $[\cdot]_{\times}$  e  $\cdot^{\vee}$  são mapeamentos inversos um do outro

$$\begin{aligned} \dot{e}_R &= \frac{1}{2}(\text{tr}(R^T R_d G)\mathbf{I}_3 - R^T R_d G)e_{\omega} = \\ &= E(R, R_d)e_{\omega} \end{aligned} \quad (\text{A.11})$$

onde  $E(R, R_d) = \frac{1}{2}(\text{tr}(R^T R_d G) \mathbf{I}_3 - R^T R_d G)$ .

Agora para o  $e_\omega$

$$\begin{aligned}
\dot{e}_\omega &= \frac{d}{dt}(\omega - R^T R_d \omega_d) = \dot{\omega} - \frac{d}{dt}(R^T R_d \omega_d) = \\
&= J^{-1}(M - [\omega]_\times J \omega) - (-[e_\omega]_\times R^T R_d \omega_d + R^T R_d \dot{\omega}_d) = \\
&= J^{-1}(M - [\omega]_\times J \omega) + [e_\omega]_\times R^T R_d \omega_d - R^T R_d \dot{\omega}_d = \\
&= J^{-1}(M - [\omega]_\times J \omega) + [\omega - R^T R_d \omega_d]_\times R^T R_d \omega_d - R^T R_d \dot{\omega}_d = \\
&= J^{-1}(M - [\omega]_\times J \omega) + ([\omega]_\times - [R^T R_d \omega_d]_\times) R^T R_d \omega_d - R^T R_d \dot{\omega}_d = \\
&= J^{-1}(M - [\omega]_\times J \omega) + ([\omega]_\times - R^T R_d [\omega_d]_\times R_d^T R) R^T R_d \omega_d - R^T R_d \dot{\omega}_d = \\
&= J^{-1}(M - [\omega]_\times J \omega) + [\omega]_\times R^T R_d \omega_d - R^T R_d [\omega_d]_\times \omega_d - R^T R_d \dot{\omega}_d \quad (\text{A.12})
\end{aligned}$$

Sabendo que  $[a]_\times b$ , para  $a, b \in \mathbb{R}^3$ , é o mesmo que  $a \times b$ . Portanto  $[\omega_d]_\times \omega_d = \omega_d \times \omega_d = 0$ . Logo

$$\begin{aligned}
\dot{e}_\omega &= J^{-1}(M - [\omega]_\times J \omega) + [\omega]_\times R^T R_d \omega_d - R^T R_d \dot{\omega}_d = \\
&= J^{-1}(M - [\omega]_\times J \omega) - \alpha_d \quad (\text{A.13})
\end{aligned}$$

onde  $\alpha_d = -[\omega]_\times R^T R_d \omega_d + R^T R_d \dot{\omega}_d$  e o termo  $\dot{\omega}_d$  é a aceleração angular desejada.

As funções  $\Phi(R, R_d)$ ,  $e_R$  e  $e_\omega$  definem os erros da dinâmica rotacional frente à referência definida por 3.2, e permitem formular uma função candidata de Lyapunov que será usada para encontrar a lei de controle  $M$  que estabiliza o sistema em torno de  $R_d(t)$ .

## A.1.2 Estabilidade do controle adaptativo

### A.1.2.1 Função candidata de Lyapunov

A função candidata de Lyapunov escolhida é

$$V = \frac{1}{2} \langle e_\omega, J e_\omega \rangle + k_R \Phi(R, R_d) + c \langle J e_\omega, e_R \rangle + \frac{1}{2k_J} \|\tilde{J}\|_F^2 \quad (\text{A.14})$$

onde  $k_R$ ,  $c$  e  $k_J$  são constantes positivas e  $\|\cdot\|_F$  é a norma de Frobenius dada por  $\|A\|_F = \sqrt{\text{tr}(A^T A)}$ . Derivando  $V$  no tempo

$$\dot{V} = \frac{1}{2} (\langle \dot{e}_\omega, J e_\omega \rangle + \langle e_\omega, J \dot{e}_\omega \rangle) + k_R \langle \dot{e}_R, e_\omega \rangle + c (\langle J \dot{e}_\omega, e_R \rangle + \langle J e_\omega, \dot{e}_R \rangle) + \frac{1}{k_J} \text{tr}(\tilde{J} \dot{\tilde{J}}) \quad (\text{A.15})$$

Como  $J$  é simétrica, é possível averiguar que  $\langle \dot{e}_\omega, J e_\omega \rangle = \langle e_\omega, J \dot{e}_\omega \rangle$

$$\langle \dot{e}_\omega, J e_\omega \rangle = \dot{e}_\omega^T J e_\omega = (J e_\omega)^T \dot{e}_\omega = e_\omega^T J \dot{e}_\omega = \langle e_\omega, J \dot{e}_\omega \rangle$$

Assim

$$\dot{V} = \langle e_\omega, J \dot{e}_\omega \rangle + k_R \langle \dot{e}_R, e_\omega \rangle + c (\langle J \dot{e}_\omega, e_R \rangle + \langle J e_\omega, \dot{e}_R \rangle) + \frac{1}{k_J} \text{tr}(\tilde{J} \dot{\tilde{J}}) \quad (\text{A.16})$$

Sabendo que  $k\langle a, b \rangle = \langle ka, b \rangle = \langle a, kb \rangle$  para  $k \in \mathbb{R}^1$ , bem como as propriedades comutativa

$$\langle a, b \rangle = \langle b, a \rangle$$

e distributiva

$$\langle a, b \rangle + \langle a, c \rangle = \langle a, (b + c) \rangle$$

do produto interno de vetores, pode-se agrupar alguns termos

$$\dot{V} = \langle e_\omega, J\dot{e}_\omega + k_R e_R \rangle + c(\langle J\dot{e}_\omega, e_R \rangle + \langle J e_\omega, \dot{e}_R \rangle) + \frac{1}{k_J} \text{tr}(\tilde{J}\dot{\tilde{J}}) \quad (\text{A.17})$$

De A.13

$$J\dot{e}_\omega = M - [\omega]_\times J\omega - J\alpha_d \quad (\text{A.18})$$

Escolhendo o torque como

$$M = -k_R e_R - k_\omega e_\omega + [\omega]_\times \bar{J}\omega + \bar{J}\alpha_d \quad (\text{A.19})$$

onde  $k_\omega$  é uma constante positiva. Substituindo A.19 na expressão de  $J\dot{e}_\omega$

$$\begin{aligned} J\dot{e}_\omega &= -k_R e_R - k_\omega e_\omega + [\omega]_\times \bar{J}\omega + \bar{J}\alpha_d - [\omega]_\times J\omega - J\alpha_d = \\ &= -k_R e_R - k_\omega e_\omega + [\omega]_\times (\bar{J} - J)\omega + (\bar{J} - J)\alpha_d = \\ &= -k_R e_R - k_\omega e_\omega - [\omega]_\times \tilde{J}\omega - \tilde{J}\alpha_d \end{aligned} \quad (\text{A.20})$$

Substituindo agora o  $J\dot{e}_\omega$  e  $\dot{e}_R$  (A.11) na derivada temporal de  $V$ :

$$\begin{aligned} \dot{V} &= \langle e_\omega, (-k_R e_R - k_\omega e_\omega - [\omega]_\times \tilde{J}\omega - \tilde{J}\alpha_d + k_R e_R) \rangle + \\ &+ c \left( \langle (-k_R e_R - k_\omega e_\omega - [\omega]_\times \tilde{J}\omega - \tilde{J}\alpha_d), e_R \rangle + \langle J e_\omega, E(R, R_d) e_\omega \rangle \right) + \frac{1}{k_J} \text{tr}(\tilde{J}\dot{\tilde{J}}) = \\ &= -k_\omega \|e_\omega\|^2 - ck_R \|e_R\|^2 - ck_\omega \langle e_R, e_\omega \rangle - \langle e_\omega, ([\omega]_\times \tilde{J}\omega + \tilde{J}\alpha_d) \rangle - \langle ce_R, ([\omega]_\times \tilde{J}\omega + \tilde{J}\alpha_d) \rangle + \\ &\quad + \langle J e_\omega, E(R, R_d) e_\omega \rangle + \frac{1}{k_J} \text{tr}(\tilde{J}\dot{\tilde{J}}) \\ \implies \dot{V} &= -k_\omega \|e_\omega\|^2 - ck_R \|e_R\|^2 - ck_\omega \langle e_R, e_\omega \rangle + c \langle J e_\omega, E(R, R_d) e_\omega \rangle + \\ &\quad - \langle (e_\omega + ce_R), ([\omega]_\times \tilde{J}\omega + \tilde{J}\alpha_d) \rangle + \frac{1}{k_J} \text{tr}(\tilde{J}\dot{\tilde{J}}) \end{aligned} \quad (\text{A.21})$$

Por conveniência  $e_A = e_\omega + ce_R$ . Pode-se então reescrever o penúltimo termo da equação acima

$$\langle e_A, ([\omega]_\times \tilde{J}\omega + \tilde{J}\alpha_d) \rangle = \langle e_A, [\omega]_\times \tilde{J}\omega \rangle + \langle e_A, \tilde{J}\alpha_d \rangle$$

Sabendo que  $\langle e_A, [\omega]_\times \tilde{J}\omega \rangle$  é o produto triplo dos vetores  $e_A$ ,  $\omega$  e  $\tilde{J}\omega$ , ou seja

$$\langle e_A, [\omega]_\times \tilde{J}\omega \rangle = e_A \cdot (\omega \times \tilde{J}\omega)$$

pode-se aplicar a propriedade da invariância desse produto triplo sob permutações cíclicas dos vetores:

$$a \cdot (b \times c) = c \cdot (a \times b) = b \cdot (c \times a)$$

para  $a, b$  e  $c \in \mathbb{R}^3$ . Assim

$$\langle e_A, [\omega]_{\times} \tilde{J}\omega \rangle = e_A \cdot (\omega \times \tilde{J}\omega) = \tilde{J}\omega \cdot (e_A \times \omega) = \langle \tilde{J}\omega, [e_A]_{\times} \omega \rangle$$

Para qualquer par de vetores  $a, b \in \mathbb{R}^3$   $\langle a, b \rangle = \text{tr}(ab^T) = \text{tr}(ba^T)$ . Substituindo o resultado acima e trocando o produto interno pelo traço

$$\begin{aligned} \langle e_A, ([\omega]_{\times} \tilde{J}\omega + \tilde{J}\alpha_d) \rangle &= \langle \tilde{J}\omega, [e_A]_{\times} \omega \rangle + \langle e_A, \tilde{J}\alpha_d \rangle = \\ &= \text{tr}[\tilde{J}\omega([e_A]_{\times} \omega)^T] + \text{tr}(\tilde{J}\alpha_d e_A^T) = \text{tr}[\tilde{J}\omega([e_A]_{\times} \omega)^T + \tilde{J}\alpha_d e_A^T] \\ \implies \langle e_A, ([\omega]_{\times} \tilde{J}\omega + \tilde{J}\alpha_d) \rangle &= \text{tr}[\tilde{J}(\omega([e_A]_{\times} \omega)^T + \alpha_d e_A^T)] \end{aligned} \quad (\text{A.22})$$

Como a inércia do quadrotor é constante  $\dot{J} = 0$ , pode-se obter de 3.17  $\dot{J} = -\dot{J}$ . Substituindo essa expressão juntamente com o resultado acima na equação A.21 e valendo-se da lineariedade do operador  $\text{tr}$

$$\begin{aligned} \dot{V} &= -k_{\omega} \|e_{\omega}\|^2 - ck_R \|e_R\|^2 - ck_{\omega} \langle e_R, e_{\omega} \rangle + c \langle J e_{\omega}, E(R, R_d) e_{\omega} \rangle + \\ &\quad - \text{tr}[\tilde{J}(\omega([e_A]_{\times} \omega)^T + \alpha_d e_A^T)] - \frac{1}{k_J} \text{tr}(\tilde{J} \dot{J}) = \\ &= -k_{\omega} \|e_{\omega}\|^2 - ck_R \|e_R\|^2 - ck_{\omega} \langle e_R, e_{\omega} \rangle + c \langle J e_{\omega}, E(R, R_d) e_{\omega} \rangle + \\ &\quad - \text{tr} \left[ \tilde{J} \left( \omega([e_A]_{\times} \omega)^T + \alpha_d e_A^T + \frac{1}{k_J} \dot{J} \right) \right] \end{aligned}$$

Fazendo  $([e_A]_{\times} \omega)^T = -\omega^T [e_A]_{\times}$

$$\begin{aligned} \dot{V} &= -k_{\omega} \|e_{\omega}\|^2 - ck_R \|e_R\|^2 - ck_{\omega} \langle e_R, e_{\omega} \rangle + c \langle J e_{\omega}, E(R, R_d) e_{\omega} \rangle + \\ &\quad - \text{tr} \left[ \tilde{J} \left( -\omega \omega^T [e_A]_{\times} + \alpha_d e_A^T + \frac{1}{k_J} \dot{J} \right) \right] \end{aligned} \quad (\text{A.23})$$

Escolhendo-se a dinâmica para a estimativa  $\bar{J}$

$$\dot{\bar{J}} = \frac{k_J}{2} (-\alpha_d e_A^T - e_A \alpha_d^T + \omega \omega^T [e_A]_{\times} - [e_A]_{\times} \omega \omega^T) \quad (\text{A.24})$$

têm-se

$$\begin{aligned} \dot{V} &= -k_{\omega} \|e_{\omega}\|^2 - ck_R \|e_R\|^2 - ck_{\omega} \langle e_R, e_{\omega} \rangle + c \langle J e_{\omega}, E(R, R_d) e_{\omega} \rangle + \\ &\quad - \text{tr} \left[ \tilde{J} \left( -\omega \omega^T [e_A]_{\times} + \alpha_d e_A^T + \frac{1}{k_J} \frac{k_J}{2} (-\alpha_d e_A^T - e_A \alpha_d^T + \omega \omega^T [e_A]_{\times} - [e_A]_{\times} \omega \omega^T) \right) \right] = \\ &= -k_{\omega} \|e_{\omega}\|^2 - ck_R \|e_R\|^2 - ck_{\omega} \langle e_R, e_{\omega} \rangle + c \langle J e_{\omega}, E(R, R_d) e_{\omega} \rangle + \\ &\quad - \text{tr} \left[ \tilde{J} \left( -\frac{1}{2} ([e_A]_{\times} \omega \omega^T + \omega \omega^T [e_A]_{\times}) + \frac{1}{2} (\alpha_d e_A^T - e_A \alpha_d^T) \right) \right] \end{aligned} \quad (\text{A.25})$$

A expressão  $[e_A]_\times \omega \omega^T + \omega \omega^T [e_A]_\times$ , como mencionado anteriormente neste capítulo, está no formato  $[a]_\times A + A^T [a]_\times = [(\text{tr}(A)\mathbf{I}_3 - A)a]_\times$ . Já o termo  $\frac{1}{2}(\alpha_d e_A^T - e_A \alpha_d^T) = P_a(\alpha_d e_A^T)$ . Assim

$$\begin{aligned} \dot{V} = & -k_\omega \|e_\omega\|^2 - ck_R \|e_R\|^2 - ck_\omega \langle e_R, e_\omega \rangle + c \langle J e_\omega, E(R, R_d) e_\omega \rangle + \\ & -\text{tr} \left[ \tilde{J} \left( -\frac{1}{2} [(\text{tr}(\omega \omega^T)\mathbf{I}_3 - \omega \omega^T) e_A]_\times + P_a(\alpha_d e_A^T) \right) \right] \end{aligned} \quad (\text{A.26})$$

Uma vez que  $a + b \in \mathfrak{so}(3)$ ,  $\forall a, b \in \mathfrak{so}(3)$  e que  $\exists A \in \mathbb{R}^3 \mid P_a(A) = \varphi, \forall \varphi \in \mathfrak{so}(3)$

$$\frac{1}{2}(A - A^T) = -\frac{1}{2} [(\text{tr}(\omega \omega^T)\mathbf{I}_3 - \omega \omega^T) e_A]_\times + P_a(\alpha_d e_A^T)$$

Pode-se reescrever o último termo de  $\dot{V}$  como

$$\begin{aligned} -\text{tr} \left[ \tilde{J} \left( -\frac{1}{2} [(\text{tr}(\omega \omega^T)\mathbf{I}_3 - \omega \omega^T) e_A]_\times + P_a(\alpha_d e_A^T) \right) \right] &= -\text{tr} \left[ \tilde{J} \frac{1}{2} (A - A^T) \right] = \\ &= -\frac{1}{2} (\text{tr}(\tilde{J}A) - \text{tr}(\tilde{J}A^T)) \end{aligned} \quad (\text{A.27})$$

Sabendo que  $\text{tr}(B) = \text{tr}(B^T)$  e que  $\text{tr}(BC) = \text{tr}(CB)$ ,  $\forall B, C \in \mathbb{R}^{n \times n}$

$$\text{tr}(\tilde{J}A^T) = \text{tr}[(\tilde{J}A^T)^T] = \text{tr}(A\tilde{J}) = \text{tr}(\tilde{J}A)$$

onde  $\tilde{J} = \tilde{J}^T$ , como dito no início desta seção. Assim

$$\begin{aligned} -\text{tr} \left[ \tilde{J} \left( -\frac{1}{2} [(\text{tr}(\omega \omega^T)\mathbf{I}_3 - \omega \omega^T) e_A]_\times + P_a(\alpha_d e_A^T) \right) \right] &= \\ &= -\frac{1}{2} (\text{tr}(\tilde{J}A) - \text{tr}(\tilde{J}A)) = 0 \end{aligned} \quad (\text{A.28})$$

Por tanto, reduz-se [A.26](#) à

$$\dot{V} = -k_\omega \|e_\omega\|^2 - ck_R \|e_R\|^2 - ck_\omega \langle e_R, e_\omega \rangle + c \langle J e_\omega, E(R, R_d) e_\omega \rangle \quad (\text{A.29})$$

### A.1.2.2 Análise da estabilidade

Até o momento uma função candidata  $V$  foi definida ([A.14](#)) e, assumindo que ela atende aos critérios de ser positiva definida, nula para o ponto de equilíbrio (quando  $\Phi(R, R_d)$ ,  $e_R$  e  $e_\omega$  são nulos) e de derivada temporal  $\dot{V}$  ([A.29](#)) negativa definida; projetou-se a entrada de controle  $M$  ([A.19](#)) e a dinâmica da estimativa da inércia  $\bar{J}$  ([A.24](#)). Faz-se necessário, portanto, provar que a função escolhida cumpre tais critérios quando o sistema opera em malha fechada

$$\begin{cases} M = -k_R e_R - k_\omega e_\omega + [\omega]_\times \bar{J} \omega + \bar{J} \alpha_d \\ \dot{\bar{J}} = \frac{k_I}{2} (-\alpha_d e_A^T - e_A \alpha_d^T + \omega \omega^T [e_A]_\times - [e_A]_\times \omega \omega^T) \end{cases} \quad (\text{A.30})$$

Como já demonstrado

$$\begin{cases} V = \frac{1}{2}\langle e_\omega, Je_\omega \rangle + k_R \Phi(R, R_d) + c\langle Je_\omega, e_R \rangle + \frac{1}{2k_J} \|\tilde{J}\|_F^2 \\ \dot{V} = -k_\omega \|e_\omega\|^2 - ck_R \|e_R\|^2 - ck_\omega \langle e_R, e_\omega \rangle + c\langle Je_\omega, E(R, R_d)e_\omega \rangle \end{cases} \quad (\text{A.31})$$

quando aplicadas as leis de [A.30](#).

No que concerne as propriedades de  $\Phi(R, R_d)$ , as afirmações

1.  $\Phi$  é localmente positiva definida em torno de  $R = R_d$ .
2. Os pontos críticos de  $\Phi$  quando  $e_R = 0$  são  $\{R_d\} \cup \{R_d \exp(\pi[s]_\times)\}$  para  $s \in \{\vec{i}, \vec{j}, \vec{k}\}$

são verdadeiras e as provas podem ser encontradas em ([BULLO; LEWIS, 2005](#)) no capítulo 11, e não farão parte do escopo desta análise. O validação da função candidata consiste portanto em encontrar as fronteiras de  $V$  e  $\dot{V}$ , o que envolve encontrar os limites inferior e superior de  $\Phi$  bem como dos demais termos presentes em [A.31](#). Mais precisamente, dado que [A.31](#) representa a dinâmica de energia do sistema em malha fechada e que  $V$  está no formato

$$V = U(q) + \frac{1}{2}\|v_q\|^2 + \epsilon \langle \dot{U}(q), v_q \rangle$$

a análise dos limites de [A.31](#) e, por consequência, a prova de sua estabilidade seguem da aplicação direta do Teorema 6.45 (([BULLO; LEWIS, 2005](#)), Capítulo 6) onde

$$V(q) \geq b_1 \|\dot{U}(q)\|^2 + \frac{1}{2}\|v_q\|^2 - \epsilon \|\dot{U}(q)\| \|v_q\| \quad (\text{A.32})$$

$$\begin{aligned} \dot{V}(q) &\leq -\|v_q\|^2 - \epsilon \|v_q\|^2 - \epsilon \|\dot{U}(q)\|^2 \\ &\quad + \epsilon \|v_q\| \|\dot{U}(q)\| \end{aligned} \quad (\text{A.33})$$

onde  $q$  representa os estados ( $R$ );  $v_q$  o vetor dos erros no espaço tangente aos estados ( $e_\omega$  e  $Je_\omega$ );  $U$  é uma função potencial dos estados ( $\Phi$ );  $\dot{U}$  é a derivada de  $U$  com relação à  $q$  ( $e_R$ ) e a constante  $\epsilon > 0$  é um peso atribuído a contribuição conjunta de  $\dot{V}$  e  $v_q$ , que no nosso caso é equivalente a  $c$ . Fica evidente a necessidade de determinar o limite superior de  $\Phi$  na forma  $b_1 \|e_R\|^2$ .

Seja  $\tilde{R} = R_d^T R = \exp([x]_\times)$ ,  $\tilde{R} \in SO(3)$ , conhecido como mapa exponencial, relaciona elementos do  $\mathfrak{so}(3)$  à elementos do  $SO(3)$ , onde  $x \in \mathbb{R}^3$  e  $\exp([x]_\times)$  pode ser obtido pela fórmula de Rodrigues

$$\exp([x]_\times) = \begin{cases} \mathbf{I}_3, & x = 0 \\ \mathbf{I}_3 + \frac{\sin \|x\|}{\|x\|} [x]_\times + \frac{1 - \cos \|x\|}{\|x\|^2} [x]_\times^2, & x \neq 0 \end{cases} \quad (\text{A.34})$$

Reescrevendo  $\Phi$  como  $\frac{1}{2}\text{tr}(G(\mathbf{I}_3 - \tilde{R}))$  e considerando que  $x \neq 0$ :

$$\begin{aligned} \Phi &= \frac{1}{2}\text{tr}(G(\mathbf{I}_3 - \exp([x]_{\times}))) = \\ &= \frac{1}{2}\text{tr} \left[ G \left( \mathbf{I}_3 - \mathbf{I}_3 - \frac{\sin \|x\|}{\|x\|} [x]_{\times} - \frac{1 - \cos \|x\|}{\|x\|^2} [x]_{\times}^2 \right) \right] = \\ &= -\frac{1}{2}\text{tr} \left[ G \left( \frac{\sin \|x\|}{\|x\|} [x]_{\times} + \frac{1 - \cos \|x\|}{\|x\|^2} [x]_{\times}^2 \right) \right] = \\ &= -\frac{1}{2} \left[ \frac{\sin \|x\|}{\|x\|} \text{tr}(G[x]_{\times}) + \frac{1 - \cos \|x\|}{\|x\|^2} \text{tr}(G[x]_{\times}^2) \right] \end{aligned}$$

Como  $G$  é diagonal  $G = \text{diag}(g_1, g_2, g_3)$ , a diagonal principal de  $G[x]_{\times}$  será nula. Portanto  $\text{tr}(G[x]_{\times}) = 0$ . Assim

$$\Phi = -\frac{1}{2} \left[ \frac{1 - \cos \|x\|}{\|x\|^2} \text{tr}(G[x]_{\times}^2) \right] \quad (\text{A.35})$$

Como para o traço só é de interesse o conteúdo da diagonal principal da matriz, calcula-se a diagonal de  $[x]_{\times}^2$

$$\begin{aligned} ([x]_{\times}^2)_{11} &= \begin{bmatrix} 0 & -x_3 & x_2 \end{bmatrix} \begin{bmatrix} 0 \\ x_3 \\ -x_2 \end{bmatrix} = -(x_2^2 + x_3^2) \\ ([x]_{\times}^2)_{22} &= \begin{bmatrix} x_3 & 0 & -x_1 \end{bmatrix} \begin{bmatrix} -x_3 \\ 0 \\ x_1 \end{bmatrix} = -(x_1^2 + x_3^2) \\ ([x]_{\times}^2)_{33} &= \begin{bmatrix} -x_2 & x_1 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ -x_1 \\ 0 \end{bmatrix} = -(x_1^2 + x_2^2) \end{aligned}$$

e a diagonal de  $G[x]_{\times}^2$

$$\begin{aligned} (G[x]_{\times}^2)_{11} &= g_1([x]_{\times}^2)_{11} = -g_1(x_2^2 + x_3^2) \\ (G[x]_{\times}^2)_{22} &= g_2([x]_{\times}^2)_{22} = -g_2(x_1^2 + x_3^2) \\ (G[x]_{\times}^2)_{33} &= g_3([x]_{\times}^2)_{33} = -g_3(x_1^2 + x_2^2) \end{aligned}$$

Logo

$$\begin{aligned} \text{tr}(G[x]_{\times}^2) &= -g_1(x_2^2 + x_3^2) - g_2(x_1^2 + x_3^2) - g_3(x_1^2 + x_2^2) = \\ &= -(g_1 + g_2)x_3^2 - (g_2 + g_3)x_1^2 - (g_3 + g_1)x_2^2 \\ &\implies \text{tr}(G[x]_{\times}^2) = - \left( \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)x_k^2 \right) \end{aligned} \quad (\text{A.36})$$

onde  $\mathcal{C} = \{(1, 2, 3), (2, 3, 1), (3, 1, 2)\}$ . Finalmente

$$\begin{aligned}\Phi &= -\frac{1}{2} \left\{ \frac{1 - \cos \|x\|}{\|x\|^2} \left[ - \left( \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2 \right) \right] \right\} \\ \implies \Phi &= \frac{1 - \cos \|x\|}{2\|x\|^2} \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2\end{aligned}\quad (\text{A.37})$$

Busca-se expressar os limites de  $\Phi$  em termos de  $\|e_R\|^2$ . Assim, usando [A.7](#) e substituindo  $R_d^T R$  por  $\tilde{R}$

$$\begin{aligned}\|e_R\|^2 &= \langle e_R, e_R \rangle = -\frac{1}{2} \text{tr}([e_R]_{\times} [e_R]_{\times}) = \frac{1}{2} \text{tr}([e_R]_{\times} [e_R]_{\times}^T) = \\ &= \frac{1}{2} \text{tr} \left[ \frac{1}{4} (G\tilde{R} - \tilde{R}^T G)(\tilde{R}^T G - G\tilde{R}) \right]\end{aligned}\quad (\text{A.38})$$

O cálculo do termo acima pode ser simplificado com aplicação das propriedades do traço, no entanto essa análise se mostrou extensa. Seguindo o mesmo procedimento de ([LEE, 2011](#)), usou-se o *Symbolic Math Toolbox* ([MathWorks, 2021](#)) do MATLAB, onde expressou-se  $\tilde{R}$  como

$$\tilde{R} = \mathbf{I}_3 + a[x]_{\times} + b[x]_{\times}^2$$

declarando primeiro um vetor  $x$  e usando a função skew do *Spatial Math Toolbox* ([CORKE, 2023](#)). Os termos  $a$  e  $b$  foram declarados como variáveis simbólicas e representam

$$a = \frac{\sin \|x\|}{\|x\|} \quad e \quad b = \frac{1 - \cos \|x\|}{\|x\|^2}$$

O script produzido segue abaixo

```
syms x [3 1]
syms a b
syms g [3 1]

X = skew(x);
G = diag(g);
R = eye(3) + a*X + b*X^2;

eR_skew = 1/2*(G*R - transpose(R)*G);
eR_norm = simplify(collect(...
    1/2*trace(eR_skew*transpose(eR_skew)), {'a', 'b'}...
));

coefficients = coeffs(eR_norm, {'a', 'b'});
```

```
a_coeff = coefficients(1)
b_coeff = coefficients(2)
```

A saída desse script resultou em

```
eR_norm =
```

```
((x1^2*(g2 + g3)^2)/4 + (x2^2*(g1 + g3)^2)/4 + (x3^2*(g1 + g2)^2)/4)*a^2 +
((x1^2*x2^2*(g1 - g2)^2)/4 + (x1^2*x3^2*(g1 - g3)^2)/4 +
(x2^2*x3^2*(g2 - g3)^2)/4)*b^2
```

```
a_coeff =
```

```
(x1^2*x2^2*(g1 - g2)^2)/4 + (x1^2*x3^2*(g1 - g3)^2)/4
+ (x2^2*x3^2*(g2 - g3)^2)/4
```

```
b_coeff =
```

```
(x1^2*(g2 + g3)^2)/4 + (x2^2*(g1 + g3)^2)/4
+ (x3^2*(g1 + g2)^2)/4
```

Finalmente

$$\|e_R\|^2 = \frac{\sin^2 \|x\|}{4\|x\|^2} \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)^2 x_k^2 + \frac{(1 - \cos \|x\|)^2}{4\|x\|^4} \sum_{(i,j,k) \in \mathcal{C}} (g_i - g_j)^2 x_i^2 x_j^2 \quad (\text{A.39})$$

onde  $\mathcal{C} = \{(1, 2, 3), (2, 3, 1), (3, 1, 2)\}$ .

Calculando  $\frac{\|e_R\|^2}{\Phi}$

$$\begin{aligned} \frac{\|e_R\|^2}{\Phi} &= \left( \frac{\sin^2 \|x\|}{4\|x\|^2} \right) \left( \frac{2\|x\|^2}{1 - \cos \|x\|} \right) \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)^2 x_k^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2} \right) + \\ &+ \left[ \frac{(1 - \cos \|x\|)^2}{4\|x\|^4} \right] \left( \frac{2\|x\|^2}{1 - \cos \|x\|} \right) \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i - g_j)^2 x_i^2 x_j^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2} \right) \end{aligned} \quad (\text{A.40})$$

Fazendo  $\sin^2 \|x\| = 1 - \cos^2 \|x\| = (1 - \cos \|x\|)(1 + \cos \|x\|)$

$$\begin{aligned} \frac{\|e_R\|^2}{\Phi} &= \left[ \frac{(1 - \cos \|x\|)(1 + \cos \|x\|)}{4\|x\|^2} \right] \left( \frac{2\|x\|^2}{1 - \cos \|x\|} \right) \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)^2 x_k^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2} \right) + \\ &+ \left[ \frac{(1 - \cos \|x\|)^2}{4\|x\|^4} \right] \left( \frac{2\|x\|^2}{1 - \cos \|x\|} \right) \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i - g_j)^2 x_i^2 x_j^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2} \right) \\ &\implies \frac{\|e_R\|^2}{\Phi} = \frac{1 + \cos \|x\|}{2} \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)^2 x_k^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2} \right) + \\ &+ \frac{1 - \cos \|x\|}{2} \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i - g_j)^2 x_i^2 x_j^2 / \|x\|^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2} \right) \end{aligned} \quad (\text{A.41})$$

É possível observar que, para  $x \neq 0$

$$\frac{\|e_R\|^2}{\Phi} \leq \frac{1}{2h_1} (1 - \cos \|x\|) h_2 + \frac{1}{2h_1} (1 + \cos \|x\|) h_3 \leq \frac{h_2 + h_3}{h_1} \quad (\text{A.42})$$

onde

$$\begin{aligned} h_1 &= \min\{g_1 + g_2, g_2 + g_3, g_3 + g_1\} \\ h_2 &= \max\{(g_1 - g_2)^2, (g_2 - g_3)^2, (g_3 - g_1)^2\} \\ h_3 &= \max\{(g_1 + g_2)^2, (g_2 + g_3)^2, (g_3 + g_1)^2\} \end{aligned}$$

resultando no seguinte limite superior de  $\Phi$  em termos de  $e_R$

$$\Phi \geq \left( \frac{h_1}{h_2 + h_3} \right) \|e_R\|^2 = b_1 \|e_R\|^2 \quad (\text{A.43})$$

Agora tomando  $\frac{\Phi}{\|e_R\|^2}$

$$\begin{aligned} \frac{\Phi}{\|e_R\|^2} &= \frac{1 - \cos \|x\|}{2\|x\|^2} \left[ \frac{4\|x\|^4}{\|x\|^2 \sin^2 \|x\| \beta + (1 - \cos \|x\|)^2 \gamma} \right] \alpha = \\ &= \frac{2\alpha(1 - \cos \|x\|)}{\sin^2 \|x\| \beta + (1 - \cos \|x\|)^2 \gamma / \|x\|^2} \end{aligned} \quad (\text{A.44})$$

onde

$$\begin{aligned} \alpha &= \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2 \\ \beta &= \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)^2 x_k^2 \\ \gamma &= \sum_{(i,j,k) \in \mathcal{C}} (g_i - g_j)^2 x_i^2 x_j^2 \end{aligned}$$

É possível observar que

$$\begin{aligned} \frac{\Phi}{\|e_R\|^2} &\leq \left[ \frac{2(1 - \cos \|x\|)}{\sin \|x\|^2} \right] \left( \frac{\alpha}{\beta} \right) \\ \implies \frac{\Phi}{\|e_R\|^2} &\leq \left[ \frac{2(1 - \cos \|x\|)}{\sin \|x\|^2} \right] \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)^2 x_k^2} \right) \end{aligned}$$

Mais uma vez substituindo  $\sin^2 \|x\|$  por  $(1 - \cos \|x\|)(1 + \cos \|x\|)$ :

$$\frac{\Phi}{\|e_R\|^2} \leq \left( \frac{2}{1 + \cos \|x\|} \right) \left( \frac{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2}{\sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j)^2 x_k^2} \right) \leq \left( \frac{2}{1 + \cos \|x\|} \right) \frac{h_4}{h_5} \quad (\text{A.45})$$

onde

$$\begin{aligned} h_4 &= \max\{g_1 + g_2, g_2 + g_3, g_3 + g_1\} \\ h_5 &= \min\{(g_1 + g_2)^2, (g_2 + g_3)^2, (g_3 + g_1)^2\} \end{aligned}$$

Seja  $\varphi$  uma constante estritamente menor que  $h_1$  tal que

$$\Phi < \varphi < h_1 \quad (\text{A.46})$$

Assim, olhando para [A.37](#)

$$\begin{aligned} h_1 - \varphi < h_1 - \Phi &\leq h_1 - \frac{1 - \cos \|x\|}{2} h_1 = \frac{h_1}{2} (1 + \cos \|x\|) \\ \implies \frac{2}{1 + \cos \|x\|} &\leq \frac{h_1}{h_1 - \varphi} \end{aligned}$$

Assim

$$\begin{aligned} \frac{\Phi}{\|e_R\|^2} &\leq \left( \frac{2}{1 + \cos \|x\|} \right) \frac{h_4}{h_5} \leq \frac{h_4 h_1}{h_5 (h_1 - \varphi)} \\ \implies \Phi &\leq \left( \frac{h_4 h_1}{h_5 (h_1 - \varphi)} \right) \|e_R\|^2 = b_2 \|e_R\|^2 \end{aligned} \quad (\text{A.47})$$

Com isso foram definidas as fronteiras de  $\Phi$  em função dos valores de  $G$ , da norma do vetor erro de orientação  $e_R$  e de uma constante  $\varphi$  que atenda ao critério [A.46](#)

$$b_1 \|e_R\|^2 \leq \Phi \leq b_2 \|e_R\|^2 \quad (\text{A.48})$$

onde

$$b_1 = \frac{h_1}{h_2 + h_3} \quad \text{e} \quad b_2 = \frac{h_4 h_1}{h_5 (h_1 - \varphi)} \quad (\text{A.49})$$

$$\begin{aligned} h_1 &= \min\{g_1 + g_2, g_2 + g_3, g_3 + g_1\} \\ h_2 &= \max\{(g_1 - g_2)^2, (g_2 - g_3)^2, (g_3 - g_1)^2\} \\ h_3 &= \max\{(g_1 + g_2)^2, (g_2 + g_3)^2, (g_3 + g_1)^2\} \\ h_4 &= \max\{g_1 + g_2, g_2 + g_3, g_3 + g_1\} \\ h_5 &= \min\{(g_1 + g_2)^2, (g_2 + g_3)^2, (g_3 + g_1)^2\} \end{aligned} \quad (\text{A.50})$$

Faz-se necessário agora avaliar as fronteiras de  $E(R, R_d) = \frac{1}{2}(\text{tr}(R^T R_d G) \mathbf{I}_3 - R^T R_d G)$  para analisar os limites de  $\dot{V}$ . Considerando a norma de Frobenius

$$\begin{aligned} \|E(R, R_d)\|_F^2 &= \text{tr}(E^T E) = \text{tr} \left( \frac{1}{4} \text{tr}(\tilde{R}^T G) \mathbf{I}_3 - \tilde{R}^T G \right) \left( \frac{1}{4} \text{tr}(\tilde{R}^T G) \mathbf{I}_3 - \tilde{R}^T G \right)^T \\ &\implies \|E(R, R_d)\|_F^2 = \frac{1}{4} \text{tr}(\tilde{R}^T G)^2 + \text{tr}(G^2) \end{aligned} \quad (\text{A.51})$$

Usando mais uma vez a fórmula de Rodrigues, expande-se o termo  $\text{tr}(\tilde{R}^T G)$

$$\begin{aligned} \text{tr}(\tilde{R}^T G) &= \text{tr} \left[ \left( \mathbf{I}_3 - \frac{\sin \|x\|}{\|x\|} [x]_{\times} + \frac{1 - \cos \|x\|}{\|x\|^2} [x]_{\times}^2 \right) G \right] = \\ &= \text{tr}(G) - \frac{\sin \|x\|}{\|x\|} \text{tr}(G[x]_{\times}) + \frac{1 - \cos \|x\|}{\|x\|^2} \text{tr}(G[x]_{\times}^2) \end{aligned}$$

Como já mencionado,  $\text{tr}(G[x]_{\times}) = 0$  e

$$\text{tr}(G[x]_{\times}^2) = - \left( \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2 \right)$$

portanto

$$\begin{aligned} \text{tr}(\tilde{R}^T G) &= \sum_{i=1}^3 g_i + \left( \frac{1 - \cos \|x\|}{\|x\|^2} \right) \left( - \sum_{(i,j,k) \in \mathcal{C}} (g_i + g_j) x_k^2 \right) = \\ &= g_1 + g_2 + g_3 - \left( \frac{1 - \cos \|x\|}{\|x\|^2} \right) [(g_1 + g_2) x_3^2 + (g_2 + g_3) x_1^2 + (g_3 + g_1) x_2^2] = \\ &= g_1 + g_2 + g_3 - \left( \frac{1 - \cos \|x\|}{\|x\|^2} \right) [g_1(x_2^2 + x_3^2) + g_2(x_3^2 + x_1^2) + g_3(x_1^2 + x_2^2)] = \\ &= g_1 \left( 1 - \frac{x_2^2 + x_3^2}{\|x\|^2} \right) + g_2 \left( 1 - \frac{x_3^2 + x_1^2}{\|x\|^2} \right) + g_3 \left( 1 - \frac{x_1^2 + x_2^2}{\|x\|^2} \right) + \\ &\quad + \cos \|x\| \left[ g_1 \left( \frac{x_2^2 + x_3^2}{\|x\|^2} \right) + g_2 \left( \frac{x_3^2 + x_1^2}{\|x\|^2} \right) + g_3 \left( \frac{x_1^2 + x_2^2}{\|x\|^2} \right) \right] \end{aligned}$$

$\|x\|^2 = x_1^2 + x_2^2 + x_3^2$ , o que significa que

$$1 - \frac{x_i^2 + x_j^2}{\|x\|^2} = \frac{\|x\|^2 - x_i^2 - x_j^2}{\|x\|^2} = \frac{x_k^2}{\|x\|^2}$$

e

$$\frac{x_i^2 + x_j^2}{\|x\|^2} = \frac{\|x\|^2 - x_k^2}{\|x\|^2} = 1 - \frac{x_k^2}{\|x\|^2}$$

Dessa forma

$$\begin{aligned} \text{tr}(\tilde{R}^T G) &= g_1 \left( \frac{x_1^2}{\|x\|^2} \right) + g_2 \left( \frac{x_2^2}{\|x\|^2} \right) + g_3 \left( \frac{x_3^2}{\|x\|^2} \right) + \\ &+ \cos \|x\| \left[ g_1 \left( 1 - \frac{x_1^2}{\|x\|^2} \right) + g_2 \left( 1 - \frac{x_2^2}{\|x\|^2} \right) + g_3 \left( 1 - \frac{x_3^2}{\|x\|^2} \right) \right] \\ &\implies \text{tr}(\tilde{R}^T G) = \sum_{i=1}^3 g_i \left( \frac{x_i^2}{\|x\|^2} \right) + \cos \|x\| \sum_{i=1}^3 g_i \left( 1 - \frac{x_i^2}{\|x\|^2} \right) \end{aligned} \quad (\text{A.52})$$

Como  $0 \leq x_i^2/\|x\|^2 \leq 1$ ,

$$\text{tr}(\tilde{R}^T G) \leq \sum_{i=1}^3 g_i = \text{tr}(G)$$

de forma que

$$\|E(R, R_d)\|_F^2 \leq \frac{1}{4}(\text{tr}(G^2) + \text{tr}(G)^2) \leq \frac{1}{2}\text{tr}(G)^2$$

logo

$$\|E(R, R_d)\|_F \leq \frac{1}{\sqrt{2}}\text{tr}(G)$$

Como  $\|E(R, R_d)\| \leq \|E(R, R_d)\|_F$  (TAYLOR, 2010)

$$\|E(R, R_d)\| \leq \frac{1}{\sqrt{2}}\text{tr}(G) \quad (\text{A.53})$$

De posse de A.48 e A.53, faz-se a análise de A.31 segundo as fórmulas A.32 e A.33, à começar por  $V$ .

Sejam  $\lambda_m$  e  $\lambda_M$  o menor e o maior autovalor de  $J$ , respectivamente (obtidos de uma estimativa prévia da inércia). Pela desigualdade de Rayleigh

$$\lambda_m \leq \frac{\langle e_\omega, J e_\omega \rangle}{\|e_\omega\|^2} \leq \lambda_M$$

$$\lambda_m \|e_\omega\|^2 \leq \|e_\omega\| \|J e_\omega\| \cos \theta \leq \lambda_M \|e_\omega\|^2$$

Assumindo que a distribuição de massa do drone seja aproximadamente simétrica em torno dos eixos do referencial  $\{B\}$ , cuja a origem está no centro de massa e cujo os planos definidos pelos eixos sempre dividem o drone em partes simétricas e equilibradas, podemos considerar que  $J$  desvia a direção  $e_\omega$  em um ângulo  $\theta \approx 0$ . Essas considerações não estão distantes da realidade, uma vez que é comum da construção dos quadrotores a distribuição simétrica do peso no plano  $xy$  do referencial do corpo, que costumeiramente é centrado na IMU, instalada o mais próximo do centro de massa possível. Esse também é o caso do Parrot Mambo, cuja a matriz de inércia indicada pelo pacote (MathWorks, 2023) é simétrica, mais especificamente diagonal, com valores de inércia principal muito próximos. Portanto, considerando  $\theta \approx 0 \implies \cos \theta \approx 1$

$$\lambda_m \|e_\omega\|^2 \leq \|e_\omega\| \|J e_\omega\| \leq \lambda_M \|e_\omega\|^2$$

$$\lambda_m \|e_\omega\| \leq \|J e_\omega\| \leq \lambda_M \|e_\omega\|$$

Aplicando esses resultados à análise de  $V$

$$V \geq b_1 k_R \|e_R\|^2 + \frac{1}{2} \lambda_m \|e_\omega\|^2 + c \lambda_M \|e_R\| \|e_\omega\| + \frac{1}{2k_J} \|\tilde{J}\|_F$$

$$\implies V \geq z^T W_{11} z \quad (\text{A.54})$$

onde  $z = [||e_R||, ||e_\omega||, ||\tilde{J}||_F]^T$  e

$$W_{11} = \begin{bmatrix} b_1 k_R & \frac{1}{2} c \lambda_M & 0 \\ \frac{1}{2} c \lambda_M & \frac{1}{2} \lambda_m & 0 \\ 0 & 0 & \frac{1}{2k_J} \end{bmatrix} \quad (\text{A.55})$$

Assim,  $V$  é positiva uma vez que os autovalores de  $W_{11}$  também o sejam. Aqui fez-se o uso de (MathWorks, 2021) mais uma vez

```
syms b1 c kJ kR lM lm
W1 = [b1*kR,c*lM/2,0;c*lM/2,lm/2,0;0,0,1/(2*kJ)];

W1eig = simplify(eig(W1));

eq = [W1eig(3) == 0];

C = solve(eq, [c])

Cuja a saída resultou em

C =

(2^(1/2)*(b1*kR*lm)^(1/2))/lM
```

Avaliando as expressões que determinam os autovalores de  $W_{11}$ , chegou-se a

$$c < \sqrt{\frac{2b_1 k_R \lambda_m}{\lambda_M^2}} \quad (\text{A.56})$$

que é o critério para o qual  $z^T W_{11} z$  é positiva e, conseqüentemente,  $V$  também o é. Também é possível estabelecer um limite superior para  $V$ , o que é particularmente útil na sintonia do controlador

$$\begin{aligned} V &\leq b_2 k_R ||e_R||^2 + \frac{1}{2} \lambda_M ||e_\omega||^2 + c \lambda_M ||e_R|| ||e_\omega|| + \frac{1}{2k_J} ||\tilde{J}||_F^2 \\ &\implies V \leq z^T W_{12} z \end{aligned} \quad (\text{A.57})$$

onde

$$W_{12} = \begin{bmatrix} b_2 k_R & \frac{1}{2} c \lambda_M & 0 \\ \frac{1}{2} c \lambda_M & \frac{1}{2} \lambda_M & 0 \\ 0 & 0 & \frac{1}{2k_J} \end{bmatrix} \quad (\text{A.58})$$

Desse segundo resultado é possível estipular uma dinâmica desejada para as normas  $||e_\omega||$ ,  $||e_R||$  e  $||\tilde{J}||$  à partir de  $W_{12}$  e considerá-la como o pior caso dadas as escolhas dos ganhos.

Agora resta analisar  $\dot{V}$ , buscando determinar que valores garantem que ela seja negativa.

$$\begin{aligned}\dot{V} &\leq -k_\omega \|e_\omega\|^2 - ck_R \|e_R\|^2 + c\lambda_M \|E(RR_d)\| \|e_\omega\|^2 + ck_\omega \|e_\omega\| \|e_R\| \\ \implies \dot{V} &\leq -\left(k_\omega - \frac{c}{\sqrt{2}}\lambda_M \text{tr}(G)\right) \|e_\omega\|^2 - ck_\omega \|e_R\|^2 + ck_\omega \|e_\omega\| \|e_R\| \\ &\implies \dot{V} \leq -\chi^T W_2 \chi\end{aligned}\tag{A.59}$$

onde agora  $\chi = [\|e_R\| \|e_\omega\|]^T$  e

$$W_2 = \begin{bmatrix} ck_R & -\frac{ck_\omega}{2} \\ -\frac{ck_\omega}{2} & k_\omega - \frac{c}{\sqrt{2}}\lambda_M \text{tr}(G) \end{bmatrix}\tag{A.60}$$

Como do caso para  $V$ , quando os autovalores de  $W_2$  são positivos,  $-\chi^T W_2 \chi$  é negativa e, por consequência  $\dot{V}$  também o é. Usando (MathWorks, 2021), foi possível chegar na seguinte condição

$$\begin{aligned}c &< \frac{\sqrt{2}k_\omega}{\lambda_M \text{tr}(G)} \\ c &< \frac{4k_R k_\omega}{k_\omega^2 + \frac{1}{\sqrt{2}}k_R \lambda_M \text{tr}(G)}\end{aligned}\tag{A.61}$$

Compondo os resultados A.56 e A.61

$$c < \min \left\{ \sqrt{\frac{2b_1 k_R \lambda_m}{\lambda_M^2}}, \frac{\sqrt{2}k_\omega}{\lambda_M \text{tr}(G)}, \frac{4k_R k_\omega}{k_\omega^2 + \frac{1}{\sqrt{2}}k_R \lambda_M \text{tr}(G)} \right\}\tag{A.62}$$

Até o momento inferiu-se sobre a estabilidade do sistema desconsiderando as singularidades da solução. A prova da estabilidade assintótica global demanda maior rigor e um conhecimento profundo de controle geométrico. Feitas essas considerações, o controle desenvolvido tem estabilidade local garantida para erros angulares suficientemente distantes das singularidades que aparecem em  $\|x\| = k\pi$ . Boas estimativas iniciais de orientação e de inércia favorecem a estabilidade e, como mostrado nas simulações, impossibilitam que a ação de controle per si leve o sistema às singularidades.

# Referências

- BULLO, F.; LEWIS, A. *Geometric Control of Mechanical Systems*. [S.l.]: Springer-Verlag, 2005. Citado 2 vezes nas páginas 62 e 68.
- CORKE, P. *Robotics, Vision, and Control: Fundamental Algorithms in MATLAB and Simulink*. 2. ed. [S.l.]: Springer, 2017. v. 118. (Springer Tracts in Advanced Robotics, v. 118). Citado 3 vezes nas páginas 30, 32 e 43.
- CORKE, P. *Spatial Math Toolbox for MATLAB*. 2023. <<https://petercorke.com/toolboxes/spatial-math-toolbox/>>. Accessed: 2023-05-27. Citado na página 70.
- FERNANDO, T. et al. Robust adaptive geometric tracking controls on  $so(3)$  with an application to the attitude dynamics of a quadrotor uav. In: IEEE. *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. Orlando, FL, USA, 2011. Citado 3 vezes nas páginas 37, 38 e 39.
- FIGUEIRÔA, M.; MOUTINHO, A.; AZINHEIRA, J. R. Attitude estimation in  $so(3)$ : A comparative case study. In: *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. [S.l.: s.n.], 2014. p. 79–84. Citado na página 28.
- HORN, B. K.; SCHUNCK, B. G. Determining optical flow. *Artificial Intelligence*, v. 17, n. 1, p. 185–203, 1981. ISSN 0004-3702. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0004370281900242>>. Citado 2 vezes nas páginas 29 e 33.
- INVERNIZZI, D.; LOVERA, M.; ZACCARIAN, L. Dynamic attitude planning for trajectory tracking in thrust-vectoring uavs. *IEEE Transactions on Automatic Control*, v. 65, n. 1, p. 453–460, 2020. Submitted on 27 Nov 2020. Disponível em: <<https://hal.science/hal-03029040>>. Citado na página 51.
- INVERNIZZI, D.; PANZA, S.; LOVERA, M. Robust tuning of geometric attitude controllers for multirotor unmanned aerial vehicles. *Aerospace Research Central*, v. 43, May 2020. Disponível em: <<https://doi.org/10.2514/1.G004457>>. Citado na página 60.
- LEE, H. et al. Backstepping control on  $se(3)$  of a micro quadrotor for stable trajectory tracking. In: SEOUL NATIONAL UNIVERSITY. *2013 IEEE International Conference on Systems, Man, and Cybernetics*. Seoul, 2013. Citado na página 51.
- LEE, S.; BANG, H. Automatic gain tuning method of a quad-rotor geometric attitude controller using a3c. *Journal of the Korean Society for Aeronautical & Space Sciences*, Nov 2019. Citado na página 60.
- LEE, T. Robust adaptive geometric tracking controls on  $SO(3)$  with an application to the attitude dynamics of a quadrotor UAV. *arXiv*, arXiv:1108.6031v1, Aug 2011. Mechanical and Aerospace Engineering, Florida Institute of Technology, Melbourne, FL 32901. Disponível em: <<https://arxiv.org/pdf/1108.6031>>. Citado 3 vezes nas páginas 37, 39 e 70.
- MAHONY, R.; HAMEL, T.; PFLIMLIN, J.-M. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, IEEE, v. 53, n. 5, p. 1203, 2008. Citado 2 vezes nas páginas 40 e 58.

- MAHONY, R.; KUMAR, V.; CORKE, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, p. 20–32, ago. 2012. Issue of September 2012. Citado 4 vezes nas páginas 24, 25, 26 e 27.
- MARTIN, P.; SARRAS, I. Partial attitude estimation from a single measurement vector. In: IEEE. *2018 IEEE Conference on Control Technology and Applications (CCTA)*. Copenhagen, Denmark, 2018. Accessed: 2024-06-02. Citado na página 58.
- MathWorks. *Symbolic Math Toolbox*. 2021. Accessed: 2023-05-27. Disponível em: <<https://www.mathworks.com/products/symbolic.html>>. Citado 3 vezes nas páginas 70, 76 e 77.
- MathWorks. *Simulink Support Package for Parrot Minidrones*. 2023. <<https://www.mathworks.com/matlabcentral/fileexchange/63318-simulink-support-package-for-parrot-minidrones>>. Accessed: 2024-05-05. Citado 4 vezes nas páginas 17, 19, 39 e 75.
- MURRAY, R. M.; LI, Z.; SASTRY, S. S. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 1994. ISBN 978-0-8493-7981-2. Citado 3 vezes nas páginas 20, 21 e 22.
- OLIVEIRA, D. N. P. de. *Projeto Baseado em Modelo Aplicado no Desenvolvimento do Sistema de Controle de Voo de um Veículo Aéreo Não Tripulado*. Dissertação (Trabalho de Conclusão de Curso de Bacharelado) — Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil, out. 2020. Orientador: Prof. Antonio Marcus Nogueira Lima, Dr. Citado na página 19.
- TAYLOR, M. *Introduction to Lie Groups*. Chapel Hill, NC: University of North Carolina, 2010. Department of Mathematics, University of North Carolina at Chapel Hill. Citado na página 75.
- VANG, B.; TRON, R. Online automatic gain tuning for geometric attitude control. In: *2022 American Control Conference (ACC)*. [S.l.: s.n.], 2022. p. 4074–4081. Citado na página 60.