

Conrado Silva Miranda

# Sintonia Automática dos Parâmetros de um Controlador para um Quadrirrotor de Modelo Desconhecido em Voo Pairado

65/2014

CAMPINAS 2014

# UNIVERSIDADE ESTADUAL DE CAMPINAS FACULDADE DE ENGENHARIA MECÂNICA

Conrado Silva Miranda

# Sintonia Automática dos Parâmetros de um Controlador para um Quadrirrotor de Modelo Desconhecido em Voo Pairado

Dissertação de Mestrado apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecânica dos Sólidos e Projeto Mecânico.

Orientador: Prof. Dr. Janito Vaqueiro Ferreira

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO Conrado Silva Miranda, E ORIENTADO PELO PROF. DR. Janito Vaqueiro Ferreira.

ASSINATURA DO ORIENTADOR

CAMPINAS 2014

#### FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Miranda, Conrado Silva, 1989-

M672s

Sintonia automática dos parâmetros de um controlador para um quadrirrotor de modelo desconhecido em voo pairado / Conrado Silva Miranda. – Campinas, SP: [s.n.], 2014.

Orientador: Janito Vaqueiro Ferreira. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Controle automático. 2. Calibração

3. Estimação de estado. I. Ferreira, Janito Vaqueiro, 1961-.

II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Sintonia automática dos parâmetros de um controlador para um quadrirrotor de modelo desconhecido em voo pairado.

Título em Inglês: Automatic tuning of a controller's parameters

for a quadrotor with unknown model during hover

Palavras-chave em Inglês: Automatic control, Calibration, State estimation

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca Examinadora: Janito Vaqueiro Ferreira [Orientador],

Ely Carneiro de Paiva, Marcelo Becker

Data da defesa: 30-06-2014

Programa de Pós Graduação: Engenharia Mecânica

## UNIVERSIDADE ESTADUAL DE CAMPINAS FACULDADE DE ENGENHARIA MECÂNICA COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA DEPARTAMENTO DE MECÂNICA COMPUTACIONAL

DISSERTAÇÃO DE MESTRADO ACADÊMICO

# Sintonia Automática dos Parâmetros de um Controlador para um Quadrirrotor de Modelo Desconhecido em Voo Pairado

Autor: Conrado Silva Miranda Orientador: Janito Vaqueiro Ferreira

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:

Prof. D. Janito Vaqueiro Ferreira, Presidente

DMC/FEM/UNICAMP

Prof. Dr. Ely Carneiro de Paiva

DSI/FEM/UNICAMP

Prof. Dr. Marcelo Becker
EESCAISP

EESC/USP

Campinas, 30 de Junho de 2014.

# Agradecimentos

Ao meu orientador, Prof. Dr. Janito Vaqueiro Ferreira, pela paciência durante a pesquisa e pela oportunidade de realizar este trabalho.

Aos colegas do LMA (Laboratório de Mobilidade Autônoma) pelas discussões e opiniões que ajudaram no desenvolvimento deste trabalho.

Aos membros das bancas de qualificação e defesa por disponibilizarem seus tempos e conhecimentos para a análise e melhoria deste trabalho.

À minha família e amigos pelo apoio que me permitiu realizar este trabalho e pela companhia nos bons e maus momentos.

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo apoio financeiro à pesquisa.

## Resumo

Este trabalho tem como objetivo desenvolver um controlador para um quadrirrotor cujos parâmetros são desconhecidos. Para o controle deste sistema, normalmente são utilizados controladores avançados que dependem do conhecimento de parâmetros do sistema ou controladores simples que necessitam de ajuste manual dos ganhos. O controlador desenvolvido possui baixa complexidade e é capaz de ajustar seus parâmetros automaticamente para minimizar uma função de custo durante o voo, sem necessitar de conhecimento de nenhum parâmetro do sistema. Entretanto, este controlador necessita dos estados do sistema, que não estão disponíveis diretamente. Portanto, uma análise da aplicação de métodos de filtragem para estimação destes estados é realizada, comparando-se diversos possíveis modelos estocásticos. Contudo, a filtragem necessita de sensores calibrados corretamente, o que levou à criação de novos algoritmos para calibração dos sensores utilizados. Os algoritmos desenvolvidos nestas três áreas representam passos na direção de criar quadrirrotores que podem operar em ambientes diversificados de modo robusto.

Palavras-chave: Controle automático, Calibração, Estimação de estado.

## **Abstract**

This work's objective is to develop a controller for a quadrotor with unknown parameters. For this system's control, usually advanced controllers that require knowledge of the system's parameters or simple controllers that require manual gain tunning are used. The controller developed has low complexity and is able to adjust its parameters automatically to minimize a cost function during flight, without requiring knowledge of any system's parameter. However, this controller requires the system's states, which aren't available directly. Hence an analysis of the use of filtering methods to estimate these states is conducted, comparing many possible stochastic models. Nonetheless, the filtering requires correctly calibrated sensors, which led to the creation of new algorithms for calibrating the sensors used. The algorithms developed in these three areas represent steps in the direction of creating quadrotors that can operate in diverse environments in a robust way.

Keywords: Automatic control, Calibration, State estimation.

# Lista de Figuras

1.1	Bambu-cóptero	2
1.2	Máquina voadora de Leonardo da Vinci	3
1.3	Helicóptero de Bothezat	4
1.4	Multirotor utilizado pela empresa omstudios.	5
1.5	Diagrama de blocos do quadrirrotor e pesquisa desenvolvida	7
2.1	Diagrama de blocos da filtragem descrita em (ELKAIM <i>e outros</i> , 2012)	14
3.1	Sistemas de coordenadas do quadrirrotor	20
4.1	Erros durante a simulação para o controlador estático e com aprendizado	37
5.1	Problema na discretização de sistemas contínuos	41
5.2	Filtragem com EKF usando modelo dinâmico	56
5.3	Filtragem com UKF usando modelo dinâmico	56
5.4	Filtragem com EKF usando modelo cinemático	57
5.5	Filtragem com UKF usando modelo cinemático	57
5.6	Tempo médio para computação de previsão e correção	58
5.7	Visualização do problema ao usar a saída do giroscópio como entrada	59
5.8	Filtragem utilizando diversos ganhos para a covariância de entrada	62
5.9	Máximo, mediana e mínimo do erro variando-se $\alpha$	64
5.10	Filtragem utilizando $\alpha=0,97.$	65
5.11	Tempo médio para computação de previsão e correção	65
5.12	Máximo, mediana e mínimo do erro variando-se $\beta$	68
5.13	Superfície de erro para $\beta=0,98.$	68
5.14	Tempo médio de para computação de previsão e correção	68
5.15	Erro de atitude utilizando o modelo de "malha aberta"	71
5.16	Erro de estimação de atitude para o modelo de "malha fechada"	71

## Lista de Figuras

5.17	Erro de estimação da velocidade angular para o modelo de "malha fechada"	72
5.18	Erro de estimação de atitude utilizando a covariância estática	73
5.19	Erro de estimação de atitude utilizando a covariância adaptativa	73
5.20	Modelo de Markov para estimação conjunta da posição e atitude	75
5.21	Erro médio para $\nu=1000.$	79
5.22	Evolução do erro máximo, mediano e mínimo de atitude para $\nu=1000.$	80
5.23	Evolução do erro máximo, mediano e mínimo de posição para $\nu=1000.$	80
5.24	Histograma do tempo para computação de previsão e correção	81
5.25	Erros para sistema saindo do equilíbrio	83
6.1	Exemplo de amostragem com $N=4$	90
6.2	Erro de reconstrução com dados de treinamento e variando a condição de parada	99
6.3	Erro de reconstrução com dados de teste e variando a condição de parada	100
6.4	Erro de reconstrução com dados de treinamento e variando o número de intervalos.	100
6.5	Erro de reconstrução com dados de teste e variando o número de intervalos	101
6.6	Tempo de calibração variando a condição de parada e quantidade de intervalos	101
6.7	Exemplo de relação entre as rotações e os sistemas de coordenadas	106
6.8	Erro na estimativa de $\mathcal{R}$ para variadas quantidades de amostra e níveis de erro	110

# Lista de Tabelas

4.1	Parâmetros para simulação do controlador	36
5.1	Parâmetros para simulação de atitude	54
5.2	Parâmetros adicionais para simulação da posição	79
6.1	Parâmetros para simulação da calibração do acelerômetro e magnetômetro	98
6.2	Parâmetros para simulação da calibração da rotação.	109

## Lista de Símbolos

#### Sistemas de coordenadas

- B Sistema de coordenadas fixo ao quadrirrotor, ver página 20.
- I Sistema de coordenadas inercial fixo, ver página 24.
- Sistema de coordenadas fixo aos sensores, ver página 94.

#### Espaços

- $\mathbb{C}$  Espaço dos números complexos, ver página 118.
- H Espaço dos quaternions, ver página 118.
- $\mathbb{N}_n$  Espaço dos naturais entre 1 e n, ou seja,  $i \in \mathbb{N}_n \Leftrightarrow i \in \{1, 2, \dots, n\}$ .
- $\mathbb{R}^n$  Espaço dos números reais de dimenção n.

#### Símbolos do filtro

- $\hat{\mathbf{x}}_{i|j}$  Estimativa da variável x no instante i utilizando informações até o instante j, ver página 41.
- $\mathcal{X}_k(i|j)$  k-ésimo estado aumentado com os ruídos no instante i utilizando informações até o instante j, ver Eq. (5.6).
- $\mathcal{Z}_k(i|j)$  Medição prevista utilizando o estado aumentado  $\mathcal{X}_k(i|j)$ , ver Eq. (5.11).
- $\nu_k$  Erro entre a leitura esperada e a real, ver Eq. (5.5).
- $\sigma_k$  Covariância do estado aumentado  $\mathcal{X}(k|k)$ , ver Eq. (5.6).
- $\tilde{\mathbf{F}}_k$  Aproximação de primeira ordem de  $\mathbf{F}_k$ , ver Eq. (5.20).

 $\pi(\mathbf{x}_t; \theta)$ 

 $\tilde{\mathbf{G}}_k$ Aproximação de primeira ordem de  $G_k$ , ver Eq. (5.20).  $f(\cdot)$ Dinâmica de evolução do estado, ver Eq. (5.2).  $h(\cdot)$ Função de medição do estado, ver Eq. (5.2).  $W_{i}$ Peso do *i*-ésimo estado aumentado, ver Eq. (5.8).  $\mathbf{F}_k$ Linearização da dinâmica em relação ao estado, ver Eq. (5.3).  $\mathbf{G}_k$ Linearização da dinâmica em relação ao ruído de processo, ver Eq. (5.3).  $\mathbf{H}_k$ Linearização da medição em relação ao estado, ver Eq. (5.3). Ganho do filtro de Kalman no instante k, ver Eq. (5.5).  $\mathbf{K}_k$  $\mathbf{L}_k$ Linearização da medição em relação ao ruído de medição, ver Eq. (5.3). Covariância da estimativa  $\hat{\mathbf{x}}_{i|j}$ , ver Eq. (5.4).  $\mathbf{P}_{i|j}$  $\mathbf{Q}_k$ Covariância do ruído de processo no instante k, ver Eq. (5.2).  $\mathbf{Q}_v$ Ruído de processo associado à variável v, ver Eq. (5.21).  $\mathbf{R}_k$ Covariância do ruído de medição no instante k, ver Eq. (5.2).  $\mathbf{R}_v$ Ruído de medição associado à variável v, ver Eq. (5.30).  $\mathbf{S}_{i|j}$ Covariância da leitura esperada  $\hat{\mathbf{z}}_{i|j}$ , ver Eq. (5.5). Entrada do sistema em um instante k, ver Eq. (5.2).  $\mathbf{u}_k$ Estado do sistema em um instante k, ver Eq. (5.2).  $\mathbf{X}_k$ Medição do estado em um instante k, ver Eq. (5.2).  $\mathbf{z}_k$ Leitura esperada com base na estimativa  $\hat{\mathbf{x}}_{i|j}$ , ver Eq. (5.5).  $\mathbf{z}_{i|j}$ Símbolos do controle Passo da otimização dos parâmetros, ver Eq. (4.4).  $\alpha$ Derivada da política  $\pi(\mathbf{x}_t; \theta)$  com relação as parâmetros da política  $\theta$ , ver Eq. (4.6).  $\Phi_t$ 

Política de controle, ver Eq. (4.3).

- $\psi$  Heading do quadrirrotor, ver Eq. (4.22).
- $\Psi_t$  Somatório parcial de  $\Phi_t$ , ver Eq. (4.11).
- $\theta$  Parâmetros da política de controle, ver Eq. (4.3).
- $\tilde{\mathbf{v}}$  Erro do vetor  $\mathbf{v}$  em relação à sua referência  $\mathbf{v}_T$ , ver Eq. (4.17).
- $C(\mathbf{x}_t, \mathbf{u}_t)$  Custo de operação do sistema em um instante de tempo, ver Eq. (4.2).
- $f(\mathbf{x}_t, \mathbf{u}_t)$  Dinâmica de evolução do estado, ver Eq. (4.1).
- H Horizonte de acúmulo de custo, ver Eq. (4.2).
- J(...) Custo de operação do sistema em um horizonte H, ver Eq. (4.2).
- $\mathbf{K}_t$  Derivada da política  $\pi(\mathbf{x}_t; \theta)$  com relação ao estado  $\mathbf{x}_t$ , ver Eq. (4.6).
- $K_p$  Ganho do controlador para erros de posição, ver Eq. (4.19).
- $K_v$  Ganho do controlador para erros de velocidade, ver Eq. (4.19).
- $Q_s$  Matriz de custo da variável s, ver Eq. (4.17).
- Q<sub>t</sub> Derivada do custo  $C(\mathbf{x}_t, \mathbf{u}_t)$  com relação ao estado  $\mathbf{x}_t$ , ver Eq. (4.6).
- $\mathbf{r}_t$  Derivada do custo  $C(\mathbf{x}_t, \mathbf{u}_t)$  com relação ao controle  $\mathbf{u}_t$ , ver Eq. (4.6).
- S Matriz com sinal das derivadas, ver Eq. (4.10).
- $S_{\omega}$  Componente da velocidade angular em S, ver Eq. (4.12).
- $S_p$  Componente da posição em S, ver Eq. (4.14).
- $S_q$  Componente da atitude em S, ver Eq. (4.13).
- S<sub>v</sub> Componente da velocidade linear em S, ver página 30.
- $\mathbf{u}_t$  Ação de controle em um instante t, ver Eq. (4.1).
- $\mathbf{x}_t$  Estado do sistema em um instante t, ver Eq. (4.1).

#### Parâmetros da modelagem

r Distância entre o centro de gravidade do quadrirrotor e o acelerômetro, ver Eq. (3.9).

- $\mathcal{R}$  Matriz de rotação que leva um vetor em  $\mathcal{B}$  para  $\mathcal{S}$ , ver página 21.
- $\omega$  Velocidade angular do quadrirrotor representada em  $\mathcal{B}$ , ver Eq. (3.1).
- $\Sigma_s$  Covariância do ruído de leitura do sensor s, ver Eq. (3.4).
- $\tau$  Torque dos motores no quadrirrotor representados em  $\mathcal{B}$ , ver Eq. (3.1).
- $\theta_s$  Parâmetros do sensor s, ver Eq. (3.6).
- d Coeficiente de arrasto das hélices, ver Eq. (3.2).
- $F_i$  Força gerada pelo motor  $i \in \mathbb{N}_4$  do quadrirrotor, ver Eq. (3.2).
- *l* Distância do rotor ao centro de gravidade do quadrirrotor, ver Eq. (3.2).
- m Massa do quadrirrotor, ver Eq. (3.1).
- $\mathbf{b}_s$  Bias do sensor s, ver Eq. (3.4).
- Força dos motores no quadrirrotor representada em  $\mathcal{B}$ , ver Eq. (3.1).
- $\mathbf{F}_z$  Componente em  $\mathbf{z}^{\mathcal{B}}$  da força  $\mathbf{F}$ , ver Eq. (3.2).
- g Gravidade atuante no quadrirrotor representada em  $\mathcal{I}$ , ver Eq. (3.11).
- h Campo magnético, ver Eq. (3.11).
- J Inércia do quadrirrotor representada em  $\mathcal{B}$ , ver Eq. (3.1).
- $K_s$  Matriz de ganhos do sensor s, ver Eq. (3.4).
- p Posição do quadrirrotor representada em  $\mathcal{I}$ , ver Eq. (3.1).
- q Quaternion entre  $\mathcal{B}$  e  $\mathcal{I}$ , de foma que  $\mathbf{q} \equiv \mathbf{q}_{\mathcal{B}}^{\mathcal{I}}$ , ver Eq. (3.1).
- $\mathbf{s}_s^R$  Leitura ideal do sensor s em  $\mathcal{S}$ , ver Eq. (3.3).
- $s_{\omega}$  Leitura do giroscópio, ver Eq. (3.7).
- $s_a$  Leitura do acelerômetro, ver Eq. (3.9).
- $s_m$  Leitura do magnetômetro, ver Eq. (3.8).
- $s_s$  Valor lido pelo sensor s, ver Eq. (3.3).

 $s_{GPS}$  Leitura do GPS, ver Eq. (3.10).

v Velocidade do quadrirrotor representada em  $\mathcal{I}$ , ver Eq. (3.1).

#### **Probabilidade**

- $\delta$  Ruído de medição do sistema.
- $\epsilon$  Ruído de processo associado à dinâmica do sistema.
- $\mathcal{N}(\mu, \Sigma)$  Distribuição normal com média  $\mu$  e covariância  $\Sigma$ .
- $\mathcal{U}([v_1, v_2])$  Distribuição uniforme no intervalo  $[v_1, v_2]$ .
- $\mathcal{U}(C)$  Distribuição uniforme no conjunto  $C = \{v_1, v_2, \dots, v_n\}$ .
- ~ Amostragem de uma distribuição de probabilidade.

#### **Quaternions e operações**

- $[\otimes q]$  Representação matricial do produto de quaternions pela direita, ver Eq. (A.4).
- $[\mathbf{q}\otimes]$  Representação matricial do produto de quaternions pela esquerda, ver Eq. (A.3).
- $\otimes$  Produto entre dois quaternions, ver Eq. (A.2).
- $\vec{q}$  Componente vetorial do quaternion q, ver Eq. (A.1).
- $q_0$  Componente escalar do quaternion q, ver Eq. (A.1).
- q Quaternion, ver Eq. (A.1).
- q\* Quaternion conjugado, ver página 119.
- $\mathbf{q}_A^B$  Quaternion de rotação da base A para a base B, ver página 121.
- $q^{-1}$  Quaternion recíproco, ver página 119.

#### Símbolos da calibração

- $\Delta_s[i]$  Quantidade de amostras de  $\mu_s[i]$ , ver Eq. (6.1).
- $\gamma$  Condição de parada da otimização, ver página 97.
- $\hat{\mu}_s[i]$  Estimativa do valor esperado da leitura do sensor s no intervalo i, ver Eq. (6.2).

#### Lista de Símbolos

 $\hat{\Sigma}_s$  Estimativa da covariância do sensor s, ver Eq. (6.4).

 $\mu_s[i]$  Valor esperado da leitura do sensor s no intervalo i, ver Eq. (6.1).

 $\theta_s$  Parâmetros para calibração do sensor s, ver Eq. (6.6).

 $\mathbf{K}_{s,ij}$  Elemento na posição (i,j) da matriz de ganhos do sensor s, ver Eq. (6.16).

 $\mathbf{s}_s[i,j]$  j-ésima amostra de  $\mu_s[i]$ , ver Eq. (6.1).

#### Matrizes, vetores e operações

 $diag(\cdots)$  Matriz diagonal formada por seus argumentos.

× Produto vetorial entre dois vetores.

 $\vec{v}$  Vetor unitário na direção v.

 $\mathbf{0}_{m \times n}$  Matriz nula no  $\mathbb{R}^{m \times n}$ , send o tamanho omitido quando implítico.

 $I_n$  Matriz identidade no  $\mathbb{R}^{n \times n}$ , send o tamanho omitido quando implítico.

R Matriz de rotação, ver página 120.

 $\mathbf{R}_A^B$  Rotação da base A para a base B, ver Eq. (A.11).

 $\mathbf{v}^A$  Vetor  $\mathbf{v}$  representado na base A.

 $\mathbf{v}_c$  Componente  $c \in \{x, y, z\}$  do vetor  $\mathbf{v}$ .

· Produto interno entre dois vetores.

# Sumário

Lista de Figuras			xiii					
Li	Lista de Tabelas							
Li	sta de	Símbolos	xvii					
1	Intr	odução	1					
	1.1	História do quadrirrotor	2					
	1.2	Motivação da pesquisa	5					
	1.3	Estrutura do trabalho e contribuições	6					
2	Rev	Revisão bibliográfica 9						
	2.1	Controle de quadrirrotores	10					
	2.2	Filtragem de estados	12					
	2.3	Calibração de sensores inerciais	14					
	2.4	Modelagem e projeto de quadrirrotores	16					
3	Mod	Modelagem matemática do quadrirrotor e dos sensores						
	3.1	Modelagem do quadrirrotor	19					
	3.2	Modelagem dos sensores	21					
		3.2.1 Giroscópio	23					
		3.2.2 Magnetômetro	23					
		3.2.3 Acelerômetro	23					
		3.2.4 GPS	24					
	3.3	Escolha do sistema inercial	24					
4	Con	trole por iteração aproximada de política	25					
	4.1	Análise teórica	26					

		4.1.1	Iteração de política	26
		4.1.2	Gradiente da política pelo sinal da derivada	28
		4.1.3	Sinal da derivada para o quadrirrotor	29
		4.1.4	Função de custo	31
		4.1.5	Política para controle do quadrirrotor	32
	4.2	Simula	ıções	34
		4.2.1	Metodologia	35
		4.2.2	Resultados	36
		4.2.3	Análise	37
	4.3	Conclu	ısão	38
5	Filtr	agem d	e estados	39
	5.1	O	e teórica	
		5.1.1	Equacionamento geral dos filtros	40
		5.1.2	Equacionamento dos modelos	45
	5.2	Simula	1¢ões	51
		5.2.1	Comparação entre modelos de atitude	52
		5.2.2	Estudo da ampliação do ruído de processo no modelo cinemático	60
		5.2.3	Adaptação automática do ruído de processo no modelo cinemático	62
		5.2.4	Estimação da velocidade angular	66
		5.2.5	Modelos de independência para estimação da atitude e velocidade angular .	69
		5.2.6	Estimação de atitude utilizando apenas magnetômetro	72
		5.2.7	Estimação da posição	74
		5.2.8	Estimação completa do sistema partindo do equilíbrio	82
	5.3	Conclu	ısão	84
6	Cali	bração (	de sensores inerciais e orientação	87
	6.1	Aceler	ômetro e magnetômetro	88
		6.1.1	Desenvolvimento teórico	89
		6.1.2	Algoritmos de calibração	96
		6.1.3	Metodologia	97
		6.1.4	Simulações	99
		6.1.5	Análise	99
	6.2	Rotaçã	o do sistema de coordenada dos sensores	103
		621	Decenvolvimento teórico	104

		5.2.2 Metodologia	08
		5.2.3 Simulações	09
		5.2.4 Análise	09
	6.3	Conclusão	11
7	Con	usão 1	13
	7.1	Direções futuras para pesquisa	14
A	Qua	ernions 1	17
	A.1	Operações entre quaternions	18
	A.2 Quaternion como uma rotação no $\mathbb{R}^3$		
		A.2.1 Ângulo entre quaternions	20
		A.2.2 Transformação de matriz de rotação para quaternion	20
		A.2.3 Transformação de quaternion para matriz de rotação	20
		A.2.4 Evolução no tempo	21
	A.3 Vantagens do quaternion		
Re	ferên	as Bibliográficas 12	25

# Capítulo 1

# Introdução

UAVs (*Unmanned Aerial Vehicles*) são veículos que surgiram há cerca de um século e tiveram grande desenvolvimento nas últimas décadas com foco em guerras, sendo peças fundamentais nas mesmas (Cook, 2007), e pesquisa. Visando reduzir o tamanho destes veículos, e consequentemente seu custo e consumo de energia, a DARPA (Agência de Projetos de Pesquisa Avançada de Defesa dos Estados Unidos) criou em 2011 uma competição para desenvolver UAVs de pequeno porte<sup>1</sup>, na tentativa de ter resultados promissores como os obtidos nos desafios de carros autônomos (SEETHARAMAN *e outros*, 2006).

O sucesso atual dos UAVs se deve ao desenvolvimento das diversas tecnologias utilizadas, como visão computacional e algoritmos de controle e geração de trajetória, e maturação dos sistemas. Satisfeitos com a qualidade dos veículos, o governo dos Estados Unidos passou a considerar sua utilização em território nacional para monitoramento de ambientes públicos<sup>2</sup>.

Dentre os UAVs, o quadrirrotor é bastante popular tanto pelas suas capacidades, como decolagem e pouso vertical, quanto pela sua simplicidade de construção e manutenção, quando comparado a aviões ou helicópteros. Estas características fizeram com que a comunidade científica e empresarial desenvolvessem interesse no quadrirrotor, gerando inúmeros projetos baseados neste veículo.

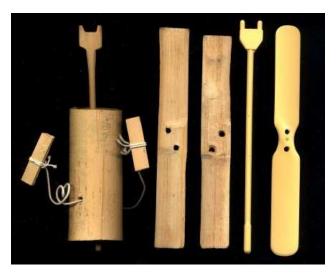
<sup>1</sup>www.uavforge.net

<sup>&</sup>lt;sup>2</sup>www.cdt.org/blogs/harley-geiger/2112drones-are-coming

Este capítulo apresenta uma visão geral da história do quadrirrotor na seção 1.1, seguida na seção 1.2 pela motivação do trabalho aqui desenvolvido. A seção 1.3 finaliza apresentando a estrutura do trabalho e as principais contribuições.

## 1.1 História do quadrirrotor

Em torno de 400 a.C., crianças chinesas brincavam com brinquedos voadores feitos de bambu, que consistiam de uma hélice de madeira presa a um eixo enrolado com cordas que, ao serem puxadas, tanto o eixo quanto a hélice levantavam voo. Atualmente estes brinquedos ainda existem e se chamam *taketombo*, no Japão, ou bambu-cóptero, sendo seu modelo moderno mostrado na figura 1.1. Em 1480 aproximadamente, Leonardo da Vinci projetou uma máquina que, teoricamente, seria capaz de voar na vertical, recebendo o nome de "parafuso aéreo" em virtude de seu desenho, apresentado na figura 1.2. Os modelos de máquinas que voavam na vertical posteriormente desenvolvidos se assemelhavam mais ao bambu-cóptero do que à maquina de da Vinci.



**Figura 1.1:** Bambu-cóptero<sup>3</sup>.

Em 1783, Christian de Launoy criou um modelo com um par de rotores que giravam em direções contrárias e utilizavam penas de peru como hélices. George Cayley se interessava muito pelo brinquedo chinês e desenvolveu vários modelos utilizando penas, assim como Launoy, e elásticos como fonte de potência. Ao fim do século XVIII, ele havia avançado para utilizar metal para as lâminas e molas para potência. Alphonse Pénaud desenvolveu helicópteros de brinquedo com

<sup>&</sup>lt;sup>3</sup>Fonte: en.wikipedia.org/wiki/Bamboo-copter

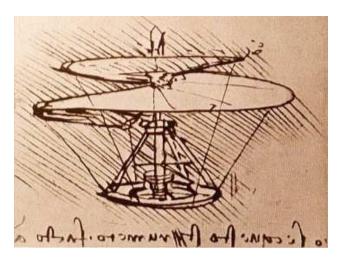


Figura 1.2: Máquina voadora de Leonardo da Vinci<sup>4</sup>.

rotores coaxiais em 1870, sendo um destes modelo dado aos Irmãos Wright por seu pai.

Gustave de Poton d'Amécourt cunhou a palavra "helicóptero" em 1891 ao nomear um pequeno modelo movido a vapor, que dava um uso inovador para o alumínio, mas nunca saiu do chão. Em 1885, Thomas Edison começou a desenvolver experimentos na tentativa de voar, tendo seu helicóptero explodido ao utilizar nitrocelulose como combustível para um motor de combustão interna. Ján Bahýl' adaptou o motor de combustão interna para seu helicóptero, que alcançou 0,5 metros do chão. Em 1905, seu helicóptero alcançou 4 metros de altitude e percorreu mais de 1,5 quilômetros. Edison posteriormente patenteou seu projeto de helicóptero movido a motor à gasolina que alcançou voo.

Em 1906, os irmãos Jacques e Louis Breguet começaram a testar aerofólios em helicópteros, resultando num modelo que levantou 0,6 metros por um minuto, sendo declarado o primeiro voo tripulado de um helicóptero. Como era extremamente instável, não foi considerado o primeiro voo livre de um helicóptero, com tal título alcançado no ano de 1907 por Paul Cornu, que construiu uma helicóptero que elevou 0,3 metros durante 20 segundos.

No início dos anos 1920, Raúl Pateras Pescara demonstrou uma das primeiras aplicações da inclinação cíclica, que permite que as hélices sejam inclinadas independentemente, permitindo que o helicóptero mova para frente e para trás sem uma propulsão extra, além de permitir movimentos laterais. Além disso, Pescara também provou o princípio da autorrotação, que permite que o

<sup>&</sup>lt;sup>4</sup>Fonte: en.wikipedia.org/wiki/Science\_and\_inventions\_of\_Leonardo\_da\_Vinci

helicóptero tenha um pouso menos abrupto no caso de falha do motor principal.

Etienne Oehmichen pesquisou máquinas rotativas nos anos 1920 e, dentre seis projetos que ele testou, um possuía quatro rotores e oito propulsores movidos por um mesmo motor. Cinco dos propulsores eram utilizados para estabilizar o helicóptero lateralmente, um para direcionar e dois para propulsão. Tal modelo apresentava um nível considerável de estabilidade e controlabilidade.

George de Bothezat construiu o quadrirrotor em 1922, chamado "helicóptero de Bothezat", para o exército dos Estados Unidos num programa que foi cancelado em 1924. Este quadrirrotor possuía quatro rotores com seis lâminas nas hélices e uma estrutura em formato de X. Dois pequenos propulsores eram utilizados para propulsão e controle de direção, sendo a inclinação controlada coletivamente. Uma foto deste primeiro quadrirrotor é mostrada na figura 1.3. Ele era mecanicamente complexo, tinha problemas de confiabilidade, baixa potência e falta de resposta.



Figura 1.3: Helicóptero de Bothezat<sup>5</sup>.

Em 1956, um quadrirrotor denominado *Convertawings Model "A" Quadrotor* foi criado como um protótipo de uma linha de quadrirrotores comerciais e civis. Possuía dois motores para os quatro rotores e asas para sustentação adicional em voo, não necessitando de motor na cauda e obtendo controle variando o empuxo entre rotores. Este protótipo provou o conceito de quadrirrotor e foi o primeiro a demonstrar voo frontal com sucesso.

Posteriormente, há um quadrirrotor desenvolvido pelo exército americano, denominado Curtiss-Wright VZ-7, com o objetivo de decolagem e pouso vertical, podendo variar o empuxo em cada um dos propulsores separadamente. Como não foi capaz de alcançar os padrões do exército, o seu desenvolvimento foi cancelado.

Atualmente, com a miniaturização e motores elétricos de alta performance, multirrotores possuem uma grande gama de aplicações, tanto por pesquisadores quanto por profissionais da in-

<sup>&</sup>lt;sup>5</sup>Fonte: en.wikipedia.org/wiki/De\_Bothezat\_helicopter

dústria. Como exemplo industrial tem-se o omcopter, mostrado na figura 1.4, que possui 8 motores para ter grande estabilidade de filmagem com sua câmera, sendo capaz de produzir imagens com altíssima qualidade para a indústria de entretenimento. Já como exemplo de pesquisa, temos a *Flying Machine Arena* (LUPASHIN *e outros*, 2011), que desenvolve aplicações como utilizar quadrirrotores para balancear um pêndulo invertido (HEHN E D'ANDREA, 2011), executar *multi-flips* durante voo (LUPASHIN *e outros*, 2010) e interagir com bolinhas (MULLER *e outros*, 2011).



**Figura 1.4:** Multirotor utilizado pela empresa omstudios<sup>6</sup>.

## 1.2 Motivação da pesquisa

O número de quadrirrotores em operação e experimentos utilizando-os cresce a cada dia. Isto se deve tanto à facilidade de construção e uso (DEVAUD *e outros*, 2012) quanto a sua diversidade de uso, como plataforma educacional (DODDS, 2011) por exemplo. Há ainda a grande capacidade de miniaturização em relação a outros veículos aéreos (BOUABDALLAH *e outros*, 2004), o que facilita seu uso em inúmeras aplicações.

Os quadrirrotores mais utilizados atualmente, como o AR.Drone da Parrot<sup>7</sup> e a linha da Ascending Technologies<sup>8</sup>, foram desenvolvidos com usos pre-determinados em mente, podendo não ser capazes de suportar modificações para as finalidades do usuário ou, quando suportam, podem ter sua estabilidade prejudicada. Em contrapartida, projetos nos quais o usuário define todo o sistema, como no Paparazzi (BRISSET *e outros*, 2006), podem requerer conhecimento prévio de como ajustar dos parâmetros do sistema, o que limita os possíveis usuários do sistema.

Para quebrar esta barreira, este trabalho propõe um conjunto de algoritmos com foco em

 $<sup>^6\</sup>mathrm{Fonte}$ : www.omstudios.de

<sup>&</sup>lt;sup>7</sup>ardrone2.parrot.com

<sup>8</sup>www.asctec.de/uav-applications/research/products/

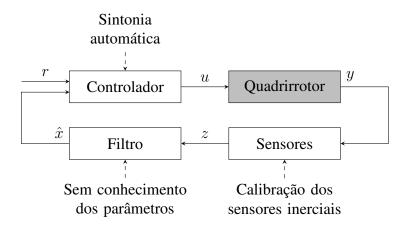
facilidade de uso sem perda de performance em três áreas do laço de controle do sistema, como mostrado na figura 1.5:

- 1. Controle do sistema: os algoritmos normalmente utilizados são extremamente simples e funcionais, como o PID. A literatura científica fornece diversas leis de controle com prova de estabilidade do sistema em malha fechada e de grande complexidade. Em ambos os casos, a determinação de parâmetros adequados para o controle, seja parâmetros de desempenho ou do sistema, é um problema. Para reduzir a distância entre estes dois lados e facilitar seu uso, deve-se buscar a utilização de técnicas com base teórica que sejam capazes de prover alto desempenho sem sobrecarregar o usuário.
- 2. Filtragem do estado: existem diversos dispositivos no mercado que realizam a estimação do estado, mas nem sempre são conhecidos seus métodos de funcionamento e se são a melhor opção para o problema a ser atacado. Portanto, um estudo comparativo é necessário e os algoritmos de estimação devem ser avaliados em situações extremas, visando reduzir a chance de falha inesperada.
- 3. Calibração de sensores inerciais: apesar destes sensores normalmente saírem de suas fábricas calibrados, o transporte e condições ambientais diferentes podem alterar o comportamento de sua leitura, fazendo com que novos algoritmos que sejam capazes de calibrar rapidamente e de maneira fácil os sensores sejam necessários.

## 1.3 Estrutura do trabalho e contribuições

Este trabalho está dividido da seguinte forma: este capítulo situa o leitor quanto ao porquê da realização deste trabalho, enquanto o capítulo 2 apresenta uma revisão bibliográfica de trabalhos relevantes encontrados na literatura. O capítulo 3 apresenta a modelagem do sistema dinâmico e sensores utilizados e os capítulos 4, 5 e 6 tratam dos problema de controle, filtragem e calibração, descritos na seção anterior. Por fim, o capítulo 7 sumariza os resultados e conclusões obtidos, sugerindo trabalhos futuros a serem realizados.

As principais contribuições deste trabalho, apresentadas na figura 1.5 junto com seus blocos afetados, podem ser divididas de acordo com os capítulos da seguinte forma:



**Figura 1.5:** Diagrama de blocos do quadrirrotor e pesquisa desenvolvida.

- 1. O uso do algoritmo de aprendizado de parâmetros do controlador no capítulo 4 é, ao conhecimento do autor, o primeiro que mostra que a adaptação faz com que quadrirrotores das mais variadas inércias passem a ter comportamentos semelhantes entre si, além de mesclar dois controladores apresentados na literatura.
- 2. A comparação entre modelo cinemático e dinâmico e entre filtros de Kalman realizada no capítulo 5 permite analisar do ponto de vista prático os efeitos do uso tanto do modelo cinemático ou dinâmico quanto de filtros baseados em teoria mais robusta.
- 3. Os algoritmos adaptativos para estimação da atitude e velocidade angular descritos no capítulo 5 fazem fortes hipóteses de independência, mas são capazes de estimar estes estados melhor do que outros filtros devido à capacidade de alterar a covariância do ruído do sistema.
- 4. A calibração do acelerômetro e magnetômetro apresentada no capítulo 6 impõe poucas restrições ao usuário se comparada com outros algoritmos do estilo, apesar de utilizar um modelo completo dos sensores. A inovação está no uso da maximização de verossimilhança para fazer uma calibração estatisticamente correta e no desenvolvimento de um algoritmo guloso e eficiente para calibração. Os resultados resultaram numa publicação do COBEM 2013 (MIRANDA E FERREIRA, 2013b).
- 5. A calibração da rotação entre o sistema de coordenadas do sensor e um sistema de coordenadas conhecido, descrita no capítulo 6, não requer uso de sensores externos para fornecer valores de referência, como é feito atualmente na literatura. Esta calibração é feita apenas com os próprios sensores que serão utilizados durante o voo. Os resultados apresentados nesta dissertação foram publicados no COBEM 2013 (MIRANDA E FERREIRA, 2013a).

Todos os algoritmos desenvolvidos para calibração e filtragem foram desenvolvidos na linguagem Python, enquanto o controlador e o simulador do modelo dinâmico utilizado na simulção foram desenvolvidos em C++.

# Capítulo 2

# Revisão bibliográfica

O controle do quadrirrotor pode ser feito através de diversas técnicas, como mostrado na seção 2.1. Estas técnicas podem ser baseadas em modelos aproximados ou testes empíricos, como o caso do PID, ou podem ter desenvolvimentos teóricos com prova de estabilidade do controlador. Em ambos os casos, os métodos apresentados podem contribuir para o entendimento de como o controle é realizado neste veículo.

Algumas variedades de algoritmos de filtragem são descritos na seção 2.2, mostrando a grande diversidade de possibilidades existentes. Quase todos os algoritmos apresentados estão preocupados com a estimação da atitude apenas, por se tratar do estado mais complexo. Com uma boa estimativa da atitude, outras técnicas podem ser usadas para posição, como visão computacional.

A seção 2.3 enumera os principais algoritmos de calibração de sensores inerciais encontrados na literatura. Não são apresentados algoritmos clássicos, como utilizar uma mesa giratória com velocidade conhecida ou posicionar o acelerômetro em rotações perfeitamente alinhadas com os eixos, para mostrar que o problema é mais difícil do que normalmente tratado.

Além de revisar as técnicas de controle, filtragem e calibração, que são o foco deste trabalho, é importante analisar também os modelos de quadrirrotor existentes na literatura. Como o modelo do quadrirrotor já foi demasiadamente abordado na literatura, a seção 2.4 faz uma coleta das melhores descrições destes modelos com seus variados níveis de complexidade. Além disso, a seção

aproveita para elucidar características construtivas de quadrirrotores, sendo de fundamental importância para usuários interessados em construir seus próprios sistemas para atingir seus objetivos.

### 2.1 Controle de quadrirrotores

Quando a literatura é analisada em busca de controles para quadrirrotores, há em geral duas possibilidades: o uso de controles extremamente simples, como PID (POUNDS E MAHONY, 2006) e LQR (HOFFMANN *e outros*, 2004) com base em modelos linearizados, ou controladores não lineares complexos, como o controle por modos deslizantes guiado por observador de modos deslizantes (BESNARD; BESNARD *e outros*, 2007; 2012). O uso de controladores lineares pode ser mais robusto, devido à menor sensitividade aos erros de estimação de estados ou parâmetros do sistema, mas possuem desempenho inferiores aos controladores não lineares, estabelecendo um compromisso de projeto. Entretanto, em ambos os casos os controladores usualmente apresentam uma grande quantidade de parâmetros (ELKAIM *e outros*, 2012), cujos valores os autores não fornecem ou propõem um método para descobrir, ficando a cargo do usuário fazer o ajuste por tentativa e erro.

Em oposição ao modelo extremamente complexo desenvolvido por Pounds *e outros* (2004), descrito posteriormente, e da análise de estabilidade e performance variando-se a altura dos rotores em (POUNDS E MAHONY, 2006), o controle é extremamente simples, utilizando apenas o modelo linearizado e um PID. Apesar da modelagem não linear completa permitir a elaboração de um modelo linear mais próximo da realidade, isto pode representar um esforço desnecessário na maioria dos casos, desmotivando pesquisadores a fazer análises tão detalhadas dos sistemas construídos.

O STARMAC (HOFFMANN *e outros*, 2004) utiliza LQR para controle da atitude, com base em seu modelo linear do sistema, e um controle por modos deslizantes simples para a altitude, considerando a gravidade como ruído. Posteriormente, um modelo mais complexo foi desenvolvido, mas o controle se tornou mais simples, com uso de PIDs em todos os estados (HOFFMANN *e outros*, 2007).

Park *e outros* (2005) utiliza duas malhas de controle com base em PIDs. O PID interno visa aproximar o comportamento da planta do comportamento desejado dado por uma planta de referência, fazendo papel de um controle adaptativo primitivo. Com base numa planta linear de referência, projeta um controlador PID externo. Apesar de ser extremamente simples, apresenta

boa qualidade de controle.

Em contrapartida com os controladores simples descritos anteriormente, Bouabdallah e Siegwart (2007) desenvolve um controle por *backstepping* com termo integral dos estados, cuja prova de estabilidade é determinada durante a construção do controlador. Além disso, o autor ainda apresenta todos os parâmetros do controlador utilizados em simulação e experimento, sendo uma referência como modelo de trabalho publicado. Entretanto, o seu controle de posição ainda utiliza aproximação de ângulos pequenos.

Castillo *e outros* (2004) utiliza um modelo linearizado para controle de posição, mas um controlador não linear baseado em saturação. Este controlador visa aproximar o ângulo de rotação do sistema de zero, levando a posição do sistema à origem em seguida. Ao comparar com o LQR, mostra que o controlador desenvolvido é mais robusto a ruídos e condições iniciais distantes da origem. Uma mistura de linearização por realimentação com controle por saturação é desenvolvido por Salazar-Cruz (SALAZAR-CRUZ *e outros*, 2005), onde o controlador é obtido de forma semelhante ao *backstepping*, ou seja, determinado construtivamente para provar a estabilidade do sistema.

Voos (2009) realiza controle do sistema utilizando linearização por realimentação, gerando portanto um controle extremamente simples mas que depende dos parâmetros do sistema. Tayebi e McGilvray (2004) prova que o controle de atitude utilizando linearização por realimentação é exponencialmente estável enquanto um controle PD é apenas assintoticamente estável, além de fornecer uma lei de controle para o comportamento dinâmico dos motores, apesar de necessitar de diversos parâmetros neste caso.

Guenard *e outros* (2005) propõe uma lei de controle visando fazer com que o erro de velocidade linear seja nulo através do uso de uma função de saturação, evitando comportamentos agressivos na posição. Esta lei garante a estabilidade exponencial da atitude e convergência assintótica da velocidade, sendo semelhante à lei proposta por Tayebi e McGilvray (2004).

Devido à presença do LQR e métodos de ajustar seus custos presentes na literatura, este trabalho descreve um controlador que visa otimizar uma função de custo quadrática durante voo. Como um compromisso entre controladores complexos, como *backstepping* e modos deslizantes, que dependem do conhecimento dos parâmetros do sistema, e controladores mais simples, que são ajustados manualmente, o algoritmo desenvolvido se baseia, de forma semelhante a controlado-

res adaptativos, no ajuste automático dos parâmetros de um controlador simples, como os usados normalmente na prática. Isto permite adaptação para diferentes sistemas sem necessidade de controladores complexos.

### 2.2 Filtragem de estados

Filtragem, no contexto de estimação do estado em sistemas autônomos, pode tratar do uso de vários sensores. Apesar da grande quantidade de pesquisas utilizando câmeras e sensores laser para localização, como apresentado por Thrun *e outros* (2005) em seu livro e diversos outros trabalhos, e.g. (GRZONKA *e outros*, 2009), usualmente unidos ao problema de localização nas técnicas denominadas *SLAM*, são escassos os avanços no uso de sistemas inerciais, que são o foco deste trabalho. Também existe a possibilidade de usar múltiplos sistemas de GPS para determinar a atitude (CRASSIDIS *e outros*, 1999) ou realizar fusão de múltiplos sensores (HOL, 2011), mas estas abordagens não serão utilizadas.

Sabatini (2006) apresenta um EKF baseado num modelo cinemático para detecção de movimento do corpo humano, sendo utilizado como base para criação dos filtros cinemáticos do capítulo 5. Para tal, ele utiliza a leitura do giroscópio como entrada do sistema e a rotação é dada por um quaternion. Além disto, monitora a leitura do acelerômetro e magnetômetro para avaliar se diferem significativamente do valor nominal, ignorando-as caso isto ocorra. Entretanto, não apresenta nenhuma sugestão ou método de escolha dos limiares de validação das leituras. Inclui ainda um estimador de bias do acelerômetro e magnetômetro no estado a ser estimado, prática comum em sistemas que devem funcionar durante muito tempo.

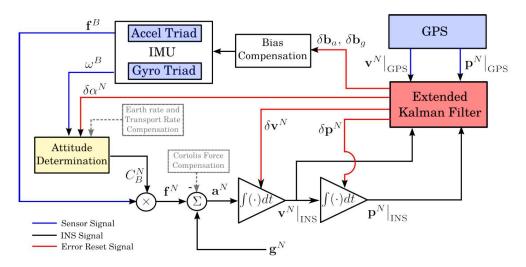
Ainda com foco na área médica, há o filtro desenvolvido por Madgwick *e outros* (2011), que se baseia num processo de otimização eficiente. Ele também utiliza a leitura do giroscópio como entrada do modelo cinemático, mas não utiliza derivados do filtro de Kalman para corrigir a estimativa. Realizando um passo na direção que minimiza a diferença entre a estimativa do campo e a leitura do sensor, o filtro constrói uma outra estimativa do quaternion correto que, ao fazer uma combinação convexa com a previsão do modelo cinemático, provê a estimativa final. Fazendo uso de algumas simplificações, o algoritmo passa a ter apenas um parâmetro, cujo valor pode ser selecionado por um método apresentado no trabalho. Num relatório técnico (MADGWICK, 2010), o desenvolvimento do filtro e os passos para determinação deste parâmetro são detalhados, além da introdução de um outro componente capaz de compensar o *bias* e *drift* do giroscópio.

Como a construção de uma estimativa da rotação do sistema através do acelerômetro e magnetômetro permitiu o desenvolvimento de um algoritmo computacionalmente eficiente no trabalho anterior, existem propostas semelhantes utilizando o EKF. Marins *e outros* (2001) determina por um processo de otimização o quaternion que se ajusta às leituras do acelerômetro e magnetômetro, usando-o juntamente com a leitura do giroscópio para determinar as estimativas do estado. Entretanto, este trabalho deixa em aberto um ponto importante: ele não descreve como a covariância do quaternion determinado pela otimização é gerada, sendo esta informação essencial para o funcionamento correto do filtro.

Gebre-Egziabher *e outros* (2000) utiliza um EKF para fazer a estimativa do quaternion de erro entre a rotação real e a estimada. Ao fazer isto, assume que o erro é pequeno e trabalha com apenas o termo vetorial do quaternion, o qual é submetido a um filtro passa-baixa guiado por um ruído. Esta formulação requer que sejam escolhidas a constante do filtro e a covariância do ruído, o que o trabalho não discute. Ao utilizar esta formulação, o filtro dispensa o uso de um giroscópio na etapa de previsão, uma vez que, segundo os autores, é um sensor custoso. No entanto, esta afirmação não é mais verdade atualmente, devido à evolução dos sensores embarcados na última década, fazendo com que o filtro seja recomendado apenas em casos particulares onde não há giroscópios ou eles não fornecem leituras corretas. Posteriormente, os autores utilizam um filtro complementar para mesclar a estimação independente de giroscópios com o uso da integração direta da leitura do giroscópio (GEBRE-EGZIABHER E ELKAIM, 2008), novamente não discutindo a escolha dos parâmetros.

Elkaim *e outros* (2012) apresentam uma proposta de filtragem utilizando EKF mais focada para veículos aéreos, representada na figura 2.1. O estado a ser estimado conta com a posição, velocidade linear, atitude e bias do acelerômetro e giroscópio, não utilizando magnetômetro. Apesar de seus experimentos apresentarem bons resultados, a não linearidade da atitude do sistema pode afetar a qualidade dos resultados quando acoplada ao GPS. O GPS possui níveis de erros elevados e frequência de amostragem baixa, quando comparado com os demais sensores, o que pode fazer com que a correção na atitude seja elevada. Além disso, a estimação dos *bias* aumenta consideravelmente a complexidade do estado sem garantia de gerar melhores resultados, contanto que a calibração seja adequada. Nos resultados apresentados, fica claro que, exceto por correções exageradas do filtro que são posteriormente revertidas, os *bias* são praticamente constantes.

Apesar da diversidade de técnicas apresentadas para estimação do sistema, usualmente focadas na estimação da atitude, elas são determinadas de maneira empírica. Tanto o filtro de Kalman



**Figura 2.1:** Diagrama de blocos da filtragem descrita em (ELKAIM *e outros*, 2012).

quanto filtros complementares têm sua base em sistemas lineares e as extensões para sistemas não lineares não possuem as garantias originais. Por serem aproximações, estes filtros devem ser avaliados objetivamente quanto à qualidade de suas estimações. Neste trabalho, serão analisadas extensamente as derivações do filtro de Kalman, devido às suas simplicidades de uso, ajustes de parâmetros e utilização abrangente na literatura.

## 2.3 Calibração de sensores inerciais

A calibração dos sensores pode ser feita de maneira online, durante seu funcionamento, ou offline. Entre os métodos online, existem abordagens (ROY E THRUN; SABATINI, 1999; 2006) que utilizam filtros para estimar os parâmetros relevantes dos sensores ou algoritmos recursivos (CRASSIDIS *e outros*; KAPALDO; PYLVANAINEN, 2005; 2005; 2008). Entretanto, se o intervalo de tempo durante o qual os sensores são utilizados for pequeno, como no caso dos quadrirrotores devido ao tempo de voo que a bateria é capaz de prover, os parâmetros permanecem praticamente constantes, favorecendo o uso de técnicas offline. Portanto, este trabalho tratará apenas de métodos offline de calibração.

Vasconcelos *e outros* (2008) prova que as leituras de um magnetômetro, na ausência de ruído, caracterizam uma elipse, o que permite sua calibração através da determinação de um *bias* e uma matriz de ganhos, que inclui os erros causados por deformações do campo magnético. Ao assumir que a covariância é conhecida e dada pela identidade multiplicada por uma constante, elabora

um algoritmo de otimização sem restrições para determinar os parâmetros do magnetômetro. No entanto, utiliza um sistema de coordenadas auxiliar arbitrário onde a calibração é feita e necessita de leituras externas, feitas por algum dispositivo capaz de provê-las nas coordenadas desejadas, para determinar a calibração no referencial desejado.

Foster e Elkaim (2008) apresenta um algoritmo para calibração do magnetômetro em dois passos: primeiro resolve um problema de mínimos quadrados para determinar variáveis auxiliares e, a partir desta solução e do conhecimento do campo magnético correto, determina os parâmetros da calibração. Entretanto, além do custo de desenvolvimento da solução algébrica dos parâmetros utilizando as variáveis auxiliares, nem sempre o valor do campo magnético ao qual o sistema está submetido é conhecido.

Ojeda *e outros* (2000) mostra que alguns tipos de giroscópios possuem não linearidades significativas. Para modelar estes erros, utiliza uma função cúbica do valor lido e quadrática da temperatura para modelar o erro entre o valor lido e o correto, supondo que o ruído de leitura é pequeno. Utilizando uma base giratória de velocidade constante, ele é capaz de computar os parâmetros para compensar o erro, fazendo com que a diferença pós-calibração seja reduzida significativamente. Em contrapartida, Aslan (ASLAN E SARANLI, 2008) mostra que existem sensores, tanto acelerômetro quanto giroscópio, que independem da temperatura e os giroscópios podem ter sua saída extremamente linear, levantando questionamento da necessidade da calibração proposta por Ojeda em sensores mais novos.

O algoritmo TWOSTEP, criado por Alonso e Shuster (2002b), é capaz de determinar bias de um magnetômetro independentemente da atitude do sistema. Para isto, apresenta uma série de aproximações justificadas para chegar a uma função de custo que possui uma componente linear e outra não linear. A partir da solução da parte linear, ele otimiza a função completa. Este algoritmo foi estendido em (ALONSO E SHUSTER, 2002a) para incluir a matriz simétrica de ganhos do sensor. Como o TWOSTEP depende de os dados estarem pré-coletados, analisando-os em batelada, Crassidis *e outros* (2005) desenvolve um método online baseado na solução da parte linear da função de custo.

Olivares *e outros* (2009) busca um método de calibração do giroscópio que não necessite do conhecimento da velocidade de rotação do sistema, como é feito normalmente através de mesas giratórias controladas. Utilizando a leitura do acelerômetro para estimar esta velocidade, o método não depende de maquinário especial, sendo utilizada como base de rotação uma roda de bicicleta.

Com a base girante inclinada, a gravidade atua ciclicamente num eixo conhecido do acelerômetro, permitindo estimar o período e a velocidade angular atuante, a qual é aproximada por um modelo linear. Sabendo em torno de qual eixo o sistema está girando, é possível utilizar a leitura do giroscópio e a estimação a partir do acelerômetro para gerar o ganho e bias daquele eixo. Apesar de não ser capaz de tratar de desalinhamento entre os eixos, a simplicidade deste método é significativa, fazendo com que seja uma base interessante para desenvolvimento de novos algoritmos de calibração de giroscópios.

Elkaim (2013) apresenta um algoritmo para calibração do alinhamento entre dois sensores, utilizando a leitura deles e o conhecimento do seu vetor unitário na coordenada inercial. Renaudin *e outros* (2010) utiliza múltiplos magnetômetros para gerar um algoritmo mais robusto a perturbações externas, permitindo uma melhor estimação da direção de movimento do sistema.

Dentre os algoritmos descritos para calibração de acelerômetros e magnetômetros, todos apresentam hipóteses fortes com relação à covariância dos ruídos, assumindo valores diagonais ou mesmo iguais para todas as direções, além de necessitarem de conhecimento da direção do campo magnético, o que nem sempre está disponível. Além disso, o sistema de coordenadas que a calibração é realizada pode não ser conhecido, necessitando de aparelhagem previamente calibrada ou restrição da orientação do sistema durante a calibração. Algoritmos para resolver ambos problemas, restrição do modelo e restrição da orientação, são descritos neste trabalho.

## 2.4 Modelagem e projeto de quadrirrotores

Existem diversos níveis de precisão na modelagem de quadrirrotor, que estabelecem um compromisso entre qualidade de predição do comportamento e tempo de computação, quantidade de parâmetros e outros problemas associados a modelos muito refinados. Apesar de não ser o foco deste trabalho, o autor considera também importante um conhecimento sobre como projetar um quadrirrotor mecanicamente, de forma a permitir que ele atinja os objetivos desejados utilizando o mínimo de recurso necessário.

Pounds *e outros* (2004) visa construir um quadrirrotor de utilidade prática, ao contrário dos sistemas utilizados normalmente que praticamente não possuem capacidade de carga. Dentre os problemas presentes, encontra a questão de como gerar empuxo suficiente e gerenciar o comportamento instável. Através de análises de aerodinâmica, projeta uma nova hélice que seja capaz

de gerar o empuxo desejado. O modelo desenvolvido é um dos mais completos encontrados na literatura, levando em conta a deformação da hélice e o comportamento dinâmico do motor. O rotor desenvolvido nesse trabalho é uma referência para pessoas interessadas em construir sistemas com capacidade de carga. Para os interessados, a leitura de (POUNDS E MAHONY; POUNDS *e outros*, 2009; 2010) é indicada, uma vez que detalham os passos do projeto desenvolvido. Em (POUNDS E MAHONY, 2006), Pounds analisa o efeito que a posição dos rotores em relação ao centro de gravidade gera na estabilidade e desempenho do sistema, mostrando o efeito positivo que existe quando o rotor está acima do centro de gravidade.

Uma modelagem mais simplificada, que desconsidera a deformação da hélice, é desenvolvida por Bouabdallah *e outros* (2004), considerando ainda o acoplamento entre a dinâmica do quadrirrotor e a velocidade de rotação dos motores e a dinâmica dos motores. Em contrapartida, o modelo utilizado pelo STARMAC (HOFFMANN *e outros*, 2004) faz aproximações de velocidades e ângulos pequenos, chegando a modelos lineares do quadrirrotor. Posteriormente, o modelo não linear é utilizado, incluindo uma análise detalhada dos efeitos aerodinâmicos na hélice (HOFFMANN *e outros*, 2007).

Neste trabalho, será utilizado um modelo simplificado que não considera modelagem dos motores ou hélices, mas que não se restringe a aproximações lineares da dinâmica do sistema. Este modelo foi escolhido por ser próximo o mais próximo do comportamento do sistema sem requerer levantamento de parâmetros aerodinâmicos.

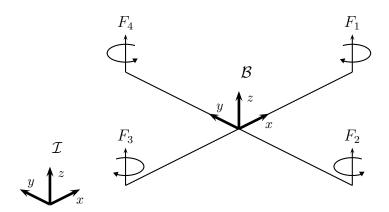
# Capítulo 3

# Modelagem matemática do quadrirrotor e dos sensores

Este capítulo apresentará a modelagem matemática do quadrirrotor na seção 3.1 e dos sensores utilizados na seção 3.2. Estas modelagens serão utilizadas nos capítulos seguintes para elaborar os algoritmos de filtragem, calibração e controle. A seção 3.3, define o sistema inercial fixo na Terra, utilizado como referência para os algoritmos de filtragem e calibração.

### 3.1 Modelagem do quadrirrotor

Quadrirrotores, assim como a maioria dos veículos aéreos, são conhecidos por terem dinâmicas extremamente complicadas. Nesta dinâmica, é possível incluir a deformação das hélices, chamado *blade flapping*, equação dinâmica dos motores, arrasto aerodinâmico do veículo e das hélices e muitos outros fatores (POUNDS *e outros*; HOFFMANN *e outros*; BOUABDALLAH *e outros*; BOUABDALLAH *e outros*; BOUABDALLAH E SIEGWART, 2010; 2007; 2004; 2007). Apesar destes modelos serem extremamente interessantes para projeto e simulação do sistema (POUNDS *e outros*; POUNDS E MAHONY, 2004; 2009), eles são usualmente impraticáveis de serem utilizados no controle e filtragem do sistema, levando ao uso de controladores lineares (POUNDS E MAHONY, 2006) e sistemas inerciais que não usam toda a dinâmica do sistema (POUNDS *e outros*, 2010). Por isso, decidiu-se utilizar neste trabalho uma dinâmica simplificada do veículo.



**Figura 3.1:** Sistemas de coordenadas do quadrirrotor.

Supondo que a resposta dos motores seja instantânea e as inércias dos rotores sejam desprezíveis, pode-se construir um modelo dinâmico composto apenas pela posição p e velocidade v do centro de gravidade do quadrirrotor num sistema de coordenadas inercial  $\mathcal{I}$ , o quaternion unitário q que leva do sistema fixo ao quadrirrotor  $\mathcal{B}$  a  $\mathcal{I}$  e pela velocidade angular  $\omega$  do sistema medida em  $\mathcal{B}$ . Os sistemas de coordenadas são apresentados na figura 3.1. O modelo é dado por:

$$\ddot{\mathbf{p}} = \dot{\mathbf{v}} = \mathbf{R}(\mathbf{q}) \frac{\mathbf{F}}{m} + \mathbf{g} + \frac{\epsilon_f}{m}, \quad \epsilon_f \sim \mathcal{N}(\mathbf{0}, \Sigma_f)$$
(3.1a)

$$\dot{\mathbf{q}} = \frac{1}{2}\omega \otimes \mathbf{q} \tag{3.1b}$$

$$\dot{\omega} = \mathbf{J}^{-1} \left( -\omega \times \mathbf{J}\omega + \tau + \epsilon_{\tau} \right), \quad \epsilon_{\tau} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\tau}), \tag{3.1c}$$

onde m é a massa do sistema,  $\mathbf{J}$  é sua inércia medida em  $\mathcal{B}$ ,  $\mathbf{g}$  é a gravidade atuante no sistema medida em  $\mathcal{I}$ ,  $\mathbf{F}$  e  $\tau$  são a forças e os torques atuantes no sistema  $\mathcal{B}$ ,  $\mathbf{R}(\mathbf{q})$  é a matriz de rotação associada ao quaternion  $\mathbf{q}$ , que pode ser computada como descrito na seção  $\mathbf{A}.2.3$ ,  $\otimes$  o produto entre quaternions, como descrito na seção  $\mathbf{A}.1$  e  $\epsilon_f$  e  $\epsilon_\tau$  são ruídos de processo de força e torque, respectivamente.

Supondo que os rotores estão alinhados com o eixo  $\mathbf{z}^{\mathcal{B}}$  e que não ocorre *blade flapping*, temse que a força  $\mathbf{F}$  possui apenas componente em  $\mathbf{z}^{\mathcal{B}}$ , denominada  $\mathbf{F}_z$ . Sendo  $F_i$  a força desenvolvida por cada um dos rotores, cujas direções de suas rotações são mostradas na figura 3.1, e assumindo que os braços do quadrirrotor possuem o mesmo tamanho e são ortogonais entre si, tem-se que

(BOUABDALLAH *e outros*, 2004)

$$\begin{bmatrix} \mathbf{F}_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -l & 0 & l \\ -l & 0 & l & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}, \tag{3.2}$$

onde l é a distância de cada rotor ao centro de gravidade do quadrirrotor e d é o coeficiente de arrasto que relaciona o empuxo  $F_i$  com o torque gerado em torno de seu eixo de rotação  $\mathbf{z}^{\mathcal{B}}$ .

Apesar deste trabalho tratar como entradas a força  $\mathbf{F}_z$  e os torques  $\tau$ , na realidade elas são utilizadas para gerar as forças  $F_i$  que, então, são utilizadas para computar a velocidade desejada dos rotores. Como este mapeamento é complexo, devido às não linearidades presentes nas hélices, e não é o foco deste trabalho, o trabalho de Pounds e outros (2004) é recomendado para uma visão detalhada do modelo.

Como é possível notar, mesmo com uma série de hipóteses simplificadoras, o modelo ainda possui uma grande quantidade de parâmetros. Estimar cada um dos parâmetros precisamente é extremamente trabalhoso, o que motiva o uso de controladores e filtros que dependam do menor número de parâmetros possível. Em todas as simulações realizadas neste trabalho foi utilizado o modelo dinâmico completo da Eq. (3.1).

## 3.2 Modelagem dos sensores

O modelo dos sensores, de uma maneira geral, é dada por:

$$\mathbf{s}_s = f_s(\mathbf{s}_s^R, \delta_s),\tag{3.3}$$

onde  $\mathbf{s}_s^R$  é o valor correto da leitura medido num referencial  $\mathcal{S}$ ,  $\delta_S$  é uma variável aleatória e  $f_s(\cdot,\cdot)$  é uma função qualquer que faz o mapeamento para o valor lido  $\mathbf{s}_s$ . O referencial  $\mathcal{S}$  deve ser introduzido pois nem sempre ele estará alinhado com o referencial do corpo  $\mathcal{B}$ . No entanto, se os sensores estiverem rigidamente presos ao corpo do quadrirrotor, então a matriz de rotação  $\mathcal{R}$  de  $\mathcal{B}$  para  $\mathcal{S}$  é fixa.

Apesar desta função  $f_s(\cdot,\cdot)$  variar para cada sensor, os sensores tratados neste trabalho são

usualmente aproximados por:

$$\mathbf{s}_s = \mathbf{K}_s \mathbf{s}_s^R + \mathbf{b}_s + \delta_s, \quad \delta_s \sim \mathcal{N}(\mathbf{0}, \Sigma_s)$$
 (3.4)

onde  $\mathbf{K}_s$  é uma matriz de ganhos,  $\mathbf{b}_s$  é o *bias* do sensor e  $\Sigma_s$  é a covariância do ruído  $\delta_s$ . Caso  $\mathbf{K}_s$  e  $\mathbf{b}_s$  sejam conhecidos, é possível obter o valor correto  $\mathbf{s}_s^R$  a partir da leitura  $\mathbf{s}_s$  usando:

$$\mathbf{s}_{s}^{R} = \mathbf{K}_{s}^{-1} \left( \mathbf{s}_{s} - \mathbf{b}_{s} \right) + \delta_{s}', \quad \delta_{s}' \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{s}^{-1} \Sigma_{s} \left( \mathbf{K}_{s}^{-1} \right)^{T})$$
(3.5)

Note que esta formulação assume que os parâmetros do sensor não variam, mas esta hipótese nem sempre é verdadeira. Existem casos onde valor lido  $s_s$  depende não linearmente do valor de referência  $s_s^R$  e da temperatura (HAKYOUNG *e outros*; OJEDA *e outros*, 2001; 2000), mas em outros casos o mesmo tipo de sensor não apresenta tais características (ASLAN E SARANLI, 2008), sendo portanto dependente da tecnologia de fabricação. O parâmetro que mais apresenta variação com o tempo é o *bias*, que pode ser compensado dinamicamente durante a filtragem (MADGWICK *e outros*; SABATINI, 2011; 2006). No entanto, se o tempo de operação do sistema for pequeno, como é o caso comum dos quadrirrotores, este parâmetro não apresenta muita variação e pode ser suposto constante.

Com a suposição de que o ruído do sistema é gaussiano, a probabilidade de uma dada leitura do sensor é dada por:

$$p(\mathbf{s}_s; \theta_s) = (2\pi)^{-\frac{3}{2}} |\Sigma_s|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \left(\mathbf{K}_s \mathbf{s}_s^R + \mathbf{b}_s - \mathbf{s}_s\right)^T \Sigma_s^{-1} \left(\mathbf{K}_s \mathbf{s}_s^R + \mathbf{b}_s - \mathbf{s}_s\right)\right)$$
(3.6a)

$$\theta_s = \left\{ \Sigma_s, \mathbf{K}_s, \mathbf{b}_s, \mathbf{s}_s^R \right\}. \tag{3.6b}$$

Esta probabilidade será importante na hora de determinar uma calibração estatisticamente correta do sistema. Os parâmetros dos sensores utilizados nas simulações foram baseados nos valores encontrados na literatura descrita nas seções 2.2 e 2.3.

#### 3.2.1 Giroscópio

O giroscópio mede diretamente a velocidade  $\omega$  do quadrirrotor, já expressa no sistema  $\mathcal{B}$ , não necessitando portanto de conversão ou ajustes. Assim, seu valor correto de leitura é dado por:

$$\mathbf{s}_{\omega}^{R} = \omega. \tag{3.7}$$

#### 3.2.2 Magnetômetro

O magnetômetro, a princípio, percebe somente o campo terrestre. Os campos perturbantes macios e duros são indistinguíveis de erros de *bias* e ganhos presentes no próprio sensor (VASCONCELOS *e outros*, 2008), fazendo com que eles sejam considerados diretamente na etapa de calibração. Assim, a leitura esperada, medida no referencia  $\mathcal{B}$  vale:

$$\mathbf{s}_m^R = \mathbf{R}(\mathbf{q})^{-1}\mathbf{h},\tag{3.8}$$

onde h é o campo magnético da Terra na posição atual do sistema medido em  $\mathcal{I}$ .

É importante lembrar que, quando usado para aplicações dentro de prédios, magnetômetros podem sofrer fortes perturbações devido às estruturas metálicas.

#### 3.2.3 Acelerômetro

A aceleração total percebida pelo acelerômetro, medida no referencial  $\mathcal{B}$  é dada por:

$$\mathbf{s}_{a}^{R} = \mathbf{R}(\mathbf{q})^{-1}\ddot{\mathbf{p}} + \omega \times (\omega \times \mathbf{r}) + \dot{\omega} \times \mathbf{r}, \tag{3.9}$$

onde r é a distância entre o centro de gravidade do quadrirrotor e o acelerômetro,  $\mathbf{R}(\mathbf{q})^{-1}\ddot{\mathbf{p}}$  é a aceleração do centro de gravidade do quadrirrotor medida em  $\mathcal{B}$  e  $\dot{\omega}$  é definida na Eq. (3.1c). É importante comentar que usualmente os acelerômetros são colocados próximos ao centro de gravidade, de forma a reduzir r a ponto de ser possível desprezá-lo.

#### 3.2.4 **GPS**

O GPS é capaz de medir a posição do sistema no sistema inercial  $\mathcal{I}$ . Além disso, por serem sistemas absolutos, sua matriz de ganho é unitária e apresenta *bias* nulo, exceto em caso de falha do sistema. Portanto, sua leitura  $\mathbf{s}_{GPS}$  é dada por:

$$\mathbf{s}_{GPS} = \mathbf{p} + \delta_{GPS}, \quad \delta_{GPS} \sim \mathcal{N}(\mathbf{0}, \Sigma_{GPS}),$$
 (3.10)

onde  $\delta_{GPS}$  é o ruído do sensor. Ao contrário dos casos anteriores, o ruído do GPS não é gaussiano, pois depende da escolha de satélites utilizados pelo sistema. No entanto, este trabalho aproxima seu ruído por uma gaussiana de covariância elevada, uma vez que os filtros utilizados dependem deste comportamento do ruído.

#### 3.3 Escolha do sistema inercial

Como o sistema inercial  $\mathcal{I}$  é arbitrário, este trabalho utiliza o único sistema de coordenadas onde a gravidade g e o campo magnético h podem ser escritos como: .

$$\mathbf{g} = \begin{bmatrix} 0 & 0 & \mathbf{g}_z \end{bmatrix}^T, \quad \mathbf{g}_z < 0 \tag{3.11a}$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_x & 0 & \mathbf{h}_z \end{bmatrix}^T, \quad \mathbf{h}_x > 0 \tag{3.11b}$$

# Capítulo 4

# Controle por iteração aproximada de política

Este capítulo apresentará o método de controle utilizado neste trabalho e os resultados obtidos. Apesar do controlador utilizado ser simples, as técnicas podem ser estendidas para controles mais complexos, mas como visto na seção 2.1, normalmente controladores mais simples são utilizados na prática. O principal problema no projeto de controladores é encontrar constantes adequadas para estabilizar o sistema e obter a performance desejada (ELKAIM *e outros*, 2012).

O controle aqui utilizado depende do conhecimento básico do funcionamento do sistema, conhecimento este que uma pessoa que o comanda é capaz de ter. A partir disto, o controlador interage com o sistema e ajusta os parâmetros do controlador visando minimizar uma determinada função de custo. O princípio de funcionamento é semelhante ao controle adaptativo, mas apresenta uma estrutura muito mais simples. Entretanto, ele depende do sistema satisfazer certas relações entre os estados e as entradas de controle, sendo estas relações comuns de serem encontradas em sistemas construídos por homens (KOLTER, 2010). O controlador desenvolvido neste capítulo, portanto, difere dos demais por ser capaz de ajustar seus parâmetros online para minimizar a função de custo escolhida pelo usuário sem que o modelo do sistema original seja conhecido.

Este capítulo está dividido da seguinte forma: a seção 4.1 apresenta toda a base teórica por traz do funcionamento do controlador, que é aplicado na seção 4.2 em sistemas simulados com

diversos parâmetros, e a seção 4.3 apresenta um resumo dos resultados obtidos.

#### 4.1 Análise teórica

Considere um sistema cuja dinâmica pode ser descrita por:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \tag{4.1}$$

onde  $f(\cdot,\cdot)$  é uma função desconhecida. Deseja-se determinar as entradas de controle  $u_t$  tais que a função de custo

$$J(\mathbf{x}_0, \mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_H) = \sum_{t=0}^{H} C(\mathbf{x}_t, \mathbf{u}_t)$$
(4.2)

seja minimizada, onde  $C(\cdot, \cdot)$  é determinada pelo usuário para satisfazer seus requisitos e H é o horizonte de otimização, utilizado para capturar a dinâmica do sistema.

A seção 4.1.1 apresenta o método de se determinar os esforços de controle  $\mathbf{u}_t$  de uma maneira geral e a seção 4.1.2 descreve uma aproximação do método, visando simplificá-lo. A seção 4.1.3 descreve como utilizar este método para o quadrirrotor, apresentando modificações necessárias para seu funcionamento. A função de custo a ser otimizada é apresentada na seção 4.1.4 e a política utilizada é descrita na seção 4.1.5.

#### 4.1.1 Iteração de política

Suponha que a lei de controle é dada por:

$$\mathbf{u}_t = \pi(\mathbf{x}_t; \theta), \tag{4.3}$$

onde a função  $\pi(\cdot, \cdot)$  é conhecida, denominada política de controle, e  $\theta$  são parâmetros do controlador, a serem determinados. Dada uma determinada política  $\pi$ , o custo da Eq. (4.2) passa a depender apenas do estado inicial  $\mathbf{x}_0$  e dos parâmetros da política  $\theta$ . Os parâmetros  $\theta$  podem ser ajustados de maneira a reduzir o funcional de custo se forem corrigidos iterativamente utilizando:

$$\theta_{k+1} = \theta_k - \alpha \frac{\partial J(\mathbf{x}_0; \theta)}{\partial \theta},\tag{4.4}$$

onde J é a mesma função de custo definida anteriormente onde as entradas de controle são escolhidas de acordo com a política  $\pi$  e  $\alpha$  é o tamanho do passo na direção do gradiente.

Aplicando-se as regras de derivação, o gradiente pode ser escrito como:

$$\frac{\partial J(\mathbf{x}_0; \theta)}{\partial \theta} = \sum_{t=0}^{H} \frac{\partial C(\mathbf{x}_t, \mathbf{u}_t)}{\partial \theta}$$
(4.5a)

$$= \sum_{t=0}^{H} \left( \frac{\partial C(\mathbf{x}_{t}, \mathbf{u}_{t})}{\partial \mathbf{x}_{t}} \frac{\partial \mathbf{x}_{t}}{\partial \theta} + \frac{\partial C(\mathbf{x}_{t}, \mathbf{u}_{t})}{\partial \mathbf{u}_{t}} \frac{\partial \pi(\mathbf{x}_{t}; \theta)}{\partial \mathbf{x}_{t}} \frac{\partial \mathbf{x}_{t}}{\partial \theta} + \frac{\partial C(\mathbf{x}_{t}, \mathbf{u}_{t})}{\partial \mathbf{u}_{t}} \frac{\partial \pi(\mathbf{x}_{t}; \theta)}{\partial \theta} \right)$$
(4.5b)

$$= \sum_{t=0}^{H} \left( (\mathbf{q}_t + \mathbf{r}_t \mathbf{K}_t) \frac{\partial \mathbf{x}_t}{\partial \theta} + \mathbf{r}_t \Phi_t \right), \tag{4.5c}$$

onde define-se:

$$\mathbf{q}_{t} = \frac{\partial C(\mathbf{x}_{t}, \mathbf{u}_{t})}{\partial \mathbf{x}_{t}}, \ \mathbf{r}_{t} = \frac{\partial C(\mathbf{x}_{t}, \mathbf{u}_{t})}{\partial \mathbf{u}_{t}}, \ \mathbf{K}_{t} = \frac{\partial \pi(\mathbf{x}_{t}; \theta)}{\partial \mathbf{x}_{t}}, \ \Phi_{t} = \frac{\partial \pi(\mathbf{x}_{t}; \theta)}{\partial \theta}.$$
(4.6)

Pela regra da cadeia, tem-se que  $\frac{\partial \mathbf{x}_t}{\partial \theta}$  é dado por:

$$\frac{\partial \mathbf{x}_t}{\partial \theta} = \sum_{t'=0}^{t-1} \frac{\partial \mathbf{x}_t}{\partial \mathbf{u}_{t'}} \frac{\partial \pi(\mathbf{x}_{t'}; \theta)}{\partial \theta},$$
(4.7)

onde, substituindo este termo na Eq. (4.5c), tem-se:

$$\frac{\partial J(\mathbf{x}_0; \theta)}{\partial \theta} = \sum_{t=0}^{H} \left( (\mathbf{q}_t + \mathbf{r}_t \mathbf{K}_t) \left( \sum_{t'=0}^{t-1} \frac{\partial \mathbf{x}_t}{\partial \mathbf{u}_{t'}} \Phi_{t'} \right) + \mathbf{r}_t \Phi_t \right). \tag{4.8}$$

Nesta expressão do gradiente, o único termo relativo à dinâmica do sistema é  $\frac{\partial \mathbf{x}_t}{\partial \mathbf{u}_{t'}}$  que, se fosse conhecido, permitiria otimizar diretamente a função. Entretanto, caso a dinâmica do sistema não seja conhecida, este termo deve ser aproximado de alguma maneira.

#### 4.1.2 Gradiente da política pelo sinal da derivada

Visando aproximar a expressão  $\frac{\partial \mathbf{x}_t}{\partial \mathbf{u}_{t'}}$ , Kolter (2010) desenvolveu um método chamado *signed derivative*. Este método é baseado em duas hipóteses, descritas a seguir.

Se t=t'+1, tem-se que  $\frac{\partial \mathbf{x}_t}{\partial \mathbf{u}_{t'}}$  é dado por:

$$\frac{\partial \mathbf{x}_{t'+1}}{\partial \mathbf{u}_{t'}} = \frac{\partial f(\mathbf{x}_{t'}, \mathbf{u}_{t'})}{\partial \mathbf{u}_{t'}}$$
(4.9)

e corresponde ao termo linearizado da função em relação às entradas de controle. Assim, estes termos representam como entradas de controle passadas afetam estados futuros. A primeira hipótese consiste em assumir que, para horizontes curtos, os termos para  $t>t^\prime+1$  são similares ao termo de um passo.

Para muitos sistemas, cada estado é afetado principalmente por uma entrada de controle, definindo uma ortogonalidade das entradas. Como exemplo, Kolter (2010) cita um carro seguindo uma trajetória: apesar da dinâmica complexa do carro, sabe-se que a posição longitudinal é principalmente afetada pela aceleração, enquanto a posição lateral e o ângulo de direção é afetada pela rotação do volante. Esta simplicidade auxilia o aprendizado humano, uma vez que sistemas mais acoplados demandam mais coordenação nos movimento, e, por isso, vários sistemas construídos por seres humanos possuem tal característica. Assim, a segunda hipótese é a de que o algoritmo sabe para cada estado qual entrada de controle mais o afeta e se este efeito é positivo ou negativo. Esta hipótese é responsável por dar nome à técnica, uma vez que o usuário fornece ao algoritmo o sinal da derivada de um estado para uma entrada de controle.

Sendo S a matriz de sinais das derivadas, possuindo apenas termos 1, 0 e -1, ela é utilizada para substituir  $\frac{\partial \mathbf{x}_t}{\partial \mathbf{u}_{t'}}$  em na Eq. (4.8), chegando a:

$$\frac{\partial J(\mathbf{x}_0; \theta)}{\partial \theta} \approx \sum_{t=0}^{H} \left( (\mathbf{q}_t + \mathbf{r}_t \mathbf{K}_t) \mathbf{S} \Psi_t + \mathbf{r}_t \Phi_t \right), \tag{4.10}$$

onde:

$$\Psi_t = \sum_{t'=0}^{t-1} \Phi_{t'}. \tag{4.11}$$

O algoritmo final consiste portanto em observar o comportamento do sistema por H+1 iterações, computando as Eqs. (4.10) e (4.11), e realizar o ajuste conforme a Eq. (4.4).

#### 4.1.3 Sinal da derivada para o quadrirrotor

Para determinar a matriz S para o quadrirrotor, seu modelo será analisado próximo à referência desejada, pois, apesar de ele não estar sempre nesta condição, isto evita conflitos na construção da matriz. Para compreender porque isto aconteceria, considere o efeito que existe na posição y ao aplicar um torque  $\tau_x$ . Supondo que o sistema possui um ângulo de direção próximo de 0, tem-se que o torque irá afetar y de forma oposta da qual o faria caso a direção fosse próxima de  $180^\circ$ . Portanto, ao considerar que o sistema está próximo de algum ponto "bem comportado", algumas simplificações podem ser feitas. Além disso, isto reduz também o acoplamento existente entre as velocidades e atitudes. Nota-se que esta aproximação não é restritiva uma vez que a referência pode ser computada por um gerador de trajetória, permitindo variações na referência para atingir o objetivo alcançado.

Assumindo que as velocidades angulares são pequenas, tem-se que elas são afetadas principalmente por seus respectivos torques. É importante ressaltar que isto não é necessariamente verdade, pois uma inércia muito reduzida em x, por exemplo, pode fazer com que um torque grande em y altere significativamente a velocidade em x. Entretanto, esta é uma aproximação razoável e frequentemente utilizada por pilotos. Além disso, um valor positivo dos torques gera valores positivos na velocidade, fazendo com que o sinal da derivada seja positivo. Portanto, tem-se que os termos de S respectivos são dados por:

$$\mathbf{S}_{\omega} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

onde as entradas foram ordenadas como  $\mathbf{u} = [F, \tau_x, \tau_y, \tau_z]^T$ , conforme a notação da Eq. (3.1).

Considerando que o quaternion está próximo de sua referência, o quaternion de erro pode ser escrito como  $\mathbf{q}_e = [1, e_x, e_y, e_z]^T$ . Cada termo  $e_i$  é principalmente afetado pela sua velocidade angular, fazendo com que suas componentes da matriz  $\mathbf{S}$  sejam os mesmos. O termo escalar praticamente não é afetado por qualquer uma das entradas, fazendo com que seu termo seja nulo para

todas as entradas. Assim, a esta parte da matriz S é dada por:

$$\mathbf{S}_{\mathbf{q}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.13}$$

A posição z do sistema é principalmente afetada pela força  ${\bf F}$ , fato este que é frequentemente explorado no projeto de controladores. Já para as posições x e y, assume-se um modelo linearizado do sistema. Supondo que o quadrirrotor está pairado, uma rotação positiva em torno do eixo x faz com que o sistema movimente-se na direção -y. Em contrapartida, uma rotação positiva em y faz o sistema se movimentar em x. Considerando que o torque  $\tau_i$  é o responsável por gerar rotações no eixo x, tem-se que a componente de  ${\bf S}$  vale:

$$\mathbf{S}_{\mathbf{p}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.14}$$

Como a velocidade possui o mesmo comportamento descrito para a posição, tem-se que  $\mathbf{S_v} = \mathbf{S_p}.$ 

É importante destacar que existem duas hipóteses com relação aos sinais utilizados:

- 1. Os termos da matriz  $S_p$  só estão corretos quando assume-se que  $F_z$  é positivo. Caso contrário, os termos referentes a x e y deveriam ser invertidos, uma vez que o sistema estaria se movendo na direção reversa. O termo z permanece inalterado, uma vez que ele não depende da direção de movimentação do sistema.
- 2. Assumiu-se, sem perda de generalidade, que o quaternion possui termo escalar positivo, resultando na análise realizada. Devido ao mapeamento duplo dos quaternions, seria possível estar numa situação onde este valor é negativo, o que faria com que  $\mathbf{S_q}$  tivesse que ser invertida.

Como este trabalho supõe que  $\mathbf{F}_z \geq 0$ , considerando a incapacidade do sistema de gerar empuxo negativo<sup>9</sup>, apenas zera-se esta entrada do sistema caso o valor desejado passe a ser negativo.

<sup>&</sup>lt;sup>9</sup>Existem quadrirrotores que podem alterar a inclinação de suas lâminas, funcionando como helicópteros e sendo

A princípio, as derivadas das funções de custo e política deveriam ser alteradas, uma vez que o sistema encontra-se em modo de saturação, mas notou-se nas simulações da seção 4.2 que isto não é necessário, pois o sistema raramente se encontra em tais situações.

Para o problema do mapeamento duplo do quaternion, leva-ase em conta o formato do controlador. Como será discutido na seção 4.1.5, o controlador faz uso do quaternion com termo escalar positivo. Portanto, caso este termo se torne negativo, ele é zerado e a computação das Eqs. (4.10) e (4.11) é reinicializada, evitando que a otimização capture uma dinâmica descontínua.

Por fim, tem-se que a matriz S para o quadrirrotor é:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\mathbf{p}} \\ \mathbf{S}_{\mathbf{v}} \\ \mathbf{S}_{\mathbf{q}} \\ \mathbf{S}_{\omega} \end{bmatrix}$$
(4.15)

onde  $S_p$ ,  $S_q$  e  $S_\omega$  são dadas pelas Eqs. (4.14), (4.13) e (4.12), respectivamente, e  $S_v = S_p$ .

#### 4.1.4 Função de custo

Como é comum de se encontrar na literatura de controle, utiliza-se um custo quadrático no erro do sistema em relação a uma referência. Para a posição, velocidade e velocidade angular, o erro é dado simplesmente pelo valor corrente menos o valor desejado. No entanto, o mesmo não ocorre para o quaternion.

Seja  $\mathbf{q}$  o quaternion corrente do sistema e  $\mathbf{q}_{des}$  o quaternion desejado para o sistema, pode-se definir o erro como:

$$\tilde{\mathbf{q}} = \mathbf{q} \otimes \mathbf{q}_{des}^*. \tag{4.16}$$

Para os sinais de controle, considera-se que a massa do sistema é conhecida, de forma que apenas a força tenha uma referência diferente de zero. Caso a massa não fosse conhecida, o esforço  $\mathbf{F}$  poderia ser separado em  $\mathbf{F}_1$  e  $\mathbf{F}_2$ , de forma que  $\mathbf{F} = \mathbf{F}_1 + \mathbf{F}_2$  e  $\mathbf{F}_2$  possuiria apenas uma constante na componente  $\mathbf{z}^{\mathcal{I}}$ , que não é penalizada. Assim, evita-se que o sistema seja forçado a manter um

capazes de gerar empuxo negativo, mas estes sistemas não serão tratados neste trabalho.

erro vertical devido à restrição do esforço de controle não permitir o ajuste devido à massa.

Uma vez definidos os erros dos estados em relação aos valores desejados e dos esforços de controle em relação ao necessário ao necessário para manter o quadrirrotor em voo, a função de custo é dada simplesmente por:

$$C(\mathbf{x}_t, \mathbf{u}_t) = \tilde{\mathbf{p}}_t^T \mathbf{Q}_{\mathbf{p}} \tilde{\mathbf{p}}_t + \tilde{\mathbf{v}}_t^T \mathbf{Q}_{\mathbf{v}} \tilde{\mathbf{v}}_t + \tilde{\mathbf{q}}_t^T \mathbf{Q}_{\mathbf{q}} \tilde{\mathbf{q}}_t + \tilde{\omega}_t^T \mathbf{Q}_{\omega} \tilde{\omega}_t + \tilde{\mathbf{u}}_t^T \mathbf{Q}_{\mathbf{u}} \tilde{\mathbf{u}}_t$$
(4.17)

Como deseja-se que a resposta de posição do sistema seja mais lenta, evitando oscilações características de sistemas subamortecidos, faz-se  $\mathbf{Q}_v = 10\mathbf{Q}_p$  e  $\mathbf{Q}_p = \mathbf{I}_3$ , onde  $\mathbf{I}_n$  é a matriz identidade de tamanho n. Pelo mesmo motivo, define-se  $\mathbf{Q}_\omega = 10\mathbf{I}_3$ . Já para o quaternion, novamente a escolha é mais delicada. Com base na aproximação de pequenos ângulos e o controlador utilizado, que será descrito na seção posterior, não faz sentido aplicar um custo à componente escalar do quaternion, uma vez que a informação está codificada nas demais componentes. Como o valor que as componentes vetoriais alcançarão são usualmente menores do que as componentes de posição, deve-se ter um custo maior do que  $\mathbf{Q}_p$ . Assim,  $\mathbf{Q}_q$  é dado por:

$$\mathbf{Q_q} = \begin{bmatrix} 0 & 0 \\ 0 & 10\mathbf{I}_3 \end{bmatrix} \tag{4.18}$$

Outro fator que auxilia a evitar oscilações é o uso de um custo para os esforços de controle, fazendo com que o sistema os aplique de maneira mais sábia. No entanto, um custo muito alto pode fazer com que a resposta do sistema seja demasiadamente lenta, levando-o à instabilidade. Assim, escolheu-se  $\mathbf{Q_u} = 10^{-2}\mathbf{I_4}$ .

Note que estes valores são arbitrários e podem ser ajustados conforme o sistema opera caso seja necessário. Os valores descritos aqui foram ajustados para reduzir o erro médio da simulação do sistema sem aprendizado, descrita na seção 4.2.

## 4.1.5 Política para controle do quadrirrotor

A política utilizada pode ser separada na política de controle vertical e de controle da rotação. Para o controle vertical, utiliza-se o método descrito em (MELLINGER E KUMAR, 2011),

enquanto o controle da rotação é dado pela lei de controle mais simples descrita em (TAYEBI E MCGILVRAY, 2004). A complexidade da lei é restrita devido à quantidade de parâmetros a serem aprendidos pelo controlador. Além disso, o uso de termos integrativos nos controladores auxiliam em manter a estabilidade e permitem que o sistema funcione fora da condição ideal. No entanto, o uso destes termos em controladores preditivos como este não é factível, devido à complexidade do termo já simplificado pelo método aqui descrito. Portanto, é comum utilizar termos constantes no controlador ao invés de integrativos, uma vez que os valores das constantes podem ser alterados durante o passo da otimização realizada.

Sejam  $\tilde{\mathbf{p}}$  e  $\tilde{\mathbf{v}}$  os erros de posição e velocidade e  $\ddot{\mathbf{p}}_T$  a aceleração da trajetória de referência naquele instante, tem-se que a força total desejada no quadrirrotor é dada por:

$$\mathbf{F}_{des} = -\mathbf{K}_{\mathbf{p}}\tilde{\mathbf{p}} - \mathbf{K}_{\mathbf{v}}\tilde{\mathbf{v}} + m\ddot{\mathbf{p}}_T + \overline{\mathbf{F}},\tag{4.19}$$

onde  $\mathbf{K_p}$  e  $\mathbf{K_v}$  são matrizes positivas definidas, m é a massa do sistema e  $\overline{\mathbf{F}}$  é um termo constante. Assume-se que  $\|\mathbf{F}_{des}\| \neq 0$ .

Como não se pode aplicar uma força em qualquer direção no quadrirrotor, componente z da força  ${\bf F}$  aplicada é dada por:

$$\mathbf{F}_z = \mathbf{F}_{des} \cdot \mathbf{z}^{\mathcal{B}},\tag{4.20}$$

onde · representa o produto interno.

Para corrigir a direção  $\mathbf{z}^{\mathcal{B}}$  do sistema, deve-se computar a rotação desejada. O valor desejado deste vetor é dado pela norma da força desejada, ou seja,

$$z_{des}^{\mathcal{B}} = \frac{\mathbf{F}_{des}}{\|\mathbf{F}_{des}\|} \tag{4.21}$$

Conforme descrito em (MELLINGER E KUMAR, 2011), para definir as demais componentes desejadas define-se um vetor auxiliar  $\mathbf{x}_{des}^{C}$  com a direção desejada do sistema:

$$\mathbf{x}_{des}^{C} = \left[\cos \psi_T, \sin \psi_T, 0\right]^T, \tag{4.22}$$

onde  $\psi_T$  é a direção desejada da frente do sistema, considerada  $\psi_T=0$  neste trabalho.

Com este vetor auxiliar, as demais coordenadas são definidas da seguinte maneira:

$$\mathbf{y}_{des}^{\mathcal{B}} = \frac{\mathbf{z}_{des}^{\mathcal{B}} \times \mathbf{x}_{des}^{C}}{\|\mathbf{z}_{des}^{\mathcal{B}} \times \mathbf{x}_{des}^{C}\|}$$
(4.23a)

$$\mathbf{x}_{des}^{\mathcal{B}} = \mathbf{y}_{des}^{\mathcal{B}} \times \mathbf{z}_{des}^{\mathcal{B}}.\tag{4.23b}$$

A partir das coordenadas desejadas, pode-se obter o quaternion desejado  $\mathbf{q}_{des}$  utilizando-se o método descrito na seção A.2.2.

Definindo o quaternion de erro conforme a Eq. (4.16), pode-se utilizar o controlador PD descrito em (TAYEBI E MCGILVRAY, 2004), dado pela seguinte equação:

$$\tau = -\beta \vec{\tilde{\mathbf{q}}} - \mathbf{K}_{\omega} \tilde{\omega} + \overline{\tau} \tag{4.24}$$

onde  $\beta$  é um escalar positivo,  $\vec{\mathbf{q}}$  é a componente vetorial do quaternion,  $\mathbf{K}_{\omega}$  é uma matriz positiva definida e  $\overline{\tau}$  é um termo constante.

Portanto, os parâmetros  $\theta$  da política de controle descrita pelas Eqs. (4.19) a (4.24) são dados por:

$$\theta = \{ \mathbf{K_p}, \mathbf{K_v}, \overline{\mathbf{F}}, \beta, \mathbf{K_\omega}, \overline{\tau} \}. \tag{4.25}$$

Como existe um acoplamento entre a política de controle de força e a política de controle de torque através das Eqs. (4.20) e (4.21), a derivada da política com relação aos estados pode ficar complexa de ser calculada e pode não gerar tantos benefícios. Ao analisar a definição e uso de  $z_{des}^{\mathcal{B}}$ , nota-se que sua contribuição no cálculo da derivada dos torques seria demasiadamente complexa, sendo simplificada como uma constante fixa a cada instante de tempo para uso do algoritmo. No entanto, o termo  $z^{\mathcal{B}}$  possui uma derivada extremamente simples com relação ao quaternion, de forma que esta relação é mantida durante a computação da Jacobiana.

### 4.2 Simulações

Nesta seção, o controlador descrito anteriormente será utilizado para controlar o quadrirrotor na presença de ruídos. A comparação de seu desempenho com outros controladores encontrados na literatura pode ser difícil, uma vez que os controladores podem ter sido ajustados com objetivos

ou funções de custos diferentes. Portanto, utilizar-se-á como base para comparação um controlador sem aprendizado submetido às mesmas condições. Os parâmetros deste controlador foram ajustados manualmente de forma a gerar um bom desempenho para diversas combinações de inércia do quadrirrotor, visando uma competição justa entre o controlador estático e o adaptativo.

#### 4.2.1 Metodologia

A referência para o controlador foi que o quadrirrotor deveria permanecer parado e sem rotação no eixo z a partir desta mesma condição, ou seja,  $\mathbf{p}_T = \mathbf{0}$  e  $\psi_T = 0$ . O tempo total de simulação foi de 1 minuto, com frequência de amostragem de 100Hz. Os parâmetros do controlador utilizados foram H = 4 e  $\alpha = 10^{-4}$ , onde o valor de  $\alpha$  deve ser pequeno para evitar variações grandes nos parâmetros devido apenas aos ruídos. Uma alternativa seria utilizar valores maiores para  $\alpha$  e fazer uma média de uma sequência de gradientes. No entanto, o sistema permaneceria utilizando a mesma política durante muito tempo, o que motivou manter o controlador online.

O quadrirrotor utilizado possui massa m=1kg e inércia em z de  $J_z=1$ kg.m². Estes valores foram empregados pela possibilidade de escalar os parâmetros na mesma proporção. Outro fator foi a ausência de dados reais de quadrirrotores. É realizada uma simulação para cada par de inércias  $J_x, J_y \in \{0,1;0,2;\ldots;1,0\}, J_x \geq J_y$  para testar o comportamento do sistema de uma maneira independente do veículo ao qual foi aplicado.

Os ruídos de torque e força foram criados com a mesma escala, variando apenas a unidade, de forma que apenas a força será descrita, onde cada componente é gerada a partir de uma combinação de ruído de onda quadrada e gaussiano. A intensidade da componente quadrada é escolhida pelo gerador  $\mathcal{U}([-5,5])$  e possui duração escolhida aleatoriamente pelo gerador  $\mathcal{U}(\{10,11,\ldots,100\})$ , fazendo com que a intensidade permaneça no máximo 1 segundo. Terminado este período, novas intensidades e durações são amostradas até que a simulação termine. Já a componente normal é gerada a partir de um ruído gaussiano de média zero e covariância 10, ou seja, a partir do gerador  $\mathcal{N}(0,10)$ . Entretanto, foi realizada uma alteração nestes valores: como foi suposto que o ruído de forças se devem a efeitos aerodinâmicos e é sabido que estes efeitos afetam menos as direções horizontais do quadrirrotor, aplica-se apenas 10% do ruído nestas direções.

Os parâmetros iniciais do controlador, também utilizados no caso sem aprendizado durante

toda a simulação, são dados por:

$$\mathbf{K}_{\mathbf{p}} = -5\mathbf{I}_{3}$$

$$\mathbf{K}_{\mathbf{v}} = -10\mathbf{I}_{3}$$

$$\overline{\mathbf{F}} = [0, 0, 9, 81]^{T}$$

$$\beta = -100$$

$$\mathbf{K}_{\omega} = -10\mathbf{I}_{3}$$

$$\overline{\tau} = [0, 0, 0]^{T}.$$

$$(4.26)$$

A tabela 4.1 apresenta um resumo dos parâmetros utilizados para gerar os resultados.

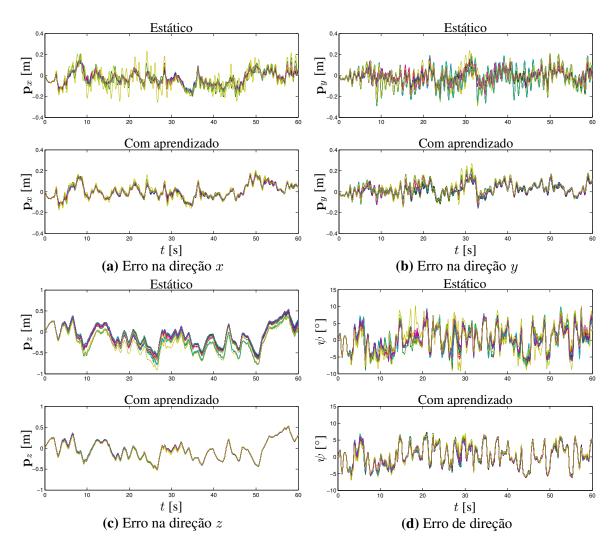
Parâmetro	Descrição	Valor
T	Tempo total de cada simulação	60s
$f_s$	Frequência dos passos da simulação	100 <b>Hz</b>
m	Massa do sistema	1kg
$J_z$	Inércia em $z$	$1 \mathrm{km.m}^2$
$J_x, J_y$	Inércias em $x$ e $y$	$J_x, J_y \in \{0, 1; 0, 2; \dots; 1, 0\}, J_x \ge J_y$
	Intensidade da ruído quadrado	$\mathcal{U}([-5,5])$
	Duração do ruído quadrado	$\mathcal{U}(\{10,11,\ldots,100\})$
	Intensidade do ruído normal	$\mathcal{N}(0,10)$
	Fator do ruído de forças em $x$ e $y$	0,1
H	Tamanho da janela do controlador	4
$\alpha$	Tamanho dos passos da otimização	$10^{-4}$
	Valor inicial dos parâmetros de controle	Ver Eq. (4.26)

**Tabela 4.1:** Parâmetros para simulação do controlador.

Para definir o erro de direcionamento do quadrirrotor, o quaternion é decomposto numa rotação em torno do eixo z e uma rotação no plano xy. Dado um quaternion  $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$ , a rotação não normalizada em torno do eixo z é facilmente calculada zerando as componentes  $q_1$  e  $q_2$  e renormalizando.

#### 4.2.2 Resultados

A figura 4.1 apresenta os resultados da simulação, onde todas as variações de inércia foram sobrepostas. As posições são dadas em metros e o direcionamento é dado em graus.



**Figura 4.1:** Erros durante a simulação para o controlador estático e com aprendizado. Cada curva corresponde a uma combinação de inércias.

#### 4.2.3 Análise

Apesar do controlador com aprendizado ter se comportado de maneira próxima ao estático em alguns casos, uma vez que os parâmetros inicias foram ajustados manualmente para terem bom desempenho, o aprendizado claramente reduziu a intensidade e oscilação dos erros.

Como o comportamento de  $\mathbf{p}_x$  e  $\mathbf{p}_y$  dependem muito da inércia do sistema, parâmetros estes que são variados na simulação, era de se esperar que apresentassem as melhores reduções de erros, mas tanto  $\mathbf{p}_z$  quanto  $\psi$  apresentaram melhoras significativas de desempenho. Em particular, a posição  $\mathbf{p}_z$  com o controlador estático apresenta comportamento próximo em todos os casos, mas

durante alguns intervalos há um *offset* entre conjuntos de inércias diferentes, o que não ocorre no caso com aprendizado.

Um resultado inesperado foi que o aprendizado fez com que o comportamento de todos as variações de inércias ficassem mais próximos entre si, apesar disto não estar especificado diretamente no controlador. É possível que este comportamento represente um ponto de baixo custo da função utilizada, fazendo com que o controlador ajuste os parâmetros para que todos os sistemas tenham comportamentos próximos a este ótimo. Esta capacidade de ajuste do controlador talvez possa ser explorada de maneira semelhante a controladores adaptativos numa malha interna, ajustando o comportamento da planta para ser controlada por um controlador externo.

#### 4.3 Conclusão

Este capítulo utilizou uma metodologia geral para aprendizado de parâmetros de controladores em sistemas dinâmicos e testou seu desempenho para controle de um quadrirrotor sujeito a ruídos. Apesar de ser capaz de controlar adequadamente o sistema, foi percebida uma certa dificuldade no ajuste da função de custo do controlador.

Com os parâmetros iniciais utilizados, frequentemente o veículo divergia antes de chegar a uma boa região dos parâmetros, levantando questionamento de como isto ocorre e como evitar tais situações. Como esta função de custo pode ser alterada durante o funcionamento do sistema, é desejável permitir que o usuário modifique o custo para satisfazer melhor seus requisitos de desempenho mas sem arriscar a estabilidade do veículo. Além disso, deve-se descobrir combinações adequadas de parâmetros para uso do controlador junto de um filtro de estados. Ao acoplar estes dois blocos, tem-se que o aprendizado não é feito com base nos estados reais, o que usualmente leva ao uso de passos de aprendizado menores. Entretanto, isto reduz a capacidade do controlador de responder rapidamente a alterações do comportamento dinâmico ao qual está habituado. Devido à importância do uso conjunto destes blocos, uma investigação detalhada dos efeitos gerados pelo uso de filtros é necessária, de forma a alcançar um melhor compromisso entre robustez a ruído e desempenho.

# Capítulo 5

# Filtragem de estados

Este capítulo apresentará as técnicas de filtragem utilizadas neste trabalho e seus resultados. Os filtros aqui utilizados são baseados no filtro de Kalman (KALMAN, 1960), uma vez que os ruídos dos sensores são principalmente gaussianos. Para tratar de outros tipos de ruídos, é possível utilizar filtros  $\mathcal{H}_{\infty}$  (STEWART, 1996) ou filtros de partículas (ARULAMPALAM *e outros*, 2002). Além disso, assumir-se-á que as matrizes de covariância são constantes, pelos motivos discutidos na seção 3.2. Para o caso de covariâncias variantes, é possível usar filtros adaptativos para corrigi-las (HAN *e outros*, 2009). Para mais detalhes das diversas técnicas de filtragem Bayesiana disponíveis, o tutorial desenvolvido por Haug (2005) é recomendado.

O filtro desenvolvido deve ser o mais preciso e eficiente computacionalmente possível. No entanto, estes requisitos são conflitantes uma vez que é possível reduzir o tempo de computação ao assumir independências entre os estados sem que elas existam, o que pode reduzir a qualidade da estimação do filtro, possivelmente chegando a resultados inconsistentes. Por estas razões, os filtros apresentados neste capítulo assumem uma série de independências entre os estados que assume-se não prejudicar significativamente a estimação, mas que aumentam significativamente sua eficiência. Todo desenvolvimento deste capítulo, exceto onde destacado, representa contribuições ao realizar uma análise extensiva de diversos modelos que podem ser utilizados para estimação do estado de um quadrirrotor.

Este capítulo está dividido da seguinte forma: a seção 5.1 apresenta o funcionamento geral

dos filtros utilizados e o equacionamento para o caso particular do quadrirrotor, a seção 5.2 traz os resultados obtidos a partir da aplicação dos filtros e suas modificações no quadrirrotor e a seção 5.3 apresenta um resumo dos resultados obtidos.

#### 5.1 Análise teórica

Esta seção apresenta o equacionamento dos filtros a serem utilizados juntamente com os modelos aos quais são aplicados, onde estes modelos são baseados naqueles desenvolvidos no capítulo 3. A seção 5.1.1 detalha o equacionamento dos filtros EKF (*Extended Kalman Filter*) e UKF (*Unscented Kalman Filter*), ambos alterações do filtro de Kalman original (KALMAN, 1960), enquanto a seção 5.1.2 desenvolve os modelos elaborados anteriormente para serem usados junto aos filtros.

#### **5.1.1** Equacionamento geral dos filtros

O objetivo da filtragem baseada em filtros de Kalman é gerar uma estimativa do estado  $\mathbf{x}_k$  a partir de uma série de medições  $\mathbf{z}_k, \mathbf{z}_{k-1}, \dots$  para o seguinte sistema:

$$\mathbf{x}_{k} = f(\mathbf{x}_{k-1}, k, \mathbf{u}_{k}, \mathbf{u}_{k-1}, \epsilon_{k-1}), \quad \epsilon_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$$

$$\mathbf{z}_{k} = h(\mathbf{x}_{k}, k, \mathbf{u}_{k}, \delta_{k}), \qquad \delta_{k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k}),$$
(5.1)

onde  $\mathbf{u}_i$  são sinais conhecidos aplicados ao sistema e  $f(\cdot)$  e  $h(\cdot)$  são funções arbitrárias do processo e da medição dos estados, respectivamente.

É importante notar que a ambas funções do sistema possuem mais termos  $\mathbf{u}_i$  do que normalmente utilizados na literatura. Isso se deve ao fato de que  $\mathbf{u}$  é um sinal arbitrário, incluindo o sinal de controle. Como o sistema real pode ser contínuo e a computação não é instantânea, a discretização permite que um dado intervalo seja influenciado por mais sinais do que o esperado. Considerando que  $\mathbf{u}$  é apenas sinal de controle, a justificativa pode ser observada na figura 5.1. Caso o tempo total de computação seja muito menor do que o período de amostragem, é possível aproximar utilizando apenas  $u_k$ . No entanto, se o tempo de computação for muito próximo do período de amostragem, então o intervalo de atuação de  $\mathbf{u}_k$  se torna desprezível.

Utilizando a aproximação de que o tempo de computação é próximo do período de amostra-

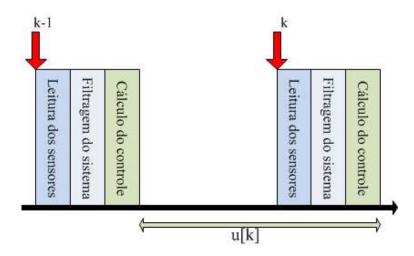


Figura 5.1: Problema na discretização de sistemas contínuos.

gem, tem-se que o sistema é dado por:

$$\mathbf{x}_{k} = f(\mathbf{x}_{k-1}, k, \mathbf{u}_{k-1}, \epsilon_{k-1}), \quad \epsilon_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$$

$$\mathbf{z}_{k} = h(\mathbf{x}_{k}, k, \mathbf{u}_{k}, \delta_{k}), \quad \delta_{k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k}).$$
(5.2)

#### 5.1.1.1 EKF

O EKF (*Extended Kalman Filter*) (MCGEE E SCHMIDT, 1985) foi um filtro desenvolvido para adaptar o filtro de Kalman para sistemas reais. Para isso, ele realiza uma aproximação linear do modelo calculando a Jacobiana em torno da melhor estimativa atual para propagar a covariância e usa o modelo não linear para estimar o próximo estado e as leituras dos sensores.

É importante destacar que, ao contrário do filtro de Kalman, que possui provas de otimalidade, o EKF não possui desempenho garantido, mas apresenta resultados bons caso o sistema seja aproximadamente linear na região de alta probabilidade da covariância.

Seja  $\hat{\mathbf{x}}_{i|j}$  a melhor estimativa de x no tempo i utilizando informações até o tempo j, as Jacobianas necessárias para o filtro são dadas pelas seguintes equações:

$$\mathbf{F}_{k} = \frac{\partial f}{\partial \mathbf{x}_{k}} \Big|_{\mathbf{x}_{k} = \hat{\mathbf{x}}_{k|k}}, \quad \mathbf{G}_{k} = \frac{\partial f}{\partial \epsilon_{k}} \Big|_{\mathbf{x}_{k} = \hat{\mathbf{x}}_{k|k}}, \quad \mathbf{H}_{k} = \frac{\partial h}{\partial \mathbf{x}_{k}} \Big|_{\mathbf{x}_{k} = \hat{\mathbf{x}}_{k|k-1}}, \quad \mathbf{L}_{k} = \frac{\partial h}{\partial \delta_{k}} \Big|_{\mathbf{x}_{k} = \hat{\mathbf{x}}_{k|k-1}}.$$
(5.3)

Para calcular a aproximação do estado futuro  $\hat{\mathbf{x}}_{k|k-1}$  utilizando a estimativa  $\hat{\mathbf{x}}_{k-1|k-1}$ , aplicase um passo semelhante à previsão do filtro de Kalman, com a diferença de que o modelo não linear é utilizado para propagar o modo da distribuição de probabilidade. Assim, sua previsão é dada por:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, k, \mathbf{u}_{k-1}, \mathbf{0})$$
(5.4a)

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T, \tag{5.4b}$$

Uma vez previsto o estado futuro, pode-se calcular a leitura esperada  $\hat{\mathbf{z}}_{k|k-1}$  e compará-la com a leitura real  $\mathbf{z}_k$ . Sua diferença determina a novidade  $\nu_k$  presente na medida, sendo usada para correção da estimativa do estado. O equacionamento para a etapa de correção é:

$$\hat{\mathbf{z}}_{k|k-1} = h(\hat{\mathbf{x}}_{k|k-1}, k, \mathbf{u}_k, 0)$$
(5.5a)

$$\nu_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \tag{5.5b}$$

$$\mathbf{S}_{k|k-1} = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T$$
(5.5c)

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_{k|k-1}^{-1} \tag{5.5d}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \nu_k \tag{5.5e}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \tag{5.5f}$$

É importante ressaltar que, no filtro de Kalman para sistemas invariantes no tempo e que satisfazem algumas hipóteses razoáveis, a matriz de ganho  $\mathbf{K}_k$  converge para um valor fixo. Por isso, vários produtos que utilizam o filtro de Kalman calculam-na num computador externo e utilizam uma versão *hard-coded* no produto, evitando o recálculo repetido. Ao fazer isso, eles permitem uma convergência mais lenta da estimativa para realizar computações mais rápidas. No entanto, a convergência pode não acontecer para sistemas variantes no tempo. Como o cálculo das Jacobianas das funções dependem da estimativa corrente, a propagação da covariância  $\mathbf{P}_{k|k}$  se comporta como um sistema variante no tempo e, portanto, não é possível pré-computar a matriz  $\mathbf{K}_k$ .

#### 5.1.1.2 UKF

O UKF (*Unscented Kalman Filter*) (JULIER *e outros*, 2000) foi criado no intuito de gerar um filtro baseado no filtro de Kalman para sistemas não lineares e mais robusto do que o EKF. Frequentemente o EKF se torna muito confidente em sua estimativa, possuindo uma matriz de covariância

menor do que a real (JULIER E UHLMANN, 1997). Para resolver isto, usualmente os ruídos têm suas matrizes de covariância amplificadas, visando evitar que a estimativa se torne inconsistente. No entanto, aumentar a covariância reduz a performance do sistema e torna a estimativa menos eficiente. O UKF visa resolver estes problemas.

O UKF se baseia nos chamados pontos sigma  $\mathcal{X}_i$ , que são amostras que representam a distribuição de probabilidade. Ao contrário das partículas do filtro de partícula, os pontos sigma são determinísticos dada uma distribuição de probabilidade. Estes pontos são uma combinação dos estados e ruídos atuantes no sistema da seguinte maneira:

$$\mathcal{X}_{i}(k|k) = \begin{bmatrix} \mathcal{X}_{i,x}(k|k) \\ \mathcal{X}_{i,\epsilon}(k|k) \\ \mathcal{X}_{i,\delta}(k+1|k+1) \end{bmatrix}$$
 (5.6a)

$$\sigma_{k} = \begin{bmatrix} \mathbf{P}_{k|k} & 0 & 0 \\ 0 & \mathbf{Q}_{k} & 0 \\ 0 & 0 & \mathbf{R}_{k+1} \end{bmatrix}.$$
 (5.6b)

Pode-se considerar casos onde os pontos sigma são utilizados para fazer apenas a predição do sistema, não possuindo os termos  $\mathcal{X}_{i,\delta}$ , ou para fazer apenas a correção, não possuindo portanto  $\mathcal{X}_{i,\epsilon}$ . A formulação aqui apresentada é a completa, mas produz os mesmos resultados de se gerar pontos para a previsão, colapsá-los, gerar pontos para correção e colapsá-los.

Os pontos são dados por:

$$\mathcal{X}_0(k-1|k-1) = \begin{bmatrix} \hat{\mathbf{x}}_{k-1|k-1}^T & 0 & 0 \end{bmatrix}^T$$
 (5.7a)

$$\mathcal{X}_i(k-1|k-1) = \mathcal{X}_0(k-1|k-1) + \left(\sqrt{(n+\kappa)\sigma_{k-1}}\right)_i, \quad i \in \mathbb{N}_n$$
 (5.7b)

$$\mathcal{X}_{i+n}(k-1|k-1) = \mathcal{X}_0(k-1|k-1) - \left(\sqrt{(n+\kappa)\sigma_{k-1}}\right)_i, \quad i \in \mathbb{N}_n,$$
 (5.7c)

onde n é a dimensão do estado aumentado, sendo portanto a soma dos tamanhos do vetor de estados, do ruído  $\epsilon$  e do ruído  $\delta$ ,  $(A)_i$  é a i-ésima linha ou coluna da matriz A, dependendo da técnica utilizada para calcular a raiz $^{10}$  e  $\kappa$  é um parâmetro a escolha do projetista, devendo ser escolhido de

 $<sup>^{-10}</sup>$ Se a raiz de A de  $\sigma_k$  for da forma  $\sigma_k = A^T A$ , então as linhas de A devem ser utilizadas. Se  $\sigma_k = AA^T$ , as colunas devem ser utilizadas. (JULIER *e outros*, 2000)

forma a aumentar a performance do filtro<sup>11</sup> (JULIER *e outros*, 1995). Cada ponto possui também um peso  $W_i$  associado, cujos valores são:

$$W_0 = \frac{\kappa}{n + \kappa} \tag{5.8a}$$

$$W_i = \frac{1}{2(n+\kappa)}, \quad i \in \mathbb{N}_{2n}. \tag{5.8b}$$

Existem formulações que utilizam um conjunto  $\overline{W}_i$  de pesos diferentes para a matriz de covariância (JULIER E UHLMANN, 2004), visando reduzir a chance da estimativa se tornar inconsistente, mas estas formulações não serão utilizadas neste trabalho.

Uma vez calculados os pontos sigma, cada um deles é utilizado para gerar uma previsão do estado futuro. Assim, o ponto sigma futuro pode ser calculado como:

$$\mathcal{X}_{i,x}(k|k-1) = f(\mathcal{X}_{i,x}(k-1|k-1), k, \mathbf{u}_{k-1}, \mathcal{X}_{i,w}(k-1|k-1)). \tag{5.9}$$

A partir destas previsões, cria-se a distribuição de probabilidade da previsão do estado futuro como:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} W_i \mathcal{X}_{i,x}(k|k-1)$$
 (5.10a)

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} W_i \left[ \mathcal{X}_{i,x}(k|k-1) - \hat{\mathbf{x}}_{k|k-1} \right] \left[ \mathcal{X}_{i,x}(k|k-1) - \hat{\mathbf{x}}_{k|k-1} \right]^T.$$
 (5.10b)

Utilizando os pontos sigma da previsão, calcula-se a medição esperada por cada um com:

$$\mathcal{Z}_i(k|k-1) = h(\mathcal{X}_{i,x}(k|k-1), k, \mathbf{u}_k, \mathcal{X}_{i,v}(k|k)). \tag{5.11}$$

De forma semelhante ao cálculo da distribuição de probabilidade da previsão, pode-se calcular a distribuição para a medida. Para fazer a correção, calcula-se também a covariância entre as

<sup>&</sup>lt;sup>11</sup>Para a gaussiana,  $n + \kappa = 3$  é ótimo.

previsões e as medidas:

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} W_i \mathcal{Z}_i(k|k-1)$$
 (5.12a)

$$\mathbf{P}_{k|k-1}^{zz} = \sum_{i=0}^{2n} W_i \left[ \mathcal{Z}(k|k-1) - \hat{\mathbf{z}}_{k|k-1} \right] \left[ \mathcal{Z}(k|k-1) - \hat{\mathbf{z}}_{k|k-1} \right]^T$$
 (5.12b)

$$\mathbf{P}_{k|k-1}^{xz} = \sum_{i=0}^{2n} W_i \left[ \mathcal{X}_{i,x}(k|k-1) - \hat{\mathbf{x}}_{k|k-1} \right] \left[ \mathcal{Z}(k|k-1) - \hat{\mathbf{z}}_{k|k-1} \right]^T.$$
 (5.12c)

Calculadas estas distribuições, é possível aplicar o filtro de Kalman para corrigir a estimativa do estado  $\mathbf{x}_k$ , alterando seu modo e covariância:

$$\mathbf{K}_{k} = \mathbf{P}_{k|k-1}^{xz} \mathbf{P}_{k|k-1}^{zz}^{-1} \tag{5.13a}$$

$$\nu_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \tag{5.13b}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \nu_k \tag{5.13c}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{k|k-1}^{zz} \mathbf{K}_k^T$$
 (5.13d)

## 5.1.2 Equacionamento dos modelos

Os modelos apresentados nesta seção serão utilizados posteriormente para filtragem. No entanto, quase todos são contínuos, enquanto os filtros apresentados anteriormente dependem de modelos discretos.

Considere o seguinte modelo contínuo:

$$\dot{\mathbf{x}}(t) = f'(\mathbf{x}(t), t, \mathbf{u}(t), \epsilon(t)). \tag{5.14}$$

Para transformar este modelo contínuo no formato discreto descrito pela Eq. (5.1), devese considerar o período de amostragem  $T_s$ . Suponha que  $\mathbf{u}(t)$  e  $\epsilon(t)$  são constantes no intervalo  $t \in [(k-1)T_s, kT_s]$ , pode-se calcular o valor de  $\mathbf{x}(kT_s)$  a partir de  $\mathbf{x}((k-1)T_s)$  da seguinte maneira:

$$\mathbf{x}(kT_s) = f(\cdot) = \mathbf{x}((k-1)T_s) + \int_{(k-1)T_s}^{kT_s} f'(\mathbf{x}(\tau), \tau, \mathbf{u}((k-1)T_s), \epsilon((k-1)T_s)) d\tau.$$
 (5.15)

No entanto, se  $f'(\cdot)$  for aproximadamente constante no intervalo considerado, a aproximação de primeira ordem:

$$\mathbf{x}(kT_s) \approx \tilde{f}(\cdot) = \mathbf{x}((k-1)T_s) + T_s f'(\mathbf{x}((k-1)T_s), kT_s, \mathbf{u}((k-1)T_s), \epsilon((k-1)T_s))$$
 (5.16)

pode ser utilizada, sendo computacionalmente mais barata. Estes são os dois modelos de discretização que serão utilizados para transformar o sistema contínuo em discreto na etapa de previsão dos filtros.

Além disso, para o EKF, deve-se calcular as Jacobianas, usadas na Eq. (5.3). Para isto, montase o modelo linearizado da Eq. (5.14) em torno de  $\mathbf{x}((k-1)T_s)$ . A linearização do modelo é dada por:

$$\dot{\mathbf{e}} = A_{k-1}\mathbf{e} + B_{k-1}\epsilon, \quad \mathbf{e} = \mathbf{x} - \mathbf{x}((k-1)T_s)$$
(5.17a)

$$\mathbf{A}_{k} = \left. \frac{\partial f'}{\partial \mathbf{x}} \right|_{\mathbf{x} = \mathbf{x}(kT_{*})} \tag{5.17b}$$

$$\mathbf{B}_{k} = \frac{\partial f'}{\partial \epsilon} \bigg|_{\mathbf{x} = \mathbf{x}(kT_{s})}.$$
 (5.17c)

A discretização de um sistema linear é bem conhecida e vale:

$$\mathbf{F}_{k} = \exp\left(\int_{kT_{s}}^{(k+1)T_{s}} \mathbf{A}_{k}(\tau) d\tau\right), \qquad \mathbf{G}_{k} = \int_{kT_{s}}^{(k+1)T_{s}} e^{\mathbf{A}_{k}((k+1)T_{s}-\tau)} \mathbf{B}_{k}(\tau) d\tau.$$
 (5.18)

Caso  $A_k$  e  $B_k$  sejam constantes no intervalo considerado, as equações simplificam para

$$\mathbf{F}_k = e^{\mathbf{A}_k T_s}, \qquad \mathbf{G}_k = \left(\int_0^{T_s} e^{\mathbf{A}_k \tau} d\tau\right) \mathbf{B}_k. \tag{5.19}$$

Novamente, este cálculo possui custo computacional elevado e é possível usar as aproximações de primeira ordem, sendo elas:

$$\tilde{\mathbf{F}}_k = \mathbf{I} + T_s \mathbf{A}_k, \qquad \tilde{\mathbf{G}}_k = T_s \mathbf{B}_k.$$
 (5.20)

Para os sensores, modelos devidamente calibrados serão utilizados, de forma que na Eq. (3.4) serão considerados  $\mathbf{K} = \mathbf{I}$ ,  $\mathbf{b} = \mathbf{0}$  e a rotação  $\mathcal{R} = \mathbf{I}$ , onde  $\mathbf{I}$  é a matriz identidade.

#### 5.1.2.1 Modelo dinâmico de atitude

O modelo de atitude a ser utilizado corresponde à parte de atitude do modelo do quadrirrotor, apresentado nas Eq. (3.1b) e (3.1c). Novamente,  $\mathbf{q}$  é um quaternion unitário que leva de um sistema inercial  $\mathcal{I}$  para o sistema fixo corpo  $\mathcal{B}$  e  $\omega$  é a velocidade angular do corpo medido em  $\mathcal{B}$ , resultando no seguinte modelo:

$$\dot{\mathbf{q}} = \frac{1}{2}\omega \otimes \mathbf{q} \tag{5.21a}$$

$$\dot{\omega} = \mathbf{J}^{-1} \left( -\omega \times \mathbf{J}\omega + \tau + \epsilon_{\tau} \right), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\tau}), \tag{5.21b}$$

onde J é a inércia do sistema e  $\tau$  é uma entrada conhecida do sistema.

Definindo o vetor de estados  $\mathbf{x} = [\mathbf{q}, \omega]^T$ , o sistema pode ser reescrito como:

$$\dot{\mathbf{q}} = f_{\mathbf{q}}(\mathbf{q}, \omega) = \frac{1}{2} \begin{bmatrix} -q_1 \omega_x - q_2 \omega_y - q_3 \omega_z \\ q_0 \omega_x - q_2 \omega_z + q_3 \omega_y \\ q_0 \omega_y + q_1 \omega_z - q_3 \omega_x \\ q_0 \omega_z - q_1 \omega_y + q_2 \omega_x \end{bmatrix}$$
(5.22a)

$$\dot{\omega} = f_{\omega}(\omega, \tau, \epsilon_{\tau}) = \mathbf{J}^{-1} \left( -\omega \times \mathbf{I}\omega + \tau + \epsilon_{\tau} \right), \tag{5.22b}$$

cujas Jacobianas são:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_{\mathbf{q}}}{\partial \mathbf{q}} & \frac{\partial f_{\mathbf{q}}}{\partial \omega} \\ \mathbf{0} & \frac{\partial f_{\omega}}{\partial \omega} \end{bmatrix} . \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \frac{\partial f_{\omega}}{\partial \epsilon_{\tau}} \end{bmatrix}, \tag{5.23}$$

onde:

$$\frac{\partial f_{\mathbf{q}}}{\partial \mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_{x} & -\omega_{y} & -\omega_{z} \\ \omega_{x} & 0 & -\omega_{z} & \omega_{y} \\ \omega_{y} & \omega_{z} & 0 & -\omega_{x} \\ \omega_{z} & -\omega_{y} & \omega_{x} & 0 \end{bmatrix}$$

$$\frac{\partial f_{\mathbf{q}}}{\partial \omega} = \frac{1}{2} \begin{bmatrix} -q_{1} & -q_{2} & -q_{3} \\ q_{0} & q_{3} & -q_{2} \\ -q_{3} & q_{0} & q_{1} \\ q_{2} & -q_{1} & q_{0} \end{bmatrix}$$
(5.24a)

$$\frac{\partial f_{\mathbf{q}}}{\partial \omega} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & q_3 & -q_2 \\ -q_3 & q_0 & q_1 \\ q_2 & -q_1 & q_0 \end{bmatrix}$$
(5.24b)

$$\frac{\partial f_{\omega}}{\partial \omega} = -\mathbf{J}^{-1} \begin{bmatrix} c(\vec{\mathbf{x}}) & c(\vec{\mathbf{y}}) & c(\vec{\mathbf{z}}) \end{bmatrix}, \quad c(\mathbf{v}) = \mathbf{v} \times \mathbf{J}\omega + \omega \times \mathbf{J}\mathbf{v}$$
 (5.24c)

$$\frac{\partial f_{\omega}}{\partial \epsilon} = \mathbf{J}^{-1} \tag{5.24d}$$

e  $\vec{x}$ ,  $\vec{y}$  e  $\vec{z}$  são os vetores unitários nas direções x, y e z, respectivamente.

Definindo estas equações, as discretizações descritas na seção anterior podem ser utilizadas juntamente com um filtro para fazer o passo da predição.

#### Modelo cinemático de atitude 5.1.2.2

O modelo cinemático é semelhante ao modelo desenvolvido na seção anterior, mas considerando apenas a equação:

$$\dot{\mathbf{q}} = f_{\mathbf{q}}(\mathbf{q}, \omega) = \frac{1}{2}\omega \otimes \mathbf{q},$$
 (5.25)

onde  $\omega$  passa a ser a entrada conhecida do sistema.

Para utilizar os filtros com o modelo cinemático, deve-se adicionar ruídos de processo condizentes. Usualmente, a prática é adicionar uma quantidade de ruídos igual à quantidade de estados (THRUN e outros, 2005), visando evitar que graus de liberdade sejam perdidos. No entanto, como existe a restrição de que o quaternion q deve ser unitário, o estado possui apenas três graus de liberdade. Portanto, é possível adicionar apenas três ruídos sem gerar distribuições de probabilidade incorretas. Além disso, os ruídos podem garantir que a norma de q permaneça unitária se forem aplicados da seguinte maneira (SABATINI, 2006):

$$\dot{\mathbf{q}} = \frac{1}{2}(\epsilon_{\omega} + \omega) \otimes \mathbf{q}, \quad \epsilon_{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\omega}).$$
 (5.26)

As Jacobianas resultantes são, portanto, dadas por:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_{\mathbf{q}}}{\partial \mathbf{q}} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \frac{\partial f_{\mathbf{q}}}{\partial \omega} \end{bmatrix}, \tag{5.27}$$

onde  $\frac{\partial f_{\bf q}}{\partial {\bf q}}$  e  $\frac{\partial f_{\bf q}}{\partial \omega}$  são dadas pelas Eqs. (5.24a) e (5.24b), respectivamente.

#### 5.1.2.3 Modelo dinâmico de posição

Com base na Eq. (3.1a), supondo que a aceleração  $\ddot{\mathbf{p}}$  do sistema é conhecida, o modelo discreto pode ser escrito como:

$$\mathbf{p}_{k} = \mathbf{p}_{k-1} + T_{s}\mathbf{v}_{k-1} + \frac{T_{s}^{2}}{2}\ddot{\mathbf{p}}$$

$$\mathbf{v}_{k} = \mathbf{v}_{k-1} + T_{s}\ddot{\mathbf{p}},$$
(5.28)

onde  $T_s$  é o intervalo de amostragem e  $\ddot{p}$  foi suposta constante durante todo o intervalo.

Utilizando a Eq. (5.3) para computar as Jacobianas, chega-se a:

$$\mathbf{F}_k = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{G}_k = \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \end{bmatrix}. \tag{5.29}$$

#### 5.1.2.4 Giroscópio

A partir das Eqs. (3.4) e (3.7), tem-se que a leitura do giroscópio é dada por:

$$\mathbf{z}_k^{\omega} = \omega_k + \delta_k^{\omega}, \quad \delta_k^{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\omega}),$$
 (5.30)

cujas Jacobianas são simplesmente

$$\mathbf{H}_k^{\omega} = \mathbf{I}_3, \qquad \mathbf{L}_k^{\omega} = \mathbf{I}_3. \tag{5.31}$$

#### Magnetômetro 5.1.2.5

Para o magnetômetros, a partir das Eqs. (3.4) e (3.8), tem-se que

$$\mathbf{z}_k^m = \mathbf{R}(\mathbf{q}_k)^{-1}\mathbf{h} + \delta_k^m, \quad \delta_k^m \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_m),$$
 (5.32)

onde  $\mathbf{R}(\mathbf{q}_k)$  é a matriz de rotação associada ao quaternion  $\mathbf{q}_k$ , que pode ser obtida utilizando-se o método descrito na seção A.2.3.

Para calcular as Jacobianas, devem-se considerar a restrição de que a norma do quaternion deve permanecer unitária. No entanto, notou-se em simulação que ignorar esta restrição não piora a qualidade do filtro, uma vez que a variância do quaternion é reduzida, e permite menor tempo de computação a cada iteração. Portanto, as Jacobianas serão calculadas como:

$$\mathbf{L}_k^m = \mathbf{I}_3,\tag{5.33a}$$

$$\mathbf{H}_{k}^{m} = \begin{bmatrix} \mathbf{H}_{k,1}^{m}(\mathbf{q}_{k}) & \mathbf{H}_{k,2}^{m}(\mathbf{q}_{k}) & \mathbf{H}_{k,3}^{m}(\mathbf{q}_{k}) & \mathbf{H}_{k,4}^{m}(\mathbf{q}_{k}) \end{bmatrix},$$
 (5.33b)

$$\mathbf{H}_{k,1}^{m}(\mathbf{q}) = \begin{bmatrix} 2\mathbf{h}_{x}q_{0} - 2\mathbf{h}_{z}q_{2} \\ -2\mathbf{h}_{x}q_{3} + 2\mathbf{h}_{z}q_{1} \\ 2\mathbf{h}_{x}q_{2} + 2\mathbf{h}_{z}q_{0} \end{bmatrix}, \qquad \mathbf{H}_{k,2}^{m}(\mathbf{q}) = \begin{bmatrix} 2\mathbf{h}_{x}q_{1} + 2\mathbf{h}_{z}q_{3} \\ 2\mathbf{h}_{x}q_{2} + 2\mathbf{h}_{z}q_{0} \\ 2\mathbf{h}_{x}q_{3} - 2\mathbf{h}_{z}q_{1} \end{bmatrix}, \qquad (5.33c)$$

$$\mathbf{H}_{k,3}^{m}(\mathbf{q}) = \begin{bmatrix} -2\mathbf{h}_{x}q_{2} - 2\mathbf{h}_{z}q_{0} \\ 2\mathbf{h}_{x}q_{1} + 2\mathbf{h}_{z}q_{3} \\ 2\mathbf{h}_{x}q_{0} - 2\mathbf{h}_{z}q_{2} \end{bmatrix}, \qquad \mathbf{H}_{k,4}^{m}(\mathbf{q}) = \begin{bmatrix} -2\mathbf{h}_{x}q_{3} + 2\mathbf{h}_{z}q_{1} \\ -2\mathbf{h}_{x}q_{0} + 2\mathbf{h}_{z}q_{2} \\ 2\mathbf{h}_{x}q_{1} + 2\mathbf{h}_{z}q_{3} \end{bmatrix}. \qquad (5.33d)$$

$$\mathbf{H}_{k,3}^{m}(\mathbf{q}) = \begin{bmatrix} -2\mathbf{h}_{x}q_{2} - 2\mathbf{h}_{z}q_{0} \\ 2\mathbf{h}_{x}q_{1} + 2\mathbf{h}_{z}q_{3} \\ 2\mathbf{h}_{x}q_{0} - 2\mathbf{h}_{z}q_{2} \end{bmatrix}, \qquad \mathbf{H}_{k,4}^{m}(\mathbf{q}) = \begin{bmatrix} -2\mathbf{h}_{x}q_{3} + 2\mathbf{h}_{z}q_{1} \\ -2\mathbf{h}_{x}q_{0} + 2\mathbf{h}_{z}q_{2} \\ 2\mathbf{h}_{x}q_{1} + 2\mathbf{h}_{z}q_{3} \end{bmatrix}.$$
(5.33d)

#### 5.1.2.6 Acelerômetro

Utilizando as Eqs. (3.4) e (3.9), tem-se:

$$\mathbf{z}_{k}^{a} = \mathbf{R}(\mathbf{q}_{k})^{-1}\ddot{\mathbf{p}} + \omega_{k} \times (\omega_{k} \times \mathbf{r}) + \dot{\omega}_{k} \times \mathbf{r} + \delta_{k}^{a}, \quad \delta_{k}^{a} \sim \mathcal{N}(0, \mathbf{R}_{a}).$$
 (5.34)

Supondo que  $r \approx 0$  e que a velocidade  $\omega$  não chega a níveis onde seus termos associados são significativos, a medição  $\mathbf{z}_k^a$  depende apenas do quaternion  $\mathbf{q}_k$ . Além disso, quando o modelo com apenas atitude é utilizado, tem-se  $\ddot{\mathbf{p}} = \mathbf{g}$ .

Suas Jacobianas são dadas por:

$$\mathbf{L}_k^a = \mathbf{I}_3 \tag{5.35a}$$

$$\mathbf{H}_k^a = \frac{\partial \mathbf{z}_k^a}{\partial \mathbf{q}_k},\tag{5.35b}$$

onde  $\frac{\partial \mathbf{z}_k^a}{\partial \mathbf{q}_k}$  pode ser calculado da mesma maneira que  $\mathbf{H}_k^m$  na Eq. (5.33b), substituindo-se  $\mathbf{h}_x = 0$  e  $\mathbf{h}_z = \mathbf{g}_z$ .

#### 5.1.2.7 GPS

A leitura do GPS é dada pela Eq. (3.10), reescrita aqui com a notação usada nos filtros:

$$\mathbf{z}_{k}^{GPS} = \mathbf{p}_{k} + \delta_{k}^{GPS}, \quad \delta^{GPS} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{GPS}),$$
 (5.36)

onde a medição  $\mathbf{z}_k^{GPS}$  é feita no referencial  $\mathcal{I}$ . As Jacobianas, considerando o modelo dado pela Eq. (5.28) e que  $\ddot{\mathbf{p}}$  contém o ruído de processo embutido, são simplesmente

$$\mathbf{L}_{k}^{GPS} = \left[ \begin{array}{cc} \mathbf{I}_{3} & \mathbf{0}_{3\times3} \end{array} \right] \tag{5.37a}$$

$$\mathbf{H}_{k}^{GPS} = \mathbf{I}_{3}.\tag{5.37b}$$

## 5.2 Simulações

Nesta seção, serão elaborados um conjunto de filtros para serem utilizados com o quadrirrotor. Para testar os filtros, os parâmetros de simulação foram escolhidos com o objetivo de manter o sistema em situações extremas, apresentando altas velocidades e acelerações. Com isso, o sistema é excitado em regiões onde modelos linearizados não funcionam bem e filtros podem apresentar respostas muito ruins. Ao ajustar os parâmetros do filtro e estudar seu comportamento nessas condições, sua robustez é aumentada, assim como a certeza de que ele funcionará corretamente mesmo em situações fora do esperado.

Uma das partes mais cruciais em veículos aéreos é a estimação correta da atitude. A atitude normalmente é utilizada por outros sistemas, como no caso de rotacionar a imagem de uma câmera para fazer associação com imagens previamente capturadas. Para isto, a seção 5.2.1 analisa

a estimação de atitude utilizando os modelos cinemático e dinâmico de atitude e os dois filtros previamente descritos.

Como a ampliação dos ruídos do sistema podem melhorar seu desempenho (THRUN *e outros*, 2005), uma análise do desempenho do modelo cinemático para diferentes níveis de ruído é realizada na seção 5.2.2. Os resultados obtidos motivam a busca na seção 5.2.3 por um algoritmo que reduza o erro de estimação da atitude nos casos em que ele é elevado, mas sem prejudicar significativamente os casos onde ele está próximo do ideal.

A seção 5.2.4 trata da estimação da velocidade angular, onde foi utilizado um algoritmo simples para previsão e correção, que é reforçado pela adaptação da covariância do ruído de processo. Como esta estimação se baseia numa independência da estimação da atitude, os desempenhos de outros modelos de independência entre estes estados são analisados na seção 5.2.5.

Uma vez analisado na seção 5.2.6 o problema de estimação da atitude e determinado como utilizar apenas o magnetômetro para estimação da atitude, é feita uma tentativa de estender a estimação de atitude para incluir estimação da posição e velocidade na seção 5.2.7. Para isso, é criado um sistema onde a estimação da atitude é considerado um sistema mestre e a estimação da velocidade é um escravo. Esta consideração permite que o filtro desenvolvido seja mais eficiente do que o uso do estado completo.

Por fim, a seção 5.2.8 analisa o comportamento do sistema quando ele sai do ponto de equilíbrio e os únicos esforços atuantes são ruídos gaussianos. Esta seção visa avaliar os filtros desenvolvidos na seção 5.2.7, uma vez que as condições de teste eram extremas e, possivelmente, não serão alcançadas durante o uso do sistema.

#### 5.2.1 Comparação entre modelos de atitude

Devido à complexidade e variedade dos modelos dinâmicos dos sistemas, usualmente IMUs (*Inertial Measurement Units*) e MARGs (*Magnetic, Angular Rate, and Gravity*) utilizam modelos cinemáticos da atitude para estimar a rotação atual do sistema. Além disso, encontra-se frequentemente na literatura modelos cinemáticos sendo usados para filtragem (THRUN *e outros*, 2005). No entanto, não foi encontrado nenhum trabalho que analisasse a qualidade desta aproximação, sempre assumindo que as velocidades são reduzidas, o que de fato faz com que os dois modelos

possuam comportamentos semelhantes.

Esta simulação tem como objetivo analisar a qualidade da estimação da atitude utilizando-se os modelos cinemático e dinâmico do quadrirrotor. Deseja-se justificar o uso do modelo cinemático pela qualidade de sua estimação e velocidade, além da ausência de qualquer parâmetro referente ao sistema, utilizando apenas modelos dos sensores. Serão testados também o EKF e o UKF para comparar seus resultados e decidir qual é melhor para o caso de estimação de atitude. Vale ressaltar que, apesar de  $n+\kappa=3$  ser ótimo para gaussianas utilizando o UKF, seu uso fez com que a matriz de covariância deixasse de ser positiva. Portanto os resultados aqui apresentados consideram  $\kappa=0$ .

#### 5.2.1.1 Metodologia

Para realizar as simulações, assumiu-se que o tempo total de funcionamento do sistema foi de 10 segundos, com a frequência de amostragem de cada sensor sendo 100Hz. O ruído de processo para o modelo dinâmico possui covariância  $\mathbf{Q}_{\tau}=0,1\mathbf{I}_{3}$ , enquanto os sensores possuem covariâncias  $\mathbf{R}_{a}=10^{-4}\mathbf{I}_{3}$ ,  $\mathbf{R}_{m}=10^{-4}\mathbf{I}_{3}$  e  $\mathbf{Q}_{\omega}=\mathbf{R}_{\omega}=10^{-2}\mathbf{I}_{3}$ .

Cada componente da entrada do modelo dinâmico é gerada a partir de duas distribuições de números aleatórios:  $u_1 \sim \mathcal{U}([-10,10])$  determina a intensidade do torque e  $u_2 \sim \mathcal{U}(\{1,2,\ldots,10\})$  determina a quantidade de períodos de amostragem dos sensores o torque  $u_1$  é aplicado no sistema, sendo este processo repetido até que cada período da simulação possua seu valor para entrada. Para o modelo cinemático, como a velocidade real do sistema não é conhecida, a entrada  $\omega$  é dada pela saída  $\mathbf{z}^{\omega}$  do giroscópio (SABATINI, 2006), que é suposta constante durante o intervalo de amostragem, e a covariância do ruído é a mesma covariância do giroscópio.

A condição inicial do quaternion foi gerada aleatoriamente conforme descrito em (KUFFNER, 2004) e a velocidade angular inicial foi escolhida aleatoriamente no intervalo [-5, 5] para cada componente. Os filtros foram inicializados com os estados corretos e com a matriz de covariância como identidade, onde a dimensão depende do modelo utilizado.

Para testar a qualidade numa grande gama de sistemas, a matriz de inércia foi dada por  $\mathbf{J} = \operatorname{diag}(J_x, J_y, J_z)$  com  $J_x, J_y \in \{0, 1; 0, 2; \dots; 1\}$  e  $J_z = 1$ . A escolha de  $J_z$  fixo não altera a resposta do sistema, uma vez que, considerando apenas a atitude, é possível trocar as coordenadas livremente sem perda de generalidade. Além disso, normalmente valores menores de torque são utilizados quando a inércia é reduzida. Portanto, os torques aplicados e seus ruídos são multiplicados pela

matriz de inércia, visando colocá-los na escala correta, e manter a covariância resultante do ruído de processo independente da inércia do sistema. Isto ocorre porque, segundo a Eq. (5.24d), a covariância do ruído será multiplicada por  $\mathbf{J}^{-1}$  durante a filtragem, o que compensa a multiplicação por  $\mathbf{J}$  Aqui descrita. Para cada conjunto de inércia, foram realizadas um total de 50 simulações pelo método de Monte Carlo, supondo conhecimento do quaternion e velocidade inicial corretos.

A tabela 5.1 apresenta um resumo dos parâmetros utilizados para gerar os resultados.

Parâmetro	Descrição	Valor
T	Tempo total de cada simulação	10s
$f_s$	Frequência de amostragem dos sensores	100 <b>Hz</b>
$\mathbf{Q}_{ au}$	Covariância do ruído no modelo dinâmico	$0,1\mathbf{I}_3$
$\mathbf{R}_a$	Covariância do acelerômetro	$10^{-4}$ <b>I</b> <sub>3</sub>
$\mathbf{R}_m$	Covariância do magnetômetro	$10^{-4}\mathbf{I}_3$
$\mathbf{Q}_{\omega}=\mathbf{R}_{\omega}$	Covariância do giroscópio	$10^{-2}\mathbf{I}_3$
$u_1$	Intensidade do esforço $ au_i$	$\mathcal{U}([-10,10])$
$u_2$	Período de aplicação de $u_1$	$\mathcal{U}(\{1,2,\ldots,10\})$
$q_0$	Quaternion inicial	Ver (KUFFNER, 2004)
$w_{0_i}$	<i>i</i> -ésima componente da velocidade angular inicial	$\mathcal{U}([-5,5])$
J	Matriz de inércia utilizada	$\operatorname{diag}(J_x,J_y,1)$
	Conjunto de inércias utilizadas	$J_x, J_y \in \{0, 1; 0, 2; \dots; 1\}$
N	Número de simulações para cada inércia	50
$\mathbf{g}_z$	Componente vertical da gravidade	-1
$\mathbf{h}_x$	Componente horizontal do campo magnético	$\cos 15^{\rm o}$
$\mathbf{h}_z$	Componente vertical do campo magnético	$\sin 15^{\circ}$

**Tabela 5.1:** Parâmetros para simulação de atitude.

Para cada simulação i, tem-se um conjunto  $\{\mathbf{q}_{J_x,J_y,i,j}\}$  de quaternions corretos e outro conjunto  $\{\hat{\mathbf{q}}_{J_x,J_y,i,j}^F\}$  de quaternions estimados pelo filtro F. Seja

$$\mathbf{e}_{J_x,J_y,i,j}^F = \mathbf{q}_{J_x,J_y,i} \otimes \left(\hat{\mathbf{q}}_{J_x,J_y,i}^F\right)^{-1}$$
(5.38)

o quaternion de erro entre seus quaternions corretos e estimados associados, ele descreve a rotação necessária para alinhar o sistema de coordenadas estimado  $\hat{\mathcal{B}}$  ao sistema de coordenadas  $\mathcal{B}$  correto. Sendo  ${}^0\mathbf{e}^F_{J_x,J_y,i,j}$  a componente escalar deste quaternion, que pode ser assumida positiva sem perda

de generalidade<sup>12</sup>, o ângulo de rotação necessário é descrito por:

$$\theta_{J_x,J_y,i,j}^F = 2\arccos\left({}^{0}\mathbf{e}_{J_x,J_y,i,j}^F\right). \tag{5.39}$$

Com base nestas rotações, será utilizado como medida de erro o ângulo médio de rotação requerido para uma dada combinação de inércia, resultando na seguinte expressão:

$$\mathbf{E}_{J_x,J_y}^F = \frac{1}{N^{\frac{3Tf_s}{4}}} \sum_{i=1}^N \sum_{j=\frac{Tf_s}{4}+1}^{Tf_s} \theta_{J_x,J_y,i,j}^F,$$
 (5.40)

onde os primeiros 25% dos dados são ignorados para evitar grande influência do regime transitório do filtro.

Com esta medida de erro, pode-se construir uma matriz  $\mathbf{E}_F$  onde cada elemento é dado por  $\mathbf{E}_{J_x,J_y}^F$ ,  $J_x,J_y\in\{0,1;0,2;\dots,1\}$ . No entanto, como notado anteriormente, o par  $J_x,J_y$  não gera resultados diferentes do par  $J_y,J_x$ , exceto pelo fato de se utilizar amostragens para gerar os resultados. Assim, ambos os resultados são válidos para 2 pares e uma matriz  $\mathbf{E}_F' = \frac{\mathbf{E}_F + \mathbf{E}_F^T}{2}$  é construída, o que permite obter uma superfície mais suave sem perder a corretude.

É importante ressaltar que o quaternion existente no estado, tanto do modelo dinâmico quanto do modelo cinemático, foi normalizado a cada vez que o estado era calculado. No entanto, a variação da norma do quaternion é tão pequena que estes ajustes não prejudicam os resultados.

### 5.2.1.2 Resultados

A figura 5.2 mostra os resultados obtidos utilizando o EKF e o modelo dinâmico de atitude, onde a figura 5.2a utiliza as Jacobianas calculadas conforme a Eq. (5.19) e a figura 5.2b utiliza a Eq. (5.20). Para o UKF utilizando o modelo dinâmico, o resultado é apresentado na figura 5.3. Em todos os casos do modelo dinâmico, foi utilizada a previsão correta da Eq. (5.15), uma vez que a versão aproximada apresentou resultados muito piores.

A figura 5.4 mostra os resultados do EKF utilizando o modelo cinemático com e sem aproximação das funções, enquanto a figura 5.5 mostra os resultados para o UKF utilizando ou não a

 $<sup>^{12}</sup>$ Caso ela seja negativa, pode-se utilizar o fato de que q e - q representam a mesma rotação.

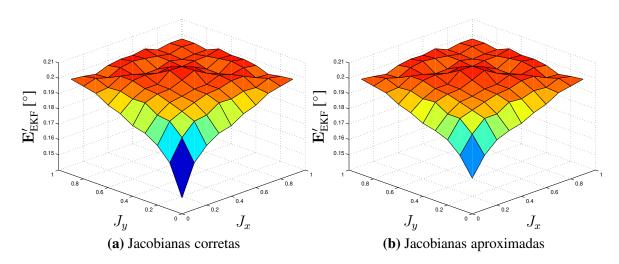


Figura 5.2: Filtragem com EKF usando modelo dinâmico.

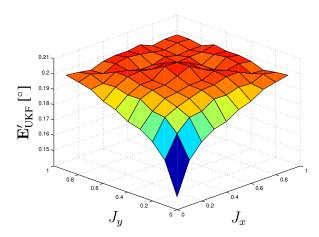


Figura 5.3: Filtragem com UKF usando modelo dinâmico.

aproximação da previsão dada pela Eq. (5.16).

Por fim, a figura 5.6 apresenta o tempo necessário por iteração para cada tipo de filtro e modelo simulados.

### **5.2.1.3** Análise

A partir das figuras 5.2 a 5.5, nota-se que todos os resultados apresentam comportamentos semelhantes para os casos onde  $J_x$  e  $J_y$  são próximos de  $J_z$ , que vale 1. Esta redução do erro pode ser associada à redução do acoplamento dinâmico entre as velocidades conforme os valores das inércias se aproximam.

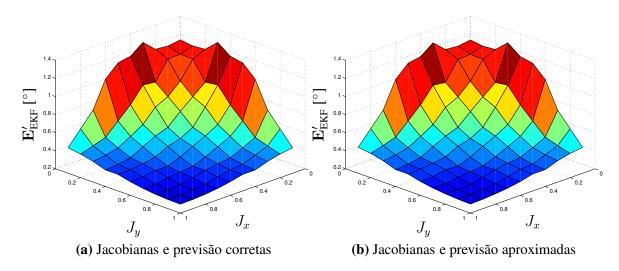


Figura 5.4: Filtragem com EKF usando modelo cinemático.

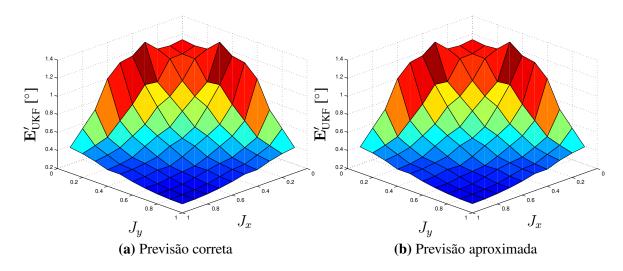
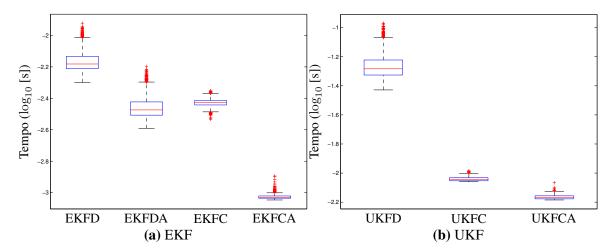


Figura 5.5: Filtragem com UKF usando modelo cinemático.

No entanto, conforme  $J_x$  e  $J_y$  diminuem, as filtragens utilizando o modelo dinâmico têm seus erros reduzidos. Isto se acentua conforme ambas inércias adquirem valores mais baixos. Este comportamento está associado à redução dos ruídos pois, apesar da covariância efetiva dos ruídos permanecer a mesma, sua influência em comparação com outros efeitos presentes no modelo dinâmico é reduzida, reduzindo assim a incerteza do sistema.

Esta redução do erro com redução das inércias é o oposto do comportamento apresentado pelos filtros utilizando o modelo cinemático. Neste caso, redução das inércias resulta em aumento significativo do erro. Conforme descrito anteriormente, valores distintos para as inércias geram o acoplamento entre as velocidades angulares, que se torna mais forte com a redução das inércias.



**Figura 5.6:** Tempo médio para computação de previsão e correção. Os sufixos 'D', 'C' e 'A' representam o uso do modelo dinâmico, cinemático e cálculo aproximado, respectivamente.

Como o modelo não possui ciência deste acoplamento, ele não pode utilizá-lo para gerar melhores estimativas. Além disso, como a leitura do giroscópio é utilizada como entrada, a entrada que teoricamente era conhecida passa a ter um valor errado.

Para entender porque isto causa problemas, considere a figura 5.7. A linha da esquerda representa a velocidade correta do sistema  $\omega$  e a distribuição de probabilidade de leituras do giroscópio está em também representada à esquerda. Suponha que a leitura resultante seja  $\mathbf{z}^{\omega}$ . Quando este valor lido é utilizado como entrada do modelo cinemático, o modelo descreve a distribuição de probabilidade da entrada pela curva central. Utilizando estas duas curvas nota-se que  $\mathbf{z}^{\omega}$  tem uma dada probabilidade na distribuição centrada em  $\mathbf{z}^{\omega}$ . No entanto, uma outra leitura  $\omega'$  possui, segundo o modelo cinemático, a mesma probabilidade de ocorrer da velocidade real  $\omega$ , apesar de ser muito distante da velocidade real. Isto motiva a ampliação e criação de ruídos como normalmente é feito ao se utilizar o modelo cinemático (Thrun *e outros*, 2005). No entanto, não existe um guia para dizer o quanto aumentar, uma vez que gera um compromisso entre velocidade de resposta e robustez do sistema. Como a frequência de oscilação das velocidades se torna mais alta conforme a inércia reduz e o modelo original é não linear, a lentidão de resposta do sistema pode fazer com que a estimativa calculada pelo filtro esteja longe do valor real, fazendo com que o filtro perca muito de sua eficiência ou mesmo divirja.

Um fato notável é que, no modelo dinâmico, o uso das Jacobinas aproximadas não gera alteração do erro medido pelo filtro, fazendo com que apenas o uso da previsão aproximada gere

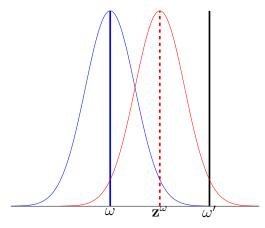


Figura 5.7: Visualização do problema ao usar a saída do giroscópio como entrada. A curva da esquerda representa a distribuição de probabilidade de uma leitura do giroscópio, centrada no valor correto de velocidade  $\omega$ . A linha tracejada representa uma leitura do giroscópio  $\mathbf{z}^{\omega}$  utilizada como entrada do modelo cinemático e sua distribuição de probabilidade correspondente, segundo o modelo. A linha preta representa uma leitura  $\omega'$  com mesma probabilidade de ocorrer que  $\omega$ , utilizando a probabilidade baseada em  $\mathbf{z}^{\omega}$ .

resultados ruins. O mesmo acontece para o modelo cinemático, mas este também possui a mesma resposta quando o modelo aproximado de previsão é usado. Como os modelos aproximados são mais rápidos de serem calculados, conforme a figura 5.6, e não apresentam alterações significativas dos resultados, eles devem ser usados ao invés dos modelos reais. O EKF com o modelo cinemático aproximado leva em torno de 1ms para computar um passo de previsão e correção. Tendo em mente que o código projetado foca na flexibilidade de testes, utilizando Python e com o filtro independendo do modelo do sistema, a otimização do código em uma linguagem de nível inferior para um caso particular deve gerar aumento significativo de performance.

Ao analisar os gráficos de erro do UKF e EKF, nota-se que eles não apresentaram diferenças perceptíveis. Apesar de o UKF ser teoricamente mais robusto, a aproximação linear para este sistema é razoável, permitindo que o EKF tenha desempenho considerável. Além disso, ao analisar o tempo do EKF e UKF para computar o modelo dinâmico, é claro que o UKF é significativamente mais lento, pois deve computar a integral do sistema para cada ponto sigma. No modelo cinemático, o uso do passo aproximado não gerou grandes benefícios para o UKF, uma vez que grande parte da computação neste caso foi gasta na parte que independe do modelo, como cálculo da raiz quadrada da matriz de covariância e cálculo da saída esperada. Por estes motivos, o EKF, neste caso, se apresenta como uma melhor alternativa.

Por fim, apesar do modelo cinemático apresentar erros, em média, de cerca de  $1,4^{\circ}$  contra  $0,2^{\circ}$  do modelo dinâmico, esta diferença é desprezível na maioria dos casos e estes erros são muito menores do que ocorre na realidade. Além disso, este modelo apresenta duas vantagens em relação ao modelo dinâmico.

Em primeiro lugar, tem-se que ele é consideravelmente mais rápido de ser computado, sendo esta diferença de aproximadamente uma ordem de grandeza. Esta diferença é mais significativa se for levado em conta que o algoritmo será utilizado num sistema de tempo real para controle. Apesar de possuir menos precisão, existe a possibilidade de ter uma malha de controle mais rápida do que se o modelo dinâmico fosse utilizado, possivelmente compensando os erros mais elevados.

Em seguida, tem-se que o sistema possui dinâmicas que não foram introduzidas no modelo dinâmico, como discutido na seção 3.1. Para não permitir que filtro utilizando o modelo dinâmico se torne inconsistente, é possível aumentar o nível de ruído considerado pelo sistema, reduzindo sua performance. Como dito anteriormente, não existe um guia para escolha desse nível de ruído, fazendo com que o uso do modelo dinâmico deva considerar o máximo de dinâmica do sistema possível. No entanto, quanto mais fatores forem considerados no modelo, mais difícil se torna a obtenção de todos os parâmetros e mais complexa se torna a etapa de previsão do sistema. O modelo cinemático, em contrapartida, funciona exatamente da mesma maneira para qualquer sistema, independente de quão complexa seja sua dinâmica.

### 5.2.2 Estudo da ampliação do ruído de processo no modelo cinemático

Na seção anterior, chegou-se à conclusão de que o modelo cinemático de atitude é mais recomendável para ser utilizado, devido à sua robustez com relação a dinâmicas não modeladas e menor tempo de computação. No entanto, sabe-se também que é uma prática comum aumentar o nível de ruído de processo no modelo cinemático, visando evitar que o filtro se torne inconsistente. Será analisado portanto o efeito gerado pelo aumento da covariância de entrada do modelo cinemático.

### 5.2.2.1 Metodologia

Para esta análise, será utilizado o mesmo método descrito na seção 5.2.1.1. Utilizar-se-á também os mesmos valores aleatórios, evitando que um novo conjunto de valores leve a observações inconsistentes. Para estimação, o modelo cinemático com EKF e computação aproximada da previsão e correção são utilizados, uma vez que esta foi uma boa combinação, conforme descrito na seção anterior. Como covariância do ruído de processo do modelo cinemático, será utilizada novamente a covariância do giroscópio  $\mathbf{R}_{\omega}$  como base. Esta covariância será multiplicada por um parâmetro  $\nu \geq 1$ , gerando uma covariância  $\mathbf{Q}_{\omega} = \nu \mathbf{R}_{\omega}$ , para aumentar o nível de ruído considerado.

Como não apenas um filtro será utilizado a partir desta seção, a notação do erro de rotação é simplificada para  $E'_{\theta}$ .

### 5.2.2.2 Resultados

A figura 5.8 traz o erro ao utilizar o filtro com diversos ganhos da matriz de covariância. O erro é calculado através da Eq. (5.40).

### **5.2.2.3** Análise

Conforme o esperado, tem-se que um aumento do ruído gerou uma redução do erro, fornecendo uma justificativa para esta prática tão comum. Ao passar de  $\nu=1$  para  $\nu=10$ , o erro foi reduzido quase pela metade nos piores casos, sem afetar significativamente a região onde os erros já eram reduzidos. No entanto, ao utilizar  $\nu=100$ , a superfície de erro se torna mais plana, ainda reduzindo o erro nos piores casos, mas gerando resultados piores nos demais.

Este aumento do erro ocorre porque, ao aumentar o ruído de processo, aumenta-se automaticamente o peso das medições. No entanto, as medições também possuem ruídos, sendo estes completamente independentes da dinâmica do sistema. Assim, o resultado corrobora com a hipótese formulada na análise da seção anterior, que disse que o erro era menor nas regiões onde as inércias são próximas devido ao comportamento da velocidade. Ao considerar a velocidade mais imprecisa, há uma elevação do erro do sistema.

Quando  $\nu=1000$ , o filtro praticamente ignora a previsão feita pelo sistema, fazendo com que a superfície de erro seja praticamente plana. Além disso, o nível de erro se eleva em todos os casos, o que leva a crer que o uso apenas das medições do acelerômetro e magnetômetro para estimação do quaternion não é recomendada.

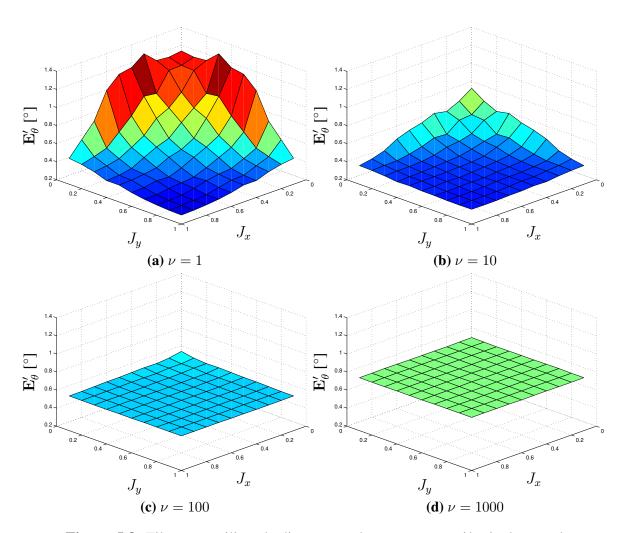


Figura 5.8: Filtragem utilizando diversos ganhos para a covariância de entrada.

### 5.2.3 Adaptação automática do ruído de processo no modelo cinemático

Ao analisar o efeito do aumento da covariância do ruído de processo, nota-se que ele visa nivelar a superfície de erro. Apesar de isto beneficiar o caso geral, levanta a questão se seria possível desenvolver um algoritmo que ajustasse conforme a necessidade de cada sistema. No caso onde as inércias são próximas, o valor da covariância deve permanecer o mesmo, mas deve ser elevado para os casos onde as inércias são menores. Além disso, como é baseado no modelo cinemático, esta adaptação não deve depender do conhecimento das inércias do sistema.

### 5.2.3.1 Desenvolvimento teórico

Considere as estimativas  $\hat{\mathbf{q}}_{k|k}$  e  $\hat{\mathbf{q}}_{k-1|k-1}$  obtidas pelo filtro. Supondo que a velocidade entre eles foi constante e que o intervalo foi de  $T_s$  segundos, a Eq. (A.16) pode ser usada para estimar a velocidade angular  $\hat{\omega}_{k-1}$  como:

$$\hat{\omega}_{k-1|k} = 2 \frac{\log\left(\hat{\mathbf{q}}_{k|k}\hat{\mathbf{q}}_{k-1|k-1}^{-1}\right)}{T_s}.$$
(5.41)

Note que a suposição de velocidade constante foi a mesma feita na construção do modelo cinemático ao utilizar a saída do giroscópio como entrada.

Seja  $\mathbf{z}_{k-1}^{\omega}$  a leitura do giroscópio utilizada para computar  $\hat{\mathbf{q}}_{k|k-1}$  a partir de  $\hat{\mathbf{q}}_{k-1|k-1}$  e considerando  $\hat{\omega}_{k-1}$  como o valor real  $\omega_{k-1}$  da velocidade, é possível estimar o ruído de entrada  $\epsilon_{k-1}^{\omega}$ . Utilizando a Eq. (5.30), chega-se a

$$\hat{\epsilon}_{k-1|k}^{\omega} = \mathbf{z}_{k-1}^{\omega} - \hat{\omega}_{k-1|k}.\tag{5.42}$$

Baseando-nos numa ideia semelhante à apresentada por Han e outros (2009), é possível adaptar uma matriz de covariância  $\mathbf{Q}_{\omega,k}$  com objetivo de melhorar a resposta do sistema, de maneira parecida com a matriz de covariância  $\mathbf{Q}_{\omega}$  descrita na seção 5.2.2.1. Esta estimativa permite que a covariância se adapte ao comportamento do sistema no momento, fornecendo estimativas mais precisas. Ela também deve ser eficiente de ser calculada, devido à restrição do tempo de computação.

Utilizando um filtro passa-baixa para as estimativas, tem-se que, partindo de uma estimativa inicial  $\mathbf{Q}_{\omega,0}$ , o algoritmo iterativo é dado por:

$$\mathbf{Q}_{\omega,k} = \alpha \mathbf{Q}_{\omega,k-1} + (1 - \alpha) \,\hat{\epsilon}_{k-1|k}^{\omega} \left(\hat{\epsilon}_{k-1|k}^{\omega}\right)^{T},\tag{5.43}$$

onde  $\alpha$  é um parâmetro definido à escolha do projetista.

### 5.2.3.2 Metodologia

Novamente, utiliza-se como base o método descrito na seção 5.2.1.1, juntamente com os mesmos valores aleatórios, evitando interpretações errôneas.

Para escolha do valor de  $\alpha$ , o intervalo [0,8;1,0] será analisado, uma vez que valores muito baixos de  $\alpha$  devem gerar comportamento mais errático do sistema. A cada  $\alpha$ , serão comparados o erro máximo, mediano e mínimo para os casos  $J_x = J_y = 0,1$  e  $J_x = J_y = 1$ , utilizando a Eq. (5.40) para cálculo do erro. A condição inicial para estimativa do ruído de entrada é dada por  $\mathbf{Q}_{\omega,0} = 100\mathbf{R}_{\omega}$ , onde  $\mathbf{R}_{\omega}$  é a covariância do giroscópio e o ganho foi escolhido por tornar o erro pequeno e uniforme na seção anterior.

### 5.2.3.3 Resultados

A figura 5.9 apresenta as medianas do erro obtido para as inércias em seus extremos e variando-se o parâmetro  $\alpha$ . Além disso, são mostrados também os mínimos e máximos do erro.

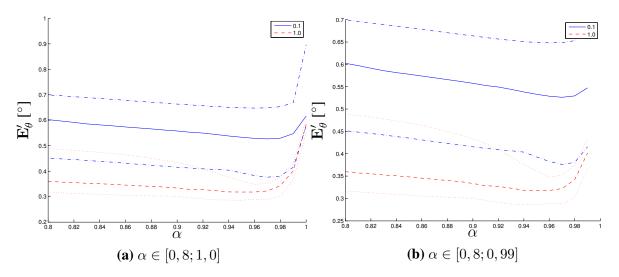
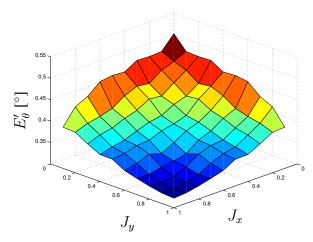


Figura 5.9: Máximo, mediana e mínimo do erro para  $J_x=J_y=0, 1$  e  $J_x=J_y=1$  variando-se  $\alpha$ .

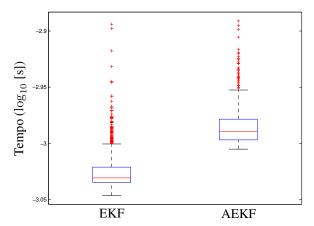
Na figura 5.10, tem-se a superfície de erro resultante para inércias variantes utilizando  $\alpha = 0,97$ . O tempo médio de computação de cada iteração é apresentado na figura 5.11 com o nome de AEKF, juntamente com o tempo do EKF da figura 5.6a.

### **5.2.3.4** Análise

Ao analisar o comportamento do erro para o parâmetro  $\alpha$  na figura 5.9a, notam-se que o caso  $\alpha=1$ , que corresponde a não haver adaptação, gera um erro muito superior para os ângulos tanto no caso de baixas quanto altas inércias. Para notar melhor os detalhes da variação de  $\alpha$ , a figura 5.9b deve ser analisada. Nesta figura, é possível notar que valores reduzidos de  $\alpha$  tendem a



**Figura 5.10:** Filtragem utilizando  $\alpha = 0,97$ .



**Figura 5.11:** Tempo médio para computação de previsão e correção. O prefixo 'A' indica a adaptação da matriz de covariância.

gerar erros maiores, provavelmente devido a passos muito grandes da adaptação. Já para valores elevados de  $\alpha$ , é claro que existe um vale em torno do qual todos os erros se tornam menores. Analisando o caso  $J_x=J_y=0,1$ , que apresenta piores respostas, o mínimo de suas curvas está próximo de  $\alpha=0,97$ , onde este valor também gera erros reduzidos para o caso onde as inércias são iguais.

Com  $\alpha=0,97$ , o algoritmo adaptativo é utilizado nas diferentes combinações de inércia. Fica claro na figura 5.10, ao compará-la com a figura 5.8, que a região da curva que apresentava pequenos erros teve seus erros elevados ligeiramente, mas o ganho de desempenho nas regiões com grandes erros foi expressiva. Ao contrário do uso de um parâmetro  $\nu$  fixo para ajustar a covariância de processo, o algoritmo adaptativo apresentado nesta seção foi capaz de reduzir os erros de pior caso sem piorar significativamente os resultados nos melhores casos. De maneira indireta, pelo

comportamento do sistema, ele foi capaz de perceber a regularidade das velocidades e se adaptar de acordo, conforme desejado.

Como o filtro é apenas um passa-baixa, seu tempo de computação é bem reduzido. Comparando-o como EKF original através da figura 5.11, o tempo de computação por iteração passou de cerca de 1ms para 1,2ms. Considerando o aumento de desempenho para pequena diferença no custo computacional, o uso do algoritmo adaptativo é justificado.

### 5.2.4 Estimação da velocidade angular

Com a atitude estimada, esta seção busca estimar a velocidade angular. As únicas duas possíveis fontes de informação são a leitura do giroscópio e estimação através de quaternions consecutivos, como feito na seção 5.2.3.

O uso de quaternions consecutivos possui o problema de que a estimação da velocidade angular  $\hat{\omega}_{k-1}$  depende das estimações do quaternion nos instantes k e k-1, ou seja, a estimação depende de um estado futuro, devendo ser atrasada. Isto poderia ser remediado estendendo-se o filtro para que ele estime  $\omega_{k-1}$  e  $\omega_k$  no mesmo vetor de estados, aumentando o custo computacional sem gerar grandes benefícios. Portanto, apenas a leitura do giroscópio será utilizada para estimar a velocidade de rotação.

### 5.2.4.1 Desenvolvimento teórico

O modelo de evolução da velocidade angular é semelhante ao utilizada para estimação de *bias*, caracterizando movimento browniano e tendo como: modelo a seguinte equação:

$$\omega_k = \omega_{k-1} + \theta_{k-1}^{\omega}, \quad \theta_{k-1}^{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\theta}^{\omega}),$$
 (5.44)

onde suas Jacobianas são  $I_3$ .

Usando a mesma ideia da seção 5.2.3, a matriz de covariância  $\mathbf{Q}_{\theta}^{\omega}$  será adaptada para que a previsão do filtro seja mais precisa. Para isto, será utilizada uma estimativa de  $\theta_{k-1}^{\omega}$  dada por:

$$\hat{\theta}_{k-1|k}^{\omega} = \hat{\omega}_{k|k} - \hat{\omega}_{k-1|k-1},\tag{5.45}$$

onde as estimativas são dadas pelo filtro. A partir de uma condição inicial  $\Sigma_{\theta,0}^{\omega}$ , a adaptação pode ser escrita como:

$$\mathbf{Q}_{\theta,k}^{\omega} = \beta \mathbf{Q}_{\theta,k-1}^{\omega} + (1-\beta)\hat{\theta}_{k-1|k}^{\omega} \left(\hat{\theta}_{k-1|k}^{\omega}\right)^{T}, \tag{5.46}$$

onde o parâmetro  $\beta$  é escolhido para o projetista.

### 5.2.4.2 Metodologia

O método descrito na seção 5.2.1.1 será utilizado como base e os valores utilizados para suas simulações. O resultado será avaliado para uma condição inicial de covariância  $\mathbf{Q}_{\theta,0}^{\omega} = \mathbf{I}_3$ , uma vez que o algoritmo adaptativo é capaz de corrigir a matriz adequadamente. A condição inicial dada ao filtro é composta pelo valor correto de velocidade e uma matriz de covariância igual à identidade.

A medida de erro será dada pela média da norma do erro entre a velocidade real e a estimada, podendo ser escrita como:

$$\mathbf{E}_{J_x,J_y}^{\omega} = \frac{1}{N^{\frac{3Tf_s}{4}}} \sum_{i=1}^{N} \sum_{j=\frac{Tf_s}{4}+1}^{Tf_s} \|\omega_{J_x,J_y,i,j} - \hat{\omega}_{J_x,J_y,i,j}\|,$$
(5.47)

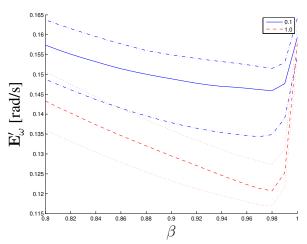
onde  $\omega_{J_x,J_y,i,j}$  são as velocidade angulares corretas e  $\hat{\omega}_{J_x,J_y,i,j}$  são suas estimativas. Utiliza-se a matriz  $\mathbf{E}'_{\omega}$  que é construída a partir dos elementos  $\mathbf{E}^{\omega}_{J_x,J_y}$  da maneira discutida na seção 5.2.1.1 para o erro de atitude.

### 5.2.4.3 Resultados

A figura 5.12 apresenta os erros de estimação para dois casos de inércia ao variar o parâmetro  $\beta$ , enquanto a figura 5.13 mostra a superfície de erro ao variar a inércia do sistema e manter  $\beta = 0,98$ . O tempo requerido pelo algoritmo é mostrado na figura 5.14.

### **5.2.4.4** Análise

Para o caso onde as inércias valem 1, 0, é evidente a partir da figura 5.12 que o valor  $\beta = 0, 98$  é o ótimo. Já para os valores de 0, 1, tem-se que este mesmo valor de  $\beta$  é o ótimo para a mediana e o máximo e possui desempenho semelhante para o mínimo, sendo portanto o escolhido.



**Figura 5.12:** Máximo, mediana e mínimo do erro para  $J_x = J_y = 0$ , 1 e  $J_x = J_y = 1$  variando-se  $\beta$ .

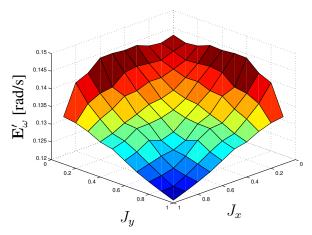


Figura 5.13: Superfície de erro para  $\beta=0,98.$ 

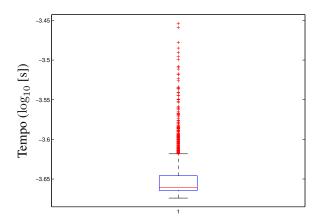


Figura 5.14: Tempo médio de para computação de previsão e correção.

A superfície de erro da figura 5.13 reforça o fato de que inércias menores gera mais acoplamento entre as velocidades, o que pode fazer com que variem mais rapidamente e sejam mais difíceis de estimar. O erro apresentado é pequeno em relação às velocidades do sistema, fazendo com que o filtro, apesar de extremamente simples, possa ser utilizado para estimar a velocidade angular. Além disso, seu tempo de computação é da ordem de  $300\mu s$ , como mostrado na figura 5.14. Este tempo de computação reduzido se deve ao fato do filtro ser linear e possuir equações extremamente simples de previsão e sensoriamento.

# 5.2.5 Modelos de independência para estimação da atitude e velocidade angular

As duas seções anteriores elaboraram filtros independentes para estimação da atitude e velocidade angular. No entanto, sabe-se a partir do modelo dinâmico que estes estados são fortemente acoplados, levantando questionamentos sobre a validade desta aproximação. Portanto, esta seção explora dois outros modelos de independência e compara os resultados obtidos.

### 5.2.5.1 Desenvolvimento teórico

O primeiro modelo visa utilizar a estimativa da velocidade angular, conforme computada na seção 5.2.4, como entrada do modelo cinemático da seção 5.2.3, substituindo a leitura do giroscópio. Este modelo possivelmente gera estimações mais precisas da velocidade do que a leitura direta do giroscópio, fazendo com que haja erros menores na estimação da atitude. Entretanto, a computação da velocidade angular permanece inalterada. Este modelo será denominado "malha direta", uma vez que o bloco de velocidade angular fornece a entrada para a atitude, mas a atitude não transmite informações de volta para a velocidade angular.

No modelo cinemático apresentado na seção 5.1.2.2, a entrada conhecida  $\omega_k$  passa a ser a estimativa  $\hat{\omega}_{k|k}$  dada pelo filtro de velocidade angular, mas a covariância  $\mathbf{Q}_{\omega}$  deve ser tratada com mais cuidado. Considere que  $\mathbf{Q}_{\omega}$  pode ser escrita como:

$$\mathbf{Q}_{\omega,k-1} = \mathbf{P}_{k-1|k-1}^{\omega} + \tilde{\mathbf{Q}}_{\omega,k-1} \tag{5.48}$$

onde  $\mathbf{P}_{k-1|k-1}^{\omega}$  é a covariância da velocidade angular estimada por seu filtro e  $\tilde{\mathbf{Q}}_{\omega,k-1}$  é um termo adicional de erro.

A cada passo da estimativa da atitude, um novo valor para  $\mathbf{Q}_{\omega,k}$  é computado de acordo com a Eq. (5.43). Como  $\mathbf{P}_{k-1|k-1}^{\omega}$  é definida pelo filtro da velocidade angular e não pode ser alterada pelo filtro de atitude, deve-se fazer

$$\tilde{\mathbf{Q}}_{\omega,k} = \mathbf{Q}_{\omega,k} - \mathbf{P}_{k-1|k-1}^{\omega},\tag{5.49}$$

para que a adaptação ocorra como esperado. Note que  $\tilde{\mathbf{Q}}_{\omega,k}$  não é necessariamente uma matriz positiva definida, uma vez que a adaptação pode escolher utilizar uma covariância menor do que a real.

O segundo modelo, denominado "malha fechada", estima simultaneamente a velocidade angular e a atitude utilizando o modelo cinemático de atitude e o modelo browniano de velocidade angular. Entretanto, notou-se durante testes que este modelo não é capaz de utilizar os algoritmos adaptativos de velocidade angular ou atitude, uma vez que a estimativa da velocidade angular atinge covariâncias muito pequenas logo no início da simulação, devido à maior precisão do acelerômetro e magnetômetro, que agora também ajustam a velocidade angular. No entanto, serão utilizadas diversas covariâncias de entrada do movimento browniano tentando identificar aquelas que resultam em menores erros.

### 5.2.5.2 Metodologia

Serão utilizados os mesmos parâmetros e métodos de avaliação discutidos nas seções 5.2.1.1 e 5.2.4.2, além de utilizar  $\alpha = 0,97$  e  $\beta = 0,98$  no caso de adaptação das matrizes de covariância.

### 5.2.5.3 Resultados

A figura 5.15 apresenta a superfície de erro de atitude para o modelo de "malha aberta". O erro da velocidade angular é o mesmo da seção anterior, uma vez que não há modificação na sua estimação, e é mostrado na figura 5.13.

As figuras 5.16 e 5.17 apresentam o erro de atitude e velocidade angular, respectivamente, para estimação simultânea destes estados.

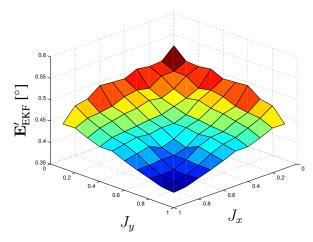


Figura 5.15: Erro de atitude utilizando o modelo de "malha aberta".

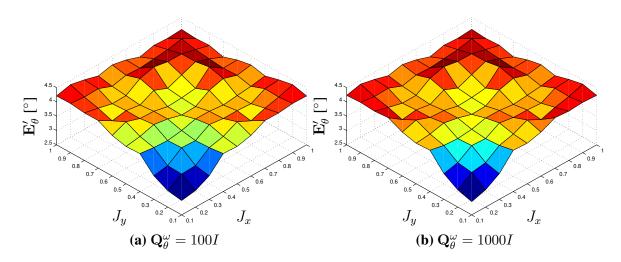


Figura 5.16: Erro de estimação de atitude para o modelo de "malha fechada".

### **5.2.5.4** Análise

Comparando as figuras 5.15 e 5.10, fica claro que o modelo que utiliza a leitura direta do giroscópio como entrada possui erros inferiores. Isto se dá ao fato da adaptação da matriz de covariância ser totalmente controlada pelo algoritmo de estimação de atitude, o que não ocorre quando a covariância da estimativa da velocidade angular é utilizada como parte do ruído de processo. Tem-se portanto que, apesar da hipótese de independência entre a velocidade angular e a atitude ser forte, ela provê uma melhor estimativa.

Este resultado é confirmado ao analisar as figuras 5.16 e 5.17. Conforme aumenta-se o ruído da entrada da velocidade angular, chega-se a um ponto onde este valor não afeta mais significati-

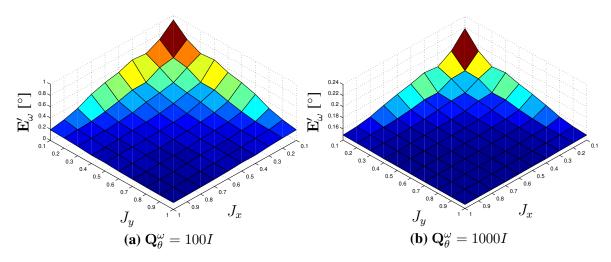


Figura 5.17: Erro de estimação da velocidade angular para o modelo de "malha fechada".

vamente a estimativa da atitude, mas a superfície de erro da velocidade angular passa a se tornar mais plana. Apesar desta uniformização do erro ser indicada para desenvolvimento de algoritmos que independam da inércia do sistema, o menor erro para  $\mathbf{Q}_{\theta}^{\omega}=1000I$  é maior do que o maior erro do modelo adaptativo apresentado na seção 5.2.4. Como o filtro acoplado não apresentou bons resultados com adaptação da covariância de sua entrada, uma vez que se tornava muito confiante nas suas estimativas de velocidade angular devido às correções do acelerômetro e magnetômetro, ele não foi capaz de competir com um filtro mais simples mas capaz desta adaptação.

### 5.2.6 Estimação de atitude utilizando apenas magnetômetro

A estimação de atitude com base em acelerômetro e magnetômetro funciona muito bem por utilizar dois vetores não co-lineares, fazendo com que, ignorando-se o ruído, haja apenas uma rotação do sistema de coordenadas que gere aquelas leituras. Entretanto, ao utilizar apenas um sensor, tem-se uma infinidade de rotações que fornecem tal leitura. Para visualizar o problema, determine uma rotação que gere a leitura. Em seguida, rotacione em torno do vetor de referência. Note que a leitura não se alterou, mas há uma nova rotação. Entretanto, o magnetômetro possui componentes não nulas em duas direções, ao contrário do acelerômetro, fazendo com que uma série de leituras limitem significativamente a quantidade de rotações possíveis para cada uma quando leva-se em conta a velocidade angular do sistema, sendo esta obtida pelo giroscópio.

A análise realizada nesta seção fornece um princípio para estimação da posição utilizando o acelerômetro, que será explorada na seção seguinte.

### 5.2.6.1 Metodologia

São utilizados os mesmos métodos das seções 5.2.2 e 5.2.3, mas as leituras do acelerômetro são desconsideradas. Para o algoritmo adaptativo, utiliza-se  $\alpha=0,97$  e, para a covariância estática,  $\mathbf{Q}_{\omega}=\nu\mathbf{R}_{\omega}$  com  $\nu\in\{1,10\}$ , onde  $\mathbf{R}_{\omega}$  é a covariância da leitura do giroscópio.

### 5.2.6.2 Resultados

As figuras 5.18 e 5.19 apresentam os resultados utilizando covariância estática e adaptativa, respectivamente.

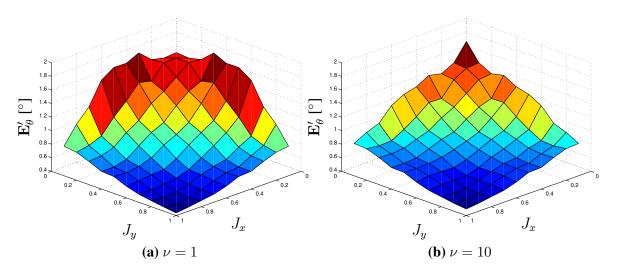


Figura 5.18: Erro de estimação de atitude utilizando a covariância estática.

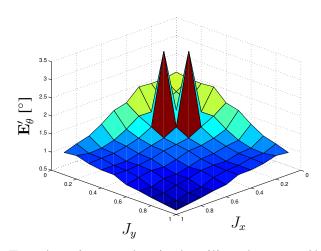


Figura 5.19: Erro de estimação de atitude utilizando a covariância adaptativa.

### **5.2.6.3** Análise

Observando a figura 5.18, nota-se que o aumento da covariância de entrada gera piores resultados para todas as combinações de inércia, o que não ocorria na seção 5.2.3. Como analisado anteriormente, o aumento da covariância visa colocar mais peso nas leituras dos sensores. Entretanto, foi discutido no início desta seção o fato da leitura apenas do magnetômetro poder ser gerada por uma infinidade de atitudes, fazendo com que a dinâmica do sistema seja mais importante durante a estimação do que quando havia dois sensores distintos para leitura.

O algoritmo adaptativo, cujo resultado é apresentado na figura 5.19, apresenta erros mais elevados do que apresentados com a covariância estática, mesmo ignorando a anomalia de elevado erro. Novamente, isto ocorre devido a estimações menos corretas da atitude, o que por sua vez gera estimações incorretas da velocidade de entrada e seu nível de ruído.

Portanto, para estimação utilizando apenas o magnetômetro, deve-se utilizar uma covariância estática e de mesmo valor do ruído do giroscópio, visando minimizar os erros de atitude. Esta conclusão será utilizada como base para escolha dos parâmetros correspondentes na seção seguinte.

### 5.2.7 Estimação da posição

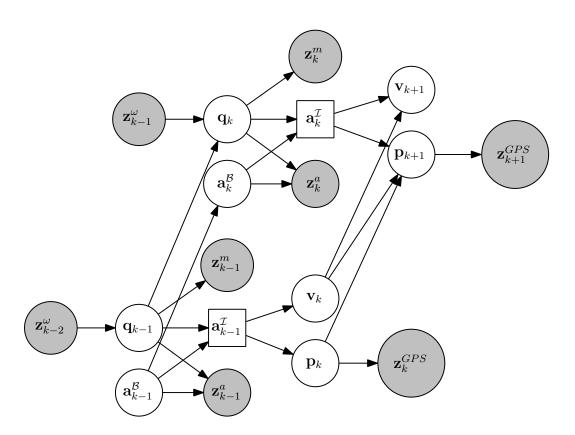
Nas seções anteriores, ao utilizar apenas o modelo de atitude, foi desconsiderado o efeito da aceleração do sistema. Se houver uma referência externa para estimação da aceleração, ela pode ser utilizada para compensar a aceleração na leitura do acelerômetro, ficando apenas com a gravidade.

Caso esta referência não exista, é possível utilizar a própria leitura do acelerômetro, mas há um custo de aumentar o erro de estimação da atitude, como discutido na seção anterior. Esta técnica é chama *dead reckoning* e é utilizada por sistemas quando apenas leituras odométricas estão presentes.

Esta seção desenvolve um filtro que é capaz de estimar a atitude e posição do sistema através do acoplamento entre dois filtros. Além dos sensores utilizados anteriormente, também é utilizado um GPS para correção da posição.

### 5.2.7.1 Desenvolvimento teórico

O GPS possui grandes erros e frequência de amostragem menor do que os outros sensores utilizados. Tais características fazem com que seu uso para correção de um modelo completo, onde há acoplamento entre a posição do sistema e sua atitude, possa ser problemática devido à aproximação linear do espaço SO(4) dos quaternions. Além disso, o modelo completo possui um custo computacional muito alto devido à dimensão do vetor de estados. Por estes motivos, foram desenvolvidos dois filtros onde a informação flui em apenas uma direção entre subsistemas. A figura 5.20 apresenta o modelo de Markov a ser utilizado, onde supõe-se que todos os sensores possuem a mesma frequência de amostragem para facilitar o desenho.



**Figura 5.20:** Modelo de Markov para estimação conjunta da posição e atitude. Os nós em branco são as variáveis escondidas, os em cinza são observações e os nós quadrados não permitem fluxo de informação para seus nós superiores.

Este modelo pode ser visto como um sistema mestre-escravo, onde o sistema de atitude e aceleração do sistema funciona de forma independente, sendo portanto o mestre. A partir dos estados

do mestre, uma função determinística é computada para chegar à aceleração no sistema inercial, a qual é fornecida de entrada para o cálculo da posição e velocidade pelo sistema escravo. Fazendo com que esta informação de aceleração só flua numa direção, os dois sistemas são desacoplados ao considerar o problema de filtragem.

Neste modelo, tem-se uma parte que é composta pelo filtro cinemático utilizado anteriormente estendido para que haja uma estimação da aceleração do sistema. A evolução da atitude do sistema permanece a mesma descrita na seção 5.1.2.2, sendo independente da evolução da estimativa da aceleração a<sup>B</sup>.

Como não há nenhuma fonte externa para fornecer informações sobre a aceleração, ela é modelada como um processo que apenas recebe um ruído aleatório e possui comportamento igual à estimação online do *bias* do acelerômetro (SABATINI, 2006). Sua evolução é dada por:

$$\mathbf{a}_k^{\mathcal{B}} = \mathbf{a}_{k-1}^{\mathcal{B}} + \theta_{k-1}^{a,\mathcal{B}}, \quad \theta_{k-1}^{a,\mathcal{B}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\theta}^{a,\mathcal{B}})$$
 (5.50)

e as Jacobianas são a matriz identidade  $I_3$ .

A leitura do acelerômetro, dada pela Eq. (5.34) e supondo  $\mathbf{r} = 0$ , pode ser reescrita como:

$$\mathbf{z}_k^a = \mathbf{a}_k^{\mathcal{B}} + \mathbf{R}(\mathbf{q}_k)^{-1}g + \delta_k^a, \quad \delta_k^a \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_a)$$
 (5.51)

e as Jacobianas originais podem ser facilmente estendidas.

Com estas modificações, tem-se um modelo para estimação da atitude  $\hat{\mathbf{q}}_{k|k}$  e da aceleração do sistema  $\hat{\mathbf{a}}_{k|k}^{\mathcal{B}}$ . Pode-se utilizar estas estimações para gerar uma estimação da aceleração  $\hat{\mathbf{a}}_{k|k}^{\mathcal{I}}$  no sistema inercial através da igualdade

$$\mathbf{a}_k^{\mathcal{I}} = \mathbf{R}(\mathbf{q}_k) \mathbf{a}_k^{\mathcal{B}} \tag{5.52}$$

e supondo que as estimativas são os valores corretos.

Para calcular a covariância da estimativa da aceleração  $\hat{\mathbf{a}}_k^{\mathcal{I}}$ , deve-se utilizar expansão da série de Taylor da Eq. (5.52). Assim, a estimativa é dada por:

$$\mathbf{P}_{k|k}^{a,\mathcal{I}} = \mathbf{J}(\hat{\mathbf{q}}_{k|k}, \hat{\mathbf{a}}_{k|k}^{\mathcal{B}}) \mathbf{P}_{k|k} \mathbf{J}_k^T (\hat{\mathbf{q}}_{k|k}, \hat{\mathbf{a}}_{k|k}^{\mathcal{B}}),$$
(5.53)

onde  $P_{k|k}$  é a matriz de covariância do filtro de atitude e aceleração e  $J(\cdot, \cdot)$  é a Jacobiana da Eq. (5.52), que, ignorando a normalização do quaternion pelo motivo descrito na seção 5.1.2.5, é dada por:

$$\mathbf{J}^{T} = \begin{bmatrix} q_{0}^{2} + q_{1}^{2} - q_{2}^{2} - q_{3}^{2} & 2q_{1}q_{2} + 2q_{0}q_{3} & 2q_{1}q_{3} - 2q_{0}q_{2} \\ 2q_{1}q_{2} - 2q_{0}q_{3} & q_{0}^{2} - q_{1}^{2} + q_{2}^{2} - q_{3}^{2} & 2q_{0}q_{1} + 2q_{2}q_{3} \\ 2q_{0}q_{2} + 2q_{1}q_{3} & 2q_{2}q_{3} - 2q_{0}q_{1} & q_{0}^{2} - q_{1}^{2} - q_{2}^{2} + q_{3}^{2} \\ 2a_{x}q_{0} - 2a_{y}q_{3} + 2a_{y}q_{2} & 2a_{x}q_{3} + 2a_{y}q_{0} - 2a_{y}q_{1} & -2a_{x}q_{2} + 2a_{y}q_{1} + 2a_{y}q_{0} \\ 2a_{x}q_{1} + 2a_{y}q_{2} + 2a_{y}q_{3} & 2a_{x}q_{2} - 2a_{y}q_{1} - 2a_{y}q_{0} & 2a_{x}q_{3} + 2a_{y}q_{0} - 2a_{y}q_{1} \\ -2a_{x}q_{2} + 2a_{y}q_{1} + 2a_{y}q_{0} & 2a_{x}q_{1} + 2a_{y}q_{2} + 2a_{y}q_{3} & -2a_{x}q_{0} + 2a_{y}q_{3} - 2a_{y}q_{2} \\ -2a_{x}q_{3} - 2a_{y}q_{0} + 2a_{y}q_{1} & 2a_{x}q_{0} - 2a_{y}q_{3} + 2a_{y}q_{2} & 2a_{x}q_{1} + 2a_{y}q_{2} + 2a_{y}q_{3} \end{bmatrix}, (5.54)$$

onde 
$$\hat{\mathbf{q}}_{k|k} = [q_0, q_1, q_2, q_3]^T$$
 e  $\hat{\mathbf{a}}_{k|k}^{\mathcal{B}} = [a_x, a_y, a_z]^T$ .

Uma vez calculada a aceleração no referencial inercial e sua covariância, ela pode ser utilizada como a entrada  $\ddot{\mathbf{p}}$  no modelo da Eq. (5.28). No entanto, é necessário chamar atenção para a conversão de escala feita. Os acelerômetros usualmente medem aceleração em relação à gravidade e este modelo foi utilizado neste trabalho ao assumir que a gravidade vale  $\mathbf{g} = [0, 0, -1]^T$ . Como a estimativa da posição é dado em metros, devido à leitura do GPS e por ser uma unidade do SI, deve-se aplicar a conversão devida. Seja  $\kappa_q$  a escala da gravidade, tem-se:

$$\hat{\ddot{\mathbf{p}}} = \kappa_{\mathbf{g}} \hat{\mathbf{a}}_{\mathbf{k}|\mathbf{k}}^{\mathcal{I}}, \qquad \mathbf{P}_{\mathbf{k}|\mathbf{k}}^{\ddot{\mathbf{p}}} = \kappa_{\mathbf{g}}^{2} \mathbf{P}_{\mathbf{k}|\mathbf{k}}^{\mathbf{a},\mathcal{I}}$$
 (5.55)

e, a partir disto, tem-se a evolução da posição como descrita na seção 5.1.2.3.

### 5.2.7.2 Metodologia

Todos os parâmetros relativos à parte angular do filtro são semelhantes aos da seção 5.2.1.1 e as mesmas variáveis aleatórias são utilizadas, permitindo a comparação com as seções anteriores. Além disso, conforme discutido na seção 5.2.6, a leitura do giroscópio é utilizada como entrada do modelo cinemático, com sua covariância como covariância do ruído de entrada.

Para cada uma das N simulações, é gerada uma matriz de covariância  $\mathbf{Q}_{\ddot{p}}$  para gerar ruídos de entrada para  $\ddot{\mathbf{p}}$  como  $\mathbf{Q}_{\ddot{p}} = \frac{10}{\kappa_g^2} \mathbf{I}_3$  de forma que, no instante k, a aceleração é dada por:

$$\ddot{\mathbf{p}}_{k} = \ddot{\mathbf{p}}_{k,1} + \epsilon_{\ddot{\mathbf{p}},k}, \quad \epsilon_{\ddot{\mathbf{p}},k} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\ddot{\mathbf{p}}}), \tag{5.56}$$

onde  $\ddot{\mathbf{p}}_{k,1}$  é constante por partes. A constante gravitacional, utilizada para conversão de valores em m/s<sup>2</sup> para g é dada por  $\kappa_g = 9,81$ .

De forma semelhante à entrada do modelo dinâmico de atitude, cada componente referência é gerada por duas distribuições de números aleatórios:  $u_1 \sim \mathcal{U}([-\frac{0.5}{\kappa_g};\frac{0.5}{\kappa_g}])$  determina a intensidade da aceleração e  $u_2 \sim \mathcal{U}(\{10,11,\ldots,100\})$  determina a quantidade de períodos de amostragem dos sensores a aceleração  $u_1$  utilizada, sendo este processo repetido até que cada período da simulação possua um valor de entrada.

Para a condição inicial, todos os estados tiveram seus valores zerados, exceto pelo quaternion, que possui o valor correto. Como assume-se que o sistema começa parado e a posição é computada em relação à sua posição inicial, a covariância do filtro de posição teve sua condição inicial zerada. Para o quaternion, foi utilizado um valor inicial para a covariância de  $\mathbf{I}_4$ , enquanto foi utilizado  $10^4\mathbf{I}_3$  para a aceleração.

A covariância do processo de estimação da aceleração é dada por  $\mathbf{Q}^{a,\mathcal{B}}_{\theta}=10^4\mathbf{I}_3$ , fazendo com que a aceleração a cada instante seja praticamente independente da anterior. Como a Eq. (5.53) utiliza a estimativa atual como valor correto para propagar a covariância, ocorre um problema semelhante ao discutido na seção 5.2.2 para o giroscópio. Portanto, utiliza-se como covariância de entrada  $\tilde{\mathbf{P}}^{a,\mathcal{I}}_{k|k}=\nu\mathbf{P}^{a,\mathcal{I}}_{k|k}$ , onde  $\nu$  é uma constante que será variada para comparar o desempenho do filtro.

O GPS foi configurado com uma frequência de amostragem de 10Hz, sendo 10 vezes menor do que os demais sensores, e uma covariância  $\mathbf{R}_{GPS}=10\mathbf{I}_3$ .

A tabela 5.2 apresenta um resumo dos parâmetros adicionais utilizados para simulação, sendo estes utilizados em conjunto com os parâmetros da tabela 5.1.

Para a estimação da atitude, a mesma Eq. (5.40) é utilizada para o erro. Já para a posição, é utilizada a média da norma dos erros, dada por:

$$\mathbf{E}_{J_x,J_y}^P = \frac{1}{N^{\frac{3Tf_s}{4}}} \sum_{i=1}^N \sum_{j=\frac{Tf_s}{4}+1}^{Tf_s} \|\mathbf{p}_{i,j} - \hat{\mathbf{p}}_{J_x,J_y,i,j}\|,$$
(5.57)

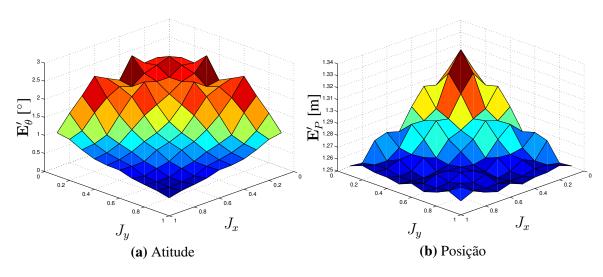
onde  $\mathbf{p}_{i,j}$  são as posições corretas obtidas pelo modelo de posição, que independe do modelo de

Parâmetro	Descrição	Valor
$u_1$	Intensidade da componente da aceleração referência $\ddot{\mathbf{p}}_{k,1}$	$\mathcal{U}([-0,5/\kappa_g;0,5/\kappa_g])$
$u_2$	Período de aplicação de $u_1$	$\mathcal{U}(\{10, 11, \dots, 100\})$
$\mathbf{Q}_{\ddot{\mathbf{p}}}$	Covariância da aceleração $\epsilon_{\ddot{\mathbf{p}},k}$	$\frac{10}{\kappa_g}\mathbf{I}_3$
$\kappa_g$	Valor da gravidade em m/s <sup>2</sup>	9,81
$\mathbf{Q}_{\theta}^{a,\mathcal{B}}$	Covariância do ruído de processo para a aceleração	$10^4 \mathbf{I}_3$
$ ilde{\mathbf{P}}_{k k}^{a,\mathcal{I}}$	Covariância do ruído de entrada na estimação da posição	$ u \mathbf{P}_{k k}^{a,\mathcal{I}}$
$f_s^{GPS}$	Frequência de amostragem do GPS	10 <b>Hz</b>
${ m R}_{GPS}$	Covariância do GPS	$10\mathbf{I}_3$

**Tabela 5.2:** Parâmetros adicionais para simulação da posição.

atitude e, portanto, das inércias, e  $\hat{\mathbf{p}}_{J_x,J_y,i,j}$  são as estimativas da posição obtidas pelo filtro. A matriz de erros  $\mathbf{E}_P'$  é construída a partir de  $\mathbf{E}_{J_x,J_y}^P$  da mesma maneira que  $\mathbf{E}_F'$  a partir de  $\mathbf{E}_{J_x,J_y}^F$  no fim da seção 5.2.1.1.

### 5.2.7.3 Resultados

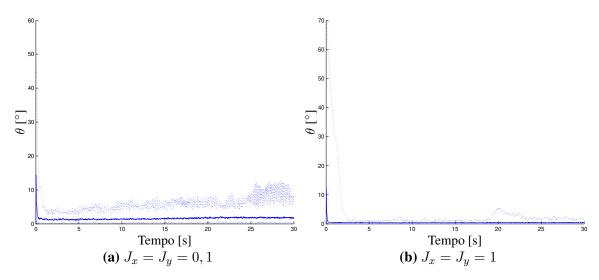


Na figura 5.21, tem-se as superfícies de erro da atitude e posição ao variar a inércia do sistema.

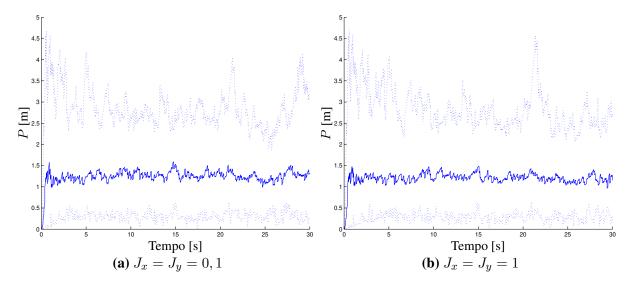
**Figura 5.21:** Erro médio para  $\nu = 1000$ .

A figura 5.22 mostra como os erros de atitude evoluem com o tempo para diferentes valores de inércias, enquanto a figura 5.23 faz o mesmo para a posição.

O histograma do tempo de computação do algoritmo é mostrado na figura 5.24.



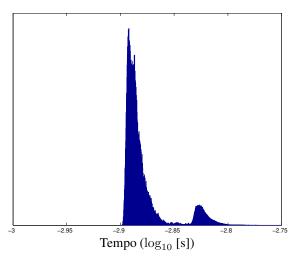
**Figura 5.22:** Evolução do erro máximo, mediano e mínimo de atitude para  $\nu=1000$ .



**Figura 5.23:** Evolução do erro máximo, mediano e mínimo de posição para  $\nu=1000$ .

### **5.2.7.4** Análise

Observando a figura 5.21, nota-se que os resultados obtidos são bem satisfatórios, chegando a baixos valores de erro de posição e atitude. Para valores de  $\nu$  maiores do que 1000 não há benefícios significativos na posição, uma vez que este valor já descarta grande parte da estimativa propagada a partir da estimação da atitude e aceleração, baseando-se principalmente na leitura do GPS. Já para valores menores, o filtro apresenta melhores estimativas para o caso mediano até  $\nu=10$ , a partir do qual se torna excessivamente confiante nas estimativas. No entanto, o pior caso do erro da estimação aumenta consideravelmente para valores de  $\nu$  abaixo de 1000, estabelecendo um



**Figura 5.24:** Histograma do tempo para computação de previsão e correção.

trade-off entre performance e robustez. Caso algum algoritmo adaptativo de estimação do ganho da entrada fosse utilizado, talvez fosse possível melhorar o desempenho, como foi feito com a estimação da atitude na seção 5.2.3.

A partir da figura 5.22, fica claro que o sistema apresenta comportamento oscilatório significativo para valores pequenos das inércias  $J_x$  e  $J_y$ , reduzindo a qualidade da estimativa. Entretanto, mesmo o pior caso do erro de atitude após o período transitório está próximo de 10 graus, o que é um valor consideravelmente baixo. Esta diferença entre as inércias não se repete na estimação da posição, apresentada na figura 5.23, uma vez que o comportamento do sistema é praticamente todo descartado devido ao alto ganho do erro de entrada, como discutido anteriormente. Apesar de se basear principalmente nas leituras do GPS, o erro obtido é considerado pequeno para voos em campo aberto.

Como a etapa de previsão da posição ocorre a cada iteração, mas a correção ocorre apenas quando há uma leitura do GPS, o que ocorre a cada 10 iterações na simulação, espera-se que a distribuição do tempo utilizado possua dois modos. A figura 5.24 confirma esta expectativa, com o pico menor sendo cerca de 9 vezes menor do que o maior. A partir desta figura, é claro que a estimativa completa do filtro pode ser feita em menos de 1,5ms mesmo quando ocorre a correção, o que somado com os  $300\mu$ s da estimação da velocidade angular, apresentada na seção 5.2.4, fornece estimação completa dos estados em menos de 2ms. Como discutido anteriormente na seção 5.2.1.3, este tempo será alterado quando otimizações das operações forem feitas no sistema final, possivelmente compensando o processador mais lento do sistema embarcado.

### 5.2.8 Estimação completa do sistema partindo do equilíbrio

Os filtros desenvolvidos nas seções anteriores foram testados com velocidades, torques e forças elevadas para avaliar sua robustez. Nesta seção, serão utilizados os mesmos filtros para avaliar o comportamento do sistema a partir de seu equilíbrio e utilizando apenas ruído gaussiano, para avaliar os erros obtidos.

### 5.2.8.1 Metodologia

O método aqui utilizado é baseado naqueles descritos nas seções 5.2.1.1, 5.2.7.2 e 5.2.4.2, realizando escolhas de parâmetros e pequenas alterações na geração dos dados.

A condição inicial em todos os casos é de velocidade angular nula e quaternion sem rotação. O parâmetro  $u_1$  da tabela 5.1 é fixado em 0, fazendo com que os únicos torques aplicados sejam ruídos gaussianos. Fazendo o mesmo para a aceleração, o parâmetro  $u_1$  da tabela 5.2 é zerado. Estas excitações representam esforços constantes aplicados ao sistema, que podem ser gerados pelo rotores ou por ventos. Assim, considerara-se aqui apenas os esforços causados por fontes variantes, representadas pelos ruídos gaussianos,

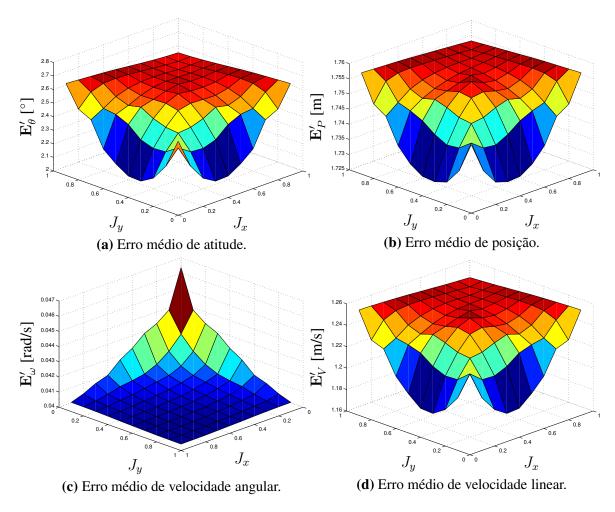
Para o parâmetro do filtro de velocidade angular, o valor  $\beta=0,98$  foi escolhido. Para avaliar a velocidade, a mesma equação de erro de posição foi utilizada, substituindo as variáveis apropriadas.

### 5.2.8.2 Resultados

A figura 5.25 apresenta os erros de estimação de todos os estados.

### **5.2.8.3** Análise

O erro de posição teve um aumento significativo em relação a quando o sistema encontrava-se sujeito ao ruído com uma componente quadrada. Apesar de não ser claro o motivo desta redução de performance, pode ser que esteja associada ao fato do ruído de processo do acelerômetro ser muito maior do que os ruídos aplicados ao sistema. Entretanto, este comportamento ainda necessita de estudos adicionais para analisar este efeito.



**Figura 5.25:** Erros para sistema saindo do equilíbrio.

Já o erro médio de atitude foi reduzido ligeiramente, além de assumir um intervalo menor de valores. Como a velocidade angular é muito inferior aos casos tratados anteriormente, o filtro é capaz de estimá-la facilmente. Como era de se esperar, os casos onde não há um forte acoplamento entre as velocidades angulares, o que evita que elas aumentem facilmente, apresentam erros semelhantes de estimação de todos os estados

O nível de erro obtido para a atitude e velocidade angular permite o uso do quadrirrotor em ambiente abertos ou fechados, contanto que não haja grandes variações do campo magnético ou alguma outra forma de estimar a atitude, como uma câmera. O erro de posição é factível com ambientes abertos, mas é muito grande para ambientes fechados. No entanto, o GPS foi desenvolvido para ambientes abertos e é sabido que ele possui erros elevados dentro de estruturas. Caso o sistema deva ser utilizado em ambiente fechado, alguma outra forma de computação da posição deve ser

utilizada, como câmeras ou sensores laser (THRUN e outros, 2005).

### 5.3 Conclusão

A partir dos resultados apresentados na seção 5.2.1, pode-se concluir que o EKF utilizando modelos aproximados para estimação da atitude do sistema é preferível em relação ao EKF usando modelo exato e UKF. Apesar do modelo cinemático apresentar erros maiores do que o modelo dinâmico, estes erros ainda são consideravelmente baixos. Como o modelo dinâmico depende do conhecimento da inércia do sistema, cujo valor incorreto pode reduzir a performance do filtro, e do ruído aplicado a ele, que pode ser variante e difícil de mensurar, uma vez que deve incluir efeitos de vento e de dinâmica presente no sistema mas não modelada, como arrasto aerodinâmico, o modelo cinemático se apresenta coma uma boa alternativa. Além disso, o tempo de computação do modelo cinemático é consideravelmente menor, o que justifica seu uso nos diversos sistemas de IMU e MARG comerciais existentes.

Na seção 5.2.2, observa-se que variar o nível de ruído na entrada do modelo cinemático nem sempre é bom, apesar de ser uma prática comum (THRUN *e outros*, 2005). Conforme o ruído aumenta, nota-se que algumas partes da superfície de erro se elevam enquanto outras abaixam, até chegar um ponto onde a entrada é praticamente desconsiderada na filtragem, o que gerou um aumento do erro em toda a superfície. Visando evitar que o erro aumente muito nas regiões que já era reduzido, mas diminua nas demais, foi desenvolvido um algoritmo de ajuste da matriz de covariância na seção 5.2.3. Este algoritmo possui apenas um parâmetro, cuja escolha segue um método apresentado na seção, e é muito eficiente. Os resultados mostraram uma grande redução no erro da estimação com um pequeno custo computacional extra, motivando o uso deste algoritmo onde for possível.

A estimação da velocidade angular é dada por um filtro linear baseado em movimento browniano, como discutido na seção 5.2.4. Os resultados mostram que o erro de estimação é reduzido para toda faixa de inércias ao utilizar a mesma ideia de adaptação utilizada para a atitude na seção 5.2.3. Apesar de a atitude e a velocidade angular serem estimadas independentemente, a seção 5.2.5 mostra que esta hipótese é plausível e obtém bons resultados em relação a dois outros modelos, principalmente pela sua capacidade de adaptação do ruído de covariância de entrada.

Como os modelos descritos levam em conta apenas a atitude, eles só podem ser utilizados se

houver uma fonte externa para informar a aceleração, que é compensada da leitura do acelerômetro. Caso contrário, a estimação da atitude deve depender praticamente só do magnetômetro, uma vez que o acelerômetro é utilizado na estimação da aceleração. A seção 5.2.6 mostrou que a ampliação do ruído de entrada, que gerou melhores resultados na seção 5.2.2, produz piores resultados ao utilizar apenas o magnetômetro, pois o comportamento dinâmico do sistema se torna mais importante para estimar a leitura esperada do sensor e, portanto, o ajuste realizado. Além disso, a adaptação da matriz de covariância também fez com que a estimativa da atitude se tornasse pior, de forma que recomenda-se o uso apenas da covariância do giroscópio como fonte de ruído na entrada do sistema.

Na seção 5.2.7 o modelo de atitude foi estendido para estimar simultaneamente a aceleração do sistema, os quais são utilizados na estimação da posição e velocidade linear, auxiliados por um GPS. Ao desacoplar o sistema de atitude e posição num formato mestre-escravo, foi obtido um algoritmo eficiente para estimação da atitude, posição e velocidade linear. Este algoritmo apresenta bons resultados tanto para a estimação da atitude quanto posição, dependendo fortemente da leitura do GPS devido à tentativa de minimizar o pior caso. A seção 5.2.8 finaliza estimando todos os estados do veículo quando este não está submetido a situações tão extremas quanto as anteriores.

Por fim, observa-se que os erros de atitude e velocidade angular do sistema são pequenos o suficiente para serem utilizados em ambientes abertos ou fechados, contanto que a leitura do magnetômetro permaneça confiável ou algum outro sensor para observação da atitude seja utilizado. Já os erros de posição e velocidade linear só são factíveis com ambientes abertos, devido à sua dimensão. No entanto, o GPS não funciona bem em ambientes fechados, o que já limita a aplicação do filtro nestas condições. Nada impede que o filtro seja utilizado com algum outro tipo de sensor que funcione melhor em ambientes internos, como câmeras ou sensores laser.

Todos os algoritmos desenvolvidos se mostraram eficientes, com um tempo máximo para uma iteração de 2ms. Como a arquitetura de software utilizada nos testes não é otimizada para os casos particulares dos sistemas, há um *overhead* computacional devido ao armazenamento de dados para análise da simulação e a linguagem utilizada foi Python, que é rápida ao fazer contas grandes com a biblioteca SciPy (Jones *e outros*, 2001–) mas lenta para fazer iterações. Uma versão otimizada para o modelo do sistema rodando num sistema embarcado e implementada em linguagem de nível inferior será capaz de realizar as estimações com grande folga de tempo.

Os algoritmos de filtragem de estados, como aqueles descritos neste capítulo, geralmente as-

sumem que as leituras dos valores utilizados para previsão e correção são dados pelos valores reais somados a ruídos. Entretanto, isto só ocorre se os sensores estiverem perfeitamente calibrados e suas imperfeições forem compensadas. Portanto, uma calibração de boa qualidade deve ser realizada para evitar que o filtro utilize informações incorretas, o que pode levar a instabilidades e falha do sistema.

# Capítulo 6

# Calibração de sensores inerciais e orientação

Este capítulo apresentará as técnicas de calibração utilizadas neste trabalho e seus resultados. As técnicas aqui desenvolvidas tem fundamento estatístico e visam construir métodos que sejam menos restritos e teoricamente mais corretos do que os encontradas na literatura, como as apresentadas na seção 2.3. Para isso, é necessário trabalhar com o caso mais geral possível e depender ao mínimo de aproximações. Além disso, as ferramentas corretas para análise dos resultados devem ser utilizadas, evitando conclusões errôneas. A grande importância da calibração correta dos sensores é percebida na Eq. (3.5), que mostra como relacionar o valor obtido pelo sensor com o valor correto de leitura.

A seção 6.1 ataca o problema de calibração simultânea de um acelerômetro e um magnetômetro, onde o modelo mais completo de parâmetros dos sensores é utilizado. São desenvolvidos dois algoritmos que, apesar de semelhantes entre si, apresentam resultados bem diferentes em alguns casos. Este algoritmo difere dos demais encontrados na literatura por assumir o caso mais geral de parametrização linear dos sensores, sendo capaz de estimar todos os parâmetros. Outras abordagens estatísticas na literatura são casos particulares, obtidas através de simplificações como covariância diagonal ou pré-conhecida ou orientações dos sensores conhecidas em cada amostragem.

A seção 6.2 trata da estimação da rotação entre o sistema de coordenadas S no qual as medições são feitas e um sistema arbitrário B definido pelo usuário. Ao invés de requisitar um alinhamento correto entre os sistemas, é importante permitir que as rotações sejam arbitrárias. Os de-

mais algoritmos de alinhamento encontrados na literatura dependem de algum referencial externo, como um sensor pré-calibrado ou outra maneira de obter os dados (posicionamento de estrelas, por exemplo). Portanto, este difere dos demais por usar apenas leituras dos próprios sensores sendo calibrados e conhecimento básico do usuário.

Por fim, a seção 6.3 apresenta uma conclusão e resumo dos resultados obtidos, destacando como os dois métodos aqui descritos podem ser utilizados nos mesmos dados para reduzir o tempo de calibração.

### 6.1 Acelerômetro e magnetômetro

Esta seção apresenta dois algoritmos para calibração simultânea de um magnetômetro e um acelerômetro, mas que podem ser estendidos facilmente para uma maior quantidade de sensores. Um dos métodos é uma adaptação do outro visando reduzir o tempo de computação através de uma aproximação, que também o torna mais robusto. Ambos são algoritmos gulosos, realizando escolhas ótimas locais durante a busca do ótimo global<sup>13</sup>, e visam maximizar a probabilidade de leitura dos sensores, dada pela Eq. (3.6). Para tal, serão utilizados os modelos dos sensores descritos na seção 3.2 e do sistema inercial da seção 3.3.

Vale ressaltar que as únicas hipóteses feitas pelo sistema de calibração são:

- 1. Os sensores são afetados por ruído gaussiano.
- 2. É possível manter os sensores parados e acoplados ao sistema com o qual serão usados durante alguns intervalos de tempo. Caso não seja possível garantir que eles se mantenham parados, mas pudermos garantir que a expectativa das perturbações medidas em cada período são nulas, então, pelo teorema do limite central, o ruído adicional pode ser considerado gaussiano com média zero se o número de amostras for suficientemente grande.

A seção 6.1.1 traz a teoria dos algoritmos, descrevendo o método de coleta de dados e os métodos de otimização utilizados. A seção 6.1.2 descreve os algoritmos que utilizam as otimizações descritas anteriormente para gerar a calibração. O método utilizado para avaliar os algoritmos

<sup>13</sup>www.nist.gov/dads/HTML/greedyalgo.html

é apresentado na seção 6.1.3 e os resultados na seção 6.1.4. Por fim, a seção 6.1.5 analisa os resultados obtidos pelos métodos de calibração apresentados.

### 6.1.1 Desenvolvimento teórico

Para aumentar a velocidade de computação, a seção 6.1.1.1 apresenta o método de resumo dos dados coletados, incluindo estimação da matriz de covariância. Com estes resumos conhecidos, pode-se definir a função de custo na seção 6.1.1.2 a ser minimizada durante a calibração. A otimização é separada em três etapas: otimização da rotação de um dado intervalo na seção 6.1.1.3, dos *bias* e campos atuantes nos sensores na seção 6.1.1.4 e das matrizes de ganho na seção 6.1.1.6.

A seção 6.1.1.5 apresenta dois problemas encontrados durante a calibração, o que faz com que existam infinitas soluções para o problema geral, descrevendo também suas possíveis soluções, que são utilizadas nas calibrações feitas.

### 6.1.1.1 Coleta dos dados para calibração

A coleta de dados de maneira apropriada é essencial para uma calibragem correta do sistema. A calibração aqui apresentada é composta de intervalos de tempo no qual o sistema encontra-se parado e fixo ao corpo ao qual fará uso dos sensores. Esta restrição visa evitar acelerações diferentes da gravidade e campos magnéticos variantes. Após capturar cada intervalo, um pré-processamento de dados pode ser feito, reduzindo a quantidade de dados a ser processada pelo resto da calibração sem perder nenhuma informação

Para cada sensor  $s \in \{a, m\}$  e cada intervalo  $i \in \mathbb{N}_N$ , onde N é a quantidade total de intervalos coletados, tem-se uma série de amostras

$$\mathbf{s}_s[i,j] \sim \mathcal{N}(\mu_s[i], \Sigma_s), \quad j \in \mathbb{N}_{\Delta_s[i]},$$
 (6.1)

onde  $\mu_s[i] = \mathbf{K}_s \mathbf{s}_s^R[i] + \mathbf{b}_s$ , conforme Eq. (3.4),  $\Sigma_s$  é a covariância do sensor, suposta constante como justificado na seção 3.2, e  $\Delta_s[i]$  é a quantidade de amostras de leituras do sensor s coletadas durante o intervalo i. A figura 6.1 mostra um exemplo dos conjuntos de amostragem para N=4. Como pode ser notado, em cada intervalo as leituras são constantes, exceto por um ruído.

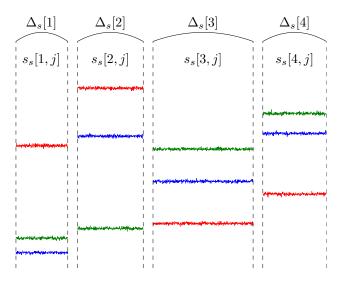


Figura 6.1: Exemplo de amostragem com N=4.

Com estes dados, é possível calcular a estimativa de menor variância de  $\mu_s[i]$ , dada por:

$$\hat{\mu}_s[i] = \frac{1}{\Delta_s[i]} \sum_{j=1}^{\Delta_s[i]} \mathbf{s}_s[i,j].$$
 (6.2)

Esta estimativa tem exatamente o papel de substituir as  $\Delta_s[i]$  leituras do intervalo sem perda nenhuma de informação. Isto pode ser observado pelo fato de que  $\mu_s[i]$  tem exatamente a mesma probabilidade, utilizando a Eq. (3.6), de todas as j amostras ocorrerem s[i,j], uma vez que

$$\hat{\mu}_s[i] \sim \mathcal{N}\left(\mu_s, \frac{\Sigma_s}{\Delta_s[i]}\right).$$
 (6.3)

Apesar de todos os algoritmos descritos neste capítulo dependerem da matriz de covariância do sensor, ela não está diretamente acessível. Portanto, seu estimador não enviesado de mínima variância, dado por:

$$\hat{\Sigma}_{s} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{\Delta_{s}[i]} (\mathbf{s}_{s}[i,j] - \hat{\mu}_{s}[i]) (\mathbf{s}_{s}[i,j] - \hat{\mu}_{s}[i])^{T}}{\left(\sum_{i=1}^{N} \Delta_{s}[i]\right) - N},$$
(6.4)

será utilizado e não será alterado pelo restante do algoritmo durante a otimização.

#### 6.1.1.2 Função de custo

Durante a calibração, o nosso objetivo é maximizar a verossimilhança dos parâmetros  $\theta_a$  e  $\theta_m$  utilizando a probabilidade descrita na Eq. (3.6), que depende dos parâmetros desconhecidos  $\mathbf{s}_a^R[i]$  e  $\mathbf{s}_m^R[i]$ . No entanto, para cada intervalo i do sistema parado, pode-se utilizar uma rotação  $\mathbf{R}_i$  e obter

$$\mathbf{s}_a^R[i] = \mathbf{R}_i \mathbf{g}, \qquad \mathbf{s}_m^R[i] = \mathbf{R}_i \mathbf{h}, \tag{6.5}$$

onde g e h são definidos pelas Eqs. (3.11a) e (3.11b), respectivamente. Note que, como não se tem conhecimento das rotações  $\mathbf{R}_i$  e elas devem ser estimadas, não importa se  $\mathbf{R}_i$  ou  $\mathbf{R}_i^{-1}$  será utilizado, contanto que haja consistência.

Assim, reescrevendo os parâmetros  $\theta_a$  e  $\theta_m$  como:

$$\theta_a = \{ \Sigma_a, \mathbf{K}_a, \mathbf{b}_a, \mathbf{g} \} \qquad \theta_m = \{ \Sigma_m, \mathbf{K}_m, \mathbf{b}_m, \mathbf{h} \}$$
 (6.6)

e adicionando um conjunto  $\{\mathbf{R}_i\}$  aos parâmetros a serem estimados, pode-se definir uma função de custo como o negativo do logaritmo da função de verossimilhança. Esta função de custo, excetuando-se as constantes, é dada por:

$$J(\theta_a, \theta_m, \{\mathbf{R}_i\}) = \sum_{i=1}^{N} \sum_{s \in \{a, m\}} \Delta_s[i] \left( \mathbf{K}_s \mathbf{R}_i \mathbf{s}_s^{\mathcal{I}} + \mathbf{b}_s - \hat{\mu}_s[i] \right)^T \Sigma_s^{-1} \left( \mathbf{K}_s \mathbf{R}_i \mathbf{s}_s^{\mathcal{I}} + \mathbf{b}_s - \hat{\mu}_s[i] \right), \quad (6.7)$$

onde  $\mathbf{s}_a^{\mathcal{I}} = \mathbf{g} \ \mathbf{e} \ \mathbf{s}_m^{\mathcal{I}} = \mathbf{h}$ . O objetivo da calibração é, portanto, determinar os parâmetros  $\theta_a$  e  $\theta_m$  que minimizam este custo.

#### 6.1.1.3 Estimação da rotação

Suponha que os parâmetros  $\theta_a$  e  $\theta_m$  sejam conhecidos e deseja-se estimar as rotações  $\{\mathbf{R}_i\}$ . Como cada rotação é independente das demais, pode-se definir um conjunto de funções de custo  $J_i$  que devem ser otimizadas independentemente. Estas funções são dadas por:

$$J_i(\mathbf{R}_i) = \sum_{s \in \{a,m\}} \Delta_s[i] \left( \mathbf{K}_s \mathbf{R}_i \mathbf{s}_s^{\mathcal{I}} + \mathbf{b}_s - \hat{\mu}_s[j] \right)^T \Sigma_s^{-1} \left( \mathbf{K}_s \mathbf{R}_i \mathbf{s}_s^{\mathcal{I}} + \mathbf{b}_s - \hat{\mu}_s[j] \right)$$
(6.8a)

$$= \sum_{s \in \{a,m\}} \Delta_s[i] \left\| \mathbf{K}_s \mathbf{R}_i \mathbf{s}_s^{\mathcal{I}} + \mathbf{b}_s - \hat{\mu}_s[j] \right\|_{\Sigma_s^{-1}}^2.$$

$$(6.8b)$$

As funções  $J_i(R_i)$  são resolvidas por otimização não linear utilizando o quaternion  $q_i$  para representar a rotação. No entanto, esta otimização é cara e não convexa, de forma que seria benéfico obter uma estimativa da rotação que fosse computacionalmente mais barata e não sofresse problemas de mínimos locais.

Para isto, considere a função de custo aproximada

$$\tilde{J}_i(\mathbf{R}_i) = \sum_{s \in \{a, m\}} \Delta_s[i] f\left(\mathbf{K}_s^T \Sigma_s^{-1} \mathbf{K}_s\right) \left\| \mathbf{R}_i \mathbf{s}_s^R + \mathbf{K}_s^{-1} \left( \mathbf{b}_s - \hat{\mu}_s[j] \right) \right\|^2, \tag{6.9}$$

onde  $f(\cdot)$  é uma função que extrai alguma característica útil da matriz de informação  $\mathbf{K}_s^T \Sigma_s^{-1} \mathbf{K}_s$ .

A função de custo aproximada é uma soma ponderada de quadrados de norma 2, cuja solução possui forma fechada utilizando quaternions (KARNEY, 2005). Para computar esta solução, a função deve ser reescrita como:

$$\tilde{J}_i(\mathbf{q}_i) = \mathbf{q}_i^T \mathbf{B}_i \mathbf{q}_i \tag{6.10a}$$

$$\mathbf{B}_{i} = \sum_{s \in \{a, m\}} \Delta_{s}[i] f\left(\mathbf{K}_{s}^{T} \Sigma_{s}^{-1} \mathbf{K}_{s}\right) \mathbf{A}_{si}^{T} \mathbf{A}_{si}$$

$$(6.10b)$$

$$A_{si} = f' \left( \mathbf{K}_s^{-1} (\hat{\mu}_s[i] - \mathbf{b}_s) + \mathbf{s}_s^{\mathcal{I}}, \ \mathbf{K}_s^{-1} (\hat{\mu}_s[i] - \mathbf{b}_s) - \mathbf{s}_s^{\mathcal{I}} \right)$$
(6.10c)

$$f'(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} 0 & -\mathbf{b}_x & -\mathbf{b}_y & -\mathbf{b}_z \\ \mathbf{b}_x & 0 & -\mathbf{a}_z & \mathbf{a}_y \\ \mathbf{b}_y & \mathbf{a}_z & 0 & -\mathbf{a}_x \\ \mathbf{b}_z & -\mathbf{a}_y & \mathbf{a}_x & 0 \end{bmatrix}.$$
 (6.10d)

Os autovalores da matriz  $\mathbf{B}_i$  são dados por  $0 \le \lambda_{i,0} \le \lambda_{i,1} \le \lambda_{i,2} \le \lambda_{i,3}$ . Tomando  $\mathbf{q}_i$  como o autovetor associado ao autovalor  $\lambda_{i,0}$ , tem-se que a função  $\tilde{J}_i(\mathbf{q}_i)$  é minimizada. Se a característica extraída pela função  $f(\cdot)$  for uma boa substituta da matriz de informação  $\Sigma_i^{-1}$ , espera-se que o quaternion  $q_i$  obtido seja próximo do quaternion  $\mathbf{q}_i^*$  que minimiza a função  $J_i(\mathbf{R}_i)$ .

Note que as computações utilizadas para chegar ao quaternion  $\mathbf{q}_i$  são fáceis de serem calculadas e produzem uma resposta fechada, não sendo sujeita a mínimos locais. Assim, esta solução  $\mathbf{q}_i$  pode ser utilizada como ponto inicial para otimização da função  $J_i(\mathbf{R}_i)$  para obter o quaternion correto.

#### 6.1.1.4 Estimação dos *bias* e campos atuantes

Suponha agora que as rotações  $\{\mathbf{R}_i\}$  e as matrizes de ganho  $\mathbf{K}_a$  e  $\mathbf{K}_m$  sejam conhecidas e deseja-se estimar os parâmetros  $\mathbf{b}_a$ ,  $\mathbf{b}_m$ , g e h.

Para isso, o método de mínimos quadrados generalizado será utilizado, sendo dado pela equação:

$$\mathbf{y} = X\beta + \epsilon, \quad \mathbb{E}[\epsilon|X] = 0, \quad \text{Var}[\epsilon|X] = \Omega,$$
 (6.11)

onde y é um vetor de observações, X é uma matriz de parâmetros observados,  $\beta$  são os valores que devem ser estimados,  $\epsilon$  são ruídos e  $\Omega$  é a covariância entre os ruídos.

A solução deste problema é dado pela solução do problema de mínimos quadrados ponderados, considerando como peso a matriz de informação  $\Omega^{-1}$ . Assim, a estimativa de  $\beta$  é dada por:

$$\hat{\beta} = \left(\mathbf{X}^T \Omega^{-1} \mathbf{X}\right)^{-1} \mathbf{X}^T \Omega^{-1} \mathbf{y} \tag{6.12}$$

Para calibração do acelerômetro, considere as seguintes matrizes:

$$\mathbf{y}_{a} = \begin{bmatrix} \hat{\mu}_{a}[1] \\ \hat{\mu}_{a}[2] \\ \vdots \\ \hat{\mu}_{a}[N] \end{bmatrix}, \mathbf{X}_{a} = \begin{bmatrix} \mathbf{K}_{a}\mathbf{R}_{1}\vec{\mathbf{z}} & \mathbf{I}_{3} \\ \mathbf{K}_{a}\mathbf{R}_{2}\vec{\mathbf{z}} & \mathbf{I}_{3} \\ \vdots & \vdots \\ \mathbf{K}_{a}\mathbf{R}_{N}\vec{\mathbf{z}} & \mathbf{I}_{3} \end{bmatrix}, \Omega_{a} = \begin{bmatrix} \frac{\Sigma_{a}}{\Delta_{a}[1]} & 0 & \cdots & 0 \\ 0 & \frac{\Sigma_{a}}{\Delta_{a}[2]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\Sigma_{a}}{\Delta_{a}[N]} \end{bmatrix}, \beta_{a} = \begin{bmatrix} \mathbf{g}_{z} \\ \mathbf{b}_{a} \end{bmatrix},$$

$$(6.13)$$

onde  $\vec{z}$  é o vetor unitário na direção z.

De maneira semelhante, pode-se definir para o magnetômetro as matrizes

$$\mathbf{X}_{m} = \begin{bmatrix} \mathbf{K}_{m} \mathbf{R}_{1} \vec{\mathbf{x}} & \mathbf{K}_{m} \mathbf{R}_{1} \vec{\mathbf{z}} & \mathbf{I}_{3} \\ \mathbf{K}_{m} \mathbf{R}_{2} \vec{\mathbf{x}} & \mathbf{K}_{m} \mathbf{R}_{2} \vec{\mathbf{z}} & \mathbf{I}_{3} \\ \vdots & \vdots & \vdots \\ \mathbf{K}_{m} \mathbf{R}_{N} \vec{\mathbf{x}} & \mathbf{K}_{m} \mathbf{R}_{N} \vec{\mathbf{z}} & \mathbf{I}_{3} \end{bmatrix}, \ \beta_{m} = \begin{bmatrix} \mathbf{h}_{x} \\ \mathbf{h}_{z} \\ \mathbf{b}_{m} \end{bmatrix},$$
(6.14)

onde  $\vec{\mathbf{x}}$  é o vetor unitário na direção  $\mathbf{x}$ . O vetor  $\mathbf{y}_m$  e a matriz  $\Omega_m$  são montadas da mesma maneira que para o acelerômetro.

Assim, utilizando as matrizes aqui definidas para o acelerômetro e magnetômetro, determinar-se os bias  $b_a$  e  $b_m$  e os campos g e h utilizando a Eq. (6.12).

### 6.1.1.5 Problemas ao estimar as matrizes de ganho

Quando não se sabe os valores corretos de g, h e das rotações  $\{\mathbf{R}_i\}$ , o que ocorre quando eles também são estimados pelo algoritmo, encontra-se dois problemas na hora de determinar as matrizes de ganho  $\mathbf{K}_a$  e  $\mathbf{K}_m$ .

O primeiro decorre do fato de existir uma infinidade de valores tais que  $\mathbf{K}_a\mathbf{R}_i\mathbf{g}$  é constante para todo  $\mathbf{R}_i$ . Esta infinidade de soluções pode afetar a otimização dos parâmetros, seja evitando a convergência ou tornando-a lenta. Para resolver este problema, estabeleceu-se que, durante a calibração de  $\mathbf{g}_z$ , seu valor será alterado para que permaneça sempre -1. Para isto, suponha que o algoritmo se encontra na iteração k e estimativa  $\mathbf{K}_a[k]$ . Após resolver a Eq. (6.12) para o acelerômetro, tem-se uma estimativa  $\hat{\mathbf{g}}$  de  $\mathbf{g}$ . Ao fazer  $\mathbf{K}_a[k+1] = |\hat{\mathbf{g}}|\mathbf{K}_a[k]$ , a escala determinada por  $\hat{\mathbf{g}}$  é mantida. Apesar de teoricamente o correto seria calcular  $\mathbf{K}_a[k+1] = -\hat{\mathbf{g}}\mathbf{K}_a[k]$ , o uso do valor absoluto se mostrou mais robusto durante os testes. De maneira semelhante, fixa-se  $\mathbf{h}_x = 1$  e faz-se  $\mathbf{K}_m[k+1] = |\hat{\mathbf{h}}_x|\mathbf{K}_m[k]$  e  $\mathbf{h}_z[k+1] = \frac{\hat{\mathbf{h}}_z}{\hat{\mathbf{h}}_x}$ , novamente mantendo a escala e a direção de  $\hat{\mathbf{h}}_x$ .

O segundo problema decorre de que é possível, segundo o teorema de decomposição polar (ALONSO E SHUSTER, 2002a), escrever  $\mathbf{K}_a$  como  $\mathbf{K}_a = \mathbf{PU}$ , onde  $\mathbf{P}$  é uma matriz semipositiva simétrica e  $\mathbf{U}$  é uma matriz unitária. Como uma rotação  $\mathbf{R}_i$  qualquer também é uma matriz unitária, tem-se que o termo  $U\mathbf{R}_i$  em  $\mathbf{K}_a\mathbf{R}_i = \mathbf{PUR}_i$  não permite distinguir entre a matriz de rotação  $\mathbf{R}_i$  e a matriz  $\mathbf{U}$  que torna  $\mathbf{K}_a$  não simétrica. Novamente, há infinitas soluções, o que gera problemas para convergência. Neste caso, pode-se definir  $\mathbf{U}$  como  $\mathbf{U} = \mathrm{diag}(\pm 1, \pm 1, \pm 1)$ , fazendo com que  $\mathbf{K}_a$  seja uma matriz simétrica. É possível ser mais restrito na hora de definir  $\mathbf{U}$ , mas isto implicaria numa otimização com restrições, além deste formato ser suficiente para o algoritmo. Caso o usuário deseje que  $\mathbf{U}$  seja a identidade, basta fazer a decomposição polar de  $\mathbf{K}_a$  após a calibração e pósmultiplicar  $\mathbf{K}_a$  e  $\mathbf{K}_m$  pela matriz  $\mathbf{U}$  resultante.

Um ponto importante de ser notado é que isto define o sistema de coordenadas S utilizado: é o sistema onde a matriz  $K_a$  é simétrica.

#### 6.1.1.6 Estimação das matrizes de ganho

Suponha finalmente que os parâmetros  $\mathbf{b}_a$ ,  $\mathbf{b}_m$ ,  $\mathbf{g}$  e  $\mathbf{h}$  e as rotações  $\{\mathbf{R}_i\}$  são conhecidos e deseja-se estimar as matrizes  $\mathbf{K}_a$  e  $\mathbf{K}_m$ . Como discutido na seção anterior, tem-se que  $\mathbf{K}_a$  deve ser simétrica.

Definindo

$$\mathbf{g}_i^{\mathcal{S}} = \mathbf{R}_i \mathbf{g}, \quad \mathbf{h}_i^{\mathcal{S}} = \mathbf{R}_i \mathbf{h},$$
 (6.15)

onde  $i \in \mathbb{N}_N$  e g e h são os parâmetros conhecidos para a gravidade e o campo magnético, o problema pode ser expressado no formato esperado pelo método de mínimos quadrados generalizado, descrito na seção 6.1.1.4.

Como  $K_a$  deve ser simétrica, tem-se que

$$\mathbf{y}_{a} = \begin{bmatrix} \hat{\mu}_{a}[1] - \mathbf{b}_{a} \\ \hat{\mu}_{a}[2] - \mathbf{b}_{a} \\ \vdots \\ \hat{\mu}_{a}[N] - \mathbf{b}_{a} \end{bmatrix}, \mathbf{X}_{a} = \begin{bmatrix} \mathbf{g}_{1,1}^{\mathcal{S}} & \mathbf{g}_{1,2}^{\mathcal{S}} & \mathbf{g}_{1,3}^{\mathcal{S}} & 0 & 0 & 0 \\ 0 & \mathbf{g}_{1,1}^{\mathcal{S}} & 0 & \mathbf{g}_{1,2}^{\mathcal{S}} & \mathbf{g}_{1,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{1,1}^{\mathcal{S}} & 0 & \mathbf{g}_{1,2}^{\mathcal{S}} & \mathbf{g}_{1,3}^{\mathcal{S}} \\ \mathbf{g}_{2,1}^{\mathcal{S}} & \mathbf{g}_{2,2}^{\mathcal{S}} & \mathbf{g}_{2,3}^{\mathcal{S}} & 0 & 0 & 0 \\ 0 & \mathbf{g}_{2,1}^{\mathcal{S}} & 0 & \mathbf{g}_{2,2}^{\mathcal{S}} & \mathbf{g}_{2,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{2,1}^{\mathcal{S}} & 0 & \mathbf{g}_{2,2}^{\mathcal{S}} & \mathbf{g}_{2,3}^{\mathcal{S}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_{N,1}^{\mathcal{S}} & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 & 0 & 0 \\ 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,1}^{\mathcal{S}} & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & 0 \\ 0 & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} \\ 0 & 0 & 0 & \mathbf{g}_{N,2}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{S}} & \mathbf{g}_{N,3}^{\mathcal{$$

e  $\Omega_a$  é o mesmo definido na Eq. (6.13).

Para o magnetômetro, cujo ganho  $K_m$  é arbitrário, as matrizes são:

$$\mathbf{y}_{m} = \begin{bmatrix} \hat{\mu}_{m}[1] - \mathbf{b}_{m} \\ \hat{\mu}_{m}[2] - \mathbf{b}_{m} \\ \vdots \\ \hat{\mu}_{m}[N] - \mathbf{b}_{m} \end{bmatrix}, \mathbf{X}_{m} = \begin{bmatrix} \mathbf{h}_{1}^{\mathcal{S}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{h}_{1}^{\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}_{1}^{\mathcal{S}} \\ \mathbf{h}_{2}^{\mathcal{S}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{h}_{2}^{\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}_{2}^{\mathcal{S}} \\ \vdots & \vdots & \vdots \\ \mathbf{h}_{N}^{\mathcal{S}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{h}_{N}^{\mathcal{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}_{N}^{\mathcal{S}} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}_{N}^{\mathcal{S}} \end{bmatrix}, \beta_{m} = \begin{bmatrix} \mathbf{K}_{m,11} \\ \mathbf{K}_{m,12} \\ \mathbf{K}_{m,21} \\ \mathbf{K}_{m,22} \\ \mathbf{K}_{m,23} \\ \mathbf{K}_{m,31} \\ \mathbf{K}_{m,31} \\ \mathbf{K}_{m,32} \\ \mathbf{K}_{m,33} \end{bmatrix}$$

$$(6.17)$$

e  $\Omega_m$  é definido de maneira semelhante à Eq. (6.13).

Com estas matrizes definidas, pode-se calcular as estimativas ótimas  $\hat{\beta}_a$  e  $\hat{\beta}_m$  utilizando a Eq. (6.12), determinando os valores das matrizes de ganho.

## 6.1.2 Algoritmos de calibração

Com base na teoria descrita na seção anterior, pode-se desenvolver dois algoritmos para calibração do magnetômetro e acelerômetro, que seguem uma lógica simples, dada pelos seguintes passos:

- 1. Preprocesse os dados, utilizando as Eq. (6.2) e (6.4).
- 2. Gere uma primeira estimativa dos parâmetros e calcule o custo J[0] associado utilizando a Eq. (6.7). A estimação inicial é dada por *bias* nulos e matrizes de ganho e rotações como identidade.
- 3. Otimize a rotação como descrito na seção 6.1.1.3.
- 4. Otimize os bias e as estimativas dos campos conforme as seções 6.1.1.4 e 6.1.1.5.
- 5. Se houve pouca alteração na estimativa dos campos, otimize os ganhos. Este condicional foi criado porque foi percebido durante os testes que otimizar os ganhos a cada iteração não

era vantajoso. Por isso, a otimização dos ganhos só é feita quando há uma alteração de  $\Delta \mathbf{h}_z < 10^{-3}$  durante o passo anterior.

- 6. Compute o custo J[k], onde k é a iteração corrente, utilizando a Eq. (6.7).
- 7. Caso  $J[k-1]-J[k]<\gamma$ , pare. Caso contrário, vá para 3.

Vale a pena ressaltar que todos os passos do algoritmo são gulosos, ou seja, calculam a melhor solução para aquele instante. Esta abordagem nem sempre é vantajosa, uma vez que pode apresentar divergências. Como há duas soluções para o problema de estimação de rotação, comparar-se-á dois algoritmos baseados nelas:

- 1. O algoritmo A1 utiliza apenas a otimização direta da Eq. (6.8b) para determinação da rotação.
- 2. O algoritmo A2 utiliza a solução da Eq. (6.9) para gerar as estimativas  $\mathbf{q}_i[k]$ . Junto com o passo 7, compara-se se J[k] > J[k-1], o que significa que houve um aumento da função de custo devido à aproximação. Se isto ocorrer, a otimização direta passa a ser utilizada nas iterações futuras, passando a se comportar como A1. É importante ressaltar que as estimativas calculadas com método aproximado são mantidas, mesmo quando elas aumentam o custo. Notou-se nos testes que ignorar a iteração k e reverter para as estimativas da iteração k-1 era mais prejudicial do que manter as estimativas atuais.

Esta separação em dois métodos visa avaliar a diferença entre um algoritmo que realiza um passo ótimo a cada iteração, sempre calculando o mínimo, de um algoritmo minimizante, que utiliza um modelo aproximado para dar um passo que minimize a função, sem necessariamente chegar ao ótimo.

## 6.1.3 Metodologia

Para realizar as simulações, assume-se que um total N de 15 períodos foram coletados, cada um com um total de  $\Delta[i] \sim \mathcal{U}(\{400,401,\ldots,600\})$  amostras coletadas, sendo iguais a quantidade de amostras para os sensores. A rotação de cada período foi gerada aleatoriamente conforme descrito em (KUFFNER, 2004) e a condição de parada dos algoritmos  $\gamma$  foi fixada em  $10^{-4}$ .

Os campos foram gerados a partir de números aleatórios nos intervalos [-1, 5; -0, 5], [0, 5; 1, 5] e [-1, 5; 1, 5] para  $\mathbf{g}_z$ ,  $\mathbf{h}_x$  e  $\mathbf{h}_z$  respectivamente. É importante ressaltar que estes parâmetros não terão seus valores recuperados, como explicado na seção 6.1.1.5.

O acelerômetro e magnetômetro tem geração de seus parâmetros iguais. Cada termo  $\mathbf{b}_{s,i}$  do bias possui um valor aleatório em [-1,1], a matriz de ganho é dada por  $\mathbf{K}_s = \mathbf{I}_3 + \delta \mathbf{K}_s$  onde  $\delta \mathbf{K}_{s,ij} \sim \mathcal{U}([-0,1;0,1])$  e as covariâncias são dadas por:

$$\Sigma_{s} = \alpha_{s} \begin{cases} \Sigma_{s,ij} \sim \mathcal{U}([0,5;2]), & \text{se } i = j \\ \Sigma_{s,ij} \sim \mathcal{U}([-0,2;0,2]), & \text{caso contrário,} \end{cases}$$
(6.18)

onde  $\alpha_s \sim \mathcal{U}([10^{-2}, 10^{-4}])$ .

Foram realizadas um total de 100 simulações pelo método de Monte Carlo, gerando novos valores a cada execução do algoritmo. A tabela 6.1 traz um resumo dos parâmetros utilizados na simulação.

TO 1 1 (1 D A	. 1 ~	1 1'1 ~	. 1 1 1 .	. ^ .
<b>Tabela 6.1:</b> Parâmetros	nara simiilacao	da cali <b>hr</b> aca	io do acelerometro	e magnetometro
Tubela 0.1. I alamenos	para siiiiaiação	da camoraça	io do deciciónicho	e magnetometro.

Parâmetro	Descrição	Valor	
N	Total de períodos	15	
$\Delta[i]$	Quantidade de amostras do período i	$\mathcal{U}(\{400, 401, \dots, 600\})$	
$\mathbf{R}_i$	Rotação referente ao período i	Ver (Kuffner, 2004)	
$\gamma$	Condição de parada	$10^{-4}$	
$\mathbf{g}_z$	Componente $z$ da gravidade	$\mathcal{U}([-1,5;-0,5])$	
$\mathbf{h}_x$	Componente $x$ do campo magnético	$\mathcal{U}([0,5;1,5])$	
$\mathbf{h}_z$	Componente $z$ do campo magnético	$\mathcal{U}([-1,5;1,5])$	
$\mathbf{b}_{s,i}$	Componente do $bias$ do sensor $s$	$\mathcal{U}([-1,1])$	
$\mathbf{K}_{s}$	Matriz de ganhos do sensor $s$	$\mathbf{I}_3 + \delta \mathbf{K}_s, \ \delta \mathbf{K}_{s,ij} \sim \mathcal{U}([-0,1;0,1])$	
$\Sigma_s$	Matriz de covariância do sensor $s$	Ver Eq. (6.18)	
	Número total de simulações	100	

Após a calibração, os parâmetros e rotações foram utilizados para reconstruir as estimativas de  $\mu_s[i]$ , gerando valores  $\tilde{\mu}_s[i]$ . Para comparar a qualidade das soluções geradas, utilizou-se a seguinte equação:

$$\delta_s = \sqrt{\frac{\sum_{i=1}^{N} \Delta[i] (\mu_s[i] - \tilde{\mu}_s[i])^T \Sigma^{-1} (\mu_s[i] - \tilde{\mu}_s[i])}{\sum_{i=1}^{N} \Delta[i]}},$$
(6.19)

que é derivada da distância de Mahalanobis (MAHALANOBIS, 1936).

Além disso, deve-se considerar o caso em que a calibração funciona bem para os dados coletados, pois está otimizando a função de custo em cima deles, mas não é capaz de aproximar adequadamente outras regiões. Este problema é conhecido como *overfitting* e deve ser analisado. Por isso, para cada simulação utilizada para a calibração, realizou-se uma segunda simulação onde novas rotações são geradas aleatoriamente, mas os demais parâmetros são os mesmos. Após a calibração, a capacidade de reconstrução  $\delta$  foi avaliada tanto para os dados de treinamento, quanto para os dados de teste. Para estimar a rotação utilizando os dados de teste, foi utilizada a solução da Eq. (6.9) como primeira aproximação seguida pela otimização da Eq. (6.8b).

### 6.1.4 Simulações

As figuras 6.2 e 6.3 mostram o erro de reconstrução ao variar o parâmetro  $\gamma$ , mantendo o número de intervalos N=15 constante, enquanto as figuras 6.4 e 6.5 mostram o erro de reconstrução com um número variável de intervalos N, mas mantendo a condição de parada  $\gamma=10^{-4}$ . A figura 6.6 apresenta o tempo total necessário para a calibração. Em todas as figuras, o algoritmo A1 está representado pela caixa larga e vazada, enquanto A2 está pela caixa fina e sólida.

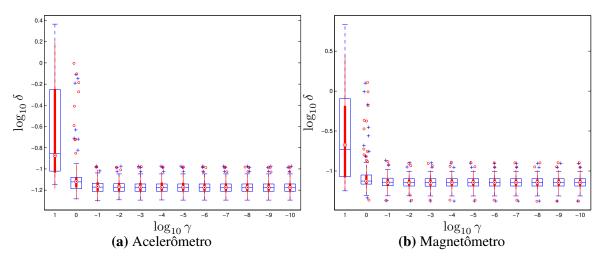


Figura 6.2: Erro de reconstrução com dados de treinamento e variando a condição de parada.

#### 6.1.5 Análise

Os resultados para condição de parada  $\gamma$  variante são dados pelas figuras 6.2 e 6.3. Nota-se claramente que, conforme a condição de parada diminui, o erro tanto de teste quanto de treinamento

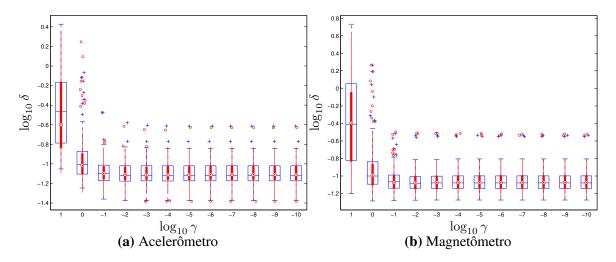


Figura 6.3: Erro de reconstrução com dados de teste e variando a condição de parada.

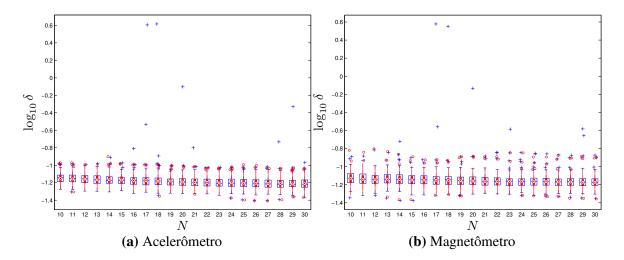


Figura 6.4: Erro de reconstrução com dados de treinamento e variando o número de intervalos.

também diminui. No entanto, um valor de  $\gamma=10^{-4}$  já é suficiente, uma vez que não há alterações significativas ao utilizar variações menores na condição de parada. Percebe-se também que os algoritmos A1 e A2 possuem desempenhos semelhantes na reconstrução dos dados de treinamento e teste, mostrando que ambos possuem capacidades semelhantes. Apesar de o erro de teste ser maior do que o erro de treinamento, como era esperado, em 75% dos casos de teste tem-se que  $\delta < 10^{-1}$ , indicando bom desempenho dos algoritmos para estimar os parâmetros dos sensores.

Ao observar o efeito da quantidade de intervalos, presente nas figuras 6.4 e 6.5, tem-se que o erro de reconstrução decresce conforme utiliza-se mais intervalos, o que também era esperado. Nota-se que a melhora do desempenho nos dados de treinamento é pequena, uma vez que a cali-

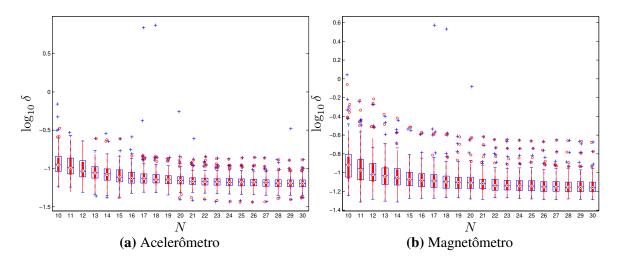
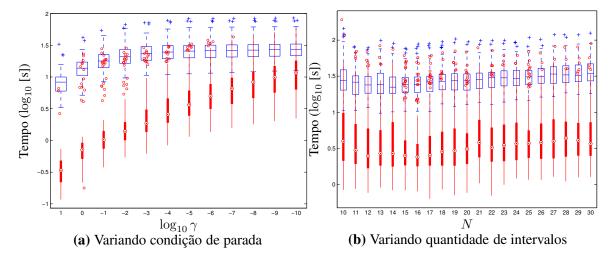


Figura 6.5: Erro de reconstrução com dados de teste e variando o número de intervalos.



**Figura 6.6:** Tempo de calibração variando a condição de parada e quantidade de intervalos.

bração com poucas amostras já é capaz de obter bons resultados, mas o erro de reconstrução dos dados de teste decresce consideravelmente conforme o número de amostras aumenta. Além disso, a variação no erro entre simulações também diminui, o que mostra que o algoritmo é mais estável com um número superior de amostras. Um fato importante de ser destacado é que A1 apresenta grandes erros tanto de treinamento quanto de teste em alguns casos. Este comportamento pode ser associado à não linearidade da otimização da rotação, uma vez que este é o único ponto que diferencia A1 de A2 e pode fazer com que o sistema convirja para mínimos locais de baixa qualidade. Com isso, tem-se que o uso da estimativa aproximada por A2 nas etapas iniciais, onde ela ainda reduz a função de custo, permite que o algoritmo encontre uma boa região do espaço de busca, permitindo que posteriormente a otimização correta atinja mínimos de melhor qualidade.

Observando o tempo requerido para calibração ao variar a condição de parada, apresentado na figura 6.6a, observa-se que o tempo de computação aumenta conforme a condição de parada diminui. Novamente este comportamento era esperado, uma vez que é necessário um número maior de iterações para atingir uma variação menor do que  $\gamma$ . Ao analisar o comportamento de A1, nota-se que sua curva permanece praticamente constante para  $\gamma \leq 10^{-4}$ . Como discutido anteriormente, o algoritmo não é capaz de extrair muito mais informações dos dados após este ponto, o que faz com que rapidamente ele atinja as condições de parada menores. Entretanto, o algoritmo A2 permanece aumentando o tempo de computação após este ponto para chegar numa iteração onde a variação da função custo seja menor do que  $\gamma$ , realizando pequenos mas custosos refinamentos nos parâmetros, mesmo que suas novas estimativas não apresentem melhoras perceptíveis no desempenho da calibração.

Já ao observar a figura 6.6b, percebe-se um resultado inesperado: o aumento do número de intervalos não altera significativamente o tempo total de computação, que apenas apresenta pequenas oscilações para A2 e um pequeno aumento para A1 com N>15. Este resultado é inesperado por dois motivos:

- 1. A equação de custo, dada pela Eq. (6.7), utilizada para avaliar a condição de parada  $\gamma$  é uma soma dos erros de cada intervalo, fazendo com que um maior número de intervalos também aumente o número de termos e, provavelmente, possua um valor mais elevado, além de apresentar maiores variações entre iterações do algoritmo, requerendo um maior número de iterações para alcançar  $\gamma$ .
- 2. Conforme o número de intervalos amostrados aumenta, todos os passos do algoritmo ficam mais lentos. A quantidade de rotações a serem estimadas é igual ao número de intervalos e as matrizes  $X_s$  e  $y_s$  crescem linearmente com N.

Analisando os resultados, conclui-se que, apesar de haver de fato um aumento do tempo para cada iteração do algoritmo, a quantidade de iterações necessárias para convergência é significativamente menor. Isto se deve ao fato de que a estimação dos parâmetros gera resultados mais próximos dos reais nas iterações, devido à quantidade maior de amostras, fazendo com que a calibração se torne mais rápida. Vale a pena ressaltar que, apesar de haver redução do tempo, esta redução não é significativa o suficiente para justificar por si só a coleta de mais dados.

É importante ressaltar que, no algoritmo A1, o passo computacionalmente mais caro é a

otimização da rotação. Por este motivo, não é necessário utilizar algoritmos que façam uso da esparsidade das matrizes  $\Omega_s$  por não melhorar o tempo de computação significativamente. Já em A2, esta otimização pode gerar tempos de computação menores. Entretanto, a figura 6.6b mostra que, para todos os valores de N, 75% dos casos possuem tempo de computação menor do que 10 segundos, o que é razoável e, portanto, a esparsidade das matrizes não foi explorada. Além disso, em todos os casos em que o tempo de A2 foi semelhante ao tempo de A1, o passo de ajuste da rotação passou para a otimização direta logo nas primeiras iterações, quando a aproximação gerou aumento do custo. Portanto, nestes casos, a otimização das operações matriciais também não gerariam melhoras significativas.

Em ambos os casos, tanto na variação da condição de parada quanto na quantidade de intervalos, o algoritmo A2 possui tempo de computação significativamente menor do que A1. Isto está associado à capacidade de uso da computação eficiente da rotação nas primeiras iterações do algoritmo, onde apenas uma melhora da estimação é suficiente, não sendo necessário atingir o mínimo naquele instante pois as demais estimativas ainda estão longe dos valores reais. Como ambos algoritmos possuem resultados semelhantes na maior parte dos casos, mas A1 é computacionalmente mais caro e apresenta problemas com mínimos locais de baixa qualidade, recomenda-se o uso de A2 e de um número não muito elevado de intervalos, uma vez que poucas amostras podem fazer com que as estimativas tenham pouca generalidade e muitas amostras não apresentam melhoras significativas de desempenho e demoram mais para serem coletadas.

## 6.2 Rotação do sistema de coordenada dos sensores

Nesta seção, será determinada a matriz de rotação  $\mathcal{R}$  que transforma vetores de  $\mathcal{B}$  para  $\mathcal{S}$ . Esta matriz é importante para a calibração pois, apesar das leituras serem feitas em  $\mathcal{S}$ , elas pouco interessam por depender da rotação em que os sensores são colocados. Apesar dos algoritmos de filtragem poderem trabalhar diretamente no sistema de coordenadas  $\mathcal{S}$ , os algoritmos de controle trabalham num sistema de coordenadas arbitrário e independente do posicionamento dos sensores.

Pode-se visualizar este problema ao utilizar um conjunto de sensores com um avião. Os filtros são capazes de trabalhar com rotações relativas a quaisquer sistemas de coordenadas, mas os algoritmos de controle são mais limitados. Considere um algoritmo de controle que foi projetado para que as coordenadas x e z estejam na direção da frente do avião e da gravidade, por exemplo. Se o sistema  $\mathcal S$  for utilizado, os sensores devem sempre estar alinhados com o sistema descrito

acima. No entanto, ao inserir a rotação  $\mathcal{R}$ , é possível utilizar os sensores em qualquer posição  $\mathcal{S}$  e especificar o controle no sistema  $\mathcal{B}$ , definido arbitrariamente.

Usualmente, uma variação da primeira abordagem é utilizada: cada coordenada de  $\mathcal{S}$  e associada a uma coordenada de  $\mathcal{B}$ , permitindo, por exemplo, que  $\mathcal{S}$  tenha sua componente z contra a gravidade. No entanto, esta metodologia ainda requer um alinhamento entre os sistemas de coordenadas. Para resolver este problema, o algoritmo aqui desenvolvido é capaz de determinar a rotação  $\mathcal{R}$  sem restrições, onerando ao mínimo o usuário do sistema.

A seção 6.2.1 apresenta a teoria do funcionamento do algoritmo e as otimizações que serão utilizadas nas simulações. A seção 6.2.2 descreve o método utilizado para avaliar o algoritmo desenvolvido, que gera os resultados presentes na seção 6.2.3. A análise dos resultados obtidos é realizada na seção 6.2.4.

#### 6.2.1 Desenvolvimento teórico

Para realizar a calibração, assume-se que, na condição normal de equilíbrio,  $\mathcal{B}$  difere de  $\mathcal{I}$  por uma rotação no eixo z apenas. Esta restrição não prejudique o uso do algoritmo, pois usualmente sabe-se como chegar ao sistema  $\mathcal{B}^*$  desejado a partir do sistema  $\mathcal{B}$  considerado pelo algoritmo.

A seção 6.2.1.1 descreve como os dados são coletados, apresentando as restrições e um exemplo de método de coleta que satisfaz estas restrições. Utilizando a descrição desta seção, pode-se definir na seção 6.2.1.2 a função de custo a ser minimizada e como isto é feito. A seção A.1 descreve como transformar produto entre quaternions em operações matriciais, onde estas matrizes são utilizadas na seção 6.2.1.3 para montar as matrizes necessárias pelo algoritmo.

#### 6.2.1.1 Coleta dos dados para calibração

Ao invés de descrever como os dados são coletados, é mais fácil descrever como eles são decompostos, fazendo com que a coleta deva seguir às restrições impostas. No fim da seção um resumo da metodologia na ordem lógica de coleta é apresentado.

Seja  $\{^i\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\}$  um conjunto de quaternions conhecidos. A notação  $\mathbf{q}_B^A$  representa um quaternion que leva de um sistema de coordenadas B para A. Estes quaternions podem ser decompostos da

seguinte maneira:

$${}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}} = \mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'},$$
 (6.20)

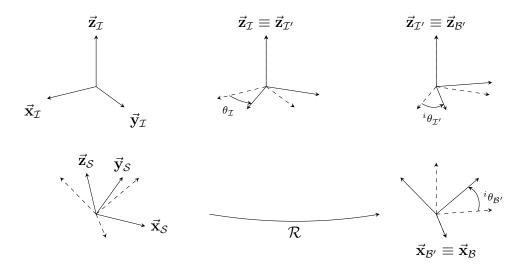
onde:

- $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$  é um quaternion formado pela rotação em torno do eixo z de um ângulo  $\theta_{\mathcal{I}}$  arbitrário. Esta rotação representa a construção de um referencial  $\mathcal{I}'$  arbitrário, uma vez que a direção do campo magnético, dada pela Eq. (3.11b), pode não ser conhecido. Portanto,  $\mathcal{I}'$  é um referencial paralelo ao plano normal à Terra.
- ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$  também é formado por uma rotação em torno do eixo z de um ângulo  $\theta_{\mathcal{I}'}^{i} \in (-\frac{pi}{2}, \frac{pi}{2})$ . Esta rotação permite que as medições possuam rotações entre si, uma vez que pode ser impraticável requerer que o sistema não possua rotações indesejadas ao movê-lo. Para adicionar esta liberdade, é necessário que o usuário informe se  $\theta_{\mathcal{I}'}^{i}$  é negativo, positivo ou nulo.
- ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$  é formado por uma rotação em torno do eixo x ou y do sistema de coordenadas  $\mathcal{B}'$  local de um ângulo  $\theta_{\mathcal{B}'}^{i} \in \left(-\frac{pi}{2}, \frac{pi}{2}\right)$ . Esta é a etapa que de fato define os eixos do sistema de coordenadas  $\mathcal{B}$ . Novamente, o usuário deve informar se o ângulo é negativo, positivo ou nulo, devendo informar ainda se a rotação foi realizada em torno do eixo x ou y local.
- $\mathbf{q}_\mathcal{B}^\mathcal{S}$  é a matriz de rotação  $\mathcal{R}$  arbitrária, sendo portanto independente da rotação atual do sistema.

A figura 6.7 mostra um exemplo de relação entre os sistemas de coordenadas e as rotações definidas, facilitando o entendimento.

Com base nestes quaternions definidos anteriormente, a seguinte metodologia para coleta dos dados é sugerida:

- 1. Posicione o robô numa superfície paralela à superfície terrestre. Determine a rotação atual e faça marcas no chão, indicando a posição do sistema.
- 2. Utilize algum apoio para rotacionar o sistema no eixo x ou y local. Ao posicionar o robô e seu apoio no chão, veja se ele está rotacionado para direita ou esquerda da posição original.
- 3. Repita o procedimento anterior algumas vezes, variando o ângulo e o eixo de rotação. É recomendável aplicar ao menos uma rotação positiva e uma negativa em cada eixo.



**Figura 6.7:** Exemplo de relação entre as rotações e os sistemas de coordenadas.

4. Opcionalmente, gere rotações apenas no eixo z, fazendo com que os quaternions  $\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$  sejam nulos.

Esta metodologia é extremamente simples e pode ser utilizada numa grande variedade de sistemas. Além disso, o passo 1 captura uma posição onde as rotações  ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$  e  ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$  são nulas, o que ajuda na convergência do algoritmo.

#### 6.2.1.2 Função de custo e sua otimização

Para determinar cada um dos quaternions, métodos de otimização de uma função de custo são utilizados. No entanto, o mapeamento duplo deve ser considerado durante esta otimização, evitando que ela convirja para valores incorretos. Para isso, seja q um quaternion qualquer, seu primeiro termo não nulo, começando a partir do escalar, deve ser positivo. Ao invés de aplicar esta restrição à otimização, a solução será determinada sem restrições e, caso o primeiro termo seja negativo, a solução inteira será negada, simplificando os algoritmos utilizados.

Considerando que um total de N amostras foram capturadas, define-se o custo

$$J\left(\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}, \{^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}\}, \{^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}\}, \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}\right) = \sum_{i=1}^{N} \left\|\mathbf{q}_{0} - \mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left({}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\right)^{-1}\right\|^{2}, \tag{6.21}$$

onde  $\mathbf{q}_0$  é o quaternion que não possui rotações e  ${}^i\mathbf{q}_\mathcal{I}^\mathcal{S}$  são as estimativas obtidas a partir de leitura

dos sensores, como descrito anteriormente.

Ao realizar a otimização de apenas um quaternion de cada vez, é possível utilizar o método de mínimos quadrados para cálculo da solução. Sejam  $\mathbf{X} \in \mathbb{R}^{n \times m}$  e  $\mathbf{y} \in \mathbb{R}^m$  conhecidos, o valor  $\theta \in \mathbb{R}^n$  que minimiza

$$J(\theta) = \|\mathbf{X}\theta - \mathbf{y}\| \tag{6.22}$$

é dado por:

$$\theta^* = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}. \tag{6.23}$$

Para isso, coloca-se X e y no formato

$$\mathbf{X} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_N \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_0 \end{bmatrix}, \tag{6.24}$$

onde as matrizes  $M_i$  dependem do quaternion estimado naquele instante. Além disso, para estimação de cada quaternion, supõe-se que os demais são conhecidos.

Dada a estimativa  $\mathbf{q}^*$  a partir dos quadrados mínimos, a estimativa passa a ser  $\hat{\mathbf{q}} = -\mathbf{q}^*$  caso o primeiro termo não nulo de  $\mathbf{q}^*$  seja negativo, uma vez que na realidade ambas soluções possuem o mesmo custo e deve-se manter a consistência entre os quaternions, como discutido anteriormente. No entanto, ainda deve-se utilizar a informação sobre o sinal da rotação, o que gera duas possibilidades: caso o sinal do termo correspondente do quaternion seja contrário ao sinal esperado, pode-se negar apenas aquele termo, invertendo a direção de rotação, ou zerá-lo. Ambas abordagens foram testadas e zerar o termo quando ele não concorda com o sinal esperado gerou problemas de convergência em todas as situações, fazendo com que apenas a troca de sinal fosse utilizada nos resultados obtidos.

#### 6.2.1.3 Estimação dos quaternions

Os valores  $M_i$  da Eq. (6.24) podem ser obtidos diretamente a partir da função de custo, presente na Eq. (6.21), utilizando a notação de multiplicação matricial, descrita na seção anterior.

Para  $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$  tem-se que as matrizes  $\mathbf{M}_i$  são dadas por:

$$\mathbf{M}_{i} = \left[ \otimes \left( {}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left( {}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}} \right)^{-1} \right) \right]$$
(6.25)

e para  $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$  elas são dadas por:

$$\mathbf{M}_{i} = \left[ \left( \mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^{i} \mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \otimes {}^{i} \mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \right) \otimes \right] \left[ \otimes \left( {}^{i} \mathbf{q}_{\mathcal{I}}^{\mathcal{S}} \right)^{-1} \right], \tag{6.26}$$

onde as notações de produtos de quaternions como matrizes são dadas pelas Eqs. (A.3) e (A.4).

Para estimar  ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$  e  ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ , deve-se levar em consideração que eles dependem apenas de suas próprias leituras, sendo portanto independentes para cada i. Para utilizar a mesma formulação anterior, faz-se  $\mathbf{M}_i = \mathbf{0}$  para  $i \in [2, 3, \dots, N]$ , utilizando somente  $\mathbf{M}_1$  na estimação. Além disso, se os passos descritos na seção 6.2.1.1 forem utilizados para coleta dos dados, então estes quaternions não serão ajustados para a amostra do passo 1, por eles já serem conhecidos. Portanto, para  ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$  tem-se:

$$\mathbf{M}_{1} = \left[\mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes\right] \left[ \otimes \left({}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'} \otimes \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left({}^{i}\mathbf{q}_{\mathcal{I}}^{\mathcal{S}}\right)^{-1}\right) \right]$$
(6.27)

e, para  ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ , tem-se

$$\mathbf{M}_{1} = \left[ \left( \mathbf{q}_{\mathcal{B}}^{\mathcal{S}} \otimes {}^{i} \mathbf{q}_{\mathcal{B}'}^{\mathcal{B}} \right) \otimes \right] \left[ \otimes \left( \mathbf{q}_{\mathcal{I}}^{\mathcal{I}'} \otimes \left( {}^{i} \mathbf{q}_{\mathcal{I}}^{\mathcal{S}} \right)^{-1} \right) \right]. \tag{6.28}$$

O algoritmo desenvolvido estima os quaternions na mesma ordem que são utilizados na Eq. (6.20).

## 6.2.2 Metodologia

Para cada simulação, gera-se o quaternion  $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$  aleatoriamente, conforme descrito em (KUFFNER, 2004). Para  $\mathbf{q}_{\mathcal{I}}^{\mathcal{I}'}$ , escolhe-se um ângulo  $\alpha$  uniformemente no intervalo  $[0,2\pi)$  e computa-se o quaternion resultante baseado na decomposição polar da Eq. (A.6), que pode ser escrita como:

$$\mathbf{q} = \cos\left(\frac{\alpha}{2}\right) + \mathbf{v}\sin\left(\frac{\alpha}{2}\right),\tag{6.29}$$

onde  ${\bf v}$  é o vetor unitário na direção de rotação, que neste caso é dado por  ${\bf z}$ . Os quaternions  ${}^i{\bf q}_{\mathcal{I}'}^{\mathcal{B}'}$  são obtidos por uma rotação  $\alpha$  aleatória no intervalo  $(-\frac{\pi}{2},\frac{\pi}{2})$  em torno do mesmo eixo  ${\bf z}$ .

Considera-se que foram feitas M medições em torno de cada eixo. Seus quaternions  ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$  são dados por rotações aleatórias no intervalo  $(-\frac{\pi}{2},\frac{\pi}{2})$  em torno de seus eixos respectivos. Supõe-se também a existência da rotação coletada no passo 1 descrito na seção 6.2.1.1, que possui rotações nulas nos quaternions  ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$  e  ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$ .

Cada medição de rotação realizada é afetada por um ruído uniforme. Para gerar cada ruído, gera-se um quaternion aleatório como descrito anteriormente e seu ângulo é multiplicado por um fator  $\gamma$ , que representa a incerteza, utilizando a decomposição polar.

Foram realizadas um total de 100 simulações pelo método de Monte Carlo, gerando novos valores a cada execução do algoritmo. A tabela 6.2 traz um resumo dos parâmetros utilizados na simulação.

Parâmetro	Descrição	Valor
M	Número de rotações para cada um dos eixos $x$ e $y$	$\{2, 5, 10\}$
	Ângulo de rotação para os quaternions ${}^{i}\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ e ${}^{i}\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$	$\mathcal{U}\left((-rac{\pi}{2},rac{\pi}{2}) ight)$
$\gamma$	Incerteza das medições	$\{50, 10, 5, 1, 0\} \times 10^{-3}$
	Número total de simulações	100

**Tabela 6.2:** Parâmetros para simulação da calibração da rotação.

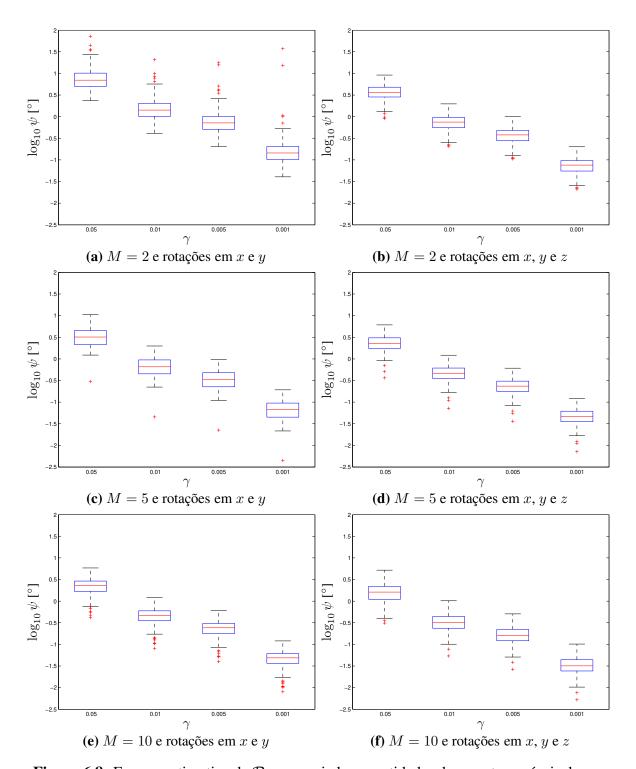
Uma vez determinadas as estimativas de cada quaternion, o valor de  $\mathcal{R}$  estimado  $\hat{\mathbf{q}}_{\mathcal{B}}^{\mathcal{S}}$  é comparado com o correto  $\mathbf{q}_{\mathcal{B}}^{\mathcal{S}}$  através da rotação entre eles, dada pela Eq. (A.8).

## 6.2.3 Simulações

A figura 6.8 apresenta os erros da estimação da rotação  $\mathcal{R}$ . É importante destacar que  $\gamma = 0,05$  equivale a um erro máximo de  $9^{\circ}$  em cada estimativa da rotação atual.

### 6.2.4 Análise

Em todos os casos, as estimativas que possuem medições com rotações em torno apenas do eixo z apresentaram resultados melhores. Apesar disso já ser esperado, uma vez que elas possuem 3M+1 amostras contra 2M+1 das demais, elas obtiveram esta melhora não só devido à quantidade de amostras.



**Figura 6.8:** Erro na estimativa de  $\mathcal{R}$  para variadas quantidades de amostra e níveis de erro.

Ao comparar as figuras 6.8b e 6.8c, nota-se que ambas possuem desempenhos semelhantes, apesar da primeira ter 7 amostras e a segunda 11. Esta melhora de performance pode ser associada

ao menor número de graus de liberdade na estimação da rotação em torno de z, uma vez que sabe-se que  $\mathbf{q}_{\mathcal{B}'}^{\mathcal{B}}$  são nulos, o que faz com que as estimativas sejam mais robustas ao ruído.

Na figura 6.8b, nota-se também que a mediana se encontra próxima à expectativa do erro das medições, mostrando que apenas 2 medições são insuficientes para fazer com que a estimação de  $\mathcal{R}$  tenha erros menores do que as estimativas originais. Aumentando o número de amostras, tem-se uma melhora lenta da qualidade de estimação, o que se deve à grande liberdade de movimentação a cada medição.

Como observado, a grande fonte de erro do algoritmo se deve ao nível de liberdade dado às medidas. Caso seja possível garantir que uma dada rotação em x e uma em z tenham o mesmo  $\mathbf{q}_{\mathcal{I}'}^{\mathcal{B}'}$ , o que pode ocorrer se o sistema se encontra num balanço, deve-se fazer uso desta informação durante os passos do algoritmo, alterando facilmente as matrizes  $\mathbf{M}_i$  para incluir ambas medidas. Entretanto, este trabalho foca em fornecer o máximo de liberdade para o usuário e evitar que fossem feitas hipóteses muito restritivas, que não podem ser satisfeitas facilmente.

## 6.3 Conclusão

Com base nos resultados da seção 6.1.4 e análise da seção 6.1.5, pode-se concluir que o algoritmo proposto para calibração é capaz de obter boas estimativas dos parâmetros para acelerômetros e magnetômetros. Concluímos também que não é necessário o uso de uma condição de parada muito restrita, uma vez que a qualidade da estimação permanece a mesma. Além disso, nota-se que o aumento do número de intervalos amostrados melhora significativamente a capacidade de generalização das estimativas dos parâmetros, resultando em erros menores nos dados de teste.

Como o aumento no número de intervalos não aumenta o tempo de computação, é recomendado capturar uma quantidade razoável de intervalos. Também recomenda-se o uso do algoritmo A2, descrito no fim da seção 6.1.2, uma vez que ele é capaz de fazer uso de uma aproximação da matriz de rotação, tornando-o mais rápido e menos sujeito a mínimos locais.

Deve-se ressaltar a importância de se utilizar dados de teste para verificar a calibração: apesar de uma maior quantidade de períodos amostrados não melhorar muito o erro para os dados utilizados na calibração, ela influencia muito na qualidade final da calibração. Esta melhora só poder ser

observada ao utilizarmos os dados de teste.

Os resultados apresentados na seção 6.2.3 e analisados na seção 6.2.4 mostram que a calibração da rotação  $\mathcal{R}$  entre  $\mathcal{B}$  e  $\mathcal{I}$  escalam bem com o nível de ruído e quantidade de amostras, apesar do algoritmo descrito impor pouquíssimas restrições ao usuário. Além disso, o algoritmo proposto pode ser utilizado com qualquer metodologia de estimação de atitude em 3 dimensões.

Destaca-se o fato de que os dois algoritmos podem ser utilizados com os mesmos dados. Uma vez feita a calibração dos parâmetros, como mostrado ao longo da seção 6.1, estima-se também as matrizes de rotação  $\mathbf{R}_i$  da leitura e seus quaternions  $\mathbf{q}_i$  associados. Comparando as equações dos sensores, dadas pelas Eqs. (3.9) e (3.8), com a Eq. (6.5), tem-se que as estimativas  $\mathbf{q}_i$  podem ser utilizadas na calibração da rotação ao fazer  ${}^i\mathbf{q}_{\mathcal{I}}^{\mathcal{S}} = \mathbf{q}_i^{-1}$ . Portanto, ao invés de realizar uma coleta de dados para a calibração dos sensores e outra para calibração da rotação  $\mathcal{R}$ , é possível realizar apenas uma coleta, possivelmente aumentando a quantidade de amostras para cada algoritmo.

# Capítulo 7

# Conclusão

O capítulo 4 utiliza um algoritmo de aprendizado de parâmetros para determinar os parâmetros de um controlador simples. Ao submeter o quadrirrotor a um ambiente ruidoso, observa-se que o aprendizado não apenas reduziu a amplitude das oscilações, como fez com que veículos com diferentes valores de inércias passassem a ter respostas semelhantes em seu comportamento. Esta observação motiva estudos futuros com relação ao uso deste aprendizado, uma vez que é extremamente interessante gerar controladores que fazem o sistema se comportar de uma determinada maneira independentemente dos seus parâmetros.

Entretanto, o controlador foi testado em ambiente de simulação com realimentação correta dos estados. Para que seja utilizado em sistemas reais, é necessário estimar o valor destes estados através de sensores. Este problema foi atacado no capítulo 5, dando maior enfoque na estimação da atitude por poder ser utilizada em parceria com diversos outros sensores. Após comparar filtros de Kalman com diferentes modelos, um algoritmo de adaptação da matriz de covariância do ruído da entrada foi proposto, o que permitiu uma redução significativa dos erros de atitude. Além disso, conclui-se que o uso desta adaptação para estimativa isolada da atitude e velocidade angular é capaz de gerar melhores resultados do que a estimação simultânea, apesar deste modelo desacoplar estados fortemente acoplados no sistema real. Uma vez tratado o problema da atitude, um novo algoritmo para estimação da posição utilizando o acelerômetro e o GPS foi descrito, sendo indicado para ambientes abertos. Devido ao desejo de gerar o melhor desempenho para o pior caso, descobriu-se que a estimação da posição por *dead reckoning* não é recomendada por sua ca-

pacidade de gerar grandes erros esporadicamente, apesar de apresentar grandes vantagens no caso geral. Para ambientes fechados, outros tipos de sensores, como câmeras e sensores laser, devem ser investigados.

Por fim, a filtragem espera que as leituras dos sensores sejam apenas versões ruidosas do valor real. Isto requer uma boa calibração dos sensores e levou ao desenvolvimento do capítulo 6. A calibração do acelerômetro e magnetômetro apresentou erros pequenos de reconstrução do sinal original, com medianas menores do que 0.1 desvios padrões. Discutiu-se ainda a importância de se utilizar dados de teste para avaliar a calibração, prática pouco comum nesta comunidade. Como a calibração destes sensores define um sistema de coordenadas arbitrário, possivelmente diferente do sistema de coordenadas desejado para especificar trajetórias e *waypoints*, foi desenvolvido em seguida um algoritmo para determinar a rotação entre estes dois sistemas de coordenadas, cujo desempenho também é satisfatório.

Assim, o controle do quadrirrotor, foco principal desta pesquisa, mostrou ser necessário pesquisa em duas áreas relacionadas, filtragem de estados e calibração de sensores, nas quais foram realizadas análises e desenvolvidos algoritmos para que o sistema como um todo possa corretamente. Estes resultados são expressivos e inovadores, tendo gerado por ora três publicações finalizadas (MIRANDA E FERREIRA; MIRANDA E FERREIRA, 2013a; 2013b; 2014) e outras em desenvolvimento, além de fornecer ideias para pesquisas posteriores. É importante destacar que, apesar do foco deste trabalho ser quadrirrotores, os avanços alcançados podem ser explorados em outras áreas, como veículos terrestres, aquáticos ou outros tipos de veículos aéreos, devido à similaridade dos sistemas e generalidade dos algoritmos.

# 7.1 Direções futuras para pesquisa

Como o desenvolvimento do controle levou ao estudo de filtragem, que por sua vez incentivou a pesquisa em calibração, as pesquisas futuras aqui descritas seguem o caminho contrário, para que desenvolvimentos numa área possam ter seus subsistemas funcionando adequadamente.

Para a calibração de acelerômetros e magnetômetros, os resultados apresentados na seção 6.1 partem de condições iniciais distantes dos valores corretos. Caso melhores estimativas iniciais fossem utilizadas, o algoritmo poderia se tornar mais rápido e mais robusto, uma vez que o número de iterações e o risco de parar em mínimo local seriam reduzidos. Além disso, um estudo da

caracterização do *bias drift* destes sensores deve ser realizada, visando verificar se este efeito é significativo o suficiente para incluir no algoritmo de calibração e para fornecer dados para que o algoritmo de filtragem seja capaz de compensar esta variação.

A calibração da rotação entre as coordenadas dos sensores e do corpo apresentada utiliza o erro quadrático como custo, o que não é o mais indicado para rotações. Além disso, o algoritmo restringe as rotações que o usuário pode realizar mais do que o necessário. Ambos problemas foram resolvidos num trabalho do autor a ser publicado (MIRANDA E FERREIRA, 2014). Outro problema é que a restrição com relação à inconsistência do sinal fornecido ao usuário e a medida pelo sistema é dura, o que pode aumentar o erro ou reduzir flexibilidade do usuário. Uma continuação da pesquisa realizada seria determinar se um relaxamento das restrições pode gerar resultados melhores e mais robustos.

Este trabalho não tratou da calibração de giroscópios, a qual normalmente é dada em aparelhos de velocidade de rotação precisamente controladas. Uma alternativa sendo pesquisada atualmente consiste na utilização de um acelerômetro para estimar a velocidade angular do sistema e utilizar esta estimativa para calibrar o giroscópio. Como a velocidade é medida indiretamente, há uma redução da qualidade de calibração, mas aparelhos mais simples e sem controle preciso podem ser utilizados.

O filtro de estados apresentado na seção 5.2.8 supunha que o sistema estava constantemente sofrendo acelerações elevadas, o que gerava o pior desempenho na estimação da atitude. Um algoritmo adaptativo, semelhante aos apresentados para ajustar a covariância de entrada, pode permitir que o sistema automaticamente se ajuste às diversas condições, possivelmente obtendo erros menores quando o sistema estiver de fato quase parado.

Outra extensão do filtro é a adição de estimadores de parâmetros dos sensores, como *bias*, ou do quadrirrotor, como inércia. Enquanto os primeiros podem ser utilizados para melhorar a qualidade da estimativa dos estados, os segundos podem ser explorados pelo controlador para se ter uma lei de controle com melhores garantias de estabilidade. Os filtros devem também ser testados com leituras obtidas a partir da calibração imperfeita dos sensores, verificando a robustez a valores incorretos mas próximos dos verdadeiros.

Por fim, o controlador apresentado no capítulo 4 supõe que o sistema está parado. Para que o controlador siga trajetórias arbitrárias, a matriz de aproximação do sinal da derivada deve ser

modificada para acompanhar a referência. Além disso, o quadrirrotor pode não ser capaz de determinar as velocidade angulares para os motores com base apenas nas forças e torques desejados, como é o caso quando os parâmetros das hélices são desconhecidos. Portanto, o controlador apresentado deve ser modificado para fornecer velocidades de rotação diretamente, podendo ser usado em qualquer quadrirrotor.

Tem-se ainda que a robustez do controlador ao utilizar estimativas dos estados, dadas por filtros, deve ser analisada. Uma possibilidade de melhoria seria o uso de outros controladores que utilizam mais informação do sistema ou que possuem mais garantias de estabilidade como base para o controle por aprendizado descrito neste trabalho. Esta frente também pode facilitar a portabilidade do algoritmo a sistemas com modelos diferentes.

Estes desenvolvimentos futuros devem ser acompanhados por experimentações em todas as etapas. Apesar desta dissertação apresentar apenas resultados simulados, prévias de resultados experimentais na parte de calibração, uma vez que esta é a primeira etapa de avaliação do sistema, mostram que os algoritmos apresentados possuem bons desempenhos.

# Apêndice A

# **Quaternions**

Este apêndice descreve o que são quaternions e suas operações, sendo um resumo de sua teoria utilizada neste trabalho. Para mais detalhes, recomenda-se a leitura de (CHOU, 1992) e (GRAF, 2008).

O quaternion é uma extensão matemática dos números complexos para quatro dimensões, podendo ser escritos como:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ \vec{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = q_0 + q_1 i + q_2 j + q_3 k, \tag{A.1}$$

onde  $q_0 \in \mathbb{R}$  é a componente escalar,  $\vec{\mathbf{q}} \in \mathbb{R}^3$  é a componente vetorial e i, j e k formam uma base ortonormal para  $\mathbb{R}^3$ .

A seção A.1 descreve as operações que podem ser realizadas com quaternions, sendo base para sua utilização como rotação, descrita na seção A.2. A seção A.3 fornece uma breve descrição das vantagens computacionais de se utilizar quaternions para representar rotações.

# A.1 Operações entre quaternions

O espaço dos quaternions, denotado por  $\mathbb{H}$ , é semelhante ao  $\mathbb{R}^4$ , sendo um espaço vetorial sobre os números reais. Portanto, a soma de dois quaternions e o produto entre um quaternion e um número escalar funciona de forma semelhante a vetores do  $\mathbb{R}^4$ . Entretanto, para produto entre quaternions, deve-se escolher uma base para representar o espaço vetorial.

Assim como o espaço  $\mathbb C$  dos números complexos tem sua base escolhida de forma que  $i^2=-1$ , o espaço  $\mathbb H$  é escolhido de forma que  $i^2=j^2=k^2=ijk=-1$ . Com esta definição do espaço, o produto entre dois quaternions, denotado  $\otimes$  neste trabalho, pode ser escrito como:

$$\mathbf{q} \otimes \mathbf{p} = (q_{0}p_{0} - q_{1}p_{1} - q_{2}p_{2} - q_{3}p_{3})$$

$$+ (q_{0}p_{1} + q_{1}p_{0} + q_{2}p_{3} - q_{3}p_{2})i$$

$$+ (q_{0}p_{2} - q_{1}p_{3} + q_{2}p_{0} + q_{3}p_{1})j$$

$$+ (q_{0}p_{3} + q_{1}p_{2} - q_{2}p_{1} + q_{3}p_{0})k$$

$$= \begin{bmatrix} q_{0}p_{0} - \vec{\mathbf{q}} \cdot \vec{\mathbf{p}} \\ q_{0}\vec{\mathbf{p}} + q_{0}\vec{\mathbf{q}} + \vec{\mathbf{q}} \times \vec{\mathbf{p}} \end{bmatrix},$$
(A.2b)

onde  $\cdot$  representa o produto interno e  $\times$  o produto vetorial no  $\mathbb{R}^3$ .

Uma alternativa para a representação do produto entre quaternions é a expressão do quaternion da esquerda como uma matriz no  $\mathbb{R}^{4\times 4}$ , dada por:

$$[\mathbf{q}\otimes] = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}, \tag{A.3}$$

ou do quaternion da diretita, dada por:

$$[\otimes \mathbf{q}] = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}, \tag{A.4}$$

de forma que  $\mathbf{q} \otimes \mathbf{p} = [\mathbf{q} \otimes] \mathbf{p} = [\otimes \mathbf{p}] \mathbf{q}$ .

O conjugado de um quaternion é dado por  $\mathbf{q}^* = q_0 - q_1 i - q_2 j - q_3 k$ , o que faz com que o conjugado de um produto possa ser escrito como  $(\mathbf{p} \otimes \mathbf{q})^* = \mathbf{q}^* \otimes \mathbf{p}^*$ . A norma é dada por  $\|\mathbf{q}\| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{\mathbf{q}^T \mathbf{q}}$ , sendo portanto a mesma definição de norma 2 no  $\mathbb{R}^4$ , e o recíproco é dado por  $\mathbf{q}^{-1} = \mathbf{q}^* / \|\mathbf{q}\|^2$ , de forma que  $\mathbf{q} \otimes \mathbf{q}^{-1} = 1$ . Por fim, o logaritmo de um quaternion é dado por:

$$\ln(\mathbf{q}) = \ln \|\mathbf{q}\| + \frac{\vec{\mathbf{q}}}{\|\vec{\mathbf{q}}\|} \arccos \frac{q_0}{\|\mathbf{q}\|}.$$
 (A.5)

# **A.2** Quaternion como uma rotação no $\mathbb{R}^3$

A decomposição polar de um quaternion é dada por:

$$\mathbf{q} = \|\mathbf{q}\|e^{\hat{\mathbf{v}}\theta} = \|\mathbf{q}\|(\cos\theta + \hat{\mathbf{v}}\sin\theta),\tag{A.6}$$

onde  $q_0 = \|\mathbf{q}\|\cos\theta \ \mathbf{e} \ \ddot{\mathbf{q}} = \hat{\mathbf{v}}\|\ddot{\mathbf{q}}\| = \hat{\mathbf{v}}\|\mathbf{q}\|\sin\theta.$ 

Caso  $\|\mathbf{q}\|=1$ , ou seja, o quaternion é unitário, ele pode representar uma rotação de um ângulo  $\psi=2\theta$  em torno da direção unitária  $\hat{\mathbf{v}}$  no  $\mathbb{R}^3$ . Neste trabalho, todos os quaternions são assumidos unitários, pois são utilizados apenas para representar rotações. Como uma rotação  $\theta=\frac{\psi+2\pi}{2}$  ao redor de  $\hat{\mathbf{v}}$  é equivalente a uma rotação  $\theta=\frac{\psi}{2}$  na mesma direção, tem-se que

$$\cos\left(\frac{\psi}{2} + \pi\right) + \vec{\mathbf{v}}\sin\left(\frac{\psi}{2} + \pi\right) = -\cos\frac{\psi}{2} - \vec{\mathbf{v}}\sin\frac{\psi}{2},\tag{A.7}$$

ou seja, os quaternion q e −q representam a mesma rotação.

A seção A.2.1 apresenta o método usado para calcular o ângulo entre dois quaternions, sendo usada como uma medida de erro. As seções A.2.2 e A.2.3 descrevem como o quaternion se relaciona com a matriz de rotação no espaço e como utilizá-lo para rotacionar vetores. Por fim, a seção A.2.4 descreve o modelo dinâmico do quaternion.

## A.2.1 Ângulo entre quaternions

Dados dois quaternions  $\mathbf{q}$  e  $\mathbf{p}$ , o quaternion que representa a distância entre eles é dado por  $\mathbf{d} = \mathbf{q} \otimes \mathbf{p}^{-1}$ . Utilizando a decomposição polar, dada pela Eq. (A.6), o ângulo de rotação entre eles pode ser calculado como:

$$\psi_{\mathbf{q},\mathbf{p}} = 2\arccos|d_0|,\tag{A.8}$$

uma vez que um ângulo negativo pode ser representado por um ângulo positivo e uma inversão da direção.

## A.2.2 Transformação de matriz de rotação para quaternion

Dada uma aproximação da matriz de rotação

$$\mathbf{R} = \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{bmatrix},$$
 (A.9)

que pode não satisfazer as restrições de ortonormalidade devido a problemas numéricos, seu quaternion equivalente pode ser obtido a partir do método descrito em (BAR-ITZHACK, 2000). Seja a matriz **K** definida como:

$$\mathbf{K} = \frac{1}{3} \begin{bmatrix} R_{xx} + R_{yy} + R_{zz} & R_{yz} - R_{zy} & R_{zx} - R_{xz} & R_{xy} - R_{yx} \\ R_{yz} - R_{zy} & R_{xx} - R_{yy} - R_{zz} & R_{xy} + R_{yx} & R_{xz} + R_{zx} \\ R_{zx} - R_{xz} & R_{xy} + R_{yx} & -R_{xx} + R_{yy} - R_{zz} & R_{yz} + R_{zy} \\ R_{xy} - R_{yx} & R_{xz} + R_{zx} & R_{yz} + R_{zy} & -R_{xx} - R_{yy} + R_{zz} \end{bmatrix},$$
(A.10)

o quaternion que representa a rotação é dado pelo autovetor associado ao maior autovalor de K. Além disso, este autovalor é 1 se e somente se R for uma matriz de rotação ortonormal.

## A.2.3 Transformação de quaternion para matriz de rotação

Sejam as bases  $\mathbf{x}^A = [1,0,0]^T$ ,  $\mathbf{y}^A = [0,1,0]^T$  e  $\mathbf{z}^A = [0,0,1]^T$  de um sistema de coordenadas A e as bases  $\mathbf{x}^B$ ,  $\mathbf{y}^B$  e  $\mathbf{z}^B$  de outro sistema de coordenadas B. Note que esta escolha de A não gera perda de generalidade, uma vez que a rotação entre A e B pode ser descrita por uma rotação entre A e A', seguida de uma rotação entre A' e B, onde A' possui as bases escolhidas. A matriz de

rotação  $\mathbf{R}_A^B$  de A para B pode ser escrita como:

$$\mathbf{R}_{A}^{B} = \left[ \begin{array}{ccc} \mathbf{x}^{B} & \mathbf{y}^{B} & \mathbf{z}^{B} \end{array} \right], \tag{A.11}$$

de forma que um vetor  $\mathbf{v}^A$  descrito na base A pode ser descrito por  $\mathbf{v}^B = \mathbf{R}_A^B \mathbf{v}^A$  na base B. Portanto, para definir a matriz de rotação a partir de um quaternion, é necessário apenas descrever como um quaternion  $\mathbf{q}_A^B$  é utilizado para representar um vetor  $\mathbf{v}^A$  na base B.

Como o espaço  $\mathbb{H}$  dos quaternions é equivalente ao  $\mathbb{R}^4$ , um vetor  $\mathbf{v}^A \in \mathbb{R}^3$  pode ser escrito como um quaternion com termo escalar nulo. A rotação então é dada por:

$$\mathbf{v}^B = \mathbf{q}_A^B \otimes \mathbf{v}^A \otimes \left(\mathbf{q}_A^B\right)^{-1}. \tag{A.12}$$

Para se determinar a rotação inversa  $\mathbf{q}_B^A$ , basta realizar manipulações algébricas com equação anterior, uma vez que

$$\mathbf{v}^A = \mathbf{q}_B^A \otimes \mathbf{v}^B \otimes \left(\mathbf{q}_B^A\right)^{-1} \tag{A.13a}$$

$$= \mathbf{q}_{B}^{A} \otimes \left(\mathbf{q}_{A}^{B} \otimes \mathbf{v}^{A} \otimes \left(\mathbf{q}_{A}^{B}\right)^{-1}\right) \otimes \left(\mathbf{q}_{B}^{A}\right)^{-1} \tag{A.13b}$$

$$= \left(\mathbf{q}_{B}^{A} \otimes \mathbf{q}_{A}^{B}\right) \mathbf{v}^{A} \left(\left(\mathbf{q}_{A}^{B}\right)^{-1} \otimes \left(\mathbf{q}_{B}^{A}\right)^{-1}\right) \tag{A.13c}$$

$$= \left(\mathbf{q}_{B}^{A} \otimes \mathbf{q}_{A}^{B}\right) \mathbf{v}^{A} \left(\mathbf{q}_{B}^{A} \otimes \mathbf{q}_{A}^{B}\right)^{-1} \tag{A.13d}$$

implica que  $\mathbf{q}_{B}^{A} \otimes \mathbf{q}_{A}^{B}$  não realiza nenhuma rotação no vetor, ou seja,  $\mathbf{q}_{B}^{A} \otimes \mathbf{q}_{A}^{B} = 1$ . Pela definição do elemento recíproco, tem-se portanto que  $\mathbf{q}_{B}^{A} = \left(\mathbf{q}_{A}^{B}\right)^{-1}$ .

## A.2.4 Evolução no tempo

Seja uma rotação, representada pelo quaternion  $\mathbf{q}_A^B$ , submetida a uma velocidade angular  $\omega$ . A derivada do quaternion é dada por:

$$\dot{\mathbf{q}}_A^B(t) = \frac{1}{2}\omega^A(t) \otimes \mathbf{q}_A^B(t) \tag{A.14a}$$

$$= \frac{1}{2} \left( \mathbf{q}_A^B(t) \otimes \omega^B(t) \otimes \left( \mathbf{q}_A^B(t) \right)^{-1} \right) \otimes \mathbf{q}_A^B(t)$$
 (A.14b)

$$= \frac{1}{2} \mathbf{q}_A^B(t) \otimes \omega^B(t), \tag{A.14c}$$

onde  $\omega^A$  e  $\omega^B$  são representações da velocidade angular nas coordenadas A e B, respectivamente. Utilizando a definição de produto entre quaternions descrita na Eq. (A.3), fica claro que a dinâmica do quaternion dada pela Eq. (A.14a) é linear.

Considere duas amostragens do quaternion  $\mathbf{q}_A^B[k+1]$  e  $\mathbf{q}_A^B[k]$  distantes  $T_s$  segundos. Seja a distância entre eles  $\mathbf{d}[k] = \mathbf{q}_A^B[k+1] \otimes (\mathbf{q}_A^B[k])^{-1}$ . Utilizando a decomposição polar, descrita na Eq. (A.6), pode-se escrever  $\mathbf{d}[k]$  como:

$$\mathbf{d}[k] = e^{\hat{\mathbf{v}}\theta} = e^{\frac{\omega^A[k]T_s}{2}},\tag{A.15}$$

onde a velocidade angular  $\omega^A[k]$  é suposta constante no intervalo  $t \in [kT_s, (k+1)T_s)$ . Através da definição do logaritmo de quaternions, dada pela Eq. (A.5), pode-se obter a velocidade angular como:

$$\omega^{A}[k] = 2 \frac{\ln \mathbf{d}[k]}{T_{s}} = 2 \frac{\ln \left(\mathbf{q}_{A}^{B}[k+1] \otimes \left(\mathbf{q}_{A}^{B}[k]\right)^{-1}\right)}{T_{s}}.$$
(A.16)

## A.3 Vantagens do quaternion

Dentre as principais vantagens ao se utilizar quaternions neste trabalho, tem-se:

- 1. Representação não singular: a representação de rotação por ângulos de Euler possui singularidades, ou seja, apesar de corretamente ser capaz de representar as rotações, existem orientações nas quais uma infinidade de parâmetros descreve a mesma rotação.
- 2. Velocidade e tamanho: apesar de quaternions requerem mais operações para se fazer rotações do a representação direta por matrizes de rotação, eles contêm menos termos, sendo mais compactos. Como a rotação poder ser descrita através de multiplicações e somas, ela é computacionalmente vantajosa em relação aos ângulos de Euler, que requerem operações trigonométricas.
- 3. Erros de arredondamento: ao se realizar rotações, erros de arredondamento se acumulam e a representação pode deixar de ser apenas uma rotação. Para o quaternion, isso é corrigido rapidamente apenas realizando sua normalização, enquanto uma matriz de rotação é difícil de ser convertida para uma matriz ortogonal minimizando o erro.

- 4. Medição de erro simplificada: a medida de erro entre rotações utilizada neste trabalho é dada pelo ângulo entre duas orientações, que pode ser calculada facilmente através de quaternions, como descrito na seção A.2.1. Esta medida é mais complicada para o caso de matrizes de rotação, enquanto o uso direto dos ângulos de Euler podem gerar erros grandes para pequenas rotações.
- 5. Dinâmica: como descrito na seção A.2.4, a dinâmica do quaternion é linear com a velocidade angular, o que permite tirar vantagem de toda a teoria de sistemas lineares, em particular da garantia de previsão ótima dada pelo filtro de Kalman.

# Referências Bibliográficas

- ALONSO, R. e SHUSTER, M.D. Complete linear attitude-independent magnetometer calibration. **Journal of the Astronautical Sciences**, v. 50, n. 4, 477–490, 2002a.
- ALONSO, R. e SHUSTER, M.D. TWOSTEP: a fast robust algorithm for attitude-independent magnetometer-bias determination. **Journal of Astronautical Sciences**, v. 50, n. 4, 433–451, 2002b.
- ARULAMPALAM, M.S.; MASKELL, S.; GORDON, N. e CLAPP, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. **IEEE Transactions on Signal Processing**, v. 50, n. 2, 174–188, 2002.
- ASLAN, G. e SARANLI, A. Characterization and calibration of MEMS inertial measurement units. In **16th European Signal Processing Conference**, Eusipeo, pp. 1–5. 2008.
- BAR-ITZHACK, I.Y. New method for extracting the quaternion from a rotation matrix. **Journal of guidance, control, and dynamics**, v. 23, n. 6, 2000.
- BESNARD, L. Control of a quadrotor vehicle using sliding mode disturbance observer. **American Control Conference**, **2007**. **ACC'07**, pp. 5230–5235, 2007.
- BESNARD, L.; SHTESSEL, Y.B. e LANDRUM, B. Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. **Journal of the Franklin Institute**, v. 349, n. 2, 658–684, março 2012.
- BOUABDALLAH, S.; MURRIERI, P. e SIEGWART, R. Design and control of an indoor micro quadrotor. In **Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on**, pp. 4393–4398, vol. 5. Ieee, 2004. ISBN 0-7803-8232-3.

- BOUABDALLAH, S. e SIEGWART, R. Full control of a quadrotor. In **Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on**, 1, pp. 153–158. Ieee, outubro 2007. ISBN 978-1-4244-0911-2.
- BRISSET, P.; DROUIN, A. e GORRAZ, M. The paparazzi solution. **MAV2006**, **Sandestin**, **Florida**, 2006.
- CASTILLO, P.; DZUL, A. e LOZANO, R. Real-time stabilization and tracking of a four-rotor mini rotorcraft. **Control Systems Technology, IEEE Transactions on**, v. 12, n. 4, 510–516, 2004.
- CHOU, J.C. Quaternion kinematic and dynamic differential equations. **Robotics and Automation**, **IEEE Transactions on**, v. 8, n. 1, 53–64, 1992.
- COOK, K.L.B. The silent force multiplier: The history and role of uavs in warfare. In **Aerospace Conference**, **2007 IEEE**, pp. 1–7. March 2007.
- CRASSIDIS, J.L.; LAI, K. e HARMAN, R.R. Real-time attitude-independent three-axis magnetometer calibration. **Journal of Guidance Control and Dynamics**, v. 28, n. 1, 115–120, janeiro 2005.
- CRASSIDIS, J.L.; MARKLEY, F.L. e LIGHTSEY, E.G. Global Positioning System Integer Ambiguity Resolution Without Attitude Knowledge. **Journal of Guidance Control and Dynamics**, v. 22, n. 2, 212–218, março 1999.
- DEVAUD, J.; NAJKO, S.; NAHEDIC, P.L.; MAUSSIRE, C.; ZANTE, E. e MARZAT, J. Full design of a low-cost quadrotor UAV by student team. In **Proceedings of the International Conference on System Engineering and Technology, ICSET 2012**. 2012.
- DODDS, Z. Can Quadrotors Succeed as an Educational Platform? In **Twenty-Fifth AAAI Conference on Artificial Intelligence**, pp. 1870–1871. 2011.
- ELKAIM, G.H. Misalignment Calibration Using Body Frame Measurements. Relatório técnico, 2013.
- ELKAIM, G.H.; LIE, F.A.P. e GEBRE-EGZIABHER, D. Principles of Guidance, Navigation and Control of UAVs. Relatório técnico, 2012.
- FOSTER, C.C. e ELKAIM, G.H. Extension of a two-step calibration methodology to include nonorthogonal sensor axes. **Aerospace and Electronic Systems, IEEE Transactions on**, v. 44, n. 3, 1070–1078, 2008.

- GEBRE-EGZIABHER, D. e ELKAIM, G.H. MAV attitude determination by vector matching. **Aerospace and Electronic Systems, IEEE Transactions on**, v. 44, n. 3, 1012–1028, julho 2008.
- GEBRE-EGZIABHER, D.; ELKAIM, G.H.; POWELL, J.D. e PARKINSON, B.W. A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors. In **Position Location and Navigation Symposium, IEEE 2000**, pp. 185–192. Ieee, 2000. ISBN 0-7803-5872-4.
- GRAF, B. Quaternions and dynamics. arXiv preprint arXiv:0811.2889, 2008.
- GRZONKA, S.; GRISETTI, G. e BURGARD, W. Towards a navigation system for autonomous indoor flying. In **Robotics and Automation**, **2009**. **ICRA'09**. **IEEE International Conference on**, Section III, pp. 2878–2883. 2009.
- GUENARD, N.; HAMEL, T. e MOREAU, V. Dynamic modeling and intuitive control strategy for an X4-flyer. In **Control and Automation, 2005. ICCA'05. International Conference on**, pp. 141–146, vol. 1. 2005.
- HAKYOUNG, C.; OJEDA, L. e BORENSTEIN, J. Accurate mobile robot dead-reckoning with a precision-calibrated fiber optic gyroscope. **Robotics and Automation, IEEE Transactions on**, v. 17, n. 1, 80–84, 2001.
- HAN, J.; SONG, Q. e HE, Y. Adaptive unscented Kalman filter and its applications in nonlinear control. Chapter of 'Applications of Kalman Filter', ..., n. April, 1–25, 2009.
- HAUG, A.J. A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian processes. **MITRE Corporation, McLean**, 2005.
- HEHN, M. e D'ANDREA, R. A flying inverted pendulum. In **Robotics and Automation (ICRA)**, **2011 IEEE International Conference on**, pp. 763–770. IEEE, 2011.
- HOFFMANN, G.; RAJNARAYAN, D.G.; WASLANDER, S.L.; DOSTAL, D.; JANG, J.S. e TOM-LIN, C.J. The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC). In **Digital Avionics Systems Conference**, **2004. DASC 04. The 23rd**. Ieee, 2004. ISBN 0-7803-8539-X.
- HOFFMANN, G.M.; HUANG, H.; WASLANDER, S.L. e TOMLIM, C.J. Quadrotor helicopter flight dynamics and control: Theory and experiment. In **Proc. of the AIAA Guidance, Navigation, and Control Conference**, August, pp. 1–20. 2007.

- HOL, J. Sensor Fusion and Calibration using Inertial Sensors, Vision, Ultra-Wideband and GPS. 2011. p. 1–165. Tese (Doutorado).
- JONES, E.; OLIPHANT, T.; PETERSON, P. *e outros*. SciPy: Open source scientific tools for Python. 2001–.
- JULIER, S.J. e UHLMANN, J.K. New extension of the Kalman filter to nonlinear systems. In **AeroSense'97**, pp. 182–193. 1997.
- JULIER, S.J. e UHLMANN, J.K. Unscented Filtering and Nonlinear Estimation. **Proceedings of the IEEE**, v. 92, n. 3, 401–422, março 2004.
- JULIER, S.J.; UHLMANN, J.K. e DURRANT-WHYTE, H.F. A new approach for filtering non-linear systems. In **American Control Conference**, **1995**. **Proceedings of the**, v. 3, pp. 1628–1632, vol. 3. American Autom Control Council, 1995. ISBN 0-7803-2445-5.
- JULIER, S.J.; UHLMANN, J.K. e DURRANT-WHYTE, H.F. A new method for the nonlinear transformation of means and covariances in filters and estimators. **Automatic Control, IEEE Transactions on**, v. 45, n. 3, 477–482, 2000.
- KALMAN, R.E. A new approach to linear filtering and prediction problems. **Journal of basic Engineering**, v. 82, n. 1, 35–45, 1960.
- KAPALDO, A. J. Gyroscope Calibration and Dead Reckoning for an Autonomous Underwater Vehicle. 2005. p. 1–63. Tese (Doutorado).
- KARNEY, C.F.F. Quaternions in molecular modeling. **Journal of Molecular Graphics and Modelling**, v. 25, n. 5, 595–604, junho 2005.
- KOLTER, J. Z. Learning and control with inaccurate models. 2010. p. 1–183. Tese (Doutorado).
- KUFFNER, J.J. Effective sampling and distance metrics for 3D rigid body path planning. In **Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on**, April, pp. 3993–3998, vol. 4. 2004. ISBN 0780382323.
- LUPASHIN, S.; SCHOLLIG, A.; HEHN, M. e D'ANDREA, R. The flying machine arena as of 2010. In **Robotics and Automation (ICRA), 2011 IEEE International Conference on**, pp. 2970–2971. IEEE, 2011.

- LUPASHIN, S.; SCHOLLIG, A.; SHERBACK, M. e D'ANDREA, R. A simple learning strategy for high-speed quadrocopter multi-flips. In **Robotics and Automation (ICRA), 2010 IEEE International Conference on**, pp. 1642–1648. IEEE, 2010.
- MADGWICK, S.O.H. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Relatório técnico, 2010.

URL: http://sharenet-wii-motion-trac.googlecode.com

- MADGWICK, S.O.H.; HARRISON, A.J.L. e VAIDYANATHAN, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. In **Rehabilitation Robotics** (**ICORR**), **2011 IEEE International Conference on**, v. 2011, pp. 1–7. junho 2011. ISBN 9781424498628.
- MAHALANOBIS, P.C. On the generalized distance in statistics. In **Proceedings of the National Institute of Sciences of India**, v. 2, pp. 49–55. New Delhi, 1936.
- MARINS, J.L.; YUN, X.; BACHMANN, E.B.; MCGHEE, R.B. e ZYDA, M.J. An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In **Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on**, v. 4, pp. 2003–2011, vol. 4. Ieee, 2001. ISBN 0-7803-6612-3.
- MCGEE, L.A. e SCHMIDT, S.F. Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry. Relatório Técnico November, 1985.
- MELLINGER, D. e KUMAR, V. Minimum snap trajectory generation and control for quadrotors. In **Robotics and Automation (ICRA), 2011 IEEE International Conference on**, pp. 2520–2525. 2011.
- MIRANDA, C.S. e FERREIRA, J.V. Greedy calibration of the rotation between sensor and body frame without external references. In **Proceedings of the ABCM International Congress of Mechanical Engineering**. ABCM, 2013a.
- MIRANDA, C.S. e FERREIRA, J.V. Greedy statistically correct simultaneous calibration of magnetometers and accelerometers. In **Proceedings of the ABCM International Congress of Mechanical Engineering**. ABCM, 2013b.
- MIRANDA, C.S. e FERREIRA, J.V. Sensor and body frames rotation calibration through attitude restriction. In **Proceedings of the IFAC World Congress (to appear)**. IFAC, 2014.

- MULLER, M.; LUPASHIN, S. e D'ANDREA, R. Quadrocopter ball juggling. In **Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on**, pp. 5113–5120. IEEE, 2011.
- OJEDA, L.; CHUNG, H. e BORENSTEIN, J. Precision calibration of fiber-optics gyroscopes for mobile robot navigation. In **Robotics and Automation**, **2000**. **Proceedings**. **ICRA'00**. **IEEE International Conference on**, pp. 2064–2069. 2000.
- OLIVARES, A.; OLIVARES, G.; GORRIZ, J.M. e RAMIREZ, J. High-efficiency low-cost accelerometer-aided gyroscope calibration. In **Test and Measurement, 2009. ICTM'09. International Conference on**, pp. 354–360, vol. 1. Ieee, dezembro 2009. ISBN 978-1-4244-4699-5.
- PARK, S.; WON, D.H.; KANG, M.S.; KIM, T.J.; LEE, H.G. e KWON, S.J. RIC (robust internal-loop compensator) based flight control of a quad-rotor type UAV. In **Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on**, pp. 3542–3547. Ieee, 2005. ISBN 0-7803-8912-3.
- POUNDS, P. e MAHONY, R. Modelling and control of a quad-rotor robot. In **Proceedings Australasian Conference on Robotics and Automation 2006**. 2006.
- POUNDS, P. e MAHONY, R. Design principles of large quadrotors for practical applications. In **Robotics and Automation, 2009. ICRA'09. IEEE International Conference on**, pp. 3265–3270. Ieee, maio 2009. ISBN 978-1-4244-2788-8.
- POUNDS, P.; MAHONY, R. e CORKE, P. Modelling and control of a large quadrotor robot. **Control Engineering Practice**, v. 18, n. 7, 691–699, julho 2010.
- POUNDS, P.; MAHONY, R.; GRESHAM, J.; CORKE, P. e ROBERTS, J. Towards dynamically-favourable quad-rotor aerial robots. In **Proceedings of the 2004 Australasian Conference on Robotics & Automation**. 2004.
- PYLVANAINEN, T. Automatic and adaptive calibration of 3D field sensors. **Applied Mathematical Modelling**, v. 32, n. 4, 575–587, abril 2008.
- RENAUDIN, V.; AFZAL, M.H. e LACHAPELLE, G. New method for magnetometers based orientation estimation. In **Position Location and Navigation Symposium (PLANS), 2010 IE-EE/ION**, D, pp. 348–356. Ieee, maio 2010. ISBN 978-1-4244-5036-7.

- ROY, N. e THRUN, S. Online self-calibration for mobile robots. In **Robotics and Automation**, **1999. Proceedings. 1999 IEEE International Conference on**, v. 3, pp. 2292–2297, vol. 3. IEEE, 1999.
- SABATINI, A.M. Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. **Biomedical Engineering, IEEE Transactions on**, v. 53, n. 7, 1346–1356, julho 2006.
- SALAZAR-CRUZ, S.; PALOMINO, A. e LOZANO, R. Trajectory tracking for a four rotor miniaircraft. In **Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on**, pp. 2505–2510. Ieee, 2005. ISBN 0-7803-9567-0.
- SEETHARAMAN, G.; LAKHOTIA, A. e BLASCH, E.P. Unmanned vehicles come of age: The darpa grand challenge. **Computer**, v. 39, n. 12, 26–29, December 2006.
- STEWART, I. From here to infinity. 1996.
- TAYEBI, A. e MCGILVRAY, S. Attitude stabilization of a four-rotor aerial robot. In **Decision** and Control, 2004. CDC. 43rd IEEE Conference on, pp. 1216–1221, vol. 2. Ieee, 2004. ISBN 0-7803-8682-5.
- THRUN, S.; BURGARD, W. e FOX, D. **Probabilistic Robotics**. Intelligent Robotics and Autonomous Agents Series. Mit Press, 2005. ISBN 9780262201629.
- VASCONCELOS, J.F.; ELKAIM, G.H.; SILVESTRE, C.; OLIVEIRA, P. e CARDEIRA, B. A geometric approach to strapdown magnetometer calibration in sensor frame. In **Navigation**, **Guidance and Control of Underwater Vehicles**, pp. 172–177, vol. 2. 2008.
- VOOS, H. Nonlinear control of a quadrotor micro-UAV using feedback-linearization. In Mechatronics, 2009. ICM 2009. IEEE International Conference on, April. 2009. ISBN 9781424441952.