Roberto Santos Inoue

**Controle robusto descentralizado de movimentos
coordenados de robôs heterogêneos¹**

Tese apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Doutor em Ciências, Programa de Engenharia Elétrica.

Área de Concentração: Sistemas Dinâmicos
Orientador: Prof. Dr. Marco Henrique Terra

São Carlos
2012

¹Trata-se da versão corrigida da tese. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

I58c

Inoue, Roberto Santos.

Controle robusto descentralizado de movimentos
coordenados de robôs heterogêneos. / Roberto Santos Inoue
; orientador Marco Henrique Terra. São Carlos, 2011.

Tese (Doutorado - Programa de Pós-Graduação em
Engenharia Elétrica e Área de Concentração em Sistemas
Dinâmicos)-- Escola de Engenharia de São Carlos da
Universidade de São Paulo, 2011.

1. Robótica. 2. Robô helicóptero. 3. Robô móvel. 4.
Controle de formação. 5. Filtro de Kalman. 5. Controle
robusto. 6. Filtro robusto. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **ROBERTO SANTOS INOUE**

Título da tese: "Controle robusto descentralizado de movimentos coordenados de robôs heterogêneos".

Data da defesa: 12/01/2012

Comissão Julgadora:

Resultado:

Prof. Associado **Marco Henrique Terra (Orientador)**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Dr. **Ramon Romankevicius Costa**
(Universidade Federal do Rio de Janeiro/UFRJ)

APROVADO

Dr. **Samuel Siqueira Bueno**
(CTI/Centro de Tecnologia da Informação Renato Archer)

APROVADO

Prof. Dr. **Adriano Almeida Gonçalves Siqueira**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Dr. **Denis Fernando Wolf**
(Instituto de Ciências Matemáticas e de Computação/ICMC)

APROVADO

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Titular **Denis Vinicius Coury**

Presidente da Comissão de Pós-Graduação:
Prof. Associado **Paulo Cesar Lima Segantine**

ESCOLA DE ENGENHARIA DE SÃO CARLOS
Serviço de Pós-Graduação
Procedimento de Julgamento de Tese
Data: 12/01/2012
Assinatura: _____

Dedicatória

*Dedico este trabalho aos meus pais
Francisco Hitoshi Inoue e Maria Selma
dos Santos com amor e gratidão.*

Agradecimentos

A Deus, que me concedeu saúde física e mental para a realização deste trabalho.

Aos meus pais Francisco Hitoshi Inoue e Maria Selma dos Santos com amor, admiração e gratidão por sua compreensão, carinho, presença e incansável apoio ao longo do período deste doutorado.

À minha vó Eliaci e meus tios Gilberto e Djalma que me ajudaram em minha permanência em São Carlos.

Ao Prof Dr. Marco Henrique Terra pela confiança, orientação, paciência e pelo tempo dedicado a este trabalho.

Ao Prof. Dr. Adriano Almeida Gonçalves Siqueira pela atenção, apoio e contribuições na realização deste trabalho.

Aos amigos da pós-graduação e do LASI: Aline, Amanda, Carolina, Daniel, Darby, Gildson, Giovana, João Paulo, Luciano, Miguel, Pedro, Rafael, Raphael, Robson, Samuel, Saulo, Tatiana, pela amizade, paciência, companheirismo e colaborações durante a realização das disciplinas e deste trabalho.

Aos professores e funcionários do Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos, pelas contribuições durante o doutorado.

À FAPESP pela concessão da bolsa de doutorado.

*“Mesmo que já tenha feito uma longa
caminhada, sempre haverá mais um
caminho a percorrer.”*

Santo Agostinho

Resumo

INOUE, R. S. Controle robusto descentralizado de movimentos coordenados de robôs heterogêneos. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2011.

Este trabalho trata da coordenação de robôs heterogêneos que consiste em um robô helicóptero e em múltiplos robôs móveis com rodas, de modo que estes sigam um líder. Para atingirmos este objetivo, neste trabalho desenvolve-se a estimação da atitude e posição de um robô utilizando filtros robustos. Os filtros realizam a fusão dos sinais de uma unidade de medida inercial de baixo custo e de um receptor GPS (Global Positioning System) considerando um modelo de corpo rígido formulado em termos de quatérnios. Resultados experimentais são apresentados baseados em um estudo comparativo entre os filtros robustos e o filtro de Kalman.

Desenvolve-se o controle de um robô móvel com rodas deslizantes e de um robô helicóptero. Para realizar o controle robusto utiliza-se um regulador robusto recursivo. Simulações do robô móvel com rodas deslizantes e do robô helicóptero são apresentadas utilizando os parâmetros de um robô móvel Pioneer 3AT e de um robô helicóptero RMAX.

Desenvolve-se também um mini robô helicóptero e uma base de movimento para realização de testes de controle do mini robô helicóptero. O mini robô helicóptero consiste de um mini helimodelo e de um piloto automático baseado em um microprocessador com conectividade Wi-Fi, uma unidade de medida inercial e uma placa de controle de servos motores.

E por fim, desenvolve-se a coordenação de robôs heterogêneos de modo que estes sigam um líder. Para isto é utilizado um controlador descentralizado e cooperativo cuja finalidade é gerar trajetórias de referência para que os robôs heterogêneos se movimentem em formação rígida.

Palavras-Chave: *Robótica, Robô helicóptero, robô móvel, controle de formação, filtro de Kalman, controle robusto e filtro robusto.*

Abstract

INOUE, R. S. Decentralized robust control of coordinated movements of heterogeneous robots. Thesis (Doctoral) - São Carlos School of Engineering, University of São Paulo, São Carlos, 2011.

This research deals with the coordination of the heterogeneous robots, consisting of a robotic helicopter and multiple wheeled mobile robots, to achieve this aim this work develops the estimation of attitude, heading and position of a robot based on robust filters. The filters perform the fusion of the signals of a low-cost inertial measurement unit and a GPS (Global Positioning System) receiver considering a rigid body model formulated in terms of quaternions. Experimental results are presented based on a comparative study of the robust filters and Kalman filter.

It develops a robust control scheme of a skid-steering mobile robot and a robotic helicopter. The robust control is performed through a the robust recursive regulator. Simulations of the skid-steering mobile robot and a robotic helicopter are presented using the parameters of a Pioneer 3AT and a robotic helicopter RMAX.

It also develops a mini robotic helicopter and a motion base to perform control test of a mini robotic helicopter. The mini robotic helicopter consists of a mini helicopter TREX 450 XL and an autopilot based on a microprocessor with Wi-Fi, an inertial measurement unit and a servo control board.

To finish, it develops the coordination of the heterogeneous robots in order to perform a leader

strategy based on the concept of rigid formation.

Palavras-Chave: *Robotics, robotic helicopter, mobile robot, formation control , Kalman filtering, robust filtering and robust control.*

Publicações

Terra, M. H., Ishihara, J. Y. and Inoue, R. S. (2011). A new approach to robust linear filtering problems In: 18th IFAC World Congress, Milão, Itália.

Caltran, C., Siqueira, A. A. G., Inoue, R. S. and Terra, M. H. (2011). Robust Filtering Applied to Position Estimation of an Active Ankle-Foot Orthosis In: 18th IFAC World Congress, Milão, Itália.

Inoue R. S. and Terra, M. H. (2011). Robust Recursive Kalman Filtering for Attitude Estimation In: 18th IFAC World Congress, Milão, Itália.

Alves Barbosa de Oliveira Vaz, D., Inoue, R. S. and Junior, V. G. (2010). Kinodynamic Motion Planning of a Skid-Steering Mobile Robot Using RRTs In: Latin American Robotics Symposium and Intelligent Robotic Meeting (LARS), Sao Bernardo do Campo.

Inoue, R. S., Terra, M. H., Siqueira, A. A. G. e Yabarrena, J. M. S. C. (2009). Filtragem Robusta para Navegação Inercial In: IX SBAI - Simpósio Brasileiro de Automação Inteligente, Brasília.

Inoue, R. S., Siqueira, A. A. G. and Terra, M. H. (2009). Experimental results on the nonlinear H_∞ control via quasi-LPV representation and Game Theory for wheeled mobile robots, Robotica 27(4), 547-553.

Ishihara, J. Y., Terra, M. H., Borges, G. A., Scandaroli, G. G., Inoue, R. S. and Junior, V. G. (2009). Applications of Robust Descriptor Kalman Filter in Robotics In: Kalman Filter: Recent Advances and Applications ed. Vienna, Austria : I-Tech Education and Publishing KG,

507-534.

Inoue, R. S., Terra, M. H. and Junior, V. G. (2008). Robust State-Space Estimation for Mobile Robot Localization In: 5º Simpósio IEEE Latino-Americano de Robótica, Salvador.

Lista de Figuras

1.1	Coordenação de robôs heterogêneos.	8
1.2	Diagrama de blocos do sistema de atitude e orientação.	8
1.3	Diagrama de blocos do sistema de posição.	9
2.1	Sistema de referência de atitude.	14
2.2	Variância do erro com Δ selecionado uniformemente entre o intervalo $[-1, 1]$. . .	28
2.3	Sequência de dados do giroscópio do eixo x (a), e desvio de Allan do giroscópio do eixo x (b).	30
2.4	Sequência de dados do giroscópio do eixo y (a), e desvio de Allan do giroscópio do eixo y (b).	30
2.5	Sequência de dados do giroscópio do eixo z (a), e desvio de Allan do giroscópio do eixo z (b).	31
2.6	Sequência de dados do acelerômetro do eixo x (a), e desvio de Allan do acelerômetro do eixo x (b).	31
2.7	Sequência de dados do acelerômetro do eixo y (a), e desvio de Allan do acelerômetro do eixo y (b).	31
2.8	Sequência de dados do acelerômetro do eixo z (a), e desvio de Allan do acelerômetro do eixo z (b).	32

2.9	IMU com <i>encoder</i> acoplado.	32
2.10	Diagrama de blocos do sistema de atitude.	33
2.11	Distúrbios do giroscópio.	35
2.12	Rolagem ϕ (a), arfagem θ (b), e guinada ψ (c) para $c = 1$	37
2.13	Rolagem ϕ (a) , arfagem θ (b), e guinada ψ (c) no intervalo (29s - 31s) para $c = 1$	37
2.14	Receptor GPS da <i>San Jose Navigation</i> 32 canais, 5 Hz com antena.	37
2.15	Diagrama de blocos do sistema de posição.	38
2.16	Estimativa da posição geodésica no sistema de coordenadas LLA.	38
2.17	Estimativa da posição geodésica no sistema de coordenadas LLA: Latitude λ (a), Longitude ϕ (b) e Altitude h (c).	39
2.18	Estimativa das velocidades no sistemas de coordenadas NED: v_N (a), v_E (b) e v_D (c).	39
3.1	Pioneer 3AT.	42
3.2	Geometria do robô móvel.	42
3.3	Forças de tração e resistivas do RMRD.	46
3.4	Diagrama de simulação do robô.	56
3.5	Distúrbios externos.	57
3.6	Trajatória do robô.	57
3.7	Torques do robô.	57
3.8	Trajatória no eixo X do robô.	58
3.9	Velocidades no eixo X do robô.	58
3.10	Trajatória no eixo Y do robô.	58
3.11	Velocidades no eixo Y do robô.	58
3.12	Deslocamento angular do robô.	58
3.13	Velocidades angulares do robô.	58
3.14	Sistema de coordenadas do corpo do helicóptero.	59

3.15 Dinâmica do helicóptero.	59
3.16 Ilustração dos planos HP e TPP.	60
3.17 <i>Flapping</i> lateral.	61
3.18 <i>Flapping</i> longitudinal.	61
3.19 Definição das forças no rotor principal.	61
3.20 Momentos e forças atuando sobre o helicóptero.	63
3.21 Diagrama de blocos do controlador cascata.	70
3.22 Trajetória do helicóptero.	71
3.23 Entradas do helicóptero.	71
3.24 Trajetória do helicóptero: no eixo X (a) , no eixo Y (b) e no eixo Z (c).	71
3.25 Guinada do helicóptero.	72
3.26 Rolagem e arfagem do helicóptero.	72
4.1 Esquemático do piloto automático desenvolvido sobre o kit de desenvolvimento do Rabbit 5400W.	74
4.2 Piloto automático desenvolvido sobre o kit de desenvolvimento do Rabbit 5400W.	75
4.3 Helimodelo com o piloto automático instalado.	76
4.4 Ambiente para aquisição de dados.	76
4.5 Plataforma de movimento de 3 graus de liberdade.	77
4.6 Base de movimento de 3 graus de liberdade: vista isométrica (a), vista isométrica com a base movimentada (b), vista frontal (c) e vista lateral (d).	78
4.7 Dimensões em mm da base de movimento de 3 graus de liberdade.	79
4.8 <i>Notebook</i> adquirindo dados do helicóptero.	79
4.9 Plataforma de movimento 3D : antiga (a) , nova (b)	80
4.10 Esquema elétrico do multiplexador controlador dos servos.	81
4.11 Circuito criado para o controle dos servos. À esquerda a face superior, e à direita a face inferior.	82

4.12	Placa de controle de servos.	82
4.13	Piloto automático com IMU, GPS, placa de controle de servos e receptor de rádio.	83
4.14	Esquema elétrico do circuito desenvolvido para leitura dos <i>encoders</i>	83
4.15	Placa de circuito impresso dupla face desenvolvida para leitura dos <i>encoders</i> . À esquerda a face superior, e à direita a face inferior.	84
4.16	Placa para leitura dos <i>encoders</i>	84
5.1	Representação de uma formação com três robôs.	88
5.2	Dígrafo sendo o robô 1 o líder.	92
5.3	Trajetoória da formação.	92
5.4	Trajetoória da formação com acompanhamento de trajetória.	92
5.5	Diagrama da formação seguindo helicóptero.	93
5.6	Dígrafo sendo o robô 1 o líder.	93
5.7	Formação com três RMRD acompanhando um RH mostrado em 3D.	94
5.8	Formação com três RMRD acompanhando um RH.	94
5.9	Trajetoória do helicóptero.	94
5.10	Trajetoória do RMRD 1.	94
5.11	Trajetoória do RMRD 2.	95
5.12	Trajetoória do RMRD 3.	95
5.13	Diagrama da formação seguindo helicóptero.	95
5.14	Dígrafo sendo o helicóptero o líder.	95
5.15	Formação com um helicóptero e dois RMRD mostrado em 3D.	96
5.16	Formação com três RMRD acompanhando um RH.	96
5.17	Trajetoória do helicóptero.	96
5.18	Trajetoória do RMRD 1.	96
5.19	Trajetoória do RMRD 2.	97
A.1	Ambiente para estimação dos erros dos sensores	113

A.2	Gráfico do desvio de Allan	114
B.1	Típico gráfico da variância de Allan [30].	118
A.1	Helimodelo TREX 450 XL (a) e Trans. Futaba 7CAP/7CHP (b) (figuras retiradas do manual do fabricante).	172
A.2	RCM5400W (figura retirada do manual do fabricante).	173
A.3	IMU 6DOF v4 (figura retirada do manual do fabricante)	174
A.4	Esquemático da placa controladora da IMU 6DOF v4 (a) e esquemático da placa de sensores da IMU 6DOF v4 (b) (esquemático retirado do manual do fabricante).	175
A.5	Menu de configuração da IMU 6-DOF v4 (figura retirada do manual do fabricante).	177
A.6	Lista de canais ativos da IMU 6-DOF v4 (figura retirada do manual do fabricante).	177
A.7	Receptor GPS da <i>San Jose Navigation</i> 32 canais, 5 Hz com antena (figuras retiradas do manual do fabricante).	178
A.8	<i>Layout</i> da placa do controlador de servos motores serial Pololu (figura retirada do manual do fabricante).	180

Lista de Tabelas

2.1	Comportamento do FKRR ($\mu \rightarrow +\infty$).	29
2.2	Parâmetros dos ruídos dos sensores inerciais.	30
2.3	Melhoria de desempenho sobre o filtro de Kalman estendido.	36
2.4	Estudo comparativo	38
3.1	Parâmetros do Pioneer 3-AT.	52

Sumário

1	Introdução	5
1.1	Estrutura do texto	10
2	Navegação Inercial	11
2.1	Estimativa da atitude	12
2.1.1	Modelo em espaço de estado	12
2.2	Estimativa da posição	17
2.2.1	Modelo em espaço de estado	17
2.3	Filtragem	20
2.3.1	Filtro de Kalman estendido	20
2.3.2	Filtro $\mathcal{H}_\infty$	21
2.3.3	Filtro BDU	21
2.3.4	Filtro de Kalman robusto recursivo	23
2.3.5	Exemplo numérico	27
2.4	Resultados experimentais	29
2.4.1	Estimativa de atitude	32
2.4.2	Estimativa de posição	36
3	Robô móvel com rodas e robô helicóptero	41
3.1	Modelo cinemático	42
3.2	Modelo dinâmico	45
3.3	Formulação do problema	48
3.3.1	Conversão de torque para comandos de velocidade	49
3.3.2	Parâmetros do robô	51

3.4	Determinação da trajetória	51
3.4.1	Trajetoária de referência	52
3.4.2	Trajetoária desejada	53
3.5	Controle robusto	54
3.5.1	Regulador robusto recursivo	54
3.6	Resultados simulados	56
3.6.1	Regulador robusto recursivo	56
3.7	Visão geral de um robô helicóptero	59
3.8	Dinâmica do corpo rígido	62
3.9	Modelo em espaço de estado identificado para voo pairado	64
3.10	Controle de posição e ângulo de proa de um robô helicóptero	67
3.11	Regulador linear quadrático	68
3.12	Controlador baseado em linearização por realimentação	68
3.13	Controlador PD	69
3.14	Lei de controle em cascata	70
3.15	Seguidor de pontos de passagem no sistema de coordenadas do corpo	70
3.16	Resultados simulados	71
4	Instrumentação e controle de um robô helicóptero	73
4.1	Projeto da placa do piloto automático e montagem do robô helicóptero	74
4.2	Ambiente para iteração com o helimodelo	76
4.3	Plataforma de movimento de 3 graus de liberdade	77
4.4	Reformulação da plataforma de movimento e desenvolvimento de placas de circuito impresso	77
5	Controle descentralizado e cooperativo de robôs heterogêneos em formação	85
5.1	Modelo	85
5.2	Definição de formação	87
5.3	Controle de formação	89
5.4	Acompanhamento de trajetória	91
5.5	Resultados simulados	92
5.5.1	Simulação 1	92
5.5.2	Simulação 2	93
5.5.3	Simulação 3	95
6	Conclusão	99
6.1	Trabalhos futuros	100

Apêndice	107
A Modelagem dos sensores inerciais	109
A.1 Modelo dos erros dos sensores inerciais	110
A.2 Estimativa dos valores dos parâmetros do modelo dos sensores	111
A.3 Ambiente para estimação dos erros dos sensores	112
B Variância de Allan	115
B.1 <i>Angle random walk</i>	118
B.2 <i>Instabilidade da polarização</i>	118
B.3 <i>Rate random walk</i>	119
B.4 <i>Rate ramp</i>	120
B.5 Ruído de quantização	120
B.6 Ruído exponencial correlacionado (<i>Markov</i>)	121
B.7 Ruído sinusoidal	122
C Quatérnios	123
C.1 Considerações geométricas	126
C.2 Operador quatérnio de rotação	127
C.3 Conversão entre quatérnios e ângulos de Euler	129
C.4 Derivada do quatérnio	130
C.5 Integração do quatérnio	133
D Mínimos quadrados regularizados	135
D.1 O problema de mínimos quadrados regularizados	135
E Código embarcado	149
Anexo	171
A Equipamentos	171
A.1 Helimodelo TREX 450 XL	171
A.2 RCM5400W	172
A.3 IMU 6DOF v4	174
A.4 GPS	178
A.5 Placa de controle de servos motores	180

CAPÍTULO 1

Introdução

Nas últimas décadas tem-se verificado um aumento significativo de pesquisas sobre grupos de robôs, em grande parte focados nas arquiteturas para o controle de robôs e em coordenação de vários robôs, veja [35, 64] e as referências contidas nelas. Neste trabalho, trataremos do problema de coordenação de robôs heterogêneos. Um grupo de robôs é caracterizado como heterogêneo se pelo menos um indivíduo do grupo for diferente dos demais em um ou mais dos seguintes atributos: mecânica, sensores, *hardware* de computação ou natureza do sistema de computação. Há várias razões para se usar heterogeneidade como característica de projeto, as principais são econômicas e tecnológicas, que surgem de restrições de aplicações complexas de multi-robôs.

Grupos de robôs heterogêneos são utilizados na realização de missões que requerem capacidades não encontradas nos membros da equipe individualmente. Nestas aplicações os membros da equipe devem decidir qual robô deve executar a tarefa, com base nas capacidades individuais de cada robô. Em configurações cooperativas, os robôs não precisam confiar unicamente em suas próprias percepções para obter informações sobre o ambiente, eles também podem receber informação sobre outros agentes. Por exemplo em [55] robôs com capacidade sensorial auxiliam na navegação de robôs com limitação de sensores. Em [51], um grupo de robôs heterogêneos foi utilizado para acompanhar um único ou múltiplos alvos móveis, onde cada robô obtinha a medida do alvo e combinava sua própria informação com a dos demais agentes para obter uma melhor estimativa da posição do alvo. Já em [64] foram realizados experimentos com um grupo

de robôs mecanicamente heterogêneos contendo um robô helicóptero e múltiplos robôs móveis com rodas, utilizando um sistema de arquitetura de controle baseado em comportamento.

A robótica aérea tem sido foco de intensa pesquisa há vários anos. Com a crescente disponibilidade de recursos computacionais de alto desempenho, avanços em tecnologias de transmissão de dados e de posicionamento global, os custos dos veículos aéreos não-tripulados (VANTs) têm diminuído. Estes aspectos também permitiram o desenvolvimento de veículos mais confiáveis e versáteis. Um dos principais tipos de VANTs é o dos robôs helicópteros (RHs).

O desenvolvimento de um RH não é algo trivial, pois requer conhecimentos interdisciplinares tais como engenharia espacial, engenharia elétrica, comunicações, ciência da computação, controle de sistema, inteligência artificial, operações em sistema em tempo real, entre outros, veja [9], [34], [36] e [59]. O RH pode realizar voos pairados, ou seja com velocidade zero, e a altas velocidades, e sobrevoar baixas e altas altitudes. Sua dinâmica é não-linear e está sujeito a perturbações externas como rajadas de vento. Além de ser um veículo com potencial risco ao operador, visto que se trata de um dispositivo mecânico controlado eletronicamente com altas velocidades de rotação em suas hélices.

Muitos trabalhos vêm sendo realizados utilizando RHs. Em [34] foram projetados controladores não lineares combinados com adaptação não linear de saídas e estabilização robusta para o controle do movimento vertical de um RH. Em [67] foram projetados controladores robustos baseados em critérios $\mathcal{H}_\infty$, síntese μ e sistemas lineares a parâmetros variantes (LPV) para um RH. Em [9] foi apresentada uma abordagem de controle de voo para um modelo dinâmico não linear de um helicóptero com seis graus de liberdade utilizando equação de Riccati dependente do estado, resultados experimentais foram mostrados utilizando um helicóptero X-Cell-90 e um Yamaha R-Max. Em [53] foi proposta uma metodologia na qual controladores baseados em redes neurais são envolvidos em uma simulação usando um modelo dinâmico qualitativamente similar a um helicóptero. Em [52] explorou-se alguns aspectos de um enxame autônomo de MVAs (micro ou miniaturas de veículos aéreos), como capacidade de explorar a troca de informação entre os indivíduos do enxame. Em [59] apresentou-se o projeto de um controle ótimo de trajetória para aterrissagem de um helicóptero sobre um alvo em movimento, simulações foram apresentadas. Outros exemplos podem ser vistos em [8, 10, 26, 28, 49, 69, 70].

Considerando que robôs móveis com rodas (RMRs) podem estar sujeitos a perturbações externas, como desníveis de superfície por onde o robô circulam, escorregamento das rodas, colisão com obstáculos ou com outros robôs e incertezas paramétricas, o controle utilizado deve

ser robusto o suficiente para a realização de uma tarefa pré-estabelecida. O critério $\mathcal{H}_\infty$ não linear vem sendo usado em controle de sistemas robóticos para solução deste tipo de problema, veja [16, 17, 21, 63] e as referências contidas nelas. Em [31] estratégias de controle $\mathcal{H}_\infty$ não linear baseadas em modelos dinâmicos com representação do robô em espaço de estado LPV foram utilizadas para controle de um robô móvel com rodas (RMR). Recentemente, em [15] foi desenvolvido um regulador robusto recursivo baseado em funções penalidades e teoria de jogos. Este regulador garante estabilidade e robustez do sistema sem a necessidade de ajuste de nenhum parâmetro auxiliar. Este regulador robusto recursivo ainda não foi utilizado para controlar RMRs.

Métodos para estimar variáveis dinâmicas têm sido fundamentais para resolver problemas relacionados com robótica aérea e terrestre. Uma das mais consagradas metodologias utilizadas para estimar variáveis dinâmicas foi desenvolvida na década de 1960. Trata-se do consagrado filtro de Kalman. Este tipo de filtro tem sido usado em uma infinidade de sistemas de medida. Um sistema bastante importante para a robótica e para a indústria aeronáutica é a unidade de medida inercial. Este tipo de unidade busca medir os ângulos relacionados com os três eixos fundamentais que definem a posição de um corpo rígido. As unidades de medida inercial utilizadas por exemplo, nos aviões, se baseiam em estimativas do tipo Kalman e são as melhores e as mais caras, veja por exemplo [24], [23] e [57].

Filtros de Kalman têm como base modelos matemáticos que descrevem um processo dinâmico. Foram deduzidos originalmente sob a hipótese de que todos os parâmetros do modelo em espaço de estado não estão sujeitos a incertezas. Esta hipótese garante uma estimativa ótima das variáveis do sistema.

Porém, quando o modelo considerado no processo de filtragem é incerto, esta premissa central do filtro de Kalman é violada. Neste caso o desempenho se deteriora, em alguns casos de maneira considerável. Este tipo de problema motiva o uso de métodos de estimativa robusta para limitar a degradação do desempenho de filtros ótimos, veja por exemplo [61], [62], [4], [32] e [33]. A melhoria das estimativas das unidades de medida inerciais, com um custo menor aos encontrados no mercado, pode atender a indústria de veículos tripulados e não-tripulados (aviões, helicópteros, barcos, carros, etc) e podem ser usadas nas mais diversas aplicações robóticas.

Assim, neste trabalho estamos propondo a coordenação de robôs heterogêneos, consistindo de um robô helicóptero e múltiplos robôs móveis com rodas, de modo que estes sigam um líder, como mostrado na Figura 1.1.

Figura 1.1: Coordenação de robôs heterogêneos.

Para atingirmos este objetivo desenvolveu-se primeiramente um sistema de referência de atitude, orientação e posição robusto de robôs, descritos pelos diagramas das figuras 1.2 e 1.3, respectivamente. Pois para a realização do controle de trajetória de robôs é indispensável a informação de localização do veículo.

A abordagem de estimativa de atitude, orientação e posição utiliza sinais de giroscópios, acelerômetros, magnetômetros e um receptor GPS (sistema de posicionamento global, do inglês *Global Positioning System*). Estes sinais são fundidos através do filtro de Kalman ou do filtro do robusto utilizando equações da derivada do quatérnio e do INS (Sistema de navegação inercial, do inglês *Inertial Navigation System*). Os parâmetros dos sinais dos sensores utilizados pelos filtros são estimados através da variância de Allan, um programa em MATLAB foi desenvolvido para a identificação destes parâmetros.

Figura 1.2: Diagrama de blocos do sistema de atitude e orientação.

Para o sistema de referência de atitude e orientação utilizou-se uma derivação preditora-

Figura 1.3: Diagrama de blocos do sistema de posição.

corretora estendida do filtro de Kalman robusto recursivo (FKRR) proposto em [66], esta derivação foi necessária pois as equações do sistema de referência de atitude e orientação é não linear. A extensão e a prova matemática do FKRR são apresentadas nesta tese. E para o sistema de referência de posição utilizou-se o filtro BDU (do inglês *Bounded Data Uncertainties*) proposto em [61]. O modelo em espaço de estado do sistema de referência de atitude, orientação e posição são modelados de forma a considerar a incerteza no sinal de cada sensor.

Um estudo comparativo entre o filtro de Kalman, o FKRR, o filtro BDU e o filtro $\mathcal{H}_\infty$ proposto em [1] foi realizado para mostrar o desempenho de cada filtro, dados de uma unidade de medida inercial e de um receptor GPS foram utilizados.

Vale ressaltar que a principal contribuição desta tese está no desenvolvimento do sistema de referência de atitude, orientação e posição robusto de robôs.

Após trabalharmos com o problema de localização de robôs, trabalhou-se com o controle do robô móvel com rodas deslizantes e do robô helicóptero.

Para realizar o controle robusto de acompanhamento de trajetória do robô móvel com rodas deslizantes utilizou-se o regulador robusto recursivo desenvolvido em [15]. O modelo em espaço de estado do robô foi escrito na formulação quase linear a parâmetros variantes utilizando a modelagem cinemática e dinâmica apresentadas em [13, 39]. As trajetórias de referência entre os pontos de passagem foram geradas por um polinômio do quinto grau. Resultados simulados do robô móvel com rodas deslizantes considerando perturbações externas são apresentados para mostrar a eficiência da metodologia proposta.

Para realizar o controle de posição e ângulo de proa do robô helicóptero, utilizou-se a metodologia apresentada em [3] da combinação em cascata do regulador linear quadrático para estabilizar os pólos do sistema, de um controlador baseado em linearização por realimentação para desacoplar os pares de entrada/saída e, finalmente, do controlador PD para assegurar o

acompanhamento de trajetória. Resultados simulados do robô helicóptero realizando acompanhamento de trajetória foram obtidos utilizando os modelos em espaço de estado identificado para voo pairado mostrado em [18, 50].

Para a realização dos testes de controle do robô helicóptero, foi desenvolvida a instrumentação de um mini helimodelo e de uma base de movimento para acoplar o helimodelo. A instrumentação do mini helimodelo e a base de movimento se encontram em desenvolvimento como pode ser visto nesta tese. O sistema do mini robô helicóptero consiste de um piloto automático baseado em um microprocessador com conectividade *Wi-Fi*, onde se encontra o código embarcado do sistema de referência de atitude, orientação e posição; uma unidade de medida inercial; um receptor GPS e uma placa de controle de servos motores. E finalmente, foi desenvolvida a coordenação de robôs heterogêneos, consistindo de um robô helicóptero e múltiplos robôs móveis com rodas de modo que estes sigam um líder. Resultados simulados da coordenação de robôs heterogêneos foram obtidos utilizando o controle de formação para gerar trajetórias de referência para os modelos dinâmicos dos robôs móveis com rodas deslizantes e do robô helicóptero.

1.1 Estrutura do texto

O texto está organizado da seguinte maneira: No Capítulo 2 são apresentados os modelos em espaço de estado da estimativa de atitude, orientação e posição, os filtros de Kalman robustos, exemplos numéricos do filtro de Kalman robusto recursivo, resultados experimentais e estudo comparativo entre os filtros utilizados para a estimativa de atitude, orientação e posição do robô. No Capítulo 3 são apresentados o modelo cinemático e dinâmico do robô móvel com rodas deslizantes, a formulação quase-LPV do modelo do robô, a determinação da trajetória, o regulador robusto recursivo utilizado, e resultados simulados do sistema de controle do robô. No Capítulo ?? é apresentado o modelo do helicóptero. No Capítulo 3.10 é abordado o controle de posição e ângulo de proa de um robô helicóptero. No Capítulo 4 apresenta-se a instrumentação e controle de um robô helicóptero. No Capítulo 5, o controle descentralizado e cooperativo de robôs heterogêneos em formação é desenvolvido. E no Capítulo 6 conclusões e trabalhos futuros são apresentados.

CAPÍTULO 2

Navegação Inercial

A navegação robótica compreende a localização do veículo em tempo real, como também o planejamento e a execução de manobras necessárias para mover-se entre objetivos [24]. Tanto veículos tripulados e não-tripulados envolvidos na navegação necessitam da determinação do estado do veículo (posição, velocidade e atitude), necessário para o controle automático, planejamento em tempo real, registro de dados, localização simultânea e mapeamento.

Este capítulo trata das estimativas de atitude, orientação, posição de um robô, necessárias para a realização de um acompanhamento de trajetória preciso. Essas estimativas são obtidas pelo filtro de Kalman estendido que combina sinais de sensores inerciais e de um receptor GPS.

O filtro de Kalman estendido fornece resultados excelentes quando se utiliza sensores precisos, [23], [24] e [57]. No entanto, algumas aplicações importantes têm utilizado sensores de baixo custo no desenvolvimento de sistemas de navegação. Veículos não-tripulados, são exemplos onde esses sistemas de navegação têm sido utilizados, [7, 14, 36, 58].

Sensores de baixo custo significam, geralmente, aumento de incertezas e ruídos [30, 73]. Essas incertezas violam a premissa central do filtro de Kalman, em que o modelo em espaço de estado é ótimo, [61]. Este tipo de problema motiva o uso da estimativa robusta para limitar a degradação do desempenho de filtros ótimos. Em [4, 32, 33, 61] foram desenvolvidos métodos para estimativa em espaço de estado quando os parâmetros do modelo linear estão sujeitos a

incertezas. Estas técnicas têm sido aplicadas em problemas práticos e os resultados obtidos têm sido promissores. Por exemplo, os filtros robustos propostos em [61] foram utilizados em [20] para estimar o acompanhamento de trajetória da ponta de um dedo em uma interface homem máquina.

Levando em conta os resultados mencionados, este capítulo propõe um algoritmo para resolver o problema de estimativa em espaço de estados, sujeito a incertezas, do estado de um veículo baseado em uma abordagem robusta. Resultados baseados em dados experimentais, e um estudo comparativo entre os filtros considerados são apresentados na Seção 2.4 para mostrar as vantagens do filtro robusto proposto.

Este capítulo está organizado da seguinte forma, a Seção 2.1 desenvolve o modelo cinemático de um corpo rígido usando quatérnios e seu modelo em espaço de estado. A Seção 2.2 desenvolve o modelo em espaço de estado da posição. Na Seção 2.3 são apresentados os filtros. E na Seção 2.4 analisam-se os resultados do estudo comparativo feito.

2.1 Estimativa da atitude

A determinação da atitude compreende as estimativas dos ângulos de rolagem ϕ , arfagem θ e guinada ψ de um corpo rígido através da combinação de sinais de sensores inerciais de uma IMU (unidade de medida inercial, do inglês *inertial measurement unit*). A abordagem utilizada nesta seção considera os sinais dos giroscópios em três eixos ortogonais, que fornecem velocidades angulares; acelerômetros em três eixos ortogonais, que fornecem acelerações lineares; e magnetômetros em três eixos ortogonais, que fornecem medidas de campo magnético, com as quais é possível obter a orientação do veículo.

2.1.1 Modelo em espaço de estado

Os modelos dos sensores inerciais consideram que todas as medidas estão sujeitas a incertezas. Neste contexto, as velocidades angulares, as acelerações da gravidade, e as medidas do magnetômetro, obtidas de uma IMU fixada em um corpo rígido, serão usadas na definição do modelo cinemático, também sujeito a incertezas. A velocidade angular do corpo rígido $\boldsymbol{\omega}$ pode ser modelada conforme mostrado no Apêndice A.1

$$\boldsymbol{\omega} = \boldsymbol{\omega}_g + \delta\boldsymbol{\omega}_g - \mathbf{b}_g - \mathbf{w}_g, \quad (2.1)$$

sendo $\boldsymbol{\omega}_g$ a medida da velocidade angular obtida pelo giroscópio, $\delta\boldsymbol{\omega}_g$ o termo incerto de $\boldsymbol{\omega}_g$, $\mathbf{b}_g$ a polarização do giroscópio e $\mathbf{w}_g$ o ruído Gaussiano branco das medidas dos giroscópios. A polarização do giroscópio $\mathbf{b}_g$ é definida em termos do processo Gauss-Markov

$$\dot{\mathbf{b}}_g = -\frac{1}{\tau_g}\mathbf{b}_g + \mathbf{w}_{b_g}, \quad (2.2)$$

sendo $\mathbf{w}_{b_g}$ o ruído Gaussiano branco das polarizações e τ_g o tempo correlacionado do processo de Gauss-Markov. A aceleração real do corpo rígido $\mathbf{a}$ é dada por

$$\mathbf{a} = \mathbf{a}_a + \delta\mathbf{a}_a - \mathbf{b}_a - \mathbf{w}_a, \quad (2.3)$$

sendo $\mathbf{a}_a$ a medida da aceleração obtida pelo acelerômetro, $\delta\mathbf{a}_a$ o termo incerto de $\mathbf{a}_a$, $\mathbf{b}_a$ a polarização do acelerômetro e $\mathbf{w}_a$ o ruído Gaussiano branco das medidas dos acelerômetros. A polarização do acelerômetro $\mathbf{b}_a$ é definida em termos do processo Gauss-Markov

$$\dot{\mathbf{b}}_a = -\frac{1}{\tau_a}\mathbf{b}_a + \mathbf{w}_{b_a}, \quad (2.4)$$

sendo $\mathbf{w}_{b_a}$ o ruído Gaussiano branco das polarizações e τ_a o tempo correlacionado do processo de Gauss-Markov. A medida do magnetômetro $\mathbf{m}_m$ é modelada como

$$\mathbf{m}_m + \delta\mathbf{m}_m = \mathbf{m} + \mathbf{m}_b + \mathbf{w}_m \quad (2.5)$$

sendo $\delta\mathbf{m}_m$ o termo incerto de $\mathbf{m}_m$, $\mathbf{m}$ o vetor de campo magnético da Terra, $\mathbf{m}_b$ o campo magnético gerado pelo veículo e $\mathbf{w}_m$ o ruído Gaussiano branco. É assumido que o magnetômetro é isolado do campo magnético do veículo, tal que o termo $\mathbf{m}_b$ pode ser assumido como sendo nulo.

O papel que as incertezas $\delta\boldsymbol{\omega}$, $\delta\mathbf{a}_a$, e $\delta\mathbf{m}_m$ desempenham na filtragem será descrito na Seção 2.3.

O modelo cinemático usado para estimar a atitude do corpo rígido será baseado em quatérnios. Quatérnios têm sido utilizados em uma grande quantidade de referências para descrever o movimento de um corpo rígido, veja [41] e suas referências internas, e o Apêndice C para maiores detalhes sobre quatérnios. Um aspecto interessante desta abordagem é a possibilidade de representar a atitude do corpo rígido livre de singularidades.

Por convenção, o quatérnio é representado como

$$q = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} \mathbf{u}, \quad q \in \mathbb{T}, \quad (2.6)$$

sendo $\mathbf{u}$ um vetor unitário, θ uma rotação sobre este eixo e $\mathbb{T} = \{q \mid q^T q = 1, q = [q_0 \ \mathbf{q}^T]^T, q_0 \in \mathbb{R}, \mathbf{q} \in \mathbb{R}^{3 \times 1}\}$. O produto de quatérnios é representado por $\otimes$ e pode ser dado por

$$p \otimes r = \begin{bmatrix} p_0 r_0 - \mathbf{p} \cdot \mathbf{r} \\ p_0 \mathbf{r} + r_0 \mathbf{p} + \mathbf{p} \times \mathbf{r} \end{bmatrix}. \quad (2.7)$$

Em relação à atitude, quatérnios representam uma rotação do sistema de coordenadas de navegação inercial, $\mathcal{I}$, ao sistema de coordenadas do corpo $\mathcal{B}$, veja Figura 2.1. Neste trabalho o sistema de coordenadas NED, do inglês *North, East and Down*, foi adotado como sistema de coordenadas de navegação inercial.

Figura 2.1: Sistema de referência de atitude.

Se $\mathbf{s}$ é um vetor expresso no sistema de coordenadas $\mathcal{I}$, então suas coordenadas em $\mathcal{B}$ são dadas por

$$\mathbf{b} = q \otimes \mathbf{s} \otimes q^{-1}, \quad (2.8)$$

sendo $\mathbf{b} = [0 \ \mathbf{b}^T]^T$, $\mathbf{s} = [0 \ \mathbf{s}^T]^T$ e $q^{-1} = [q_0, -\mathbf{q}^T]^T$. Então, (2.8) pode ser escrita como

$$\mathbf{b} = \mathcal{R}(q) \mathbf{s}, \quad (2.9)$$

sendo $\mathcal{R}(q)$ a matriz de rotação dos ângulos de Euler (ψ, θ, ϕ) , com convenção (zyx) , definida

como

$$\mathcal{R}(q) = (q_0^2 - \mathbf{q}^T \mathbf{q}) I_{3 \times 3} + 2(\mathbf{q} \mathbf{q}^T - q_0 [\mathbf{q}^\times]). \quad (2.10)$$

A derivada do quatérnio pode ser escrita matricialmente como

$$\dot{q} = \frac{1}{2} q \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2} \Omega(\boldsymbol{\omega}) q = \frac{1}{2} \Xi(q) \boldsymbol{\omega}, \quad (2.11)$$

sendo

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -[\boldsymbol{\omega}^\times] \end{bmatrix}, \quad \Xi(q) = \begin{bmatrix} -\mathbf{q}^T \\ I_{3 \times 3} q_0 + [\mathbf{q}^\times] \end{bmatrix},$$

$\boldsymbol{\omega}$ o vetor velocidade angular no sistema de coordenadas fixo do corpo, $[\boldsymbol{\omega}^\times]$ e $[\mathbf{q}^\times]$ as matrizes representativas do produto vetorial

$$\mathcal{E}^\times = \begin{bmatrix} 0 & -\mathcal{E}_3 & \mathcal{E}_2 \\ \mathcal{E}_3 & 0 & -\mathcal{E} \\ -\mathcal{E}_2 & \mathcal{E} & 0 \end{bmatrix} \quad (2.12)$$

e $I_{3 \times 3} \in \mathbb{R}^{3 \times 3}$ a matriz identidade.

Substituindo (2.1) em (2.11), tem-se que

$$\dot{q} = \frac{1}{2} \Xi(q) (\boldsymbol{\omega}_g + \delta \boldsymbol{\omega}_g - \mathbf{b}_g) - \frac{1}{2} \Xi(q) \mathbf{w}_g, \quad (2.13)$$

$$= \frac{1}{2} \Omega(\boldsymbol{\omega}_g) q + \frac{1}{2} \delta \Omega(\delta \boldsymbol{\omega}_g) q - \frac{1}{2} \Xi(q) \mathbf{b}_g - \frac{1}{2} \Xi(q) \mathbf{w}_g. \quad (2.14)$$

Considerando que $\Xi(q)$ tem um termo incerto $\delta \Xi(\delta q)$ devido a influência das incertezas dos sensores, a Equação (2.14) pode ser escrita como

$$\dot{q} = \frac{1}{2} \Omega(\boldsymbol{\omega}_g) q + \frac{1}{2} \delta \Omega(\delta \boldsymbol{\omega}_g) q - \frac{1}{2} \Xi(q) \mathbf{b}_g - \frac{1}{2} \Xi(q) \mathbf{w}_g - \frac{1}{2} \delta \Xi(\delta q) \mathbf{b}_g - \frac{1}{2} \delta \Xi(\delta q) \mathbf{w}_g. \quad (2.15)$$

Combinando as equações (2.2) e (2.15) a formulação em espaço de estado para o sistema

sujeito a incertezas é dada por

$$\begin{aligned}\dot{\mathbf{x}} &= (A + \delta A)\mathbf{x} + (B + \delta B)\mathbf{w}, \\ \mathbf{z} &= \mathbf{h}(\mathbf{x}) + \delta \mathbf{h}(\mathbf{x}) + \mathbf{v},\end{aligned}\tag{2.16}$$

sendo

$$\begin{aligned}A &= \begin{bmatrix} \frac{1}{2}\Omega(\boldsymbol{\omega}_g) & -\frac{1}{2}\Xi(q) \\ 0_{3 \times 4} & -\frac{1}{\tau_g}I_{3 \times 3} \end{bmatrix}, B = \begin{bmatrix} -\frac{1}{2}\Xi(q) & 0_{4 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \\ \delta A &= \begin{bmatrix} \frac{1}{2}\delta\Omega(\delta\boldsymbol{\omega}_g) & -\frac{1}{2}\delta\Xi(\delta q) \\ 0_{3 \times 4} & 0_{3 \times 3} \end{bmatrix}, \delta B = \begin{bmatrix} -\frac{1}{2}\delta\Xi(\delta q) & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix},\end{aligned}$$

$\mathbf{w} = [\mathbf{w}_g^T \ \mathbf{w}_{b_g}^T]^T \in \mathbb{R}^{6 \times 1}$ o processo Gaussiano de média zero com matriz de covariância Q , $\mathbf{x} = [q^T \ \mathbf{b}_g^T]^T \in \mathbb{R}^{7 \times 1}$ o estado que descreve o sistema de atitude, $\mathbf{h}(\mathbf{x}) = [\mathbf{m}^T \ \mathbf{a}^T]^T \in \mathbb{R}^{6 \times 1}$ a medida de saída, $\delta \mathbf{h}(\mathbf{x}) = [-\delta \mathbf{m}_m^T \ -\delta \mathbf{a}_a^T]^T \in \mathbb{R}^{6 \times 1}$, $\mathbf{v} = [\mathbf{w}_m^T \ \mathbf{w}_a^T]^T \in \mathbb{R}^{6 \times 1}$ o processo Gaussiano de média zero com matriz de covariância R , $\mathbf{m} \in \mathbb{R}^{3 \times 1}$ dado por

$$\mathbf{m} = \mathcal{R}(q)\mathbf{m}_e = \begin{bmatrix} m_e(q_0^2 + q_1^2 - q_2^2 - q_3^2) \\ 2m_e(q_1q_2 - q_0q_3) \\ 2m_e(q_1q_3 + q_0q_2) \end{bmatrix},\tag{2.17}$$

sendo $\mathbf{m}_e = [m_e \ 0 \ 0]^T$ e m_e a magnitude do campo magnético da Terra, e $\mathbf{a} \in \mathbb{R}^{3 \times 1}$ calculado como

$$\mathbf{a} = \mathcal{R}(q)\mathbf{g}_e = \begin{bmatrix} 2g_e(qq_3 - q_0q_2) \\ 2g_e(q_2q_3 + q_0q) \\ g_e(q_0^2 - q^2 - q_2^2 + q_3^2) \end{bmatrix},\tag{2.18}$$

sendo $\mathbf{g}_e = [0 \ 0 \ g_e]^T$ e g_e a constante de gravidade da Terra.

Os filtros de Kalman apresentados na Seção 2.3 são baseados em sistemas discretos no tempo. Nesse sentido (2.16) é discretizada considerando um tempo de amostragem T

$$\begin{aligned}\mathbf{x}_{k+1} &= (F_k + \delta F_k)\mathbf{x}_k + (G_k + \delta G_k)\mathbf{w}_k, \\ \mathbf{z}_k &= (H_k + \delta H_k)\mathbf{x}_k + \mathbf{v}_k,\end{aligned}\tag{2.19}$$

sendo

$$\begin{aligned} F_k &\simeq I + AT, \quad \delta F_k \simeq \delta AT, \\ G_k &\simeq BT^{\frac{1}{2}}, \quad \delta G_k \simeq \delta BT^{\frac{1}{2}}, \\ H_k &= \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}, \quad \delta H_k = \frac{\partial \delta \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}. \end{aligned}$$

2.2 Estimativa da posição

Estimativa de posição com receptores GPS (Sistema de posicionamento global, do inglês *Global Positioning System*) podem fornecer informações de posição e velocidade de plataformas móveis com precisão em tarefas de agrimensura. Porém, a utilização do receptor GPS em tarefas de navegação não é uma solução satisfatória [6, 56]. Visto que os sinais do GPS estão sujeitos a interferências, podem ser facilmente obstruídos, suas medidas são transmitidas a baixas frequências e perdem precisão com a perda de satélites. Assim, torna-se interessante integrar o INS (Sistema de navegação inercial, do inglês *Inertial Navigation System*) ao receptor GPS para uma melhora na autonomia da navegação.

O INS é baseado em medidas de acelerações lineares e velocidades angulares do veículo obtidas através de acelerômetros e giroscópios, respectivamente. Estas medidas por sua vez não são sensíveis a interferências ou obstruções e permitem uma alta taxa de amostragem. A posição e atitude do veículo no INS são obtidas através de uma dupla integração das acelerações lineares e uma simples integração das velocidades angulares, respectivamente. Contudo, sua desvantagem está no crescimento do erro de estimativa de posição e atitude devido à integração numérica das medidas dos sensores e seus erros inerentes (ruídos e polarizações).

Assim, combinando um receptor GPS e o INS, o INS fornece o posicionamento no caso em que os dados do receptor GPS se encontram indisponíveis ou são ruins, e o GPS manterá os erros de navegação em uma certa faixa especificada.

2.2.1 Modelo em espaço de estado

Conforme mostrado em [23], se as velocidades no sistema de coordenadas NED (*North, East e Down*), $\mathbf{v} = [v_N \ v_E \ v_D]^T$ são conhecidas e as posições geodésicas no sistema de coordenadas LLA (latitude, longitude e altitude), $\mathbf{p} = [\lambda \ \phi \ h]^T$ são as desejadas, então o vetor de variação

geodésica LLA é relacionado com a velocidade NED da seguinte forma:

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\phi} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_{\lambda}+h} & 0 & 0 \\ 0 & \frac{1}{(R_{\phi}+h)\cos\lambda} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix}, \quad (2.20)$$

sendo R_{λ} o raio de curvatura no meridiano em uma latitude dada,

$$R_{\lambda} = \frac{a(1-e^2)}{(1-e^2\sin^2\lambda)^{1.5}}, \quad (2.21)$$

R_{ϕ} o raio transversal da curvatura,

$$R_{\phi} = \frac{a}{(1-e^2\sin^2\lambda)^{0.5}}, \quad (2.22)$$

$a = 6378137.0$ m o raio equatorial, e $e = 0.0818$ a excentricidade elíptica da Terra.

A equação da aceleração NED é dada da seguinte forma:

$$\mathbf{a}_{NED} = \begin{bmatrix} a_N \\ a_E \\ a_D \end{bmatrix} = \mathcal{R}^T(q)\mathbf{a} \quad (2.23)$$

sendo $\mathbf{a}$, dada por (2.3), as medidas dos acelerômetros no sistema de coordenadas do corpo rígido e $\mathcal{R}(q)$ a matriz de cossenos diretores em termos do quatérnio q entre o sistema de coordenadas $\mathcal{I}$ e do corpo rígido $\mathcal{B}$.

A equação da variação da velocidade NED como mostrado em [23] é dada por

$$\begin{bmatrix} \dot{v}_N \\ \dot{v}_E \\ \dot{v}_D \end{bmatrix} = \begin{bmatrix} a_N \\ a_E \\ a_D \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g_e \end{bmatrix} + \begin{bmatrix} -\frac{v_E^2 \sin\lambda}{(R_{\phi}+h)\cos\lambda} + \frac{v_N v_D}{R_{\lambda}+h} \\ \frac{v_E v_N \sin\lambda}{(R_{\phi}+h)\cos\lambda} + \frac{v_E v_D}{(R_{\phi}+h)} \\ -\frac{v_E^2}{R_{\phi}+h} - \frac{v_N^2}{R_{\lambda}+h} \end{bmatrix}, \quad (2.24)$$

sendo g_e a constante de gravidade da Terra.

Substituindo (2.3) em (2.23) e em seguida em (2.24), tem-se

$$\begin{bmatrix} \dot{v}_N \\ \dot{v}_E \\ \dot{v}_D \end{bmatrix} = \mathcal{R}^T(q)[\mathbf{a}_a + \delta \mathbf{a}_a - \mathbf{b}_a - \mathbf{w}_a] - \begin{bmatrix} 0 \\ 0 \\ g_e \end{bmatrix} + \begin{bmatrix} -\frac{v_E^2 \sin \lambda}{(R_\phi + h) \cos \lambda} + \frac{v_N v_D}{R_\lambda + h} \\ \frac{v_E v_N \sin \lambda}{(R_\phi + h) \cos \lambda} + \frac{v_E v_D}{(R_\phi + h)} \\ -\frac{v_E^2}{R_\phi + h} - \frac{v_N^2}{R_\lambda + h} \end{bmatrix}. \quad (2.25)$$

Define-se o estado do sistema dinâmico como $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}_{NED}^T \ \mathbf{b}_a^T]^T \in \mathbb{R}^{9 \times 1}$, $\mathbf{w} = \mathbf{w}_a^T \in \mathbb{R}^{3 \times 1}$ como o ruído branco gaussiano e a equação de medida observável do GPS como $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \in \mathbb{R}^{6 \times 1}$, sendo $\mathbf{h}(\mathbf{x}) = [\mathbf{p}^T \ \mathbf{v}_{NED}^T]^T$, $\mathbf{v}$ o processo gaussiano de média zero não correlacionado. Linearizando (2.20) e (2.25) em torno de de um ponto de equilíbrio, tem-se a dinâmica linearizada

$$\begin{aligned} \dot{\mathbf{x}} &= (A + \delta A)\mathbf{x} + (B + \delta B)\mathbf{w}, \\ \mathbf{z} &= H\mathbf{x} + \mathbf{v}, \end{aligned} \quad (2.26)$$

sendo

$$\begin{aligned} A &= \begin{bmatrix} A_{11} & A_{12} & 0_{3 \times 3} \\ A_{21} & A_{22} & -R^T(q) \\ 0_{3 \times 3} & 0_{3 \times 3} & -\frac{1}{\tau_a} I_{3 \times 3} \end{bmatrix}, \quad A_{11} = \begin{bmatrix} 0 & 0 & -\frac{v_N}{(R_\lambda + h)^2} \\ \frac{v_E \sin \lambda}{(R_\phi + h)(\cos \lambda)^2} & 0 & -\frac{v_E}{(R_\phi + h)^2 \cos \lambda} \\ 0 & 0 & 0 \end{bmatrix}, \\ A_{12} &= \begin{bmatrix} (R_\lambda + h)^{-1} & 0 & 0 \\ 0 & ((R_\phi + h) \cos \lambda)^{-1} & 0 \\ 0 & 0 & -1 \end{bmatrix}, \\ A_{21} &= \begin{bmatrix} -\frac{v_E^2}{R_\phi + h} - \frac{v_E^2 (\sin \lambda)^2}{(R_\phi + h)(\cos \lambda)^2} & 0 & \frac{v_E^2 \sin \lambda}{(R_\phi + h)^2 \cos \lambda} - \frac{v_N v_D}{(R_\lambda + h)^2} \\ \frac{v_E v_N}{R_\phi + h} + \frac{v_E v_N (\sin \lambda)^2}{(R_\phi + h)(\cos \lambda)^2} & 0 & -\frac{v_E v_N \sin \lambda}{(R_\phi + h)^2 \cos \lambda} - \frac{v_E v_D}{(R_\phi + h)^2} \\ 0 & 0 & \frac{v_E^2}{(R_\phi + h)^2} + \frac{v_N^2}{(R_\lambda + h)^2} \end{bmatrix}, \\ A_{22} &= \begin{bmatrix} \frac{v_D}{R_\lambda + h} & -2 \frac{v_E \sin \lambda}{(R_\phi + h) \cos \lambda} & \frac{v_N}{R_\lambda + h} \\ \frac{v_E \sin \lambda}{(R_\phi + h) \cos \lambda} & \frac{v_N \sin \lambda}{(R_\phi + h) \cos \lambda} + \frac{v_D}{R_\phi + h} & \frac{v_E}{R_\phi + h} \\ -2 \frac{v_N}{R_\phi + h} & -2 \frac{v_E}{R_\phi + h} & 0 \end{bmatrix}, \\ B &= \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ -R^T(q) & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \text{ e } H = \begin{bmatrix} I_{6 \times 6} & 0_{6 \times 3} \end{bmatrix}. \end{aligned}$$

Discretizando (2.26), considerando o tempo de amostragem T , temos:

$$\begin{aligned}\mathbf{x}_{k+1} &= (F_k + \delta F_k)\mathbf{x}_k + (G_k + \delta G_k)\mathbf{w}_k, \\ \mathbf{z}_k &= H_k\mathbf{x}_k + \mathbf{v}_k,\end{aligned}\tag{2.27}$$

sendo

$$\begin{aligned}F_k &\simeq I + AT, \quad \delta F_k \simeq \delta AT, \\ G_k &\simeq BT^{\frac{1}{2}}, \quad \delta G_k \simeq \delta BT^{\frac{1}{2}}, \quad H_k = H.\end{aligned}$$

2.3 Filtragem

Esta seção apresenta o filtro de Kalman estendido, o filtro $\mathcal{H}_\infty$ proposto em [62], o filtro BDU (*Bounded Data Uncertainties*) proposto em [61], e o filtro de Kalman robusto recursivo (FKRR) baseado em [66] para estimar atitude, posição e velocidade de um veículo sujeito a incertezas nos parâmetros das matrizes. Também é considerado que o sistema está sujeito a distúrbios limitados. De forma a apresentar um estudo comparativo destes filtros, é importante enfatizar as principais propriedades de cada filtro que definem seu desempenho. O filtro de Kalman estendido não leva em conta as incertezas nas matrizes do sistema (2.16). O filtro $\mathcal{H}_\infty$ apresenta um nível de robustez no sentido de rejeição de distúrbios, o qual depende de um parâmetro γ que precisa ser ajustado *off-line*. O filtro não foi deduzido para considerar de forma exata as incertezas de (2.16). Os filtros BDU e o FKRR são definidos para rejeitar distúrbios e incertezas nos parâmetros das matrizes (2.16), porém o filtro BDU não considera a incerteza da matriz de observação.

2.3.1 Filtro de Kalman estendido

O filtro de Kalman estendido clássico, [11], pode ser implementado pelo seguinte algoritmo

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= F_k \hat{\mathbf{x}}_{k|k}, \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + G_k Q_k G_k^T, \\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + K_{k+1} [\mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k})], \\ P_{k+1|k+1} &= [I - K_{k+1} H_{k+1}] P_{k+1|k}, \\ K_{k+1} &= P_{k+1|k} H_{k+1}^T [H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}]^{-1},\end{aligned}$$

sendo $\hat{\mathbf{x}}_{k+1|k}$ a estimativa preditora e $\hat{\mathbf{x}}_{k+1|k+1}$ a estimativa filtrada.

2.3.2 Filtro $\mathcal{H}_\infty$

O objetivo do filtro $\mathcal{H}_\infty$ [1] é estimar uma saída auxiliar, definida por uma combinação algébrica de estados, dada por

$$\mathbf{y}_k = L_k \mathbf{x}_k, \quad (2.28)$$

sendo L_k uma matriz definida pelo projetista para obter $\hat{\mathbf{x}}_{k+1}$. A função custo conhecida a ser minimizada para a solução deste problema é dada por

$$J = \frac{\sum_{k=0}^{N-1} \|\mathbf{y}_k - \hat{\mathbf{y}}_k\|_{S_k}^2}{\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|_{P_0^{-1}}^2 + \sum_{k=0}^{N-1} \left(\|\mathbf{w}_k\|_{Q_k^{-1}}^2 + \|\mathbf{v}_k\|_{R_k^{-1}}^2 \right)}, \quad (2.29)$$

sendo P_0 , Q_k , R_k , e S_k matrizes simétricas definidas positivas. A função custo pode ser feita para ser menor que $1/\gamma$ (um limite especificado pelo projetista) com a seguinte estratégia de estimação

$$\begin{aligned} \bar{S}_k &= L_k^T S_k L_k, \\ K_k &= P_k \left[I - \gamma \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} H_k^T R_k^{-1}, \\ \hat{\mathbf{x}}_{k+1} &= F_k \hat{\mathbf{x}}_k + F_k K_k [\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k)], \\ P_{k+1} &= F_k P_k \left[I - \gamma \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} F_k^T + G_k Q_k G_k^T, \end{aligned}$$

garantindo que a condição que segue é assegurada para todo passo do tempo k

$$P_k^{-1} - \gamma \bar{S}_k + H_k^T R_k^{-1} H_k > 0. \quad (2.30)$$

2.3.3 Filtro BDU

Nesta seção é apresentado o filtro BDU proposto em [61]. Para isto considere o sistema com incertezas (2.27), e suas perturbações modeladas como

$$[\delta F_k \quad \delta G_k] = M \Delta_k [N_F \quad N_G], \quad (2.31)$$

sendo $\|\Delta_k\| < 1$, e os ruídos dados por $\mathbf{w}_k = N(0, Q_k)$ e $\mathbf{v}_k = N(0, R_k)$.

O modelo (2.31) permite que o projetista restrinja as fontes de distorção selecionando as entradas $\{N_F \ N_G\}$ apropriadamente [61].

Este filtro robusto procura minimizar o erro de estimativa para o pior caso possível criado pela faixa de incertezas δF_k e δG_k . As estimativas $\{\hat{\mathbf{x}}_{k+1|k}, \hat{\mathbf{x}}_{k+1|k+1}\}$ no algoritmo que segue, podem ser obtidas por iterações recursivas entre os passos 2 e 3. Para o caso quando $\hat{\lambda}_k = 0$, os passos 2 e 3 são reduzidos para o tempo padrão e para as equações de atualização de medida do filtro de Kalman. O algoritmo do filtro robusto é descrito a seguir:

Passo 1:

Se $H_{k+1}M_k = 0$, então configure $\hat{\lambda}_k = 0$ (filtro não robusto). Caso contrário, selecione α (tipicamente $0 \leq \alpha_f \leq 1$) e configure $\hat{\lambda}_k = (1 + \alpha_f) \cdot \|M_k^T H_{k+1}^T R_{k+1}^{-1} H_{k+1} M_k\|$.

Passo 2:

Substitua $\{Q_k, R_k, P_{k|k}, G_k, F_k\}$ por:

$$\begin{aligned}\hat{Q}_k^{-1} &= Q_k^{-1} + \hat{\lambda}_k N_{G_k}^T [I + \hat{\lambda}_k N_{F_k} P_{k|k} N_{F_k}^T]^{-1} N_{G_k} \\ \hat{R}_{k+1} &= R_k - \hat{\lambda}_k^{-1} H_{k+1} M_k M_k^T H_{k+1}^T \\ \hat{P}_{k|k} &= (P_{k|k}^{-1} + \hat{\lambda}_k N_{F_k}^T N_{F_k})^{-1} \\ &= P_{k|k} - P_{k|k} N_{F_k}^T (\hat{\lambda}_k^{-1} I + N_{F_k} P_{k|k} N_{F_k}^T)^{-1} N_{F_k} P_{k|k} \\ \hat{G}_k &= G_k - \hat{\lambda}_k F_k \hat{P}_{k|k} N_{F_k}^T N_{G_k} \\ \hat{F}_k &= (F_k - \hat{\lambda}_k \hat{G}_k \hat{Q}_k N_{G_k}^T N_{F_k}) (I - \hat{\lambda}_k \hat{P}_{k|k} N_{F_k}^T N_{F_k})\end{aligned}$$

Passo 3:

Atualize $\{\hat{\mathbf{x}}_{k|k}, P_{k|k}\}$ como segue:

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= \hat{F}_k \hat{\mathbf{x}}_{k|k} \\ P_{k+1|k} &= F_k \hat{P}_{k|k} F_k^T + \hat{G}_k \hat{Q}_k \hat{G}_k^T, \\ K_{k+1} &= P_{k+1|k} H_{k+1}^T [H_{k+1} P_{k+1|k} H_{k+1}^T + \hat{R}_{k+1}]^{-1} \\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + K_{k+1} [\mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k})] \\ P_{k+1|k+1} &= [I - K_{k+1} H_{k+1}] P_{k+1|k}\end{aligned}$$

2.3.4 Filtro de Kalman robusto recursivo

O filtro de Kalman robusto recursivo (FKRR) proposto nesta seção é uma extensão do filtro robusto apresentado em [66]. Para isto considere o seguinte sistema com incertezas (2.19), cujas incertezas δF_k , δG_k , e δH_k são consideradas estruturadas da seguinte forma

$$\begin{bmatrix} \delta F_k & \delta G_k \\ \delta H_k & 0 \end{bmatrix} = \begin{bmatrix} M_{1_k} & 0 \\ 0 & M_{2_k} \end{bmatrix} \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix} \begin{bmatrix} N_{F_k} & N_{G_k} \\ N_{H_k} & 0 \end{bmatrix}, \quad (2.32)$$

sendo Δ_1 e Δ_2 termos aleatórios cujas normas são definidas por $\|\Delta_i\| < 1$. As incertezas (2.32) permitem restringir as fontes de distorção através da seleção apropriada das entradas $\{M_{1_k} \ M_{2_k}\}$ e $\{N_{F_k} \ N_{G_k} \ N_{H_k}\}$.

Uma interpretação determinística é dada às variáveis $\mathbf{w}_k$ e $\mathbf{v}_k$ do modelo em espaço de estado (2.19). Elas são consideradas geralmente na literatura de filtro de Kalman, como variáveis estocásticas sendo Q_k e R_k as variâncias dos respectivos ruídos. Além disso, é considerado que não existe nenhuma matriz de ponderação entre $\{\mathbf{x}_0, \mathbf{w}_0, \mathbf{v}_1\}$. Em uma interpretação estocástica significa que elas são variáveis não correlacionadas.

Dado o funcional

$$\begin{aligned} J_k := & \begin{bmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{w}_k \\ \mathbf{v}_k \\ \mathbf{x}_{k+1} \end{bmatrix}^T \begin{bmatrix} P_{k|k-1}^{-1} & 0 & 0 & 0 \\ 0 & Q_k^{-1} & 0 & 0 \\ 0 & 0 & R_k^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{w}_k \\ \mathbf{v}_k \\ \mathbf{x}_{k+1} \end{bmatrix} \\ & + \left(\left(\begin{bmatrix} F_k & G_k & 0 & -I \\ H_k & 0 & I & 0 \end{bmatrix} + \begin{bmatrix} \delta F_k & \delta G_k & 0 & 0 \\ \delta H_k & 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{w}_k \\ \mathbf{v}_k \\ \mathbf{x}_{k+1} \end{bmatrix} \right. \\ & \left. - \left(\begin{bmatrix} -F_k \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{z}_{k+1} - H_k \hat{\mathbf{x}}_{k|k-1} \end{bmatrix} + \begin{bmatrix} -\delta F_k \hat{\mathbf{x}}_{k|k-1} \\ -\delta H_k \hat{\mathbf{x}}_{k|k-1} \end{bmatrix} \right) \right)^T \times \mu I \left(\begin{bmatrix} \bullet \\ \bullet \end{bmatrix} \right), \quad (2.33) \end{aligned}$$

o problema de filtragem robusta é encontrar $\hat{\mathbf{x}}_{k|k}$ que minimize J_k , considerando o pior caso das

perturbações, ou seja

$$\min_{\mathbf{x}_k, \mathbf{x}_{k+1}} \max_{\delta_k} J_k \quad (2.34)$$

sendo $\delta_k := \{\delta F_k, \delta G_k, \delta K_k, \delta H_k\}$.

Para apresentarmos o FKRR as seguintes variáveis são definidas:

$$\begin{aligned} \mathcal{F}_k &= \begin{bmatrix} F_k \\ H_k \end{bmatrix}, \quad \mathcal{G}_k = \begin{bmatrix} G_k & 0 \\ 0 & I \end{bmatrix}, \quad \mathcal{E}_k = \begin{bmatrix} -I \\ 0 \end{bmatrix}, \\ N_{\mathcal{F}_k} &= \begin{bmatrix} N_{F_k} \\ N_{H_k} \end{bmatrix}, \quad N_{\mathcal{G}_k} = \begin{bmatrix} N_{G_k} & 0 \\ 0 & 0 \end{bmatrix}, \quad N_{\mathcal{E}_k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ \mathbf{b}_k &= \begin{bmatrix} -F_k \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k-1} \end{bmatrix}, \quad N_{\mathbf{b}_k} = \begin{bmatrix} -N_{F_k} \hat{\mathbf{x}}_{k|k-1} \\ -N_{H_k} \hat{\mathbf{x}}_{k|k-1} \end{bmatrix}, \quad \hat{\boldsymbol{\nu}}_{k|k} = \begin{bmatrix} \hat{\mathbf{w}}_{k|k} \\ \hat{\mathbf{v}}_{k|k} \end{bmatrix} \\ \mathcal{R}_k &= \begin{bmatrix} Q_k & 0 \\ 0 & R_{k+1} \end{bmatrix}, \quad \text{e } M_k = \begin{bmatrix} M_{1k} & 0 \\ 0 & M_{2k} \end{bmatrix}. \end{aligned} \quad (2.35)$$

Teorema 2.3.1. *A estimativa filtrada e preditora do FKRR para o sistema (2.19), e a correspondente equação recursiva de Riccati são dadas por (2.37) e (2.38), respectivamente. A estimativa preditora também pode ser obtida por (2.39).*

O parâmetro escalar ótimo $\hat{\lambda}_k$ é obtido pela minimização do funcional (D.8) no intervalo

$$\hat{\lambda}_k = (1 + \alpha_f) \|M_k^T \mu I M_k\|, \quad (2.36)$$

para um dado valor de $\mu > 0$, e $\alpha_f > 0$.

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{v}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}^T \begin{bmatrix} P_{k|k-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & \mathcal{R}_k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & \hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu^{-1} I & 0 & 0 & M_k & \mathcal{F}_k & \mathcal{G}_k & \mathcal{E}_k \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & N_{\mathcal{F},k} & N_{\mathcal{G},k} & N_{\mathcal{E},k} \\ 0 & 0 & I & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & M_k^T & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & \mathcal{F}_k^T & N_{\mathcal{F},k}^T & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & \mathcal{G}_k^T & N_{\mathcal{G},k}^T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{E}_k^T & N_{\mathcal{E},k}^T & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{b}_k \\ N_{\mathbf{b}_k} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2.37)$$

$$P_{k+1|k} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I \end{bmatrix}^T \begin{bmatrix} P_{k|k-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & \mathcal{R}_k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & \hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu^{-1} I & 0 & 0 & M_k & \mathcal{F}_k & \mathcal{G}_k & \mathcal{E}_k \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & N_{\mathcal{F},k} & N_{\mathcal{G},k} & N_{\mathcal{E},k} \\ 0 & 0 & 0 & I & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & M_k^T & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & \mathcal{F}_k^T & N_{\mathcal{F},k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & \mathcal{G}_k^T & N_{\mathcal{G},k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{E}_k^T & N_{\mathcal{E},k}^T & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -I \end{bmatrix}, \quad \text{e} \quad (2.38)$$

$$\hat{\mathbf{x}}_{k+1|k} = F_k \hat{\mathbf{x}}_{k|k} + G_k \hat{\mathbf{w}}_{k|k}. \quad (2.39)$$

Prova: Veja no Apêndice D.

◇

Observação 2.3.1. A matriz de malha fechada associada à estimativa filtrada é dada por

$$L_{k|k} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} P_{k|k-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & \mathcal{R}_k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & \hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu^{-1} I & 0 & 0 & M_k & \mathcal{F}_k & \mathcal{G}_k & \mathcal{E}_k \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & N_{\mathcal{F},k} & N_{\mathcal{G},k} & N_{\mathcal{E},k} \\ 0 & 0 & 0 & I & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & M_k^T & 0 & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 & \mathcal{F}_k^T & N_{\mathcal{F},k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & \mathcal{G}_k^T & N_{\mathcal{G},k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{E}_k^T & N_{\mathcal{E},k}^T & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2.40)$$

E a matriz de malha fechada associada à estimativa preditora é dada por

$$L_{k+1|k} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I \end{bmatrix}^T \begin{bmatrix} P_{k|k-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & \mathcal{R}_k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & \hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\hat{\lambda}_k^{-1} I & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu^{-1} I & 0 & 0 & M_k & \mathcal{F}_k & \mathcal{G}_k & \mathcal{E}_k \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & N_{\mathcal{F},k} & N_{\mathcal{G},k} & N_{\mathcal{E},k} \\ 0 & 0 & 0 & I & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & M_k^T & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & \mathcal{F}_k^T & N_{\mathcal{F},k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & \mathcal{G}_k^T & N_{\mathcal{G},k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{E}_k^T & N_{\mathcal{E},k}^T & 0 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2.41)$$

Observação 2.3.2. É considerado que $\mu \rightarrow \infty$ e, em consequência, $\mu^{-1} \rightarrow 0$, $\hat{\lambda}_k^{-1} \rightarrow 0$, [66]. Observe que este filtro depende da inversa destas variáveis que tendem para zero. Como estamos considerando neste trabalho uma versão estendida deste filtro para lidar com o sistema não linear, μ pode ser considerado tão grande quanto possível. A natureza robusta, estabilidade e convergência deste filtro continuam válidas para este caso.

Para o caso em que $\mu \rightarrow \infty$, depois de alguma álgebra, pode-se mostrar que (2.37) é equivalente à (2.42). Neste caso, a condição de existência deste filtro é garantida se

$$\begin{bmatrix} \mathcal{F}_k & \mathcal{G}_k & \mathcal{E}_k \\ N_{\mathcal{F},k} & N_{\mathcal{G},k} & N_{\mathcal{E},k} \end{bmatrix}$$

tiver posto linha pleno. Observe que nesta abordagem o filtro robusto não depende de parâmetros auxiliares para ser ajustado, somente de parâmetros e ponderações de matrizes que são conhecidos

a-priori. Este resultado é útil para aplicações em tempo real.

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{v}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}^T \begin{bmatrix} P_{k|k-1} & 0 & 0 & 0 & I & 0 & 0 \\ 0 & \mathcal{R}_k & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & \mathcal{F}_k & \mathcal{G}_k & \mathcal{E}_k \\ 0 & 0 & 0 & 0 & N_{\mathcal{F},k} & N_{\mathcal{G},k} & N_{\mathcal{E},k} \\ I & 0 & \mathcal{F}_k^T & N_{\mathcal{F},k}^T & 0 & 0 & 0 \\ 0 & I & \mathcal{G}_k^T & N_{\mathcal{G},k}^T & 0 & 0 & 0 \\ 0 & 0 & \mathcal{E}_k^T & N_{\mathcal{E},k}^T & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \mathbf{b}_k \\ N_{\mathbf{b}_k} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2.42)$$

$$P_{k+1|k} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I \end{bmatrix}^T \begin{bmatrix} P_{k|k-1} & 0 & 0 & 0 & I & 0 & 0 \\ 0 & \mathcal{R}_k & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & \mathcal{F}_k & \mathcal{G}_k & \mathcal{E}_k \\ 0 & 0 & 0 & 0 & N_{\mathcal{F},k} & N_{\mathcal{G},k} & N_{\mathcal{E},k} \\ I & 0 & \mathcal{F}_k^T & N_{\mathcal{F},k}^T & 0 & 0 & 0 \\ 0 & I & \mathcal{G}_k^T & N_{\mathcal{G},k}^T & 0 & 0 & 0 \\ 0 & 0 & \mathcal{E}_k^T & N_{\mathcal{E},k}^T & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -I \end{bmatrix}. \quad (2.43)$$

Na Seção 2.3.5 é apresentada uma comparação entre o filtro robusto recursivo dado pelas equações (2.42) e (2.43), para o caso em que $\mu \rightarrow \infty$, com o filtro BDU proposto em [61], a solução da equação algébrica de Riccati $P_{k+1|k}$ quando $k \rightarrow +\infty$ e os pólos do sistema em malha aberta e fechada.

2.3.5 Exemplo numérico

Considere o sistema apresentado em [61]

$$\begin{aligned} \mathbf{x}_{k+1} &= (F_k + \delta F_k) \mathbf{x}_k + G_k \mathbf{w}_k, \\ \mathbf{z}_{k+1} &= H_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1}, \end{aligned}$$

sendo

$$F_k = \begin{bmatrix} 0.9802 & 0.0196 \\ 0 & 0.9802 \end{bmatrix}, \quad G_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_{k+1} = \begin{bmatrix} 1 & -1 \end{bmatrix},$$

e as covariâncias de $\mathbf{w}_k$ e $\mathbf{v}_k$ (ou matrizes de ponderação) são dadas respectivamente por

$$Q_k = \begin{bmatrix} 1.9608 & 0.0195 \\ 0.0195 & 1.9605 \end{bmatrix}, \quad R_{k+1} = 1.$$

As incertezas paramétricas são modeladas por

$$M_1 = \begin{bmatrix} 0.198 \\ 0 \end{bmatrix}, \quad N_f = \begin{bmatrix} 0 & 5 \end{bmatrix}. \quad (2.44)$$

Na Figura 2.2 é comparado o filtro de Kalman robusto recursivo dado pelas equações (2.42), e (2.43), com o filtro BDU proposto em [61]. O parâmetro $\alpha_f = 0.5$ do filtro BDU foi escolhido de maneira empírica. Observe que o FKRR não depende de nenhum parâmetro *offline* para ser ajustado.

Os resultados da Figura 2.2 mostram uma equivalência de ambos os filtros e uma significativa deterioração de desempenho do filtro de Kalman padrão. Estas curvas representam a variância do erro definida como sendo

$$\mathcal{E} \|x_k - \hat{x}_k\| \approx \frac{1}{T} \sum_{j=1}^T \|x_k^{(j)} - \hat{x}_k^{(j)}\|.$$

Cada ponto no instante k em cada curva de variância é o conjunto de médias calculadas sobre

Figura 2.2: Variância do erro com Δ selecionado uniformemente entre o intervalo $[-1, 1]$.

T experimentos ($T = 5000$ trajetórias com $N = 1000$ pontos gerados). Para cada experimento j , as matrizes Δ_1 e Δ_2 com normas menores ou igual a um são selecionadas randomicamente e fixadas para todo k . E a curva ótima é obtida calculando-se o filtro de Kalman padrão com a

matriz $(F + \delta F)$, ou seja, o filtro de Kalman padrão com a informação da incerteza.

Na Tabela 2.1, são apresentados valores de convergência de $P_{k+1|k}$ para diferentes casos de μ do filtro de Kalman robusto recursivo dado pelas equações (2.37) e (2.38).

Tabela 2.1: Comportamento do FKRR ($\mu \rightarrow +\infty$).

μ	$P_{k+1 k}$
10^2	$\begin{bmatrix} 0.2972 & -0.2941 \\ -0.2941 & 2.5873 \end{bmatrix}$
10^{10}	$\begin{bmatrix} 0.2876 & -0.2951 \\ -0.2951 & 2.5688 \end{bmatrix}$
10^{15}	$\begin{bmatrix} 0.1871 & -0.2216 \\ -0.2216 & 2.4095 \end{bmatrix}$
∞	$\begin{bmatrix} 0.0018 & -0.0587 \\ -0.0587 & 1.9574 \end{bmatrix}$

Observe que para valores suficientemente elevados de μ , a sequência matricial $\{P_{k+1|k}\}_{k=0}^{+\infty}$ converge para a solução $P_{k+1|k}$ da equação algébrica de Riccati (2.43), quando $k \rightarrow +\infty$.

2.4 Resultados experimentais

Para a realização dos experimentos de estimativa de atitude e posição, foram identificados os ruídos dos sensores da IMU apresentada no Apêndice A.3, utilizando o método apresentado no Apêndice A.2 e o programa apresentado no Apêndice A.3. Segue abaixo a Tabela 2.2 com os parâmetros dos ruídos identificados; as figuras 2.3.a, 2.4.a, 2.5.a 2.6.a, 2.7.a e 2.8.a mostram os dados utilizados para a identificação e as figuras 2.3.b, 2.4.b, 2.5.b 2.6.b, 2.7.b e 2.8.b mostram o desvio de Allan utilizado para a identificação. Os dados dos sensores foram obtidos com o sensor parado; porém, em algumas figuras é possível observar uma variação do valor da medida do sensor, pois os dados dos sensores não foram obtidos em um ambiente com temperatura controlada.

Os dados de saída da IMU são obtidos no formato digital (RS-232) a uma taxa de 25 Hz. E os algoritmos dos filtros foram executados em MATLAB a uma taxa de amostragem $T = 0.04$ s.

Nas duas próximas seções são apresentados os resultados experimentais da estimativa de atitude e posição, respectivamente.

Giroscópios	σ_g^2	σ_{bg}^2	τ_g
x	$0.1105 \text{ (}^\circ/\text{s)}^2$	$12.6653 \text{ (}^\circ/\text{s)}^2$	1759.07 s
y	$0.0760 \text{ (}^\circ/\text{s)}^2$	$14.7800 \text{ (}^\circ/\text{s)}^2$	1621.95 s
z	$0.1170 \text{ (}^\circ/\text{s)}^2$	$0.5600 \text{ (}^\circ/\text{s)}^2$	1093.75 s
Acelerômetros	σ_a^2	σ_{ba}^2	τ_a
x	$0.0040 \text{ (m/s}^2\text{)}^2$	$0.0003 \text{ (m/s}^2\text{)}^2$	242.24 s
y	$0.0036 \text{ (m/s}^2\text{)}^2$	$0.0100 \text{ (m/s}^2\text{)}^2$	4132.23 s
z	$0.0024 \text{ (m/s}^2\text{)}^2$	$0.0050 \text{ (m/s}^2\text{)}^2$	4444.44 s
Magnetômetros	σ_m^2		
x	0.00098 G^2		
y	0.00098 G^2	-	-
z	0.00098 G^2		

Tabela 2.2: Parâmetros dos ruídos dos sensores inerciais.

Figura 2.3: Sequência de dados do giroscópio do eixo x (a), e desvio de Allan do giroscópio do eixo x (b). Figura 2.4: Sequência de dados do giroscópio do eixo y (a), e desvio de Allan do giroscópio do eixo y (b).

Figura 2.5: Sequência de dados do giroscópio do eixo z (a), e desvio de Allan do giroscópio do eixo z (b).

Figura 2.6: Sequência de dados do acelerômetro do eixo x (a), e desvio de Allan do acelerômetro do eixo x (b).

Figura 2.7: Sequência de dados do acelerômetro do eixo y (a), e desvio de Allan do acelerômetro do eixo y (b).

Figura 2.8: Sequência de dados do acelerômetro do eixo z (a), e desvio de Allan do acelerômetro do eixo z (b).

2.4.1 Estimativa de atitude

O ângulo de referência para o estudo comparativo foi obtido através de um *encoder* fixo a uma IMU, como mostrado na Figura 2.9. Os dados da IMU são obtidos através de um porta serial RS-232 com frequência de 25 Hz. Maiores detalhes sobre a IMU podem ser obtidos no Apêndice A.3. Os filtros foram executados em MATLAB®.

Figura 2.9: IMU com *encoder* acoplado.

Durante o período de tempo de operação, o sistema mantém uma *flag* indicando *verdadeiro* (ou seja, 1) se o veículo não estiver acelerando e *falso* (ou seja, 0) caso contrário, veja [24] para mais detalhes. No tempo t , baseado nos dados da IMU, esta *flag* é definida para ser verdadeira quando a condição lógica

$$(\mu(t) < \beta_\mu) \text{ and } (\|\omega_g\| < \beta_g) \text{ and } (\mu_f(t) < \beta_f) \quad (2.45)$$

é verdadeira. O sinal $\mu(t)$ é definido por $\mu(t) = \|\mathbf{a}_a - \mathbf{g}_e\|$, e os parâmetros β_μ , β_g , e β_f foram ajustados como os valores 2, 4 e 1.2, respectivamente.

Quando o veículo está acelerando, somente os sinais do magnetômetro são usados com saída medida. Um diagrama de blocos do sistema de atitude implementado é mostrado na Figura 2.10.

Figura 2.10: Diagrama de blocos do sistema de atitude.

Filtro de Kalman estendido

As matrizes de ponderações, Q e R são dadas por

$$Q = \begin{bmatrix} Q_g & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_{bg} \end{bmatrix}, R = \begin{bmatrix} R_m & 0_{3 \times 3} \\ 0_{3 \times 3} & R_a \end{bmatrix},$$

$$Q_g = \text{diag}(\sigma_{g,x}^2, \sigma_{g,y}^2, \sigma_{g,z}^2), Q_{bg} = \text{diag}(\sigma_{bg,x}^2, \sigma_{bg,y}^2, \sigma_{bg,z}^2),$$

$$R_m = \text{diag}(\sigma_{m,x}^2, \sigma_{m,y}^2, \sigma_{m,z}^2), \text{ e } R_a = \text{diag}(\sigma_{a,x}^2, \sigma_{a,y}^2, \sigma_{a,z}^2).$$

Os valores destas variâncias são apresentados na Tabela 2.2.

Filtro $\mathcal{H}_\infty$

As matrizes de ponderações, Q e R , são as mesmas utilizadas no filtro de Kalman estendido e $\gamma = 0.1$.

Filtro de Kalman robusto recursivo

As matrizes de ponderações, Q e R , são as mesmas utilizadas no filtro de Kalman estendido e as matrizes restantes foram consideradas como

$$\begin{aligned} N_{F_k} &= 10^2 [f_1 \ \cdots \ f_7], \\ N_{G_k} &= 10^2 [g_1 \ \cdots \ g_3 \ 0 \ 0 \ 0], \\ N_{H_k} &= [h_1 \ \cdots \ h_7], \\ M_{1_k} &= 10^{-3} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T, \\ M_{2_k} &= 10^{-3} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T, \\ \mu &= 450000, \text{ e } \alpha_f = 1, \end{aligned}$$

sendo

$$\begin{aligned} f_l &= \frac{\sum_{i=1}^4 |\bar{F}_k(i, l)|}{4} \text{ com } l = 1, 2, \dots, 7, \\ g_l &= \frac{\sum_{i=1}^4 |G_k(i, l)|}{4} \text{ com } l = 1, 2 \text{ e } 3, \\ h_l &= \frac{\sum_{i=1}^6 |H_k(i, l)|}{6} \text{ com } l = 1, 2, \dots, 7, \text{ e} \\ \bar{F}_k &= F_k - I_{7 \times 7}. \end{aligned} \tag{2.46}$$

Estudo comparativo

Um estudo comparativo entre o filtro de Kalman estendido (FKE), o filtro $\mathcal{H}_\infty$, e o filtro de Kalman robusto recursivo (FKRR) proposto, foi desenvolvido baseado na média de 10 amostras de dados consecutivos de cada ângulo de Euler, rolagem (ϕ), arfagem (θ), e guinada (ψ). Para cada amostra, como a precisão do sensor depende da temperatura, incertezas e distúrbios foram adicionadas aos sinais do giroscópio para simular mudanças estáticas e abruptas de temperatura no sensor como

$$\boldsymbol{\omega} = (1 + c)\boldsymbol{\omega}_g + \mathbf{d}_g - \mathbf{b}_g, \tag{2.47}$$

sendo $c = -0.1, -0.05, 0, 0.05, 0.1$, $\mathbf{d}_g = [d_{g_x} \ d_{g_y} \ d_{g_z}]^T$, mostradas na Figura 2.11, e

$$d_{g_{x,y,z}} = 5e^{-(t-t_d)^4} \sin(1.3\pi t), \quad (2.48)$$

sendo $t_d = 10s$ para d_{g_x} , $t_d = 20s$ para d_{g_y} , e $t_d = 30s$ para d_{g_z} . Os valores do parâmetro c foi escolhido baseado no *datasheet* do giroscópio IDG-300 que compõe a IMU.

Figura 2.11: Distúrbios do giroscópio.

As normas $\mathcal{L}_2$ das estimativas de erros dos ângulos foram utilizadas para comparar o desempenho dos filtros

$$\mathcal{L}_2[\tilde{\rho}(t)] = \left(\frac{1}{(t_r - t_0)} \int_{t_0}^{t_r} \|\tilde{\rho}(t)\|_2^2 dt \right)^{\frac{1}{2}}, \quad (2.49)$$

sendo $\|\cdot\|_2$ a norma Euclideana, $\tilde{\rho}(t)$ o erro de estimativa do ângulo no tempo t , $t_0 = 0$, e $t_r = 120$ s o tempo do experimento. A Tabela 2.4 mostra os desempenhos dos filtros robustos sobre o filtro de Kalman estendido baseado na Equação (2.49).

Como esperado, o FKRR apresentou melhores resultados que o filtro de Kalman estendido e o filtro $\mathcal{H}_\infty$. O filtro de Kalman robusto recursivo teve melhora de aproximadamente 8 % sobre o desempenho do filtro de Kalman estendido.

As figuras 2.12a, 2.12b, e 2.12c mostram a estimativa de atitude obtida dos filtros considerando o sistema de coordenadas da Figura 2.1. É interessante observar nestas figuras que a estimativa dos ângulos deteriora quando o efeito do distúrbio (2.48) aparece. Detalhes deste efeito podem ser feitos vistos nas Figuras 2.13a, 2.13b e 2.13c. Elas mostram um *zoom* das estimativas no intervalo (29s para 31s), quando a estimativa dos ângulos estão sob o efeito dos distúrbio d_{g_z} .

Tabela 2.3: Melhoria de desempenho sobre o filtro de Kalman estendido.

Ângulo	c	Filtro $\mathcal{H}\infty$	FKRR	
ϕ	-0.1	3.3614%	7.9188%	
	-0.5	2.5340%	8.0225%	
	0	3.7153%	9.4971%	
	0.5	4.8189%	11.0529%	
	1	4.0851%	10.2806%	
		3.7030%	9.3544%	Média
θ	-0.1	4.4218%	4.4518%	
	-0.5	4.6228%	5.5199%	
	0	3.0332%	3.7590%	
	0.5	-0.3283%	1.8966%	
	1	0.5814%	1.8831%	
		2.4662%	3.5021%	Média
ψ	-0.1	-1.3726%	7.9830%	
	-0.5	-0.2406%	6.4680%	
	0	6.2734%	9.7917%	
	0.5	17.7794%	18.4639%	
	1	9.7537%	13.5324%	
		6.4387%	11.2478%	Média

2.4.2 Estimativa de posição

Os dados de saída da IMU e do GPS são obtidos no formato digital (RS-232). Maiores detalhes sobre o GPS, mostrado na Figura 2.14, podem ser obtidos no Apêndice A.4. E os algoritmos dos filtros foram executados em MATLAB a uma taxa de amostragem $T = 0.4$ s para o filtro de Kalman padrão e para o filtro BDU [61].

Para a estimativa de posição foram utilizados o filtro de Kalman e o filtro BDU. As matrizes de ponderações, Q e R , são as mesmas para ambos os filtros. As matrizes de ponderações específicas do filtro BDU, M , N_F e N_G , foram selecionadas com base nos valores das variâncias das matrizes F e G , sendo o parâmetro α_f ajustado em 1000. As Figuras 2.17 a 2.18 apresentam os resultados obtidos. Um diagrama de blocos do sistema de posição implementado é mostrado na Figura 2.15.

Figura 2.12: Rolagem ϕ (a), arfagem θ (b), e guinada ψ (c) para $c = 1$.

Figura 2.13: Rolagem ϕ (a), arfagem θ (b), e guinada ψ (c) no intervalo (29s - 31s) para $c = 1$.

Figura 2.14: Receptor GPS da *San Jose Navigation* 32 canais, 5 Hz com antena.

Estudo comparativo

Um estudo comparativo entre o filtro de Kalman e o filtro BDU é mostrado na Tabela 2.4. A norma $\mathcal{L}_2$ do vetor do erro de observação foi utilizada para comparar os algoritmos:

$$\mathcal{L}_2[\tilde{\mathbf{z}}(t)] = \left(\frac{1}{(t_r - t_0)} \int_{t_0}^{t_r} \|\tilde{\mathbf{z}}(t)\|_2^2 dt \right)^{\frac{1}{2}},$$

sendo $\|\cdot\|_2$ a norma Euclideana, $\tilde{\mathbf{z}}(t)$ o erro de observação de posição e $t_r = 342.60$ s o tempo de experimento.

Observe que o filtro robusto obteve uma melhoria de 5 % no erro de observação da posição sobre o filtro de Kalman. Mas graficamente pouca diferença é vista entre os filtros, como pode

Figura 2.15: Diagrama de blocos do sistema de posição.

Algoritmos	$\mathcal{L}_2[\tilde{\mathbf{z}}(t)]$
Filtro de Kalman	0.0947
Filtro BDU	0.0895
Melhoria (%)	5.4932

Tabela 2.4: Estudo comparativo

ser observado nas figuras 2.16- 2.18.

Figura 2.16: Estimativa da posição geodésica no sistema de coordenadas LLA.

Figura 2.17: Estimativa da posição geodésica no sistema de coordenadas LLA: Latitude λ (a), Longitude ϕ (b) e Altitude h (c).

Figura 2.18: Estimativa das velocidades no sistemas de coordenadas NED: v_N (a), v_E (b) e v_D (c).

CAPÍTULO 3

Robô móvel com rodas e robô helicóptero

Neste capítulo são apresentados os modelos individuais de um robô móvel com rodas e um robô helicóptero. Inicialmente são apresentados o modelo cinemático e o modelo dinâmico de um robô móvel com rodas deslizantes, RMRD, baseado no robô Pioneer 3AT mostrado na Figura 3.1. A modelagem do robô está baseada em [13, 39], e o modelo dinâmico é reescrito em espaço de estado na forma quase linear a parâmetros variantes (quase-LPV). Depois mostramos como se determina a trajetória do robô móvel ponto a ponto, e apresentamos resultados de simulações utilizando um controlador robusto apresentado na Seção 3.5. Em seguida apresentamos uma visão geral de um modelo de helicóptero e suas equações dinâmicas de corpo rígido. Apresentamos também o modelo em espaço de estado linearizado e identificado descrito em [50] e [18] que usaremos nas simulações. O controle de posição e ângulo de proa de um robô helicóptero, utilizando a metodologia apresentada em [3] é considerado nesta tese. Ela considera a combinação em cascata do regulador linear quadrático para estabilizar os pólos do sistema, de um controlador baseado em linearização por realimentação para desacoplar os pares de entrada/saída e de um controlador PD para assegurar o acompanhamento de trajetória. E finalmente, são apresentados resultados de simulações do robô helicóptero utilizando a metodologia de controle apresentada em [3] .

Figura 3.1: Pioneer 3AT.

3.1 Modelo cinemático

A geometria do RMRD é mostrada na Figura 3.2, sendo (X, Y, Z) o sistema de coordenadas

Figura 3.2: Geometria do robô móvel.

de navegação, definido aqui pelo sistema de coordenadas geográficas NED (do inglês, *North, East, Down*), (X^B, Y^B) o sistema de coordenadas do corpo, CG o centro de gravidade do robô, (x, y) as coordenadas do robô no centro de gravidade no sistema de coordenadas de navegação, ψ o ângulo entre o eixo X^B e o eixo X , CIR o centro instantâneo de rotação, (x_{CIR}^B, y_{CIR}^B) as coordenadas do centro instantâneo de rotação, a a distância entre o eixo de simetria da roda atuada frontal paralelo ao eixo Y^B e o centro de gravidade, b a distância entre o eixo de simetria paralelo ao eixo Y^B da roda atuada traseira e o centro de gravidade, c a distância entre o eixo

de simetria paralelo ao eixo X^B da roda atuada e o eixo X^B e r o raio das rodas atuadas.

No sistema de coordenadas do corpo a velocidade linear do robô é dada por $\mathbf{v} = [v_{x^B} \ v_{y^B} \ 0]^T$, sendo v_{x^B} e v_{y^B} as componentes de $\mathbf{v}$ no eixo X^B e Y^B , e a velocidade angular por $\boldsymbol{\omega} = [0 \ 0 \ \omega]$, sendo $\omega = \dot{\psi}$. As velocidades lineares e angulares de cada roda atuada são dadas por v_i e ω_i , sendo $i = 1, 2, \dots, 4$ a i -ésima roda atuada.

As velocidades absolutas $\dot{x}$, $\dot{y}$ e $\dot{\psi}$ no sistema inercial de coordenadas são dadas por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{x^B} \\ v_{y^B} \\ \omega \end{bmatrix} = R(\psi) \begin{bmatrix} v_{x^B} \\ v_{y^B} \\ \omega \end{bmatrix}. \quad (3.1)$$

Diferenciando a equação acima em relação ao tempo tem-se

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\psi} \end{bmatrix} = R(\psi) \begin{bmatrix} \dot{v}_{x^B} - \dot{\psi}v_{y^B} \\ \dot{v}_{y^B} + \dot{\psi}v_{x^B} \\ \dot{\omega} \end{bmatrix} = R(\psi) \begin{bmatrix} a_{x^B} \\ a_{y^B} \\ \dot{\omega} \end{bmatrix}. \quad (3.2)$$

Da Figura 3.2, como mostrado em [39], obtém-se

$$v_L = v_{1x^B} = v_{3x^B},$$

$$v_R = v_{2x^B} = v_{4x^B},$$

$$v_F = v_{1y^B} = v_{2y^B},$$

$$v_B = v_{3y^B} = v_{4y^B},$$

$$\begin{bmatrix} v_L \\ v_R \\ v_F \\ v_B \end{bmatrix} = \begin{bmatrix} 1 & c \\ 1 & -c \\ 0 & -x_{CIR}^B + a \\ 0 & -x_{CIR}^B - b \end{bmatrix} \begin{bmatrix} v_{x^B} \\ \omega \end{bmatrix}, \quad (3.3)$$

$$\boldsymbol{\omega}_w = \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_L \\ v_R \end{bmatrix}, \text{ e} \quad (3.4)$$

$$\boldsymbol{\eta} = \begin{bmatrix} v_{x^B} \\ \omega \end{bmatrix} = r \begin{bmatrix} \frac{\omega_L + \omega_R}{2} \\ \frac{\omega_L - \omega_R}{2c} \end{bmatrix}, \quad (3.5)$$

sendo v_L , v_R , ω_L , ω_R , $\boldsymbol{\eta}$ as velocidades lineares das rodas do lado esquerdo e direito do robô, as velocidades angulares das rodas do lado esquerdo e direito do robô e a nova entrada de controle, respectivamente. Escrevendo matricialmente (3.5) e isolando o vetor $\boldsymbol{\omega}_w$, tem-se

$$\begin{bmatrix} v_x \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2c} & -\frac{r}{2c} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix}, \text{ e} \quad (3.6)$$

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{c}{r} \\ \frac{1}{r} & -\frac{c}{r} \end{bmatrix} \begin{bmatrix} v_{x^B} \\ \omega \end{bmatrix}. \quad (3.7)$$

Diferente da maioria dos robôs móveis, a velocidade lateral do RMRD é geralmente diferente de zero, isso vem da estrutura mecânica do RMRD que faz o deslizamento lateral necessário se o veículo mudar de orientação, e quando a velocidade do robô $v_{y^B} = 0$ não há deslocamento lateral. A velocidade angular ω e a velocidade lateral v_{y^B} ambas desaparecem quando o robô movimentar-se em linha reta, e o CIR vai para infinito ao longo do eixo Y^B . Já em uma trajetória em curva o CIR desloca x_{CIR}^B ao longo do eixo X^B . Se x_{CIR}^B ultrapassar a base das rodas do robô, o veículo desliza drasticamente com perda de estabilidade de movimento. Assim, para completar o modelo cinemático do RMRD, a restrição não-holonômica abaixo foi introduzida em [13]

$$v_{y^B} = x_{CIR}^B \dot{\psi}, \text{ com } x_{CIR}^B \in (-b, a), \quad (3.8)$$

de (3.1) obtém-se

$$v_{y^B} = -\sin \psi \dot{x} + \cos \psi \dot{y} \quad (3.9)$$

e substituindo a equação acima em (3.8) e escrevendo na forma Pfaffian, tem-se

$$\begin{bmatrix} -\sin \psi & \cos \psi & -x_{CIR}^B \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = A(\mathbf{q})\dot{\mathbf{q}} = 0, \quad (3.10)$$

sendo $\mathbf{q} = [x \ y \ \psi]^T$.

As velocidades $\dot{\mathbf{q}}$ podem ser expressas como

$$\dot{\mathbf{q}} = S(\mathbf{q})\boldsymbol{\eta}, \quad (3.11)$$

sendo $\boldsymbol{\eta} = [\omega_L \ \omega_R]^T$ e $S(\mathbf{q})$ é uma matriz cujas colunas estão no espaço nulo de $A(\mathbf{q})$, ou seja

$$S^T(\mathbf{q})A^T(\mathbf{q}) = 0, \quad (3.12)$$

e

$$S(\mathbf{q}) = \begin{bmatrix} \frac{r}{2} \left(\cos \psi - \frac{x_{CIR}^B}{c} \sin \psi \right) & \frac{r}{2} \left(\cos \psi + \frac{x_{CIR}^B}{c} \sin \psi \right) \\ \frac{r}{2} \left(\sin \psi + \frac{x_{CIR}^B}{c} \cos \psi \right) & \frac{r}{2} \left(\sin \psi - \frac{x_{CIR}^B}{c} \cos \psi \right) \\ \frac{r}{2c} & -\frac{r}{2c} \end{bmatrix}. \quad (3.13)$$

3.2 Modelo dinâmico

A Figura 3.3 mostra as forças de tração F_{ixB} e as forças resistivas longitudinais R_{ixB} que atuam no RMRD. E como há um acoplamento através de uma correia entre as rodas do lado esquerdo do RMRD e entre as rodas do lado direito do RMRD, assume-se que a atuação das rodas é igual em cada lado, ou seja, $F_{1xB} = F_{3xB}$ e $F_{2xB} = F_{4xB}$. As forças laterais R_{iyB} atuam sobre as rodas em consequência do deslizamento lateral. E o momento resistivo M_r em volta do centro de massa, induzido pelas forças R_{iyB} e R_{ixB} , se opõe ao momento M .

Para representar as forças longitudinais R_{xB} , as forças laterais R_{yB} , e o momento resistivo M_r , deve-se considerar mg como a carga gravitacional, sendo m a massa do robô e g a aceleração da gravidade, que é dividida sobre as rodas do robô, e introduzir o modelo Coulomb de atrito para o contato das rodas com a superfície. Deste modo, desde que o robô tenha uma linha

Figura 3.3: Forças de tração e resistivas do RMRD.

mediana longitudinal de simetria, obtém-se

$$\begin{aligned} R_{1zB} &= R_{2zB} = \frac{b}{2(a+b)}mg, \\ R_{3zB} &= R_{4zB} = \frac{a}{2(a+b)}mg. \end{aligned} \quad (3.14)$$

Em velocidades baixas, a carga lateral transferida devido às forças centrífugas sobre trajetórias curvadas pode ser desconsiderada. No caso de superfícies duras, pode-se assumir que o contato entre as rodas e o solo é retangular e que a carga vertical produz uma pressão uniforme distribuída. Nestas condições, $R_{ixB} = f_r R_{izB} \text{sgn}(v_{ixB})$, sendo f_r o coeficiente resistivo de rolagem, $\text{sgn}(\cdot)$ a função sinal, assumido independentemente da velocidade. Assim, a força resistiva longitudinal total é

$$R_{xB} = \sum_{i=1}^4 R_{ixB} = f_r \frac{mg}{2} (\text{sgn}(v_L) + \text{sgn}(v_R)). \quad (3.15)$$

Introduzindo o coeficiente de atrito lateral μ , a força lateral atuando sobre cada roda será $R_{iyB} = \mu R_{izB} \text{sgn}(v_{iyB})$. A força lateral total será

$$R_{yB} = \sum_{i=1}^4 R_{iyB} = \mu \frac{mg}{a+b} (b \text{sgn}(v_F) + a \text{sgn}(v_B)), \quad (3.16)$$

enquanto o momento resistivo será

$$\begin{aligned} M_r &= a(R_{1y^B} + R_{2y^B}) - b(R_{3y^B} + R_{4y^B}) + c[(R_{1x^B} + R_{3x^B}) - (R_{2x^B} + R_{4x^B})] \\ &= \mu \frac{abmg}{a+b} (\text{sgn}(v_F) - \text{sgn}(v_B)) + f_r cmg (\text{sgn}(v_L) - \text{sgn}(v_R)). \end{aligned} \quad (3.17)$$

As equações de movimento no sistema de coordenada do corpo para o RMRD mostrado na Figura 3.3 de massa m e de inércia I no centro de massa, são dadas por

$$\begin{aligned} ma_{x^B} &= 2F_{x^B1} + 2F_{x^B2} - R_{x^B}, \\ ma_{y^B} &= -R_{y^B}, \\ I\ddot{\psi} &= 2c(F_{x^B1} - F_{x^B2}) - M_r. \end{aligned} \quad (3.18)$$

Reescrevendo (3.18) de forma matricial, tem-se

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} a_{x^B} \\ a_{y^B} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 2F_{1x^B} + 2F_{2x^B} - R_{x^B} \\ -R_{y^B} \\ 2c(F_{1x^B} - F_{2x^B}) - M_r \end{bmatrix}. \quad (3.19)$$

Multiplicando a equação acima pela matriz de rotação $R(\psi)$, para escrever a equação de movimento no sistema de coordenadas inerciais, tem-se

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} R(\psi) \begin{bmatrix} a_{x^B} \\ a_{y^B} \\ \ddot{\psi} \end{bmatrix} = R(\psi) \begin{bmatrix} 2F_{1x^B} + 2F_{2x^B} - R_{x^B} \\ -R_{y^B} \\ 2c(F_{1x^B} - F_{2x^B}) - M_r \end{bmatrix}. \quad (3.20)$$

Substituindo (3.2) na equação acima e fazendo as devidas manipulações algébricas, obtém-se

$$M\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = E(\mathbf{q})\tau, \quad (3.21)$$

sendo

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \cos \psi R_{x^B} - \sin \psi R_{y^B} \\ \sin \psi R_{x^B} + \cos \psi R_{y^B} \\ M_r \end{bmatrix}, E(\mathbf{q}) = \begin{bmatrix} \frac{\cos \psi}{r} & \frac{\cos \psi}{r} \\ \frac{\sin \psi}{r} & \frac{\sin \psi}{r} \\ \frac{c}{r} & -\frac{c}{r} \end{bmatrix},$$

$$\tau = \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} = \begin{bmatrix} \tau_1 + \tau_3 \\ \tau_2 + \tau_4 \end{bmatrix} = \begin{bmatrix} 2\tau_1 \\ 2\tau_2 \end{bmatrix} = \begin{bmatrix} 2\tau_3 \\ 2\tau_4 \end{bmatrix}, \text{ e } \tau_i = rF_{ix}^B.$$

Incluindo a restrição não-holonômica no modelo dinâmico (3.21), como introduzido em [13], tem-se

$$M\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = E(\mathbf{q})\tau + A^T(\mathbf{q})\lambda, \quad (3.22)$$

sendo λ o vetor de multiplicadores de Lagrange correspondente a Equação (3.8).

3.3 Formulação do problema

O modelo dinâmico mostrado na Seção 3.2 é descrito em espaço de estado na forma quase linear a parâmetros variantes (quase-LPV), ou seja, os parâmetros das matrizes do sistema dependem dos estados. Diferenciando (3.11) com relação ao tempo, tem-se

$$\ddot{\mathbf{q}} = \dot{S}(\mathbf{q})\dot{\boldsymbol{\eta}} + S(\mathbf{q})\ddot{\boldsymbol{\eta}}. \quad (3.23)$$

Multiplicando (3.21) por $S^T(\mathbf{q})$ à esquerda

$$S^T(\mathbf{q})M(\mathbf{q})\ddot{\mathbf{q}} + S^T(\mathbf{q})C(\mathbf{q}, \dot{\mathbf{q}}) = S^T(\mathbf{q})E(\mathbf{q})\tau + S^T(\mathbf{q})A^T(\mathbf{q})\lambda. \quad (3.24)$$

Note que nesta passagem o termo $A^T\lambda$ é eliminado, pois $S^T(\mathbf{q})A^T = 0$ (as colunas de $S(\mathbf{q})$ estão no espaço nulo de $A(\mathbf{q})$). Substituindo (3.23) na equação acima, tem-se

$$\overline{M}\dot{\boldsymbol{\eta}} + \overline{N}\boldsymbol{\eta} + \overline{C} = \overline{E}\tau, \quad (3.25)$$

sendo

$$\overline{M} = S^T(\mathbf{q})M(\mathbf{q})S(\mathbf{q}),$$

$$\overline{N} = S^T(\mathbf{q})M(\mathbf{q})\dot{S}(\mathbf{q}),$$

$$\overline{C} = S^T(\mathbf{q})C(\mathbf{q}), \text{ e}$$

$$\overline{E} = S^T(\mathbf{q})E(\mathbf{q}).$$

Isolando $\dot{\eta}$ e acrescentado um distúrbio de torque $\mathbf{w} = [w_L \ w_R]^T$ em (3.25), obtém-se

$$\dot{\eta} = -\overline{M}^{-1}\overline{N}\eta - \overline{M}^{-1}\overline{C} + \overline{M}^{-1}\overline{E}\tau + \overline{M}^{-1}\overline{E}\mathbf{w}. \quad (3.26)$$

Somando e subtraindo $\dot{\eta}^d$ e $\overline{M}^{-1}\overline{N}\eta^d$ e definindo o estado como

$$\tilde{\mathbf{x}} = \begin{bmatrix} \omega_L - \omega_L^d \\ \omega_R - \omega_R^d \\ \delta_L - \delta_L^d \\ \delta_R - \delta_R^d \end{bmatrix}, \quad (3.27)$$

a representação quase-LPV, em espaço de estado, para o controle de acompanhamento de trajetória do RMRD é dada por

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} A(\mathbf{q}) & 0 \\ I & 0 \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} I \\ 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{w}, \quad (3.28)$$

sendo

$$A(\mathbf{q}) = -\overline{M}^{-1}\overline{N}, \quad (3.29)$$

$$B = \overline{M}^{-1}\overline{E}, \quad (3.30)$$

$$\mathbf{u} = -\dot{\eta}^d + A(\mathbf{q})\eta^d - \overline{M}^{-1}\overline{C} + B\tau, \quad (3.31)$$

ou

$$\tau = B^{-1}(\mathbf{u} + \dot{\eta}^d - A(\mathbf{q})\eta^d + \overline{M}^{-1}\overline{C}). \quad (3.32)$$

3.3.1 Conversão de torque para comandos de velocidade

Em virtude de certos robôs permitirem apenas comandos de velocidade linear e angular, deve-se converter comandos de torque em comandos de velocidade linear e angular, veja [19] e [46]. Considerando que o robô seja acionado por quatro motores DC, a equação do i -ésimo motor é dada por

$$u_i = Ri_i + L\frac{di_i}{dt} + u_{emf,i} \quad (3.33)$$

sendo u_i a tensão nos terminais do motor, i_i a corrente elétrica que circula na bobina do rotor, R e L a resistência e a indutância do fio da bobina do rotor e $u_{emf,i}$ a força contra- eletromotriz.

A força contra-eletromotriz é proporcional à velocidade do motor

$$u_{emf,i} = k_E \omega_{m,i} \quad (3.34)$$

sendo k_E a constante de tensão e $\omega_{m,i}$ a velocidade angular do rotor. Já o torque gerado pelo motor é proporcional à corrente que circula pelas suas bobinas

$$\tau_{m,i} = k_T i_i \quad (3.35)$$

sendo $\tau_{m,i}$ o torque do motor, k_T a constante de torque.

Como os atuadores são equipados com relação de engrenagem dada por n , tem-se

$$\tau_i = n k_T i_i, \quad (3.36)$$

$$\omega_{m,i} = n \omega_i. \quad (3.37)$$

Como se está interessado na operação do motor em regime permanente, quando a corrente i_i é constante, o termo $L \frac{di_i}{dt} = 0$, assim a Equação (3.33), torna-se

$$u_i - R i_i - u_{emf,i} = 0. \quad (3.38)$$

Substituindo (3.38) em (3.36) e usando (3.37) e (3.34), obtém-se

$$\tau_i = \frac{n k_T}{R} (u_i - n k_E \omega_i). \quad (3.39)$$

Partindo de (3.39), a tensão que deve ser aplicada nos motores para a obtenção do torque desejado nas rodas do robô é

$$u_L = \frac{R \tau_L}{2 n k_T} + n k_E \omega_L \quad (3.40)$$

$$u_R = \frac{R \tau_R}{2 n k_T} + n k_E \omega_R \quad (3.41)$$

sendo $u_L = u_1 = u_3$ e $u_R = u_2 = u_4$. As velocidades angulares de controle das rodas são dadas

por

$$\begin{aligned}\omega_{L,u} &= \frac{1}{nk_E} \left(u_L - \frac{R}{2k_T n} \tau_{L,a} \right), \\ \omega_{R,u} &= \frac{1}{nk_E} \left(u_R - \frac{R}{2k_T n} \tau_{R,a} \right),\end{aligned}\tag{3.42}$$

sendo $\tau_{L,a}$ e $\tau_{R,a}$ os torques atuais do robô. As velocidades de controle linear e angular são

$$\begin{bmatrix} v_{x^B,u} \\ \omega_u \end{bmatrix} = r \begin{bmatrix} \frac{\omega_{L,u} + \omega_{R,u}}{2} \\ \frac{-\omega_{L,u} + \omega_{R,u}}{2} \end{bmatrix}.\tag{3.43}$$

Para se obter as velocidades linear e angular de referência que serão enviadas como comandos para o robô, consideram-se as equações do controlador PD abaixo

$$\begin{bmatrix} v_{x^B,u} \\ \omega_u \end{bmatrix} = \begin{bmatrix} k_{PT}(v_{x^B,ref} - v_{x^B}) - k_{DT}\dot{v}_{x^B} \\ k_{PR}(\omega_{ref} - \omega) - k_{DR}\dot{\omega} \end{bmatrix},\tag{3.44}$$

sendo $v_{x^B,ref}$ e ω_{ref} as velocidades linear e angular de referência, k_{PT} e k_{DT} os ganhos proporcional e derivativo de translação e k_{PR} e k_{DR} os ganhos proporcional e derivativo de rotação, estes são os ganhos que estão no robô. As variáveis $\dot{v}_{x^B,ref}$ e $\dot{\omega}_{ref}$ foram omitidas para simplificar o modelo.

Isolando a velocidade linear $v_{x^B,ref}$ e angular ω_{ref} de referência, tem-se

$$\begin{aligned}v_{x^B,ref} &= \frac{v_{x^B,u}}{k_{PT}} + v_{x^B} + \frac{k_{DT}}{k_{PT}}\dot{v}_{x^B}, \\ \omega_{ref} &= \frac{\omega_u}{k_{PR}} + \omega + \frac{k_{DR}}{k_{PR}}\dot{\omega}.\end{aligned}\tag{3.45}$$

3.3.2 Parâmetros do robô

Os parâmetros do Pioneer 3-AT retirados do *datasheet* do motor do robô e da dissertação de mestrado [40], seguem abaixo

3.4 Determinação da trajetória

Como o robô móvel com rodas deslizantes possui restrições de movimentos a determinação da trajetória é realizada em duas etapas. A primeira é a geração da trajetória de referência e a

	Parâmetro	Valor	Unidade
Motor	k_T	0.0230	Nm/A
	k_E	0.0230	V s/rad
	n	38.3	
	R	0.71	Ω
Robô	I	0.413	Kgm ²
	m	40	Kg
	f_r	0.043	
	μ	0.506	
	x_{CIR}	0.008	m
	a	0.138	m
	b	0.122	m
	c	0.1975	m
	r	0.1075	m

Tabela 3.1: Parâmetros do Pioneer 3-AT.

segunda a geração da trajetória desejada, o qual o robô pode executar.

3.4.1 Trajetória de referência

Neste trabalho, define-se a trajetória de referência do robô móvel de roda deslizante através de um polinômio de quinto grau descrito por

$$q_i^{ref}(t) = a_{i0} + a_{i1}\Delta t + a_{i2}\Delta t^2 + a_{i3}\Delta t^3 + a_{i4}\Delta t^4 + a_{i5}\Delta t^5 \quad (3.46)$$

sendo $\Delta t = t - t_0$ e $i = 1, 2, \dots, l$, sendo l o número de eixos que se deseja gerar trajetórias de referência. E suas derivadas são dadas por

$$\dot{q}_i^{ref}(t) = a_{i1} + 2a_{i2}\Delta t + 3a_{i3}\Delta t^2 + 4a_{i4}\Delta t^3 + 5a_{i5}\Delta t^4 \quad (3.47)$$

$$\ddot{q}_i^{ref}(t) = 2a_{i2} + 6a_{i3}\Delta t + 12a_{i4}\Delta t^2 + 20a_{i5}\Delta t^3 \quad (3.48)$$

$$\ddot{\ddot{q}}_i^{ref}(t) = 6a_{i3} + 24a_{i4}\Delta t + 60a_{i5}\Delta t^2 \quad (3.49)$$

satisfazendo as seguintes condições

$$\begin{aligned}
 q_i^{ref}(t_0) &= q_{i0} & q_i^{ref}(t_f) &= q_{if} \\
 \dot{q}_i^{ref}(t_0) &= \dot{q}_{i0} & \dot{q}_i^{ref}(t_f) &= \dot{q}_{if} \\
 \ddot{q}_i^{ref}(t_0) &= \ddot{q}_{i0} & \ddot{q}_i^{ref}(t_f) &= \ddot{q}_{if} \\
 \ddot{\ddot{q}}_i^{ref}(t_0) &= \ddot{\ddot{q}}_{i0} & \ddot{\ddot{q}}_i^{ref}(t_f) &= \ddot{\ddot{q}}_{if}.
 \end{aligned} \tag{3.50}$$

Desse modo, podemos obter os parâmetros do polinômio do quinto grau pela equação abaixo

$$\mathbf{a}_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & \Delta t & \Delta t^2 & \Delta t^3 & \Delta t^4 & \Delta t^5 \\ 0 & 1 & 2\Delta t & 3\Delta t^2 & 4\Delta t^3 & 5\Delta t^4 \\ 0 & 0 & 2 & 6\Delta t & 12\Delta t^2 & 20\Delta t^3 \end{bmatrix}^{-1} \mathbf{q}_i \tag{3.51}$$

sendo $\mathbf{q}_i = [q_{i0} \ \dot{q}_{i0} \ \ddot{q}_{i0} \ q_{if} \ \dot{q}_{if} \ \ddot{q}_{if}]^T$ e $\mathbf{a}_i = [a_{i0} \ a_{i1} \ a_{i2} \ a_{i3} \ a_{i4} \ a_{i5}]^T$.

Para o caso do robô móvel de roda deslizante $l = 2$ pois deseja-se gerar trajetórias para o eixo X e o eixo Y , levando em consideração a condição inicial $x_0, \dot{x}_0, \ddot{x}_0, \ddot{\ddot{x}}_0, y_0, \dot{y}_0, \ddot{y}_0, \ddot{\ddot{y}}_0$ e a condição final que se deseja alcançar $x_f, \dot{x}_f, \ddot{x}_f, \ddot{\ddot{x}}_f, y_f, \dot{y}_f, \ddot{y}_f, \ddot{\ddot{y}}_f$. E através da variação das equações (3.46) - (3.49) no tempo t obtém-se a trajetória de referência do robô $x^{ref}(t)$ e $y^{ref}(t)$ e suas derivadas a cada instante.

3.4.2 Trajetória desejada

Nesta seção, utiliza-se o controlador baseado na cinemática proposto por [37], para gerar as velocidades desejadas para o problema de acompanhamento de trajetória de referência. Considere o erro $q_e = [x_e \ y_e \ \psi_e]^T$, entre a postura de referência $P^{ref} = [x^{ref}, y^{ref}, \psi^{ref}]^T$, e a postura real do RMR $P = [x, y, \psi]^T$, dada por

$$\begin{aligned}
 x_e &= \cos\psi(x_r - x_c) + \sin\psi(y_r - y_c), \\
 y_e &= -\sin\psi(x_r - x_c) + \cos\psi(y_r - y_c), \\
 \psi_e &= \psi^{ref} - \psi,
 \end{aligned} \tag{3.52}$$

sendo $[x^{ref}, y^{ref}]^T = traj^{ref}$ a trajetória de referência escolhida e $\psi^{ref} = \tan^{-1}(\dot{y}^{ref}/\dot{x}^{ref})$. As velocidades desejadas linear (v^d) e angular (w^d) de um RMR são dadas por

$$\begin{aligned} v^d &= v^{ref} \cos \psi_e + k_x x_e, \\ w^d &= w^{ref} + v^{ref} (k_y y_e + k_\psi \sin \psi_e), \end{aligned} \quad (3.53)$$

sendo k_x, k_y, k_ψ constantes,

$$v^{ref} = \sqrt{(\dot{x}^{ref})^2 + (\dot{y}^{ref})^2} \text{ e } w^{ref} = \dot{\psi}^{ref}. \quad (3.54)$$

3.5 Controle robusto

Esta seção apresenta o regulador robusto recursivo desenvolvido em [15]. Este controlador será usado para o controle robusto do robô móvel de roda deslizante apresentado neste capítulo.

3.5.1 Regulador robusto recursivo

Considere o seguinte sistema linear de tempo discreto sujeito a incertezas paramétricas:

$$\mathbf{x}_{k+1} = (F_k + \delta F_k) \mathbf{x}_k + (G_k + \delta G_k) \mathbf{u}_k \quad (3.55)$$

sendo $F_k \in \mathbb{R}^{n \times n}$ e $G_k \in \mathbb{R}^{n \times m}$ as matrizes de parâmetros nominais, $\mathbf{x}_k \in \mathbb{R}^n$ o vetor de estado, $\mathbf{u}_k \in \mathbb{R}^m$ a entrada de controle e estado inicial $\mathbf{x}_0$ assumido conhecido. Sejam $\delta F_k \in \mathbb{R}^{n \times n}$ e $\delta G_k \in \mathbb{R}^{n \times m}$ as matrizes de incertezas desconhecidas modeladas como

$$\begin{bmatrix} \delta F_k & \delta G_k \end{bmatrix} = H_k \Delta_k \begin{bmatrix} E_{F_k} & E_{G_k} \end{bmatrix} \quad (3.56)$$

sendo $H_k \in \mathbb{R}^{n \times k}$, $E_F \in \mathbb{R}^{l \times n}$, $E_{G_k} \in \mathbb{R}^{l \times m}$ matrizes conhecidas arbitrárias com $|\Delta_k| \leq 1$.

Considere o seguinte problema de otimização:

$$\min_{\mathbf{x}_{k+1}, \mathbf{u}_k} \max_{\delta F_k, \delta G_k} \{J_k(\mathbf{x}_{k+1}, \mathbf{u}_k, \delta F_k, \delta G_k)\} \quad (3.57)$$

sendo J_k o funcional custo quadrático definido por

$$J_k(\mathbf{x}_{k+1}, \mathbf{u}_k, \delta F_k, \delta G_k) = \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_k \end{bmatrix}^T \begin{bmatrix} P_{k+1} & 0 \\ 0 & R_k \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_k \end{bmatrix} + \\ + \left\{ \left(\begin{bmatrix} 0 & 0 \\ I & -G_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ I & -\delta G_k \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_k \end{bmatrix} - \left(\begin{bmatrix} -I \\ F_k \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0 \\ \delta F_k \end{bmatrix} \mathbf{x}_k \right) \right\}^T \begin{bmatrix} Q_k & 0 \\ 0 & \mu I \end{bmatrix} \{\bullet\}$$

com $\mu > 0$ fixado, $P_{i+1} > 0$, $Q_k > 0$ e $R_k > 0$. Então, a solução ótima do problema para $\mu > 0$ é dada por

$$\begin{bmatrix} \mathbf{x}_{k+1}^*(\mu) \\ \mathbf{u}_k^*(\mu) \end{bmatrix} = \begin{bmatrix} L_k \\ K_k \end{bmatrix} \mathbf{x}_k \quad (3.58)$$

$$\begin{bmatrix} L_k \\ K_k \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{k+1}^{-1} & 0 & 0 & 0 & I & 0 \\ 0 & R_k^{-1} & 0 & 0 & 0 & I \\ 0 & 0 & Q_k^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum_k(\mu, \hat{\lambda}) & \mathcal{I} & \mathcal{G}_k \\ I & 0 & 0 & \mathcal{I}^T & 0 & 0 \\ 0 & I & 0 & -\mathcal{G}_k^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ -I \\ \mathcal{F}_k \\ 0 \\ 0 \end{bmatrix} \quad (3.59)$$

sendo

$$\sum_k = \begin{bmatrix} \mu^{-1}I - \hat{\lambda}_k^{-1}H_k H_k^T & 0 \\ 0 & \hat{\lambda}_k^{-1} \end{bmatrix}, \mathcal{I} = \begin{bmatrix} I \\ 0 \end{bmatrix}, \mathcal{G}_k = \begin{bmatrix} G_k \\ E_{G_k} \end{bmatrix}, \mathcal{F}_k = \begin{bmatrix} F_k \\ E_{F_k} \end{bmatrix} \quad (3.60)$$

e $\hat{\lambda}_k$ é dado sobre o intervalo $[\|\mu H_k^T H_k\|, +\infty)$. Além disso, o custo ótimo $J_k^*(\mu)$ é dado por

$$J_k^*(\mu) := J_k(\mathbf{x}_{k+1}^*, \mathbf{u}_k^*, \delta F_k, \delta G_k) = \mathbf{x}_k^T (L_k^T P_{k+1} L_k + K_k^T R_{k+1} K_k + \\ + Q_k + (\mathcal{I} L_k - \mathcal{G}_k K_k - \mathcal{F}_k)^T \sum_k^{-1}(\mu, \hat{\lambda}_k) (\mathcal{I} L_k - \mathcal{G}_k K_k - \mathcal{F}_k)) \mathbf{x}_k. \quad (3.61)$$

3.6 Resultados simulados

3.6.1 Regulador robusto recursivo

O sistema em espaço de estado (3.28) é discretizado e o controlador robusto apresentado no Capítulo 3.5 é utilizado para realizar o controle do robô móvel de roda deslizante. A simulação foi realizada conforme diagrama simplificado mostrado na Figura 3.4. O robô inicia na posição $x_0 = 0$, $y_0 = 0$ e orientação $\psi_0 = \pi$, em seguida segue para os pontos de passagem $x_f = 3$ e $y_f = 3$, $x_f = -3$ e $y_f = 3$, $x_f = -3$ e $y_f = -3$ e $x_f = 0$ e $y_f = 0$, com a orientação definida pelo movimento. A trajetória de referência é gerada pelo polinômio de quinto grau e em seguida a trajetória desejada que o robô pode realizar é obtida. O erro da trajetória utilizado pelo controlador é composto da seguinte forma

$$\tilde{\mathbf{x}} = \begin{bmatrix} \omega_e \\ \delta_e \end{bmatrix}, \quad (3.62)$$

sendo

$$\omega_e = \begin{bmatrix} \omega_L - \omega_L^d \\ \omega_R - \omega_R^d \end{bmatrix}, \quad \delta_e = \begin{bmatrix} \delta_L - \delta_L^d \\ \delta_R - \delta_R^d \end{bmatrix} \approx S^+ \begin{bmatrix} x - x^d \\ y - y^d \\ \psi - \psi^d \end{bmatrix},$$

e S^+ a pseudo inversa de (3.13). Obtém-se a lei de controle e em seguida os torque são calculados e enviados para o robô. A simulação considera o controle interno e saturação do torque do robô Pioneer 3AT.

Figura 3.4: Diagrama de simulação do robô.

Os ganhos do controlador cinemático foram escolhidos de maneira heurística e valem

$$k_x = 0.5, \quad k_y = 3, \quad k_\psi = 0.5. \quad (3.63)$$

E as matrizes do controlador robusto foram escolhidas também de maneira heurística, da seguinte forma

$$Q = \begin{bmatrix} I_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & I_{2 \times 2} \end{bmatrix}, \quad R = I_{2 \times 2}, \quad E_F = [0.1 \quad 0.1 \quad 0 \quad 0], \quad E_G = [0.1 \quad 0.1], \quad (3.64)$$

e $\mu = \infty$. Na simulação foram aplicados distúrbios externos como mostrados na Figura 3.5.

Figura 3.5: Distúrbios externos.

Os resultados da simulação podem ser vistos nas figuras 3.6 - 3.13. E conforme observado nos gráficos, verifica-se que o robô móvel com rodas deslizante consegue atingir o seu objetivo.

Figura 3.6: Trajetória do robô.

Figura 3.7: Torques do robô.

Figura 3.8: Trajetória no eixo X do robô.

Figura 3.9: Velocidades no eixo X do robô.

Figura 3.10: Trajetória no eixo Y do robô.

Figura 3.11: Velocidades no eixo Y do robô.

Figura 3.12: Deslocamento angular do robô.

Figura 3.13: Velocidades angulares do robô.

3.7 Visão geral de um robô helicóptero

A descrição do comportamento de voo de um helicóptero, como o da Figura 3.14, sendo (X^B, Y^B, Z^B) o sistema de coordenadas do corpo, em um modelo matemático, apresenta dificuldades desafiadoras [54]. O robô helicóptero pode ser visto como um arranjo complexo de subsistemas interagindo. E estes subsistemas podem ser divididos em quatro partes [38], dinâmica dos atuadores, dinâmica das hélices, geração de força e torque, e dinâmica do corpo rígido, como pode ser visto na Figura 3.15.

Figura 3.14: Sistema de coordenadas do corpo do helicóptero.

Figura 3.15: Dinâmica do helicóptero.

O primeiro subsistema recebe as quatro entradas de controle do helicóptero que são $\mathbf{u} = [u_{lat} \ u_{long} \ u_{col} \ u_{ped}]^T$. O comando lateral u_{lat} faz com que o helicóptero rotacione sobre o eixo X^B realizando o movimento de rolagem. O comando longitudinal u_{long} faz com que o helicóptero rotacione sobre o eixo Y^B realizando o movimento de arfagem. Os comandos u_{lat} e u_{long} formam os comandos cíclicos. O comando de coletivo u_{col} faz com que o helicóptero realize um movimento vertical, elevação. Esta elevação é um resultado das forças aerodinâmicas exercidas sobre as pás das hélices do rotor principal sobre o helicóptero. O comando de leme u_{ped} faz com que o helicóptero rotacione sobre o eixo Z^B realizando o movimento de guinada.

As entradas de $\mathbf{u}$ são definidas entre $-1 \leq u_{(.)} \leq 1$ e são repassadas para os atuadores do helicóptero, que são servos motores acionados por sinais PWM (do inglês *pulse-width modulation*, modulação por largura de pulso). Estes servos por sua vez geram desvios angulares $\delta = [\delta_{lat} \ \delta_{lon} \ \delta_{col} \ \delta_{ped}]^T$ que atuam na dinâmica das asas.

O segundo subsistema trata da dinâmica das hélices, o qual descreve o movimento *flapping* do rotor principal, da barra estabilizante, e dos empuxos do rotor principal T_{MR} e da cauda T_{TR} .

Definindo o plano HP (do inglês *hub plane*) paralelo aos eixos X^B e Y^B com origem no eixo do rotor, e o plano TPP (do inglês *tip path plane*) definido quando o eixo do rotor é inclinado devido a alteração da orientação do empuxo. Estes planos podem ser vistos na Figura 3.16. O ângulo de azimuth Ψ é usado para descrever a posição das hélices, definido como 0° na direção da cauda do helicóptero e aumentando com a rotação da hélice no sentido horário.

Figura 3.16: Ilustração dos planos HP e TPP.

Os ângulos de *flapping* lateral do rotor principal e da barra estabilizante, β_{1s} e κ_{1s} , respectivamente, e o ângulo da bailarina A são ilustrados na Figura 3.17.

Os ângulos de *flapping* longitudinal do rotor principal e da barra estabilizante, β_{1c} e κ_{1c} , respectivamente, e o ângulo da bailarina B são ilustrados na Figura 3.18.

As forças do rotor principal são ilustradas na Figura 3.19. As forças de empuxo são definidas perpendicularmente ao TPP, enquanto as forças de elevação e propulsão são aos componentes vertical e horizontal, respectivamente.

O terceiro subsistema trata das forças, $\mathbf{f}^B$, e torques, $\boldsymbol{\tau}^B$, que são gerados pelo rotor principal

Figura 3.17: *Flapping* lateral.

Figura 3.18: *Flapping* longitudinal.

e de cauda, gravidade, e arrasto sobre a fuselagem. As forças e torques devido à gravidade e ao arrasto são obtidas diretamente, no entanto as forças e torques geradas pelos rotores são de descrição complexa, visto que o fluxo de ar através dos rotores são fatores que influenciam na descrição das forças e torques gerados pelos rotores.

E o quarto subsistema trata da dinâmica do corpo rígido. As forças, $\mathbf{f}^B$, e torques, $\boldsymbol{\tau}^B$, causam acelerações angulares e de translação. Essas equações são então derivadas através das equações de movimento de Newton e Euler. Desse modo obtendo a posição $\mathbf{P}$ e velocidade $\mathbf{V}^B$ no centro de gravidade do helicóptero, a atitude $\boldsymbol{\Theta}$ e as velocidades angulares $\boldsymbol{\omega}$.

Devido a complexidade da dinâmica do helicóptero, será considerado o modelo do helicóptero

Figura 3.19: Definição das forças no rotor principal.

como um corpo rígido incorporado com o processo de geração de forças e momentos que será mostrado na próxima seção. Atualmente, encontra-se em estudo um modelo que leve em conta os rotores, fuselagem, dinâmica dos atuadores, motor e hélices. Este modelo será utilizado para a simulação de voo do helicóptero, simulações de controladores de voo, e futuramente como uma ferramenta de auxílio de projeto da estrutura física de um helicóptero autônomo.

3.8 Dinâmica do corpo rígido

Considere o helicóptero descrito na Figura 3.20. O conjunto de forças e momentos atuando sobre o helicóptero pode ser organizado por componentes: $()_{MR}$ para rotor principal, $()_{TR}$ para rotor de cauda, $()_{FUS}$ para fuselagem, $()_{VT}$ para o estabilizador vertical e $()_{HT}$ para o estabilizador horizontal. As variáveis x , y e z descrevem a posição da origem do sistema de coordenada do corpo B no sistema de coordenada de navegação N . Os ângulos de Euler, ϕ , θ e ψ , definem os ângulos entre o sistema de coordenada do corpo e o sistema de coordenada de navegação. No sistema de coordenada do corpo as variáveis u , v e w descrevem as velocidades de translação e p , q e r as velocidades angulares que descrevem os movimentos de rolagem, arfagem e guinada.

As equações de movimento de um helicóptero pode ser escrita com relação ao sistema de coordenada do corpo, o qual é fixado no centro de gravidade, CG , do helicóptero. Como mostrado em [54], as equações do corpo rígido sujeitas às forças $\mathbf{f}^B = [X \ Y \ Z]^T$ e torques $\boldsymbol{\tau}^B = [L \ M \ N]^T$ do corpo aplicadas ao centro de gravidade são dadas por Newton-Euler, e podem ser escritas como

$$\begin{aligned}
 \dot{u} &= vr - wq - g\sin\theta + X/m, \\
 \dot{v} &= wp - ur + g\cos\theta\sin\phi + Y/m, \\
 \dot{w} &= uq - vp + g\cos\theta\cos\phi + Z/m, \\
 \dot{p} &= qr(I_{yy} - I_{zz})/I_{xx} + L/I_{xx}, \\
 \dot{q} &= rp(I_{zz} - I_{xx})/I_{yy} + M/I_{yy}, \\
 \dot{r} &= pq(I_{xx} - I_{yy})/I_{zz} + N/I_{zz}, \\
 [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T &= \Psi(\phi, \theta, \psi)[p \ q \ r]^T, \text{ e} \\
 [\dot{x} \ \dot{y} \ \dot{z}]^T &= R(\phi, \theta, \psi)[u \ v \ w]^T,
 \end{aligned} \tag{3.65}$$

sendo m a massa do helicóptero, g a gravidade, $R(\phi, \theta, \psi) \in SO(3)$ uma matriz de transformação

Figura 3.20: Momentos e forças atuando sobre o helicóptero.

da velocidade linear

$$R = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \theta \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}, \quad (3.66)$$

$\Psi(\phi, \theta, \psi)$ descreve uma transformação na velocidade angular

$$\Psi = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix}, \quad (3.67)$$

I_{xx} , I_{yy} , I_{zz} são momentos de inércia, e

$$\begin{aligned}
X &= X_{MR} + X_F, \\
Y &= Y_{MR} + Y_{FUS} + Y_{TR} + Y_{VT}, \\
Z &= Z_{MR} + Z_{FUS} + Z_{HT}, \\
L &= L_{MR} + L_{VT} + L_{TR}, \\
M &= M_{MR} + M_{HT} + M_{TR}, \text{ e} \\
N &= -Q_e + N_{VT} + N_{TR},
\end{aligned} \tag{3.68}$$

são forças e momentos induzidas do rotor, veja [27].

3.9 Modelo em espaço de estado identificado para voo pairado

Nesta seção é apresentado o modelo identificado para voo pairado mostrado em [18, 50]. Conforme mostrado em [50] as equações do corpo rígido do helicóptero Eq. 3.65 podem ser reescritas da seguinte forma

$$\begin{aligned}
\dot{u} &= (-w_0q + v_0r) - g\theta + X_uu + X_vv + X_w w + X_rr + X_{\beta_{1c}}\beta_{1c} + X_{\delta_{ped}}\delta_{ped} \\
&\quad + X_{\delta_{col}}\delta_{col}, \\
\dot{v} &= (-u_0r + w_0p) - g\phi + Y_uu + Y_vv + Y_w w + Y_rr + Y_{\beta_{1c}}\beta_{1c} + Y_{\delta_{ped}}\delta_{ped} \\
&\quad + Y_{\delta_{col}}\delta_{col}, \\
\dot{w} &= (-v_0p + u_0q) - g\theta_0\theta + Z_uu + Z_vv + Z_w w + Z_pp + Z_qq + Z_rr + Z_{\delta_{lat}}\delta_{lat}, \\
&\quad + Z_{\delta_{lon}}\delta_{lon} + Z_{\delta_{ped}}\delta_{ped} + \frac{-p\pi R^2(\Omega R)^2}{m}C_T, \\
\dot{p} &= L_uu + L_vv + L_w w + L_rr + L_{\beta_{1s}}\beta_{1s} + L_{\delta_{ped}}\delta_{ped} + L_{\delta_{col}}\delta_{col}, \\
\dot{q} &= M_uu + M_vv + M_w w + M_rr + M_{\beta_{1c}}\beta_{1c} + M_{\delta_{ped}}\delta_{ped} + M_{\delta_{col}}\delta_{col}, \\
\dot{r} &= N_uu + N_vv + N_w w + N_pp + N_qq + N_rr + N_{\delta_{lat}}\delta_{lat} + N_{\delta_{lon}}\delta_{lon} + Z_{\delta_{ped}}\delta_{ped} \\
&\quad + N_{\delta_{col}}\delta_{col}, \text{ e} \\
\dot{r}_{fb} &= K_rr + K_{r_{fb}}r_{fb},
\end{aligned} \tag{3.69}$$

sendo r_{fb} uma taxa de realimentação de guinada

$$C_T = \frac{0.543}{C_0\Omega^2 R}\dot{\nu} + \frac{4\bar{\nu}_0}{\Omega R}\nu + \frac{4\bar{\nu}_0}{3\Omega}\dot{\beta}_0, \tag{3.70}$$

o coeficiente de empuxo, $\bar{\nu}$ a relação do influxo, R o raio do rotor, Ω a velocidade do rotor, $Y_{\beta_{1c}}$ e $X_{\beta_{1s}}$ as derivadas das forças do rotor, $M_{\beta_{1c}}$ e $L_{\beta_{1s}}$ as derivadas dos momentos do rotor, e $()_u$, $()_v$, $()_w$, $()_p$, $()_q$, $()_r$ as derivadas das velocidades devido ao efeito aerodinâmico. E $()_0$ valores de estado fixo.

As equações da dinâmica quase-fixa são acopladas à dinâmica *coning*/influxo através do coeficiente de empuxo C_T

$$\ddot{\beta}_0 = \frac{\Omega\gamma}{8}\dot{\beta}_0 - \Omega^2\beta_0 - \frac{\Omega\gamma}{6R}\nu + \frac{\Omega^2\gamma}{8}K_{\theta_0}\delta_{col}, \text{ e} \quad (3.71)$$

$$\dot{\nu} = -\frac{75\pi\Omega}{32}\left(\bar{\nu}_0 + \frac{C_{L\alpha}\sigma}{16}\right)C_0\nu - \nu_{\dot{\beta}_0}\dot{\beta}_0 + \frac{25\pi\Omega^2R}{32}\left(\frac{C_{L\alpha}\sigma}{8}\right)C_0K_{\theta_0}\delta_{col}, \quad (3.72)$$

sendo $C_{L\alpha}$ a inclinação da curva de elevação, C_0 a constante de influxo da teoria Carpenter-Fridovich, γ o número de bloqueio e K_{θ_0} a transformação da barra do coletivo para o ângulo de arfagem da hélice.

As dinâmicas do rotor são modeladas como equações de primeira ordem acopladas e simplificadas da equações dinâmicas acopladas de segunda ordem do plano TPP

$$\dot{\beta}_{1c} = \frac{1}{\tau_f}\left(-\beta_{1c} + M_{f\beta_{1s}}\beta_{1s} + \tau_f q + M_{f\delta_{lon}}\delta_{lon} + M_{f\delta_{lat}}\delta_{lat} + M_{f\kappa_{1c}}\kappa_{1c}\right), \text{ e} \quad (3.73)$$

$$\dot{\beta}_{1s} = \frac{1}{\tau_f}\left(-\beta_{1s} + L_{f\beta_{1c}}\beta_{1c} + \tau_f p + L_{f\delta_{lon}}\delta_{lon} + L_{f\delta_{lat}}\delta_{lat} + L_{f\kappa_{1s}}\kappa_{1s}\right), \quad (3.74)$$

sendo τ_f a constante de tempo do rotor principal, $M_{()}$ e $L_{()}$ são derivadas.

A barra estabilizante é modelada como um rotor secundário conectado ao eixo do rotor principal através de uma articulação sem restrição *teetering*

$$\dot{\kappa}_{1c} = \frac{1}{\tau_s}\left(-\kappa_{1c} + \tau_s q + M_{s\delta_{lon}}\delta_{lon}\right), \text{ e} \quad (3.75)$$

$$\dot{\kappa}_{1s} = \frac{1}{\tau_s}\left(-\kappa_{1s} + \tau_s p + L_{s\delta_{lat}}\delta_{lat}\right), \quad (3.76)$$

sendo τ_s a constante de tempo da barra estabilizante, $M_{s\delta_{lon}}$ e $L_{s\delta_{lat}}$ derivadas. Os valores das derivadas são identificados em [18] através do método de identificação no domínio da frequência utilizando o programa CIFER[®]. Utilizando as eqs. (3.69)-(3.76), pode-se escrever o seguinte modelo em espaço de estado

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (3.77)$$

sendo $\mathbf{x}$ o estado e $\mathbf{u}$ a entrada dados por:

$$\mathbf{x} = \begin{bmatrix} u & v & w & p & q & r & r_{fb} & \beta_{1c} & \beta_{1s} & \kappa_{1c} & \kappa_{1s} & \dot{\beta}_0 & \beta_0 & \nu \end{bmatrix}^T, \text{ e} \quad (3.78)$$

$$\mathbf{u} = [\delta_{lat} \quad \delta_{lon} \quad \delta_{col} \quad \delta_{ped}]^T. \quad (3.79)$$

Assim, obtém-se a descrição do modelo em espaço de estado identificado de voo pairado. Este modelo será utilizado nas simulações apresentadas neste trabalho com o helicóptero.

3.10 Controle de posição e ângulo de proa de um robô helicóptero

Para o controle de posição e ângulo de proa de um robô helicóptero, pretende-se utilizar a metodologia apresentada em [3] da combinação em cascata de três metodologias de controle bem estabelecidas. Inicialmente, utiliza-se um regulador linear quadrático (LQR, em inglês) para estabilizar os pólos do sistema, depois utiliza-se um controlador baseado em linearização por realimentação (FLC, em inglês) para desacoplar os pares de entrada/saída e, finalmente, utiliza-se um controlador PD para assegurar um bom acompanhamento de trajetória. A saída do estimador foi empregada como entrada para a etapa de controle. Para isso, utiliza-se um modelo reduzido, que pode ser obtido a partir da Eq. (3.77) eliminando-se as estimativas dos estados associados aos pólos menos dominantes do sistema. Assim, o novo estado contém apenas posição, atitude, velocidades lineares e angulares e a taxa de realimentação da guinada

$$\mathbf{x}_c = [x^B \ y^B \ z^B \ u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r \ r_{fb}]^T, \quad (3.80)$$

sendo

$$x^B = \int u dt, \quad y^B = \int v dt, \quad z^B = \int w dt. \quad (3.81)$$

Dessa forma, o modelo para o projeto de controle do helicóptero é dado por

$$\dot{\mathbf{x}}_c = A_c \mathbf{x}_c + B_c \mathbf{u}_c, \text{ e} \quad (3.82)$$

$$\mathbf{y}_c = C_c \mathbf{x}_c, \quad (3.83)$$

com

$$\begin{aligned} \mathbf{u}_c &= [\delta_{lat} \ \delta_{lon} \ \delta_{col} \ \delta_{ped}]^T, \text{ e} \\ \mathbf{y}_c &= [x^B \ y^B \ z^B \ \psi]^T. \end{aligned} \quad (3.84)$$

O termo r_{fb} não pode ser medido diretamente em aplicações praticas, assim ele é estimado pela equação abaixo

$$\hat{r}_{fb} = -\frac{K_r}{K_{r_{fb}}}. \quad (3.85)$$

3.11 Regulador linear quadrático

O regulador linear quadrático consiste em encontrar uma matriz de ganho fixo K_{LQR} para a lei de controle de realimentação de estado

$$\mathbf{u}_c = -K_{LQR}\mathbf{x}_c \quad (3.86)$$

minimize o funcional de custo

$$J = \int_{t_0}^{\infty} (\mathbf{x}_c^T Q \mathbf{x}_c + \mathbf{u}_c^T R \mathbf{u}_c) dt \quad (3.87)$$

para que o sistema (3.82) alcance um estado desejado $\mathbf{x}^d = 0$, resultando em

$$\begin{aligned} \dot{\mathbf{x}}_c &= A_c \mathbf{x}_c + B_c \mathbf{u}_c, \\ &= A_c \mathbf{x}_c - B_c K_{LQR} \mathbf{x}_c, \\ &= (A_c - B_c K_{LQR}) \mathbf{x}_c. \end{aligned} \quad (3.88)$$

Desde que o sistema de controle seja estável sob a realimentação de estados via LQR, pode-se utilizar outra metodologia de controle para guiar o sistema para qualquer saída admissível.

3.12 Controlador baseado em linearização por realimentação

Controle através de linearização por realimentação de estados consiste em determinar uma transformação e uma lei de realimentação com duas funções: linearizar o sistema e desacoplar os estados, permitindo o controle de cada saída independentemente das outras através de um simples controle linear. Como o sistema considerado é linear, este controlador foi utilizado apenas com o objetivo de desacoplar os estados e assim poder fechar a malha de controle com um controlador proporcional-derivativo. De um ponto de vista simplificado, a metodologia requer que se calcule sucessivas derivadas, em relação ao tempo, das saídas do sistema até que todas as entradas apareçam explicitamente nas equações diferenciais resultantes. A esta altura, a equação é invertida e a entrada é calculada como uma função explícita das matrizes de parâmetros do estado do sistema. Escolhe-se uma entrada de controle $\mathbf{u}_c$ como uma combinação da realimentação

de estados do regulador e uma entrada auxiliar $\mathbf{v}_c$

$$\mathbf{u}_c = \mathbf{v}_c - K_{LQR}\mathbf{x}_c. \quad (3.89)$$

Substituindo a expressão acima em (3.82), obtém-se

$$\dot{\mathbf{x}}_c = (A_c - B_c K_{LQR})\mathbf{x}_c + B_c \mathbf{v}_c, \quad (3.90)$$

$$\mathbf{y}_c = C_c \mathbf{x}_c. \quad (3.91)$$

Diferenciando (3.91) e manipulando algebricamente, chega-se em

$$\mathbf{v}_c = [C_c (A_c - B_c K_{LQR}) B_c]^{-1} \left[\dot{\mathbf{y}}_c - C_c (A_c - B_c K_{LQR})^2 \mathbf{x}_c \right], \quad (3.92)$$

portanto o comportamento do laço fechado de $\mathbf{y}_c$ é ditado por $\ddot{\mathbf{y}}_c = \ddot{\mathbf{v}}_c$.

3.13 Controlador PD

Finalmente, escolhe-se $\bar{\mathbf{v}}_c$ como um controle proporcional-derivativo da forma

$$\bar{\mathbf{v}}_c = \ddot{\mathbf{y}}_c^{ref} + K_d (\dot{\mathbf{y}}_c^{ref} - \dot{\mathbf{y}}_c) + K_p (\mathbf{y}_c^{ref} - \mathbf{y}_c) \quad (3.93)$$

sendo $\mathbf{y}_c^{ref}$ a saída desejada. O comportamento dinâmico completo do erro $\mathbf{e}_c = \mathbf{y}_c^{ref} - \mathbf{y}_c$ torna-se

$$\ddot{\mathbf{e}}_c + K_d \dot{\mathbf{e}}_c + K_p \mathbf{e}_c. \quad (3.94)$$

Na ausência de erros de modelagem, uma seleção cuidadosa das matrizes de ganho K_p e K_d garante $\mathbf{e}_c \rightarrow 0$, ou seja, $\mathbf{y}_c$ converge para seu valor de referência com uma taxa de referência determinada pelos elementos das matrizes de ganho. Escolhendo matrizes diagonais e definidas positivas, cada saída é controlada independentemente das outras e sua dinâmica pode ser determinada arbitrariamente.

3.14 Lei de controle em cascata

O controlador final, portanto é obtido pela combinação das estratégias descritas anteriormente

$$\mathbf{u}_c = [C_c (A_c - B_c K_{LQR}) B_c]^{-1} \left[\ddot{\mathbf{y}}_c^{ref} + K_d \dot{\mathbf{e}}_c + K_p \mathbf{e}_c - C_c (A_c - B_c K_{LQR})^2 \mathbf{x}_c \right] - K_{LQR} \mathbf{x}_c. \quad (3.95)$$

A Figura 3.21 exibe um diagrama de blocos ilustrando o sistema.

Figura 3.21: Diagrama de blocos do controlador cascata.

3.15 Seguidor de pontos de passagem no sistema de coordenadas do corpo

Retornando as variáveis x^B , y^B , z^B presentes na estimativa do vetor x_c , elas representam a posição do helicóptero em um sistema de coordenadas que rotaciona juntamente com o corpo do helicóptero mas cuja origem é a mesma do sistema de coordenadas de navegação. Como o guia de pontos de passagem (destinos de um trecho de navegação) é geralmente definido no sistema de coordenadas de navegação, que não rotaciona mas é rigidamente fixo a Terra, é necessário transformar os pontos de passagem para o sistema de coordenadas do corpo no qual o controlador foi projetado. Para obter tal resultado, usa-se o fato de que a posição no sistema de coordenadas do corpo é igual a

$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = R_N^B \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.96)$$

sendo a matriz de transformação do sistema de navegação para corpo R_N^B , dada por (3.66)

Portanto, o seguidor de pontos de passagem em sistemas de coordenadas do corpo x_B , y_B , z_B é alcançado utilizando como saída de referência

$$\left(y_c^{ref}\right)^B = \begin{bmatrix} R_N^B & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \left(y_c^{ref}\right)^N. \quad (3.97)$$

3.16 Resultados simulados

A simulação foi realizada utilizando o controlador em cascata descrito na Figura 3.21. O helicóptero inicia na posição $x_0 = 0$, $y_0 = 0$, $z_0 = 0$ e orientação $\psi_0 = 0$, em seguida decola para o *waypoint* $x_f = 0$, $y_f = 0$ e $z_f = -4$, e então segue os pontos de passagem $x_f = -3$ e $y_f = -3$, $x_f = -3$ e $y_f = 3$ e $x_f = 3$ e $y_f = 3$, rotacionando o helicóptero conforme o movimento. A trajetória de referência para o controle do helicóptero é gerada pelo polinômio do quinto grau descrito na Seção 3.4.1. Os resultados obtidos são mostrados nas figuras 3.22 - 3.23.

Figura 3.22: Trajetória do helicóptero.

Figura 3.23: Entradas do helicóptero.

Figura 3.24: Trajetória do helicóptero: no eixo X (a) , no eixo Y (b) e no eixo Z (c).

Figura 3.25: Guinada do helicóptero.

Figura 3.26: Rolagem e arfagem do helicóptero.

CAPÍTULO 4

Instrumentação e controle de um robô helicóptero

O desenvolvimento do sistema de um robô helicóptero não é algo trivial, pois requer conhecimentos interdisciplinares tais como engenharia espacial, engenharia elétrica, comunicações, ciência da computação, controle de sistema, inteligência artificial, operações de sistemas em tempo real, entre outros. Trata-se de um dispositivo mecânico controlado eletronicamente viajando a altas velocidades e altitudes, com altas velocidades de rotações de suas hélices.

Para o desenvolvimento do robô helicóptero optou-se partir de um helimodelo comercial de pequeno porte e de fácil manuseio. O modelo escolhido foi o TREX 450 XL que será descrito no Anexo A.1. Para o piloto automático do robô helicóptero será usado o microprocessador RCM5400W com conectividade *Wi-Fi* descrito no Anexo A.2, a IMU 6DOF v4 da Sparkfun de seis graus de liberdade descrito no Anexo A.3 e uma placa de controle de servos Pololu descrito no Anexo A.5. Maiores detalhes da integração do helimodelo com o piloto automático pode ser visto na Seção 4.1.

No microprocessador embarcou-se os algoritmos do filtro de Kalman apresentado no Capítulo 2 e as principais funções escritas em *Dynamic C* são mostradas no Apêndice E. O piloto automático se comunica com o computador terrestre via Wi-Fi, o qual receberá os dados de posição, velocidade e atitude do helicóptero. Quando o controle estiver implementado enviará comandos de posição e velocidade para que o helicóptero se desloque de um ponto a outro. Atualmente

os dados são adquiridos pelo Matlab® via Wi-Fi, veja Seção 4.2. Como medida de segurança, caso ocorra perda de comunicação com o sistema do piloto automático, o helicóptero poderá ser operado via rádio por um piloto de helimodelo.

Para realizar os primeiros testes de estabilização com o helicóptero, pretende-se fixar o helimodelo em uma plataforma de 3 graus de liberdade juntamente com o piloto automático. A plataforma será usada para a identificação e estabilização do helicóptero, além de ser uma medida de segurança para a realização dos primeiros testes. Portanto, foi projetada uma plataforma de 3 graus de liberdade em Solid Edge®, a plataforma já foi construída conforme pode ser visto na Seção 4.3.

4.1 Projeto da placa do piloto automático e montagem do robô helicóptero

Primeiramente desenvolveu-se o circuito do piloto automático sobre o kit de desenvolvimento do Rabbit 5400W, como pode ser visto no esquemático do circuito na Figura 4.1 e na Figura 4.2.

Figura 4.1: Esquemático do piloto automático desenvolvido sobre o kit de desenvolvimento do Rabbit 5400W.

Como visto o circuito montado no kit de desenvolvimento é composto pelo microprocessador Rabbit 5400W descrito no Anexo A.2, pela IMU descrita no Anexo A.3, pelo GPS descrito no Anexo A.4, e pela placa de controle de servos descrita no Anexo A.5. Devido às diferenças de

Figura 4.2: Piloto automático desenvolvido sobre o kit de desenvolvimento do Rabbit 5400W.

tensão das portas seriais do GPS, da placa de servos e do RCM5400W foi utilizado um *buffer* para adequar a tensão entre suas conexões. Para as portas seriais do IMU não foi necessário pois ambos trabalham em 3.3 V. A bateria de lítio polímero do helimodelo TREX 450 XL mostrado no Anexo A.1 alimenta tanto o kit de desenvolvimento do Rabbit 5400W como o motor e os servos do helimodelo. O kit de desenvolvimento fornecem tensões de 5 V e 3.3 V, os quais alimentam o GPS, a IMU, a placa de servos e o *buffer*. As saídas de controle de servos da placa Pololu são conectados aos servos, ao giroscópio e ao controle de velocidade do helimodelo.

O RCM5400W possui um programa denominado *Rabbit I/O Utility*, que gera arquivos do tipo *lib* para configuração de suas portas. Utilizando o *Rabbit I/O Utility* configurou-se as seguintes portas do RCM5400W:

- A porta PC3 como receptora da serial F com 38400 de *baud rate*, utilizada para obtenção dos dados do GPS;
- A porta PC2 como transmissora da serial F com 38400 de *baud rate*, utilizada para enviar comandos para a placa dos servos;
- A porta PE1 como receptora da serial D com 115200 de *baud rate*, utilizada para obtenção dos dados do IMU;
- A porta PC1 como transmissora da serial D com 115200 de *baud rate*, utilizada para enviar comandos para a IMU e
- A porta PB3 como I/O utilizada como *reset* da placa de servos.

Na Figura 4.3, tem-se o piloto automático desenvolvido no kit da Rabbit instalado no helimodelo, o GPS foi instalado na calda do helimodelo.

Figura 4.3: Helimodelo com o piloto automático instalado.

4.2 Ambiente para iteração com o helimodelo

Foi desenvolvido um ambiente para aquisição dos dados dos filtros implementados no Rabbit 5400W e para enviar comandos para os servos do helimodelo. O ambiente foi desenvolvido utilizando a ferramenta *guide* do Matlab®. Na Figura 4.4 abaixo é mostrado o programa adquirindo os dados do piloto automático via Wi-Fi, em ambiente fechado.

Figura 4.4: Ambiente para aquisição de dados.

4.3 Plataforma de movimento de 3 graus de liberdade

Para realizar os primeiros testes com o helimodelo construiu-se uma plataforma de movimento de 3 graus de liberdade conforme mostrado na Figura 4.5.

Figura 4.5: Plataforma de movimento de 3 graus de liberdade.

A plataforma foi projetada em Solid Edge conforme mostrado nas figuras 4.6. e 4.7. Os *encoders* estão em preto na figura e são utilizados para a leitura dos ângulos de Euler (rolagem, arfagem e guinada).

A plataforma de movimento será utilizada para a realização dos experimentos da estimativa da atitude e de controle do helicóptero. Sob a plataforma será utilizado o helimodelo TREX 450XL juntamente com a placa do piloto automático. E a comunicação com o computador será feita via Wi-Fi. Na Figura 4.8 é mostrado como se pretende utilizar a base.

4.4 Reformulação da plataforma de movimento e desenvolvimento de placas de circuito impresso

A reformulação da plataforma de movimento e desenvolvimento de placas de circuito impresso foram realizadas com auxílio de um aluno de iniciação científica, para maiores detalhes veja [12].

(a)

(b)

(c)

(d)

Figura 4.6: Base de movimento de 3 graus de liberdade: vista isométrica (a), vista isométrica com a base movimentada (b), vista frontal (c) e vista lateral (d).

Figura 4.7: Dimensões em mm da base de movimento de 3 graus de liberdade.

Figura 4.8: *Notebook* adquirindo dados do helicóptero.

A plataforma de movimento do helicóptero foi reformulada devido a dificuldade de estabilização com o mini-helicóptero. Este problema foi gerado por causa do peso da base do helicóptero, complexidade mecânica das conexões com os *encoders*, da baixa potência do rotor do mini-helicóptero e principalmente da grande distância entre a base onde o helicóptero está fixado e o ponto de giro do mesmo. A Figura 4.9 mostra respectivamente a plataforma antiga e a nova, onde nota-se estes e outros detalhes melhorados, como por exemplo, a simplificação na fixação dos *encoders*, e a não utilização de um *encoder* “vazado”.

Como medida de segurança, caso ocorra perda de comunicação com o sistema do piloto automático, o helicóptero poderá ser operado via rádio por um piloto de helimodelo. Para selecionar entre o modo automático e manual o piloto aciona uma chave seletora no rádio do helimodelo. E para realizar a leitura do sinal PWM (modulação por largura de pulso, do inglês *Pulse-width modulation*) do receptor do rádio do helimodelo, utilizou-se um microcontrolador *ATTINY13-20PU* para identificar a largura do pulso do sinal PWM e converter em um sinal

Figura 4.9: Plataforma de movimento 3D : antiga (a) , nova (b) .

digital. Este sinal digital foi ligado na porta de seleção de dois multiplexadores 74LS157 que selecionam entre os sinais do rádio do helimodelo e o piloto automático. A Figura 4.10 mostra o esquema elétrico proposto. Um *buffer* foi necessário para elevar a tensão de saída do *Rabbit* de 3,3V para 5V devido aos quesitos da placa do Pololu. O conector J6 recebe os comandos dos servos vindos do Pololu. Já o J1 recebe os comandos vindos do receptor de rádio Futaba. Os conectores J7, J8 e J9 são as saídas, onde serão ligados os servos. J5 alimenta o circuito, enquanto J2 recebe o sinal serial de comando do piloto automático vindo do *Rabbit*. O soquete U4 irá conter o microcontrolador *Attiny13*.

A Figura 4.11 mostra a placa de circuito impresso gerada. Na Figura 4.12 tem-se a placa finalmente montada. Note que a placa Pololu já está conectada à placa de controle de servos, e como foi montada com soquetes, sua remoção se dá com extrema facilidade. Na Figura 4.13 é possível ver o piloto automático com a placa de controle de servos.

Também foi desenvolvida uma placa de circuito impresso para aquisição dos dados dos *encoders* da plataforma de movimento, veja Figura 4.14. Nesta placa estão contidos os *buffers* para nivelar níveis de tensão além de um decodificador de quadratura para o terceiro *encoder* (O kit de desenvolvimento do *Rabbit* possui apenas 2 decodificadores de quadratura internos). O diagrama do circuito elétrico e a placa do circuito impresso são mostrados nas figuras 4.14 e 4.15. A placa pronta pode ser vista na Figura 4.16.

Figura 4.10: Esquema elétrico do multiplexador controlador dos servos.

No diagrama da Figura 4.14 é visto no canto direito três conectores verdes, que como estão assinalados, são os conectores que receberão os três *encoders*. Note que os *encoders* 1 e 2 ligam-se ao *buffer* U1A, que é o C.I. *SN74S244N* a fim de reduzir a tensão de 5V para 3,3V (compatível com o Rabbit), que ligam-se ao conector J2, que por fim irão conectar-se ao Rabbit. Já o *encoder* 3 é ligado ao U3 que é o C.I. decodificador de quadratura. Este C.I. funciona com alimentação de 5V, e como esperado, sua saída antes de ir para o Rabbit deve passar por um *buffer* (U2 - *SN74S244N*) para assim reduzir sua tensão para 3,3V. De maneira análoga, o nível de tensão dos sinais dos comandos enviados pelo Rabbit para o decodificador também precisam ser aumentados de 3,3V para 5V, isso também é feito com um *buffer*, U4 - *SN74S244N*. O pequeno circuito analógico no canto esquerdo do diagrama é um oscilador que funciona na frequência do cristal montado. Este oscilador é necessário para o decodificador de quadratura, pois este irá operar na frequência deste sinal de entrada gerado.

Figura 4.11: Circuito criado para o controle dos servos. À esquerda a face superior, e à direita a face inferior.

Figura 4.12: Placa de controle de servos.

Figura 4.13: Piloto automático com IMU, GPS, placa de controle de servos e receptor de rádio.

Figura 4.14: Esquema elétrico do circuito desenvolvido para leitura dos *encoders*.

Figura 4.15: Placa de circuito impresso dupla face desenvolvida para leitura dos *encoders*. À esquerda a face superior, e à direita a face inferior.

Figura 4.16: Placa para leitura dos *encoders*.

CAPÍTULO 5

Controle descentralizado e cooperativo de robôs heterogêneos em formação

Uma formação de robôs autônomos refere-se a um grupo de robôs espacialmente distribuídos cuja dinâmica do estado são acopladas através de uma lei de controle comum. Neste capítulo é apresentado o controlador descentralizado e cooperativo de robôs heterogêneos proposto por [43, 47, 48], cuja finalidade é gerar trajetórias de referência para que os robôs heterogêneos se movimentem em formação rígida, como apresentado em [65] para robôs móveis com rodas.

5.1 Modelo

Considerando que todos os robôs de uma formação de N robôs estejam sob o mesmo sistema de coordenadas de navegação (X, Y, Z) , aqui definido pelo sistema de coordenadas geográficas NED (do inglês, *North, East, Down*). Assim o i -ésimo robô da formação pode ser modelado como um sistema linear invariante no tempo, como mostrado abaixo

$$\dot{\mathbf{x}}_i = A_r \mathbf{x}_i + B_r \mathbf{u}_i, \quad (5.1)$$

sendo

$$A_r = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} & \cdots & 0 & a_{2(2k)} & \cdots & 0 & a_{2(2n)} \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & a_{42} & 0 & a_{44} & \cdots & 0 & a_{4(2k)} & \cdots & 0 & a_{4(2n)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 & 0 \\ 0 & a_{(2k)2} & 0 & a_{(2k)4} & \cdots & 0 & a_{(2k)(2k)} & \cdots & 0 & a_{(2k)(2n)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 \\ 0 & a_{(2n)2} & 0 & a_{(2n)4} & \cdots & 0 & a_{(2n)(2k)} & \cdots & 0 & a_{(2n)(2n)} \end{bmatrix}, \quad (5.2)$$

$$B_r = I_{n \times n} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (5.3)$$

$\mathbf{x}_i \in \mathbb{R}^{2n \times 1}$ o estado composto por n variáveis de configuração do robô i e suas derivadas, $\mathbf{u}_i$ a entrada de controle, $\otimes$ a representação do Produto de Kronecker, $k = 1, 2, \dots, n$ e $i = 1, 2, \dots, N$.

Como as entradas pares de $\mathbf{x}_i$ representam as velocidades das entradas ímpares, as linhas ímpares de A_r e B_r assumem a forma apresentada em (5.2). Já os zeros das colunas ímpares de A_r são necessários para garantir a convergência dos robôs para a formação conforme mostrado em [42] Proposição 3.1 e [44] Proposição 4.2. Esses zeros indicam que as velocidades não devem ser afetadas pelas posições dos robôs, o que é necessário para a formação permanecer invariante. As entradas da forma $a_{(2k)(2k)}$ afetam a aceleração da formação como um todo: quando negativa, os robôs alcançam a formação e param, quando zero, os robôs alcançam a formação enquanto deslizam e, quando positivo, os robôs alcançam a formação mas a formação como um todo se acelera [44]. As outras entradas estão relacionadas a um movimento de rotação da formação [71] e resultam de acoplamentos cruzados entre as entradas do estado.

Definindo as entradas ímpares de $\mathbf{x} = [x_1, \dots, x_N]^T$ como variáveis de posição e as entradas

pares como variáveis de velocidade representadas respectivamente por

$$\mathbf{x}_p = [x_{p_1}, \dots, x_{p_N}], \text{ e}$$

$$\mathbf{x}_v = [x_{v_1}, \dots, x_{v_N}],$$

o vetor de estado i pode ser definido como

$$\mathbf{x}_i = \mathbf{x}_{p_i} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \mathbf{x}_{v_i} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (5.4)$$

Para o caso $n = 2$ tem-se $\mathbf{x}_{p_i} = [x_{r_i} \ y_{r_i}]^T$, $\mathbf{x}_{v_i} = [\dot{x}_{r_i} \ \dot{y}_{r_i}]^T$, $\mathbf{x}_{r_i} = [x_{r_i} \ \dot{x}_{r_i} \ y_{r_i} \ \dot{y}_{r_i}]^T$,

$$A_r = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & 0 & a_{44} \end{bmatrix} \text{ e } B_r = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \quad (5.5)$$

A trajetória do líder pode ser alterada variando os elementos da matriz A_r . Por exemplo, se $a_{22} = a_{24} = a_{42} = a_{44} = 0$, temos uma reta com inclinação dada pelas condições iniciais; se $a_{22} = a_{44} = 0$ e $a_{24} = -p$ e $a_{42} = p$, com $p > 0$, temos uma elipse. E a variação pode ser realizada ao longo do tempo para se criar uma trajetória variável do líder.

Agrupando as equações para todos os robôs em um único sistema no espaço de estado tem-se

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad (5.6)$$

sendo $A = I_{N \times N} \otimes A_r$ e $B = I_{N \times N} \otimes B_r$.

5.2 Definição de formação

Neste trabalho, uma formação será definida por um vetor

$$\mathbf{h} = \mathbf{h}_p \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^{2nN \times 1}, \quad (5.7)$$

sendo que $\mathbf{h}_p = [h_{p1} \dots h_{pN}]^T$ identifica as posições descritas no sistema de coordenadas de navegação (X, Y, Z) , sendo $\mathbf{h}_{pi} = [h_{xi} \ h_{yi}]^T$. Assim, os N robôs estão em formação $\mathbf{h}$ no tempo t se existirem vetores $\mathbf{q}, \mathbf{v} \in \mathbb{R}^n$ tal que

$$\mathbf{x}_{p_i}(t) - \mathbf{h}_{p_i} = \mathbf{q}(t) \text{ e } \mathbf{x}_{v_i}(t) = \mathbf{v}. \quad (5.8)$$

Uma vez que $\mathbf{q}(t) = [q_x(t) \ q_y(t)]^T$ representa a distância entre a posição atual do robô i e a posição do mesmo robô definida pela formação $\mathbf{h}$, esta distância deve ser a mesma para cada robô. Da mesma forma, $\mathbf{v}$ representa a velocidade da formação, sendo que todos os robôs devem possuir a mesma velocidade.

Os robôs convergem para a formação $\mathbf{h}$ se existirem funções $\mathbf{q}(\cdot), \mathbf{v}(\cdot) \in \mathbb{R}^n$, tais que,

$$\mathbf{x}_{p_i}(t) - \mathbf{h}_{p_i} - \mathbf{q}(t) \rightarrow 0 \text{ e } \mathbf{x}_{v_i}(t) - \mathbf{v}(t) \rightarrow 0 \quad (5.9)$$

quando $t \rightarrow \infty$. A Figura 5.1 ilustra este conceito de formação, sendo representada apenas a análise com relação à coordenada x do sistema de coordenadas de navegação (X, Y, Z) .

Figura 5.1: Representação de uma formação com três robôs.

A topologia de comunicação entre os robôs é representada por um dígrafo $\Gamma = (\mathcal{V}, \mathcal{E})$, definido por vértices $\mathcal{V}$ e arestas $\mathcal{E}$ que captura a conectividade entre os robôs. Cada vértice representa um

robô e há uma aresta direcionada de um vértice ao outro se houver comunicação entre os robôs. O robô que envia a informação é considerado vizinho do robô que está recebendo a informação. O par (i, j) pertence ao conjunto de arestas $\mathcal{E}$ se i é vizinho de j . Sendo que J_i indica o número de vizinhos do i -ésimo robô e $|J_i|$ o número de vizinhos visíveis. E para implementação do controle de formação, utiliza-se uma matriz Laplaciana que tem o objetivo de agrupar as informações que definem quais robôs se comunicam entre si.

Seja Γ um dígrafo com conjunto de vértices $\mathcal{V}$ e um conjunto de arestas $\mathcal{E}$. A matriz de adjacência de Γ é a matriz quadrada Q com entradas

$$q_{ij} = \begin{cases} 1, & \text{se } (j, i) \in \mathcal{E}, \\ 0, & \text{caso contrário.} \end{cases} \quad (i, j \in \mathcal{V}). \quad (5.10)$$

A matriz de grau interno de Γ , definida como sendo diagonal, é denotada por D sendo formada por

$$d_{ii} = |\{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}|, \quad (i \in \mathcal{V}). \quad (5.11)$$

E a matriz Laplaciana para a comunicação dos grafos pode ser definida como em [47]

$$L_{\Gamma} = D - Q. \quad (5.12)$$

O robô considerado líder da formação é caracterizado por não receber nenhuma informação de outro robô. Isto significa que os outros robôs são forçados a arranjar suas posições em resposta ao movimento desse robô [44]. Para que haja esse líder, o dígrafo de comunicação é uma árvore. Trata-se de um dígrafo Γ que não tem ciclos e admite um vértice v (a raiz) tal que há um trajeto direcionado de v a todos os outros vértices em Γ . Nesse caso, a linha da matriz de adjacência Q referente ao vértice do líder tem que ser zero.

5.3 Controle de formação

O problema de controle cooperativo é encontrar uma lei de controle $\mathbf{u}$ para que todos os robôs converjam para a formação $\mathbf{h}$. O aspecto da distribuição desse controle é baseado na dependência de $\mathbf{u}$ em informações locais e relativas ao estado, o qual é recebido pelos vizinhos J_i de cada robô.

Portanto, um controlador central não é necessário. Além disso, $\mathbf{u}$ é a única ligação dinâmica do robô com a formação.

Resolver o problema de controle de formação para a Eq. (5.6), que representa todos os robôs, compreende a solução de um problema de estabilização envolvendo a realimentação da saída. Tem-se que vetor de saída $\mathbf{z}$ pode ser escrito como $\mathbf{z} = L(\mathbf{x} - \mathbf{h})$ sendo $L = L_\Gamma \otimes I_{2n \times 2n}$ e L_Γ a matriz direcionada Laplaciana do dígrafo Γ [44]. E a lei de controle, dada a existência de uma matriz de realimentação F , pode ser dada por

$$\mathbf{u} = F\mathbf{z} = FL(\mathbf{h} - \mathbf{x}). \quad (5.13)$$

Consequentemente a formação controlada é dada por

$$\dot{\mathbf{x}} = A\mathbf{x} + BFL(\mathbf{x} - \mathbf{h}), \quad (5.14)$$

cujos robôs convergem para formação $\mathbf{h}$ [72].

Considerando o aspecto de implementação, torna-se necessário a conversão da solução acima apresentada, para todos os robôs, para uma solução que considera apenas as leis de controle locais de cada robô. Assim, os robôs autônomos alcançarão seus objetivos em comum usando somente leis de controle locais, baseando-se nas informações compartilhadas entre seus vizinhos. O objetivo comum é a convergência para a formação $\mathbf{h}$. Portanto é necessário expressar a Eq. (5.14) para o nível local de um robô. Assim, as informações de cada robô podem ser combinadas de tal forma que os robôs ajustem seus movimentos com relação ao líder da formação. As funções de saída z_i podem ser calculadas pela média dos deslocamentos relativos (e velocidades relativas) da vizinhança dos respectivos robôs móveis como sendo (veja [25] e [44] para maiores detalhes)

$$\mathbf{z}_i = \begin{cases} \frac{1}{|J_i|} \sum_{j \in J_i} ((\mathbf{x}_i - \mathbf{h}_i) - (\mathbf{x}_j - \mathbf{h}_j)), & \text{se } (|J_i| \neq 0) \\ 0 & \text{caso contrário} \end{cases}. \quad (5.15)$$

Considerando as estruturas de A , B e L tem-se F da forma $F = I_{N \times N} \otimes F_r$ (que é uma lei de controle descentralizada aplicada a todos os robôs móveis). Neste caso, pode-se reescrever a Eq. (5.6) como

$$\dot{\mathbf{x}} = A\mathbf{x} + L_\Gamma \otimes B_r F_r (\mathbf{x} - \mathbf{h}). \quad (5.16)$$

Note que o conjunto de robôs móveis convergirá para uma dada formação $\mathbf{h}$ se o Laplaciano do grafo direcionado L_Γ possuir pelo menos dois autovalores nulos, [44]. Para obtermos o controlador, a matriz de realimentação, F_r , pode ser representada da seguinte forma

$$F_r = \begin{bmatrix} f_1 & f_2 & 0 & 0 \\ 0 & 0 & f_1 & f_2 \end{bmatrix}. \quad (5.17)$$

Em [72] mostram-se que as condições necessárias e suficientes para o sistema convergir para a formação são obtidas com $f_1 < 0$ e $f_2 < 0$.

A principal finalidade do controle de formação é encontrar a matriz de realimentação F_r que garanta a convergência para a formação, dada a estrutura de A_r . Os elementos da matriz A_r são os parâmetros que definem a trajetória de referência para o robô líder. O controle mostrado acima gera as trajetórias que os demais robôs devem realizar para entrar em formação com o líder. Estas curvas são utilizadas como trajetórias de referência para os robôs heterogêneos.

5.4 Acompanhamento de trajetória

Conforme apresentado em [47], o acompanhamento de trajetória para formação de robôs autônomos pode ser feito a partir de extensão dos algoritmos existentes de planejamento de trajetória e acompanhamento de trajetória de um robô individual para o contexto de um grupo. O algoritmo combina o acompanhamento do líder e o controle cooperativo da formação, sendo um robô designado como líder. O sistema dinâmico do veículo controlado, incluindo tanto os controladores cooperativos e de acompanhamento, são descritos por

$$\dot{\mathbf{x}} = A\mathbf{x} + BFL(\mathbf{x} - \mathbf{h}) + K(\mathbf{r} - \mathbf{h} - \mathbf{x}) \quad (5.18)$$

sendo $\mathbf{r}$ a trajetória desejada ou de referência e K os ganhos de acompanhamento.

5.5 Resultados simulados

5.5.1 Simulação 1

Para a simulação da formação foi considerada a seguinte matriz de formação

$$\begin{aligned} \mathbf{h}_1 &= [6 \ 0], \quad \mathbf{h}_2 = [3 \ 2], \quad \mathbf{h}_3 = [3 \ -2], \\ \mathbf{h}_4 &= [0 \ 4], \quad \mathbf{h}_5 = [0 \ 0], \quad \mathbf{h}_6 = [0 \ -4], \end{aligned} \quad (5.19)$$

com matriz de realimentação

$$F_r = \begin{bmatrix} -100 & -100 & 0 & 0 \\ 0 & 0 & -100 & -100 \end{bmatrix}. \quad (5.20)$$

Os resultados das simulações para o dígrafo mostrado na Figura 5.2 são mostrados na Figura 5.3 e na Figura 5.4. Estas curvas serão utilizadas como trajetórias de referência para os robôs.

Figura 5.2: Dígrafo sendo o robô 1 o líder.

Figura 5.3: Trajetória da formação.

Figura 5.4: Trajetória da formação com acompanhamento de trajetória.

5.5.2 Simulação 2

Nesta simulação uma formação de robôs segue um robô helicóptero, conforme mostrado no diagrama da Figura 5.5. A seguinte matriz de formação foi considerada

$$\mathbf{h}_1 = [1 \ 0], \quad \mathbf{h}_2 = [0 \ 1], \quad \mathbf{h}_3 = [0 \ -1], \quad (5.21)$$

com matriz de realimentação

$$F_r = \begin{bmatrix} -0.0001 & -0.0012 & 0 & 0 \\ 0 & 0 & -0.0001 & -0.0012 \end{bmatrix}. \quad (5.22)$$

Figura 5.5: Diagrama da formação seguindo helicóptero.

Foi utilizado para esta simulação o dígrafo descrito na Figura 5.6.

Figura 5.6: Dígrafo sendo o robô 1 o líder.

O helicóptero inicia na posição $x_0 = -3$, $y_0 = 0$, $z_0 = 0$ e orientação $\psi_0 = 0$, em seguida decola para o *waypoint* $x_f = 0$, $y_f = 0$ e $z_f = -4$, e então segue os pontos de passagem $x_f = -4$ e $y_f = -4$, $x_f = -4$ e $y_f = 4$ e $x_f = 4$ e $y_f = 4$, rotacionando o helicóptero conforme o movimento. A trajetória de referência do helicóptero é gerada pelo polinômio do quinto grau descrito na Seção 3.4.1. E o helicóptero utiliza o sistema de controle apresentado na Figura 3.21. A posição do helicóptero é então repassada como trajetória de referência para o algoritmo de controle de formação descrito na Seção 5.4. E o controle de formação gera trajetórias de referências para os RMRDs. O RMRD 1 inicia na posição $x_0 = -2$, $y_0 = -1$ e orientação

$\psi_0 = 0$, o RMRD 2 inicia na posição $x_0 = -2$, $y_0 = 0.6$ e orientação $\psi_0 = 0$, e o RMRD 3 inicia na posição $x_0 = -2$, $y_0 = 2$ e orientação $\psi_0 = 0$. Os RMRDs utilizam a estrutura de controle apresentada na Figura 3.4.

Os resultados da simulação são mostrados nas figuras 5.7 - 5.8. E as trajetórias individuais do helicóptero, do RMRD 1, do RMRD 2 e do RMRD 3 são mostradas nas figuras 5.9 - 5.12, respectivamente. Observe que os RMRDs, devido à restrição não-holonômica apresentada no Capítulo 3 acompanham uma trajetória desejada obtida a partir da trajetória de referência gerada pelo controle de formação, o que não é necessário para o helicóptero uma vez que o mesmo não possui restrição de movimento.

Figura 5.7: Formação com três RMRD acompanhando um RH mostrado em 3D.

Figura 5.8: Formação com três RMRD acompanhando um RH.

Figura 5.9: Trajetória do helicóptero.

Figura 5.10: Trajetória do RMRD 1.

Figura 5.11: Trajetória do RMRD 2.

Figura 5.12: Trajetória do RMRD 3.

5.5.3 Simulação 3

Nesta simulação o helicóptero faz parte da formação conforme mostrado no diagrama da Figura 5.13. A seguinte matriz de formação foi considerada

$$\mathbf{h}_1 = [0 \ 0], \quad \mathbf{h}_2 = [0 \ 1], \quad \mathbf{h}_3 = [1 \ -1], \quad (5.23)$$

com matriz de realimentação

$$\mathbf{F}_r = \begin{bmatrix} -0.0001 & -0.0012 & 0 & 0 \\ 0 & 0 & -0.0001 & -0.0012 \end{bmatrix}. \quad (5.24)$$

Figura 5.13: Diagrama da formação seguindo helicóptero.

Foi utilizado para esta simulação o dígrafo descrito na Figura 5.14.

Figura 5.14: Dígrafo sendo o helicóptero o líder.

A formação percorre os seguintes pontos de passagem $x_f = 0$ e $y_f = 0$, $x_f = 3$ e $y_f = 3$, $x_f = -3$ e $y_f = 3$, $x_f = -3$ e $y_f = -3$, e $x_f = 3$ e $y_f = -3$. A trajetória de referência para o controle de formação é gerada pelo polinômio do quinto grau descrito na Seção 3.4.1. O helicóptero inicia na posição $x_0 = -2$, $y_0 = -1$, $z_0 = 0$ e orientação $\psi_0 = 0$, o RMRD 1 inicia na posição $x_0 = -2$, $y_0 = -1$ e orientação $\psi_0 = 0$, o RMRD 2 inicia na posição $x_0 = -2$, $y_0 = 0.6$ e orientação $\psi_0 = 0$, e o RMRD 3 inicia na posição $x_0 = -2$, $y_0 = 2$ e orientação $\psi_0 = 0$.

Os resultados da simulação são mostrados nas figuras 5.15 - 5.16. E as trajetórias individuais do helicóptero, do RMRD 1 e do RMRD 2 são mostradas nas figuras 5.9 - 5.11, respectivamente.

Figura 5.15: Formação com um helicóptero e dois RMRD mostrado em 3D.

Figura 5.16: Formação com três RMRD acompanhando um RH.

Figura 5.17: Trajetória do helicóptero.

Figura 5.18: Trajetória do RMRD 1.

Figura 5.19: Trajetória do RMRD 2.

CAPÍTULO 6

Conclusão

Neste trabalho, foi desenvolvido um filtro de Kalman robusto recursivo (FKRR) baseado em [66] para sistemas não lineares. Também foi desenvolvido um sistema de estimativa de atitude e posição utilizando filtros robustos. Estes filtros minimizam os efeitos das incertezas do modelo na estimativa dos valores do estado do veículo. O filtro proposto foi implementado utilizando dados experimentais de uma IMU e um GPS. Um estudo comparativo mostrou que o FKRR apresentou uma melhora média de 8 % sobre o filtro de Kalman estendido para a estimativa de atitude, e o filtro BDU uma melhora de 5 % para a estimativa de posição no erro de observação da posição sobre o filtro de Kalman, para os casos apresentados neste trabalho.

Também foi desenvolvido um modelo em espaço de estado no sistema de coordenadas de navegação NED (*North, East, Down*) do robô com rodas deslizantes, resultados de simulação foram apresentados utilizando o regulador robusto para trajetórias geradas a partir de pontos de passagem. É apresentada também uma visão geral do modelo do helicóptero, e a simulação de um helicóptero, baseada no modelo linearizado identificado para voo pairado utilizando o controlador em cascata. Também foram apresentadas simulações da formação com os modelos dinâmicos do robô helicóptero e do robô móvel com rodas deslizantes.

Um código embarcado já foi desenvolvido usando o kit de desenvolvimento da Rabbit, sendo necessário agora apenas realizar a otimização do código. E a placa de circuito impresso do piloto

automático do helicóptero se encontra em desenvolvimento. Uma base de movimento onde se pretende realizar os primeiros testes de estabilização do helicóptero também foi desenvolvida.

6.1 Trabalhos futuros

Para trabalhos futuros pretende-se implementar o sistema de controle no robô Pioneer 3-AT, finalizar o piloto automático e montagem do helicóptero, e implementar o sistema proposto de coordenação. Desenvolver o algoritmo *array* do FKRR para implementação do sistema de estimativa de atitude e posição em ponto fixo em uma FPGA (*Field Programmable Gate Array*). Utilizar um regulador robusto recursivo e controladores $\mathcal{H}_\infty$ para realizar o controle do robô helicóptero, e utilizar a base de movimento para realização de testes de controle do robô helicóptero.

Referências Bibliográficas

- [1] Abdelkrim, N., N. Aouf, A. Tsourdos, e B. White (2008). Robust nonlinear filtering for ins/gps uav localization. In *16th Mediterranean Conference on Control and Automation Congress Centre*, Ajaccio, France.
- [2] Albert, A. (1972). *Regression and the Moore-Penrose Pseudoinverse*. New York and London: Academic Press.
- [3] Bergerman, M., O. Amidi, J. R. Miller, N. Vallidis, e T. Dudek (2007). Cascaded position and heading control of a robotic helicopter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA.
- [4] Bianco, A., J. Y. Ishihara, e M. H. Terra (2008). Optimal robust prediction for general discrete-time singular systems. In *Tenth International Conference on Control, Automation, Robotics and Vision - ICARV*, Hanoi, Vietnam.
- [5] Bianco, A. F. (2009). *Filtros de Kalman Robustos para Sistemas Dinâmicos Singulares em Tempo Discreto*. Tese de Doutorado, University of São Paulo.
- [6] Bijker, J. (2006). Development of an attitude heading reference system for an airship. Dissertação de Mestrado, University of Stellenbosch.
- [7] Bijker, J. e W. Steyn (2008). Kalman filter configurations for a low-cost loosely integrated inertial navigation system on an airship. *Control Engineering Practice* 16, 1509–1518.
- [8] Bogdanov, A., M. Carlsson, G. Harvey, J. Hunt, R. Kiebert, R. van der Merwe, e E. Wan

- (2003). State-dependent *Riccati* equation control of a small unmanned helicopter. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas*.
- [9] Bogdanov, A., E. Wan, e G. Harvey (2004). *SDRE* flight control for x-cell and R-max autonomous helicopters. In *Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*.
- [10] Bogdanov, A. A. e E. A. Wan (2001). Model predictive neural control of a high-fidelity helicopter model. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada*.
- [11] Brown, R. G. e P. Y. C. Hwang (1997). *Introduction to random signal and applied Kalman filtering: with Matlab exercises and solutions* (3rd ed.). New York, USA: John Wiley & Sons.
- [12] Caldas, L. C. (2011). Relatório final de iniciação científica - instrumentação de um helicóptero não tripulado. Technical report, Universidade de São Paulo.
- [13] Caracciolo, L., A. D. Luca, e S. Iannitti (1999). Trajectory tracking control of a four-wheel differentially driven mobile robot. In *IEEE International Conference on Robotics & Automation*, Detroit, Michigan.
- [14] Castellanos, J. F. G., S. Lesecq, e N. Marchand (2005). A low-cost air data attitude heading reference system for the tourism airplane applications. In *IEEE Sensors*, Irvine, CA, USA, pp. 4.
- [15] Cerri, J. P. (2009). Regulador robusto recursivo para sistemas lineares de tempo discreto no espaço de estado. Dissertação de Mestrado, Escola de Engenharia de São Carlos da Universidade de São Paulo.
- [16] Chang, Y. C. (2005). Intelligent robust control for uncertain nonlinear time-varying systems and its application to robotic systems. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* 35(6), 1108–1119.
- [17] Chen, B. S., T. S. Lee, e J. H. Feng (1994). A nonlinear $\mathcal{H}_\infty$ control design in robotic systems under parameter perturbation and external disturbance. *International Journal of Control* 59(2), 439–461.
- [18] Cheng, R. P., M. B. Tischler, e G. J. Schulein (2005). R-max helicopter state-space model identification for hover and forward-flight. *Journal of the American Helicopter Society* 51, 202–210.

-
- [19] Cruz, C. D. L. e R. Carelli (2006). Dynamic modeling and centralized formation control of mobile robots. In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*.
 - [20] Dominguez, S. M., T. Keaton, e A. H. Sayed (2006). A robust tracking method for multi-modal wearable computer interfacing. *IEEE Transactions on Multimedia* 8(5), 956–972.
 - [21] dos Reis, G. A., A. A. G. Siqueira, e M. H. Terra (2005). Nonlinear $\mathcal{H}_\infty$ control via quasi-LPV representation and game theory for wheeled mobile robots. In *Proceedings of the Mediterranean Conference on Control and Automation, Limassol, Cyprus*, pp. 686–691.
 - [22] El-Sheimy, N., H. Hou, e X. Niu (2008). Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement* 57, 140–149.
 - [23] Farrel, J. e M. Barth (1998). *The global positioning system and inertial navigation*. McGraw-Hill.
 - [24] Farrel, J. A. (2008). *Aided Navigation GPS with High Rate Sensors*. The McGraw-Hill Companies.
 - [25] Fax, J. e R. Murray (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control* 49(9), 1465–1476.
 - [26] Garcia-Pardo, P. J., G. S. Sukhatme, e J. F. Montgomery (2001). Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems* 38(1), 19–29.
 - [27] Gavrillets, V. (2003). *Autonomous Aerobatic Maneuvering of Miniature Helicopters*. Tese de Doutorado, Massachusetts Institute of Technology.
 - [28] Harbick, K., J. F. Montgomery, e G. S. Sukhatme (2004). Planar spline trajectory following for an autonomous helicopter. *Journal of Advanced Computational Intelligence - Computational Intelligence in Robotics and Automation* 8(3), 237–242.
 - [29] Hou, H. (2004). Modeling inertial sensors errors using allan variance. Dissertação de Mestrado, University of Calgary.
 - [30] IEEE (1997). Guide and test procedure for single axis interferometric fiber optic gyros. *IEEE Std 952*.
 - [31] Inoue, R. S. (2007). Controle robusto de robôs moveis com rodas. Dissertação de Mestrado, Universidade de São Paulo.

- [32] Ishihara, J. Y. e M. H. Terra (2008). Robust state prediction for descriptor systems. *Automatica* 44(8), 2185 – 2190.
- [33] Ishihara, J. Y., M. H. Terra, e J. C. T. Campos (2006). Robust Kalman filter for descriptor systems. *IEEE Transaction on Automatic Control* 51(8), 1354–1358.
- [34] Isidori, A., L. Marconi, e A. Serran (2003). Robust nonlinear motion control of a helicopter. *IEEE Transactions on Automatic Control* 48, 413–426.
- [35] Jadbabaie, A., J. Lin, e A. S. Morse (2003). Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules. *IEEE Transactions on Automatic Control* 48(6), 988–1001.
- [36] Jung, D. e P. Tsiotras (2007). Inertial attitude and position reference system development for a small UAV. In *American Institute of Aeronautics and Astronautics*, Rohnert Park, California.
- [37] Kanayama, Y., Y. Kimura, F. Miyazaki, e T. Noguchi (1990). A stable tracking control method for an autonomous mobile robot. In *Proc. IEEE International Conference on Robotics and Automation*, pp. 384 – 389.
- [38] Koo, T. J. e S. Sastry (1998). Output tracking control design of a helicopter model based on approximate linearization. In *IEEE Conference on Decision and Control*, Tampa, FL , USA.
- [39] Kozłowski, K. e D. Pazderski (2004). Modeling and control of a 4-wheel skid-steering mobile robot. *International Journal of Applied Mathematics and Computer Science* 14(4), 477–496.
- [40] Krishnamurthy, D. A. (2008). Modeling and simulation of skid steered robot pioneer - 3at. Dissertação de Mestrado, Florida State University College of Engineering.
- [41] Kuipers, J. B. (1998). *Quaternions and rotation sequences*. Princenton, New Jersey: Princenton University Press.
- [42] Lafferriere, G., J. Caughman, e A. Williams (2004). Graph theoretic methods in the stability of vehicle formations. In *American Control Conference*, Volume 4, Boston, pp. 3724–3729.
- [43] Lafferriere, G. e K. Mathia (2008). Control of formations under persistent disturbances. In *American Control Conference*, Washington, USA.
- [44] Lafferriere, G., A. Williams, J. Caughman, e J. Veerman (2005). Decentralized control of vehicle formations. *Systems & Control Letters* 54(9), 899 – 910.

-
- [45] Luenberger, D. G. (2003). *Linear and Nonlinear Programming*. Kluwer Academic Publishers.
 - [46] Martins, F. N., W. C. Celeste, R. Carelli, M. Sarcinelli-Filho, e T. F. Bastos-Filho (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice* 16, 1354–1363.
 - [47] Mathia, K., G. Lafferriere, e T. Tittensor (2007). Cooperative control of uav platoons - a prototype. In *Euro UAV 2007 Conference and Exhibition*, Paris, France.
 - [48] Mathia, K., G. Lafferriere, e A. Williams (2006). Cooperative control of unmmanned vehicle formations. In *Euro UAV 2006 Conference and Exhibition*, Paris, France.
 - [49] Mejias, L. O., S. Saripalli, P. Cervera, e G. S. Sukhatme (2006). Visual servoing of an autonomous helicopter in urban areas using feature tracking. *Journal of Field Robotics* 23(3), 185–199.
 - [50] Mettler, B., M. B. Tischler, e T. Kanade (2001). System identification modeling of a small-scale unmanned rotorcraft for flight control design. *Journal of the American Helicopter Society*, 50–63.
 - [51] Morbidi, F., C. Ray, e G. L. Mariottini (2011). Cooperative active target tracking for heterogeneous robots with application to gait monitoring. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA.
 - [52] Nardi, R. D., O. Holland, J. Woods, e A. Clark (2006). Swarmav: A swarm of miniature aerial vehicles. In *Proceedings of the 21st Bristol UAV Systems Conference, Bristol, UK*.
 - [53] Nardi, R. D., J. Togelius, O. E. Holland, e S. M. Lucas (2006). Evolution of neural networks for helicopter control: why modularity matters. In *Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, Canada*.
 - [54] Padfield, G. D. (1996, 2007). *Helicopter Flight Dynamics: the theory and application of flying qualities and simulation modellign*. Blackwell Publishing.
 - [55] Parker, L. E., B. Kannan, F. Tang, e M. Bailey (2004). Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan.
 - [56] Rezaie, J., B. Moshiri, B. N. Araabi, e A. Asadian (2007). Gps/ins integration using nonlinear blending filters. In *SICE Annual Conference*, Kagawa University, Japan.

- [57] Rogers, R. M. (2003). *Applied mathematics in integrated navigation systems* (2nd ed.). Reston, VA: AIAA education series.
- [58] Roumeliotis, S. I., G. S. Sukhatme, e G. A. Bekey (1998). Smoother based 3d attitude estimation for mobile robot localization. Technical report, USC, <http://iris.usc.edu/>.
- [59] Saripalli, S. e G. S. Sukhatme (2007). Landing a helicopter on a moving target. In *Submitted to IEEE International Conference on Robotics and Automation, Roma, Italy*.
- [60] Sayed, A., V. Nascimento, e F. Cipparrone (2002). A regularized robust design criterion for uncertain data. *SIAM Journal on Matrix Analysis and Its Applications* 23(4), 1120–1142.
- [61] Sayed, A. H. (2001). A framework for state-space estimation with uncertain models. *IEEE Transaction on Automatic Control* 46, 998–1013.
- [62] Simon, D. (2006). *Optimal state estimation Kalman, $\mathcal{H}_\infty$, and nonlinear approaches*. John Wiley & Sons.
- [63] Siqueira, A. A. G. e M. H. Terra (2004). Nonlinear and markovian $\mathcal{H}_\infty$ controls of underactuated manipulators. *IEEE Transactions on Control System Tecnology* 12(6), 811–826.
- [64] Sukhatme, G., J. Montgomery, e R. Vaughan (2001). Experiments with cooperative aerial-ground robots. *Scientific Literature Digital Library - CiteSeer*.
- [65] Terra, M., A. A. G. Siqueira, e T. B. R. Francisco (2010). Controle robusto de robôs móveis em formação sujeitos a falhas. *Controle & Automação* 21(1), 29–42.
- [66] Terra, M. H., J. Y. Ishihara, e R. S. Inoue (2011). A new approach to robust linear filtering problems. In *18th World Congress of the International Federation of Automatic Control (IFAC)*, Milano, Italy.
- [67] Voorsluijs, G. M., S. Bennani, e C. W. Scherer (2004). Linear and parameter-dependent robust control techniques applied to a helicopter UAV. In *Proceedings of the 38-th AIAA Guidance, Navigation and Control Conference, Providence, RI; USA*.
- [68] Wall, J. e D. Bevy (2006). Characterization of inertial sensor measurements for navigation performance analysis. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS*, Fort Worth, Texas.
- [69] Wan, E. A. e A. A. Bogdanov (2001). Model predictive neural control with applications to a 6 dof helicopter model. In *Proceedings of the American Control Conference, Arlington, VA*.

-
- [70] Wan, E. A., A. A. Bogdanov, R. Kiebert, A. Baptista, M. Carlsson, Y. Zhang, e M. Zulauf (2003). Model predictive neural control for aggressive helicopter maneuvers. *Scientific Literature Digital Library - Citeseer*.
- [71] Williams, A., G. Lafferriere, e J. Veerman (2005a). Stable motions of vehicle formations. In *IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain.
- [72] Williams, A., G. Lafferriere, e J. Veerman (2005b). Stable motions of vehicles formations. In *IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain.
- [73] Xing, Z. e D. Gebre-Egziabher (2008). Modeling and bounding low cost inertial sensor errors. In *IEEE/ION Position, Location and Navigation Symposium*.
- [74] Zhang, X., Y. Li, P. Mumford, e C. Rizos (2008). Allan variance analysis of error characteristics of mems inertial sensors for an fpga-based gps/ins system. In *International Symposium on GPS/GNSS*, Yokohama, Japan.

APÊNDICE A

Modelagem dos sensores inerciais

As medidas de IMUs (unidade de medida inercial, do inglês *inertial measurement unit*) são geralmente corrompidas por diferentes tipos de fontes de erros, ruídos presentes nos sinais de sensores, fator de escala, e variações da polarização com a temperatura, entre outros [30]. Realizando a integração numérica das medidas de uma IMU em um algoritmo de navegação, estes erros acumulam, levando a um desvio significativo nas saídas de posição, velocidade e atitude. Portanto, para uma estimativa precisa da informação de navegação, uma modelagem estocástica dos erros de saída dos sensores inerciais é necessária, [22, 68]. Porém, são poucas as informações fornecidas pelos fabricantes destes sensores para elaboração de um modelo de erros apropriado para a análise de desempenho de sistemas de navegação inercial (do inglês, *inertial navigation system*, INS). Neste contexto, a Variância de Allan, veja Apêndice B para maiores detalhes, tem sido utilizada como um método simples e eficiente para verificar e modelar os erros de sensores inerciais, veja [22, 29, 68, 73, 74].

Assim, segue abaixo o modelo dos erros de saída dos sensores inerciais, a metodologia proposta por [73] para estimativa do modelo dos erros dos sensores e o desenvolvimento de um programa para a estimativa dos erros dos sensores.

A.1 Modelo dos erros dos sensores inerciais

O modelo geral da saída de um sensor inercial após calibração pode ser dado por

$$y_m = (1 + k)y_t + b(t), \quad (\text{A.1})$$

sendo y_m a saída medida do sensor, y_t o valor exato da medida do sensor, k erro de fator de escala e $b(t)$ a polarização. O erro de fator de escala é geralmente modelado como uma constante randômica, veja [73]. Porém, neste trabalho o termo ky_t será considerado como uma incerteza δy_t que será atenuada pelo filtro de Kalman robusto, veja Seção 2.1.

Sem perda de generalidade o termo da polarização pode ser considerado como

$$b(t) = b_0 + b_R(t), \quad (\text{A.2})$$

sendo b_0 uma constante de desvio nulo, ou seja, invariante no tempo, e $b_R(t)$ representa o desvio da polarização randômico ou o resíduo da polarização após a calibração, que é variante no tempo.

O termo $b_R(t)$ é dado por

$$b_R(t) = b_w(t) + \sum_{i=1}^N b_i(t), \quad (\text{A.3})$$

sendo $b_w(t)$ o processo de ruído de largura de banda e $b_i(t)$ o i -ésimo processo de erro correlacionado no domínio do tempo. Como sensores de baixo custo são normalmente usados em configurações onde eles são integrados como um sensor de auxilio e são usados por momentos breves, o modelo detalhado de longos termos de erros de características são desnecessários. Deste modo, como proposto em [73] pode-se escrever

$$b_R(t) = b_w(t) + b_1(t), \quad (\text{A.4})$$

sendo $b_1(t)$ o processo Gauss-Markov escalar, veja [11] para maiores detalhes, dado por

$$\dot{b}_1(t) = -\frac{1}{\tau_{b_1}}b_1(t) + w_{b_1}. \quad (\text{A.5})$$

A variável τ_{b_1} é a constante de tempo do processo ou o tempo de correlação, w_{b_1} é o ruído do processo branco.

A.2 Estimativa dos valores dos parâmetros do modelo dos sensores

Esta seção descreve a metodologia proposta em [73] para a estimativa dos valores do modelo dos sensores.

A estimativa do termo b_0 pode ser calculada tirando a média da sequência de dados do sensor ou usando um estimador que seja parte do sistema de navegação. A variância de b_0 , $\sigma_{b_0}^2$, é utilizada para estabelecer as condições iniciais da matrizes de ponderações do estado em algoritmos de fusão de vários sensores como o filtro de Kalman estendido. Desde que o algoritmo aprimore a variância do estado continuamente, um conhecimento preciso deste termo não é necessário.

Para encontrar o desvio padrão de b_w , σ_{b_w} , plota-se a variância de Allan de uma sequência de dados do sensor. O valor de σ_{b_w} será quando $\tau_a = T$, sendo T a taxa de amostragem do sensor.

Em seguida plota-se a variância de Allan do ruído correlacionado $\beta_{b_1}^2(\tau_a)$, subtraindo a variância de Allan do ruído branco $\beta_w^2(\tau_a)$ com variância $\sigma_{b_w}^2$ da variância de Allan da sequência de dados do sensor $\beta_{Total}^2(\tau_a)$. Isto é feito da seguinte forma

$$\beta_{Total}^2(\tau_a) = \beta_w^2(\tau_a) + \beta_{b_1}^2(\tau_a). \quad (\text{A.6})$$

Depois calcula-se a amplitude do ruído de *Markov* q_c através da equação

$$\beta^2 = \frac{q_c^2}{3} \tau_a, \quad (\text{A.7})$$

obtida em (B.24).

Quando τ_a é pequeno comparado com τ_{b_1} , a variância de Allan do processo Gauss-Markov de primeira ordem tem uma inclinação de $+\frac{1}{2}$ no gráfico log-log. A posição desta linha é afetada somente por q_c . Esta linha é usada para limitar o ruído correlacionado no gráfico log-log.

Em seguida, filtra-se o dado $b_R(t)$ agrupando os dados em janelas de 1 segundo e tirando-se a média, para abrandar o efeito do ruído de banda larga. Durante essa filtragem, o ruído decorrelacionado vai continuar decorrelacionado com uma variância de $\sigma_{b_w}^2/f_s$, sendo f_s a frequência de amostragem. Além disto, o ruído correlacionado permanece inalterado, desde que seu tempo de correlação esteja em uma faixa mais larga que 1 segundo. A variância total do

dado filtrado será a soma da banda larga filtrada e do ruído correlacionado. Assim,

$$\sigma_{b_1}^2 = \sigma_{Total}^2 - \frac{\sigma_{b_w}^2}{f_s}. \quad (A.8)$$

Usando o valor de σ_{b_1} e q_c , pode-se plotar o ruído correlacionado exponencial sobre o gráfico da variância de Allan. E então pode-se ajustar σ_{b_1} para obter um limitante superior. E depois, pode-se determinar τ_{b_1} a partir desta relação

$$\tau_{b_1} = \frac{2\sigma_{b_1}^2}{q_c^2}, \quad (A.9)$$

como mostrado em [73].

A.3 Ambiente para estimação dos erros dos sensores

Baseado nas seções anteriores foi desenvolvido um programa para a estimação do erros dos sensores. O programa foi desenvolvido em MATLAB® utilizando a ferramenta *guide* e o *simulink*. A Figura A.1 mostra a tela principal do programa.

Como mostrado na Figura A.1, o canto esquerdo da tela principal é usado para mostrar os gráficos gerados pelo programa, o botão ao lado *Hold on* habilita a opção de plotar outros gráficos na mesma janela e o botão *Hold off* desabilita esta opção. No canto direito da tela principal há uma tabela *Data of the plot* onde pode-se visualizar os dados que estão sendo plotados.

No painel *Simulation model* pode-se simular o sinal do sensor usando o modelo dos erros do sensores apresentado no Anexo A.1. Para isto basta configurar o valor de cada parâmetro e clicar no botão *Simulate*. Após a simulação, pode plotar os ruídos clicando no botão de plotar de cada sinal, por exemplo no botão *Plot ym*. O botão *Save* salva os valores dos parâmetros do modelo, o botão *Load* carrega os valores dos parâmetros do modelo alterados no *Command Window* do MATLAB® ou que estão no *Worksapce* do MATLAB®.

No painel *Data information* são colocados o tempo de experimento e frequência dos dados que serão analisados. Pode-se salvar os dados no *Workspace* através do botão *Save* ou carregá-los através do botão *Load*. O botão *Copy Data* é utilizado para copiar os dados da simulação para análise.

O dado para análise deve ser colocado dentro da estrutura *data.ym.data*. Feito isso e preenchido os valores do experimento no painel *Data information*, pode se utilizar o painel *Error*

Figura A.1: Ambiente para estimação dos erros dos sensores

model structure estimation para a identificação dos parâmetros do modelo do erro do sensor. A sequência para estimativa dos parâmetros é a mesma apresentada no Apêndice A.2 e os valores obtidos e os dados da variância de Allan podem ser salvos em um arquivo *m-file* através do botão *Save*.

Foi simulado o exemplo de um giroscópio mostrado em [73] que apresenta desvio nulo b_0 igual a 1 grau/segundo, desvio padrão do ruído de banda larga $\sigma_w = 0.40$ graus/segundo. O erro correlacionado é modelado pela superposição de dois processos randômicos. O primeiro é um processo de Gauss-Markov de primeira ordem com $\sigma_{b_1} = 0.02$ graus/segundo e tempo de correlação de $\tau = 150$ segundos. O segundo também é um processo de Gauss-Markov de primeira

ordem com $\sigma_{b_1} = 0.015$ graus/segundo e tempo de correlação de $\tau = 1000$ segundos. Obteve-se os dados de 3 horas de simulação a uma frequência de 50 Hz.

Na identificação dos valores, obteve-se $\sigma_w = 0.4042$ graus/segundo, $q_c = 0.0035$ graus/segundo ^{$\frac{3}{2}$} , $\sigma_{b_1} = 0.0221$ graus/segundo e tempo de correlação $\tau_b = 120.5986$ segundos, os quais são similares aos valores encontrados em [73]. A Figura A.2 mostra o gráfico do desvio de Allan utilizado para a identificação dos parâmetros do modelo do sensor.

Figura A.2: Gráfico do desvio de Allan

APÊNDICE B

Variância de Allan

A variância de Allan β^2 é uma técnica de análise no domínio do tempo originalmente desenvolvida para o estudo de estabilidade da frequência de osciladores. Pode ser usada para determinar as características essenciais de processos randômicos de origem dos dados ruidosos, o que ajuda identificar a fonte de um termo ruidoso do dado. Assim a variância de Allan pode ser usada para identificar as propriedades dos ruídos de giroscópios e acelerômetros, como mostrado em [30, 22, 73].

Como mostrado em [22], assume-se que há N pontos consecutivos do dado, cada um tendo um tempo de amostragem T , formando um grupo de n consecutivos pontos de dado, com $n < \frac{N}{2}$, cada membro do grupo é um agrupamento associado a um tempo $\tau_a = nT$. Se a taxa de saída instantânea do sensor inercial é $\Omega(t)$, a média do agrupamento é definida como

$$\bar{\Omega}_k(\tau_a) = \frac{1}{\tau_a} \int_{t_k}^{t_k + \tau_a} \Omega(t) dt, \quad (\text{B.1})$$

sendo $\bar{\Omega}_k(\tau_a)$ o agrupamento médio da taxa de saída para um agrupamento que inicia no k -ésimo ponto de dado e contém n pontos de dados. A definição do agrupamento médio subsequente é dada por

$$\bar{\Omega}_{k+1}(\tau_a) = \frac{1}{\tau_a} \int_{t_{k+1}}^{t_{k+1} + \tau_a} \Omega(t) dt, \quad (\text{B.2})$$

sendo $t_{k+1} = t_k + \tau_a$.

Desenvolvendo a diferença para cada dois agrupamentos adjacentes, obtém-se a diferença

$$\xi_{k+1,k} = \bar{\Omega}_{k+1}(\tau_a) - \bar{\Omega}_k(\tau_a). \quad (\text{B.3})$$

Para cada agrupamento de tempo τ_a , o grupo de ξ , definido por (B.3), forma um conjunto de variáveis randômicas. A quantidade de interesse é a variância de ξ sobre todos os agrupamentos de mesmo tamanho que pode ser formado da totalidade do dado.

Assim, a variância de Allan de comprimento τ_a é dada por

$$\beta^2(\tau_a) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} (\bar{\Omega}_{k+1}(\tau_a) - \bar{\Omega}_k(\tau_a))^2. \quad (\text{B.4})$$

Obviamente, para qualquer número finito de pontos de dados (N), um número finito de agrupamentos de tamanho fixo τ_a pode ser formado. No entanto, (B.4) representa a estimação de uma quantidade $\beta^2(\tau_a)$ cuja qualidade da estimação depende do número de agrupamentos independentes de tamanho fixo que podem ser formados.

Resumindo, a variância de Allan é obtida da seguinte forma: primeiro obtém-se uma longa sequência de dados do sensor e divide-se os dados em agrupamentos baseados em tempos médios τ_a . Obtém-se a média do dado em cada agrupamento, tira-se a diferença entre a média dos sucessivos agrupamentos, eleva-se ao quadrado a diferença, faz-se o somatório destes valores, e divide-se por um fator de escala. Tira-se a raiz do resultado, e então se obtém-se a medida quantitativa de quanto o valor médio muda a cada valor de tempo médio. Aumenta-se o valor de τ_a , e inicia-se novamente o processo de cálculo. Continua-se a fazer isto até obter mais de nove agrupamentos.

A variância de Allan pode também ser definida em termos da saída ângulo ou velocidade como

$$\theta(t) = \int^t \Omega(t) dt. \quad (\text{B.5})$$

O limite inferior da integração não é especificado porque somente as diferenças de ângulo e velocidade são empregadas nas definições. As medidas de ângulo ou velocidade são feitas em tempo discreto dado por $t = kT$, $k = 1, 2, 3, \dots, N$. Adequadamente, a notação é simplificada

para $\theta_k = \theta(kT)$.

Equações (B.1) e (B.2) podem ser definidas por

$$\bar{\Omega}_k(\tau_a) = \frac{\theta_{k+n} - \theta_k}{\tau_a}, \quad (\text{B.6})$$

e

$$\bar{\Omega}_{k+1}(\tau_a) = \frac{\theta_{k+2n} - \theta_{k+n}}{\tau_a}. \quad (\text{B.7})$$

Assim a variância de Allan é estimada como

$$\beta^2(\tau_a) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} (\bar{\Omega}_{k+1}(\tau_a) - \bar{\Omega}_k(\tau_a))^2. \quad (\text{B.8})$$

Há uma única relação entre $\beta^2(\tau_a)$ e a densidade espectral de potência de processos randômicos intrínsecos. Esta relação é dada por

$$\beta^2(\tau_a) = 4 \int_0^\infty S_\Omega(f) \frac{\sin^4(\pi f \tau_a)}{(\pi f \tau_a)^2}, \quad (\text{B.9})$$

sendo $S_\Omega(f)$ a densidade espectral de potência do processo randômico $\Omega(\tau_a)$.

A equação (B.9) será usada para calcular a variância de Allan a partir da densidade espectral de potência do ruído. A densidade espectral de potência de qualquer processo randômico fisicamente significativo pode ser substituído na integral, e uma expressão da variância de Allan $\beta^2(\tau_a)$ em função do comprimento do agrupamento pode ser obtida. Como $\beta^2(\tau_a)$ é mensurável, um gráfico log-log de $\beta(\tau_a)$ versus τ_a fornece uma indicação direta do tipo do processo randômico que existe no dado de um sensor inercial.

Conforme mostrado em [30], um típico gráfico da variância de Allan como o da Figura B.1, pode ser usado para a identificação de diferentes termos de ruídos. Estes termos aparecem em diferentes regiões de τ_a e são estatisticamente independentes e a variância de Allan para qualquer τ_a é a soma dos seus processos randômicos para um mesmo τ_a . As subseções que seguem descrevem os principais ruídos que podem ser identificados através da variância de Allan e foram baseadas em [30].

Figura B.1: Típico gráfico da variância de Allan [30].

B.1 Angle random walk

Termos de ruídos de alta frequência com tempo de correlação muito menor que o tempo de amostragem, contribuem para o *angle random walk* do giroscópio e do acelerômetro, e podem ser eliminados por projeto. Estes termos podem ser caracterizados por um espectro de ruído branco sobre a saída do sensor. A taxa da densidade espectral de potência associada a este ruído é representada por

$$S_{\Omega}(f) = N^2, \quad (\text{B.10})$$

sendo N o coeficiente *angle random walk*.

Substituindo (B.10) em (B.9) e desenvolvendo a integração resulta

$$\beta^2(\tau_a) = \frac{N^2}{\tau_a}. \quad (\text{B.11})$$

Como mostrado na Figura B.1, a Equação (B.11) tem uma inclinação de $-\frac{1}{2}$. Além disso, o valor numérico de N pode ser obtido diretamente lendo a inclinação em $\tau_a = 1$.

B.2 Instabilidade da polarização

A origem desse ruído é a eletrônica, ou outros componentes sujeitos a *random flickering*. Por causa da sua natureza de baixa frequência, indica como a polarização flutua no dado. A taxa da densidade espectral de potência associada a este ruído é

$$S_{\Omega} = \begin{cases} \left(\frac{B^2}{2\pi}\right)^{\frac{1}{f}} & f \leq f_0 \\ 0 & f > f_0 \end{cases}, \quad (\text{B.12})$$

sendo B o coeficiente da instabilidade da polarização e f_0 a frequência de corte.

Substituindo (B.12) em (B.9) e desenvolvendo a integração resulta

$$\begin{aligned} \beta^2(\tau_a) = \frac{2B^2}{\pi} \left[\ln 2 - \frac{\sin^3 x}{2x^2} (\sin x + 4x \cos x) \right. \\ \left. + Ci(2x) - Ci(4x) \right], \end{aligned} \quad (\text{B.13})$$

sendo $x = \pi f_0 \tau_a$ e Ci a função cosseno integral.

Como mostrado na Figura B.1, a Equação (B.13) mostra a variância de Allan para a instabilidade da polarização. A região plana do gráfico pode ser examinada para estimar o limite da instabilidade da polarização como também a frequência de corte fundamental do ruído *flicker*.

B.3 Rate random walk

Este é um processo randômico de incerteza na origem, possivelmente um caso limitado de um ruído correlacionado exponencial com tempo de correlação muito longo. A taxa da densidade espectral de potência associada a este ruído é

$$S_{\Omega}(f) = \left(\frac{K}{2\pi}\right)^2 \frac{1}{f^2}, \quad (\text{B.14})$$

sendo K o coeficiente *rate random walk*.

Substituindo (B.14) em (B.9) e desenvolvendo a integração resulta

$$\beta^2(\tau_a) = \frac{K^2 \tau_a}{3}. \quad (\text{B.15})$$

Como visto na Figura B.1 o *rate random walk* é representado por uma inclinação de $+\frac{1}{2}$ no gráfico log-log de $\beta(\tau_a)$ versus τ_a . A magnitude do ruído pode ser encontrada lendo a linha de inclinação em $\tau_a = 3$.

B.4 Rate ramp

Para intervalos de tempos longos, mas finitos, *rate ramp* é mais um erro de determinístico do que um ruído. Mas é útil para determinar o comportamento de $\beta(\tau_a)$ sobre erros determinísticos sistemáticos. *Rate ramp* é definido como

$$\Omega = Rt, \quad (\text{B.16})$$

sendo R o coeficiente *rate ramp*. Formando e operando sobre os agrupamentos de dados uma entrada dada pela Equação (B.16), obtém-se

$$\beta^2(\tau_a) = \frac{R^2 \tau_a^2}{2}. \quad (\text{B.17})$$

Como mostrado na Figura B.1 o ruído *rate ramp* tem inclinação de +1 no gráfico log-log de $\beta(\tau_a)$ versus τ_a . A magnitude de *rate ramp* R pode ser obtida a partir da linha de inclinação em $\tau_a = \sqrt{2}$. A taxa da densidade espectral de potência deste ruído é dada por

$$S_\Omega(f) = \frac{R^2}{(2\pi f)^3}. \quad (\text{B.18})$$

Deve-se observar que há um ruído de aceleração *flicker* com densidade espectral de potência $\frac{1}{f^3}$ que leva a mesma variância de Allan dependente de τ_a , veja [30].

B.5 Ruído de quantização

O ruído de quantização é um dos erros introduzidos dentro de um sinal analógico pela codificação na forma digital. Este ruído é causado por pequenas diferenças entre a amplitude real dos pontos amostrados e a resolução do *bit* do conversor analógico digital. A taxa da densidade espectral de potência é dada por

$$S_\Omega(f) = \begin{cases} \frac{4Q^2}{\tau_0} \sin^2(\pi f \tau_0) \\ \approx (2\pi f)^2 \tau_0 Q^2 & f < \frac{1}{2\tau_0} \end{cases}, \quad (\text{B.19})$$

sendo Q o coeficiente do *quantization noise*.

Substituindo (B.19) em (B.9) e desenvolvendo a integração resulta

$$\beta^2(\tau_a) = \frac{3Q^2}{\tau_a^2}. \quad (\text{B.20})$$

Como mostrado na Figura B.1 o ruído de quantização tem inclinação de -1 no gráfico log-log de $\beta(\tau_a)$ versus τ_a . A magnitude desse ruído pode ser obtida a partir da linha de inclinação em $\tau_a = \sqrt{3}$.

Deve-se observar que há outros termos de ruídos com diferentes características espectrais, tal como o ruído de ângulo *flicker* e o ruído de ângulo branco, que leva a mesma variância de Allan dependente de τ_a , veja [30].

B.6 Ruído exponencial correlacionado (Markov)

Este ruído é caracterizado por uma função de decaimento exponencial com um tempo de correlação finito. A taxa de densidade espectral de potência para este processo é

$$S_\Omega(f) = \frac{(q_c \tau_b)^2}{1 + (2\pi f \tau_b)^2}, \quad (\text{B.21})$$

sendo q_c a amplitude do ruído e τ_c o tempo de correlação. Substituindo (B.21) em (B.9) e desenvolvendo a integração resulta em

$$\beta^2(\tau_a) = \frac{(q_c \tau_c)^2}{\tau_a} \left[1 - \frac{\tau_c}{2\tau_a} \left(3 - 4e^{-\frac{\tau_a}{\tau_c}} + e^{-\frac{2\tau_a}{\tau_c}} \right) \right]. \quad (\text{B.22})$$

O ruído de *Markov* pode ser visto na Figura B.1. É instrutivo examinar os limites desta equação. Para τ_a muito longo, o tempo de correlação é

$$\beta^2 \Rightarrow \frac{(q_c \tau_c)^2}{\tau_a} \quad \tau_a \gg \tau_c, \quad (\text{B.23})$$

que é a variância de Allan para o *angle random walk*, sendo $N = q_c \tau_c$ o coeficiente do *angle random walk*. Para τ_a muito menor que o tempo de correlação, a Equação (B.22) se reduz para

$$\beta^2 \Rightarrow \frac{q_c^2}{3} \tau_a \quad \tau_a \ll \tau_c, \quad (\text{B.24})$$

que é a variância de Allan para *rate random walk*.

B.7 Ruído sinusoidal

A densidade espectral deste ruído é caracterizada por uma ou mais frequências distintas. Uma fonte de frequência baixa pode ser o movimento lento do veículo devido às mudanças no ambiente. Uma representação da densidade espectral de potência desse ruído contendo uma única frequência é dada por

$$S_{\Omega}(f) = \frac{1}{2}\Omega_0^2 [\delta(f - f_0) + \delta(f + f_0)], \quad (\text{B.25})$$

sendo Ω_0 a amplitude, f_0 a frequência e $\delta(x)$ a função delta de Dirac.

Erros de múltiplas frequências sinusoidais podem ser similarmente representados por uma soma de termos como da Equação (B.25), com suas respectivas frequências e amplitudes. Substituindo (B.25) em (B.9) e desenvolvendo a integração resulta em

$$\beta^2(\tau_a) = \Omega_0^2 \left(\frac{\sin^2 \pi f_0 \tau_a}{\pi f_0 \tau_a} \right)^2. \quad (\text{B.26})$$

O ruído sinusoidal pode ser visto na Figura B.1.

APÊNDICE C

Quatérnios

Conforme mostrado em [41], o quatérnio foi desenvolvido por William Rowan Hamilton em 1843 como uma generalização dos números complexos. É representado por quatro parâmetros de números reais (q_0, q_1, q_2, q_3) . E pode ser definido pela soma

$$q = q_0 + \mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}, \quad (\text{C.1})$$

sendo q_0 e $\mathbf{q}$ a parte escalar e vetorial do quatérnio, respectivamente. E os escalares q_0, q_1, q_2 e q_3 os componentes do quatérnio. As seguintes operações são definidas:

1. Dois quatérnios $p = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}$ e $q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ são *iguais* se e somente se os seus componentes são exatamente os mesmos. Ou seja, $p = q$ se e somente se

$$p_0 = q_0,$$

$$p_1 = q_1,$$

$$p_2 = q_2,$$

$$p_3 = q_3,$$

2. A *soma* de dois quatérnios p e q acima é definida pela adição dos componentes correspon-

dentes:

$$\begin{aligned} p + q &= (p_0 + q_0) + (p_1 + q_1)\mathbf{i} + (p_2 + q_2)\mathbf{j} \\ &\quad + (p_3 + q_3)\mathbf{k}. \end{aligned} \tag{C.2}$$

3. O *produto* de um escalar c e um quatérnio q é dado por

$$cq = cq_0 + cq_1\mathbf{i} + cq_2\mathbf{j} + cq_3\mathbf{k}. \tag{C.3}$$

4. O *produto* de dois quatérnios é definido pelos produtos dos elementos de base:

$$\begin{aligned} \mathbf{i}^2 &= \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, \\ \mathbf{ij} &= \mathbf{k} = -\mathbf{ji}, \\ \mathbf{jk} &= \mathbf{i} = -\mathbf{kj}, \\ \mathbf{ki} &= \mathbf{j} = -\mathbf{ik}. \end{aligned} \tag{C.4}$$

Assim, tem-se

$$\begin{aligned} p \otimes q &= (p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k})(q_0 + q_1\mathbf{i} + q_2\mathbf{j} \\ &\quad + q_3\mathbf{k}), \\ &= p_0q_0 - (p_1q_1 + p_2q_2 + p_3q_3) \\ &\quad + p_0(q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}) + q_0(p_1\mathbf{i} + p_2\mathbf{j} \\ &\quad + p_3\mathbf{k}) \\ &\quad + (p_2q_3 - p_3q_2)\mathbf{i} + (p_3q_1 - p_1q_3)\mathbf{j} \\ &\quad + (p_1q_2 - p_2q_1)\mathbf{k}, \end{aligned} \tag{C.5}$$

sendo $\otimes$ definido como a multiplicação entre dois quatérnios.

Lembrando o produto escalar e o produto vetorial da álgebra de vetores em três dimensões.

Dados os vetores

$$\mathbf{a} = (a_1, a_2, a_3) \text{ e } \mathbf{b} = (b_1, b_2, b_3), \tag{C.6}$$

então o produto escalar é dado por:

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3, \quad (\text{C.7})$$

e o produto vetorial é definido como

$$\begin{aligned} \mathbf{a} \times \mathbf{b} &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}, \\ &= (a_2 b_3 - a_3 b_2) \mathbf{i} + (a_3 b_1 - a_1 b_3) \mathbf{j} \\ &\quad + (a_1 b_2 - a_2 b_1) \mathbf{k}, \\ &= \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \mathbf{b} = [\mathbf{a}^\times] \mathbf{b}. \end{aligned} \quad (\text{C.8})$$

Usando estes resultados, pode-se escrever o produto de dois quatérnios p e q da seguinte forma

$$\begin{aligned} p \otimes q &= p_0 q_0 - \mathbf{p} \cdot \mathbf{q} + p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q}, \\ &= \begin{bmatrix} p_0 q_0 - \mathbf{p} \cdot \mathbf{q} \\ p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q} \end{bmatrix}, \\ &= \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \end{aligned} \quad (\text{C.9})$$

5. O *conjugado complexo* de um quatérnio é definido por

$$q^* = q_0 - \mathbf{q} = q_0 - q_1 \mathbf{i} - q_2 \mathbf{j} - q_3 \mathbf{k}. \quad (\text{C.10})$$

6. O *módulo* do quatérnio é definido por

$$|q| = \sqrt{q \otimes q^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \quad (\text{C.11})$$

7. Todo quatérnio diferente de zero tem um *multiplicativo inverso*. Pela definição de inversa, tem-se

$$q^{-1} \otimes q = q \otimes q^{-1} = 1. \quad (\text{C.12})$$

Assim, pré e pós multiplicando pelo complexo conjugado q^* , pode-se escrever:

$$q^{-1} \otimes q \otimes q^* = q^* \otimes q \otimes q^{-1} = q^*. \quad (\text{C.13})$$

E desde que $q \otimes q^* = |q|^2$, tem-se:

$$q^{-1} = \frac{q^*}{|q|^2}. \quad (\text{C.14})$$

Perceba que se q é um quatérnio unitário ou normalizado, $|q| = 1$, então a inversa é simplesmente o complexo conjugado

$$q^{-1} = q^*. \quad (\text{C.15})$$

C.1 Considerações geométricas

Sabe-se que $\cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} = 1$. Então, desde que $q_0^2 + |\mathbf{q}|^2 = 1$, tenha módulo 1, pode-se chamar $\cos^2 \frac{\theta}{2} = q_0^2$, e $\sin^2 \frac{\theta}{2} = |\mathbf{q}|^2$, para $\frac{\theta}{2}$ que satisfaz a restrição $-\pi < \frac{\theta}{2} \leq \pi$.

Dessa forma, define-se um vetor unitário $\mathbf{u}$ que representa a direção de $\mathbf{q}$

$$\mathbf{u} = \frac{\mathbf{q}}{|\mathbf{q}|} = \frac{\mathbf{q}}{\sin \frac{\theta}{2}}. \quad (\text{C.16})$$

Pode-se escrever o quatérnio unitário q em função do ângulo $\frac{\theta}{2}$ e do vetor unitário $\mathbf{u}$ como

$$q = q_0 + \mathbf{q} = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} \mathbf{u}. \quad (\text{C.17})$$

Desse modo, o quatérnio q será sempre um quatérnio *unitário* ou *normalizado*, ou seja com módulo 1,

$$q_0^2 + |\mathbf{q}|^2 = 1. \quad (\text{C.18})$$

Observe que substituindo $\frac{\theta}{2}$ por $-\frac{\theta}{2}$, obtém-se o complexo conjugado de q , que é

$$\begin{aligned} \cos\left(-\frac{\theta}{2}\right) + \operatorname{sen}\left(-\frac{\theta}{2}\right)\mathbf{u} &= \cos\frac{\theta}{2} + (-\operatorname{sen}\frac{\theta}{2})\mathbf{u}, \\ &= \cos\frac{\theta}{2} - \operatorname{sen}\frac{\theta}{2}\mathbf{u} = q^*. \end{aligned} \quad (\text{C.19})$$

Assim, por convenção, o quatérnio

$$q = q_0 + \mathbf{q}, \quad (\text{C.20})$$

é representado por

$$q = \cos\frac{\theta}{2} + \operatorname{sen}\frac{\theta}{2}\mathbf{u}. \quad (\text{C.21})$$

C.2 Operador quatérnio de rotação

O resultado da rotação q (representado como um quatérnio unitário) sobre algum ponto $\mathbf{v}$ (representado como um vetor) é dado por:

$$\begin{aligned} \mathbf{w} &= q \otimes \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} \otimes q^*, \\ &= (q_0 + \mathbf{q})(0 + \mathbf{v})(q_0 - \mathbf{q}), \\ &= (q_0\mathbf{v} + \mathbf{q}\mathbf{v})(q_0 - \mathbf{q}), \\ &= q_0^2\mathbf{v} - q_0\mathbf{v}\mathbf{q} + q_0\mathbf{q}\mathbf{v} - \mathbf{q}\mathbf{v}\mathbf{q}, \\ &= q_0^2\mathbf{v} - q_0(-\mathbf{v} \cdot \mathbf{q} + \mathbf{v} \times \mathbf{q}) \\ &\quad + q_0(-\mathbf{q} \cdot \mathbf{v} + \mathbf{q} \times \mathbf{v}) - (-\mathbf{q} \cdot \mathbf{v} + \mathbf{q} \times \mathbf{v})\mathbf{q}, \\ &= q_0^2\mathbf{v} + q_0(\mathbf{v} \cdot \mathbf{q}) - q_0(\mathbf{v} \times \mathbf{q}) - q_0(\mathbf{q} \cdot \mathbf{v}) \\ &\quad + q_0(\mathbf{q} \times \mathbf{v}) + (\mathbf{q} \cdot \mathbf{v})\mathbf{q} - (\mathbf{q} \times \mathbf{v})\mathbf{q}, \\ &= q_0^2\mathbf{v} + 2q_0(\mathbf{q} \times \mathbf{v}) + (\mathbf{q} \cdot \mathbf{v})\mathbf{q} - (\mathbf{q} \times \mathbf{v})\mathbf{q}, \\ &= q_0^2\mathbf{v} + 2q_0(\mathbf{q} \times \mathbf{v}) + (\mathbf{q} \cdot \mathbf{v})\mathbf{q} + (\mathbf{q} \times \mathbf{v}) \cdot \mathbf{q} \\ &\quad - \mathbf{q} \times \mathbf{v} \times \mathbf{q}. \end{aligned}$$

Como $(\mathbf{q} \times \mathbf{v}) \cdot \mathbf{q} = 0$ e $\mathbf{q} \times \mathbf{v} \times \mathbf{q} = (\mathbf{q} \cdot \mathbf{q})\mathbf{v} - (\mathbf{q} \cdot \mathbf{v})\mathbf{q}$, o operador $\mathbf{w}$ torna-se

$$\begin{aligned}
\mathbf{w} &= q_0^2 \mathbf{v} + 2q_0(\mathbf{q} \times \mathbf{v}) + (\mathbf{q} \cdot \mathbf{v})\mathbf{q} - (\mathbf{q} \cdot \mathbf{q})\mathbf{v} + (\mathbf{q} \cdot \mathbf{v})\mathbf{q}, \\
&= (q_0^2 - \mathbf{q} \cdot \mathbf{q})\mathbf{v} + 2(\mathbf{q} \cdot \mathbf{v})\mathbf{q} + 2q_0(\mathbf{q} \times \mathbf{v}), \\
&= (q_0^2 - |\mathbf{q}|^2)\mathbf{v} + 2(\mathbf{q} \cdot \mathbf{v})\mathbf{q} + 2q_0(\mathbf{q} \times \mathbf{v}), \\
&= ((q_0^2 - \mathbf{q}^T \mathbf{q})I_{3 \times 3} + 2\mathbf{q}\mathbf{q}^T + 2q_0[\mathbf{q}^\times])\mathbf{v}, \\
&= \left((q_0^2 - q_1^2 - q_2^2 - q_3^2)I_{3 \times 3} + 2 \begin{bmatrix} q_1^2 & q_1 q_2 & q_1 q_3 \\ q_1 q_2 & q_2^2 & q_2 q_3 \\ q_1 q_3 & q_2 q_3 & q_3^2 \end{bmatrix} + 2q_0 \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \right) \mathbf{v}, \\
&= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \mathbf{v}. \tag{C.22}
\end{aligned}$$

Utilizando (C.18) em (C.22), pode-se escrever o operador $\mathbf{w}$ como (C.23),

$$\begin{aligned}
\mathbf{w} &= \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \mathbf{v}, \\
&= Q\mathbf{v}, \tag{C.23}
\end{aligned}$$

sendo Q a matriz de rotação.

Um resultado equivalente pode ser obtido para (C.24).

$$\begin{aligned}
\mathbf{w} &= q^* \otimes \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} \otimes q, \\
&= ((q_0^2 - \mathbf{q}^T \mathbf{q})I_{3 \times 3} + 2\mathbf{q}\mathbf{q}^T - 2q_0[\mathbf{q}^\times])\mathbf{v}, \\
&= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}, \\
&= \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \\
&= Q^T \mathbf{v}. \tag{C.24}
\end{aligned}$$

C.3 Conversão entre quatérnios e ângulos de Euler

O operador quatérnio de rotação é equivalente a matriz de rotação para a sequência de ângulos de Euler aeroespacial:

$$\psi = \hat{\text{Ângulo de proa}}$$

$$\theta = \hat{\text{Ângulo de arfagem}}$$

$$\phi = \hat{\text{Ângulo de rolagem.}}$$

Os quatérnios constituintes são:

$$\begin{aligned} q_z &= \cos \frac{\psi}{2} + \operatorname{sen} \frac{\psi}{2} \mathbf{k}, \\ q_y &= \cos \frac{\theta}{2} + \operatorname{sen} \frac{\theta}{2} \mathbf{j} \quad \text{e} \\ q_x &= \cos \frac{\phi}{2} + \operatorname{sen} \frac{\phi}{2} \mathbf{i}, \end{aligned}$$

assim

$$q = q_z \otimes q_y \otimes q_x = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}, \quad (\text{C.25})$$

sendo

$$\begin{aligned} q_0 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} + \operatorname{sen} \frac{\psi}{2} \operatorname{sen} \frac{\theta}{2} \operatorname{sen} \frac{\phi}{2}, \\ q_1 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \operatorname{sen} \frac{\phi}{2} - \operatorname{sen} \frac{\psi}{2} \operatorname{sen} \frac{\theta}{2} \cos \frac{\phi}{2}, \\ q_2 &= \cos \frac{\psi}{2} \operatorname{sen} \frac{\theta}{2} \cos \frac{\phi}{2} + \operatorname{sen} \frac{\psi}{2} \cos \frac{\theta}{2} \operatorname{sen} \frac{\phi}{2} \quad \text{e} \\ q_3 &= \operatorname{sen} \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} - \cos \frac{\psi}{2} \operatorname{sen} \frac{\theta}{2} \operatorname{sen} \frac{\phi}{2}. \end{aligned}$$

Lembrando que a sequência ângulo/eixo aeroespacial, $(\psi, \theta, \phi) \rightarrow (zyx)$, é somente uma das vinte sequências possíveis. A matriz de rotação em termos de cossenos diretores, correspondente

a rotação dos ângulos de Euler (ψ, θ, ϕ) , com convenção (zyx) é dada por

$$\begin{aligned}
 R &= [rij] = R_\phi^x R_\theta^y R_\psi^z, \\
 &= \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix}, \\
 &= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}, \\
 &= \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix}. \tag{C.26}
 \end{aligned}$$

Para os ângulos de Euler tem-se

$$\begin{aligned}
 \tan\psi &= \frac{r_{12}}{r_{11}} = \frac{2(q_1q_2 + q_0q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \\
 &= \frac{2(q_1q_2 + q_0q_3)}{2(q_0^2 + q_1^2) - 1}, \tag{C.27}
 \end{aligned}$$

$$\sin\theta = -r_{13} = -2(q_1q_3 - q_0q_2), \tag{C.28}$$

$$\begin{aligned}
 \tan\phi &= \frac{r_{23}}{r_{33}} = \frac{2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \\
 &= \frac{2(q_2q_3 + q_0q_1)}{2(q_0^2 + q_3^2) - 1}. \tag{C.29}
 \end{aligned}$$

C.4 Derivada do quatérnio

Pode-se relacionar $q(t)$ e $q(t + \Delta t)$ como segue

$$\begin{aligned}
 q(t + \Delta t) &= q(t)\Delta r(t), \\
 \text{sendo } \Delta r(t) &= \cos\frac{\Delta\alpha}{2} + \sin\frac{\Delta\alpha}{2}\mathbf{r}(t), \tag{C.30}
 \end{aligned}$$

o quatérnio de transição incremental. Seu ângulo de rotação é $\Delta\alpha$ sobre o eixo definido pelo vetor unitário $\mathbf{r}(t)$. Assumindo $\Delta t \rightarrow 0$, tem-se

$$\begin{aligned}\Delta \mathbf{r}(t) &= \cos \frac{\Delta\alpha}{2} + \sin \frac{\Delta\alpha}{2} \mathbf{r}(t), \\ &= 1 + \frac{\Delta\alpha}{2} \mathbf{r}(t).\end{aligned}\tag{C.31}$$

Então

$$\begin{aligned}q(t + \Delta t) &= q(t) \left(1 + \frac{\Delta\alpha}{2} \mathbf{r}(t) \right), \\ q(t + \Delta t) - q(t) &= \frac{\Delta\alpha}{2} \begin{bmatrix} 0 \\ \mathbf{r}(t) \end{bmatrix}.\end{aligned}$$

Dividindo por Δt e utilizando limite, obtém-se

$$\begin{aligned}\dot{q}(t) &= \lim_{\Delta t \rightarrow 0} \frac{q(t + \Delta t) - q(t)}{\Delta t}, \\ &= \lim_{\Delta t \rightarrow 0} \frac{q(t) \otimes \begin{bmatrix} 0 \\ \mathbf{r}(t) \end{bmatrix} \frac{\Delta\alpha}{2}}{\Delta t},\end{aligned}\tag{C.32}$$

$$= q(t) \otimes \begin{bmatrix} 0 \\ \mathbf{r}(t) \end{bmatrix} \dot{\alpha}(t) = \frac{1}{2} q(t) \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}(t) \end{bmatrix},\tag{C.33}$$

sendo $\boldsymbol{\omega}(t) = \dot{\alpha} \mathbf{r}(t)$ o vetor velocidade angular do quatérnio Δ_r .

Na forma de matriz, a derivada do quatérnio pode ser escrita como

$$\begin{aligned}
 \dot{q}(t) &= \frac{1}{2}q(t) \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}(t) \end{bmatrix}, \\
 &= \frac{1}{2} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \\
 &= \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \\
 &= \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{C.34}
 \end{aligned}$$

Similarmente, para a derivada do conjugado, tem-se

$$\begin{aligned}
 \dot{q}^*(t) &= -\frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}(t) \end{bmatrix} \otimes q^*(t) \\
 &= \frac{1}{2} \begin{bmatrix} 0 & \omega_1 & \omega_2 & \omega_3 \\ -\omega_1 & 0 & \omega_3 & -\omega_2 \\ -\omega_2 & -\omega_3 & 0 & \omega_1 \\ -\omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix}, \\
 &= \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ -q_1 & -q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & -q_0 & q_1 \\ -q_3 & q_2 & -q_1 & -q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \\
 &= \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ -q_0 & q_3 & -q_2 \\ -q_3 & -q_0 & q_1 \\ q_2 & -q_1 & -q_0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{C.35}
 \end{aligned}$$

C.5 Integração do quatérnio

A abordagem do quatérnio usando um vetor de quatro parâmetros com uma restrição de norma unitária permite três graus de liberdade. Devido a restrição de norma unitária, os quatérnios são elementos de uma esfera unitária em quatro espaços. Esta esfera unitária não é um espaço vetorial Euclidiano onde se pode aplicar as definições de adição de vetores e escalonamento. Portanto, deve-se tomar cuidado na integração de equações diferenciais de quatérnios. Como mostrado em [24], se a taxa de amostragem é alta o suficiente e precisa para considerar a taxa de rotação como constante sobre o intervalo de amostragem T , então é possível encontrar e implementar uma solução de forma fechada para a equação (C.34).

A equação (C.34) pode ser reescrita como

$$\dot{q}(t) = \frac{1}{2}Q(t)q(t), \quad (\text{C.36})$$

como pretende-se encontrar $q(t_2)$ para $q(t_1)$ conhecido e $Q(t)$ constante para $t \in (t_1, t_2]$, sendo $T = t_2 - t_1$, então

$$Q(t) = Q = \frac{1}{2} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}. \quad (\text{C.37})$$

Definindo o fator de integração como $e^{-\int_{t_1}^t Q(\tau)d\tau}$ e multiplicando-o do lado esquerdo de (C.36) tem-se

$$\begin{aligned} e^{-\int_{t_1}^t Q(\tau)d\tau} \dot{q}(t) - e^{-\int_{t_1}^t Q(\tau)d\tau} Qq(t) &= 0, \\ \frac{d}{dt} \left(e^{-\int_{t_1}^t Q(\tau)d\tau} q(t) \right) &= 0. \end{aligned} \quad (\text{C.38})$$

Integrando ambos os lados sobre o intervalo, tem-se

$$\begin{aligned} e^{-\int_{t_1}^{t_2} Q(\tau)d\tau} q(t_2) &= e^{-\int_{t_1}^{t_1} Q(\tau)d\tau} q(t_1), \\ q(t_2) &= e^{\int_{t_1}^{t_2} Q(\tau)d\tau} q(t_1), \end{aligned} \quad (\text{C.39})$$

sendo identificado que $\left(e^{-\int_{t_1}^t Q(\tau)d\tau} \right)^{-1} = e^{\int_{t_1}^t Q(\tau)d\tau}$.

Para escrever (C.39) de uma forma mais conveniente, define-se $\mathbf{w} = [W_1 \ W_2 \ W_3]^T$ e

$$W = \begin{bmatrix} 0 & -W_1 & -W_2 & -W_3 \\ W_1 & 0 & -W_3 & W_2 \\ W_2 & W_3 & 0 & -W_1 \\ W_3 & -W_2 & W_1 & 0 \end{bmatrix}, \quad (\text{C.40})$$

sendo $W_1 = \frac{1}{2} \int_{t_1}^t \omega_1(\tau) d\tau$, $W_2 = \frac{1}{2} \int_{t_1}^t \omega_2(\tau) d\tau$, $W_3 = \frac{1}{2} \int_{t_1}^t \omega_3(\tau) d\tau$.

Com estas definições, a Equação (C.39) é equivalente a

$$q(t_2) = e^{\int_{t_1}^{t_2} Q(\tau) d\tau} q(t_1). \quad (\text{C.41})$$

Expandindo e^W usando séries de Taylor, tem-se

$$\begin{aligned} e^W &= I + W + \frac{W^2}{2!} + \frac{W^3}{3!} + \dots, \\ &= \left(I + \frac{W^2}{2!} + \frac{(W^2)^2}{4!} + \dots \right) \\ &\quad + W \left(I + \frac{W^2}{3!} + \frac{(W^2)^2}{5!} + \dots \right), \\ &= \cos(\|\mathbf{w}\|)I + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}W. \end{aligned} \quad (\text{C.42})$$

Como a dedução acima pode ser repetida para qualquer intervalo de duração T para qual é assumido que a taxa angular constante é válida, tem-se a equação geral

$$q(t_k) = \left(\cos(\|\mathbf{w}\|)I + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}W \right) q(t_{k-1}). \quad (\text{C.43})$$

sendo $\mathbf{w}$, $\|\mathbf{w}\|$ e W envolve a integral da taxa angular sobre o intervalo de amostragem $t \in (t_{k-1}, t_k]$. A solução é em forma fechada dada a suposição que $Q(t)$ é constante sobre o intervalo de tamanho T . E a solução preserva a norma pois o lado direito da Equação (C.43) é igual a $\|q(t_{k-1})\|$, conforme mostrado em [24].

APÊNDICE D

Mínimos quadrados regularizados

Neste apêndice é apresentado o problema de mínimos quadrados regularizados com incertezas nos parâmetros e sua solução é dada na forma de blocos matriciais. Este capítulo é baseado em [5, 66].

D.1 O problema de mínimos quadrados regularizados

Considere o seguinte problema de otimização

$$\min_{\mathbf{x}} \max_{\{\delta A, \delta \mathbf{b}\}} f(\mathbf{x}), \quad (\text{D.1})$$

$$f(\mathbf{x}) := \{\|\mathbf{x}\|_Q^2 + \|(A + \delta A)x - (\mathbf{b} + \delta \mathbf{b})\|_W^2\}, \quad (\text{D.2})$$

sendo A a matriz de dados, $\mathbf{b}$ o vetor de medida assumido como conhecido, $\mathbf{x}$ o vetor desconhecido, $Q = Q^T \geq 0$ and $W = W^T > 0$ são matrizes de ponderações dadas, δA , $\delta \mathbf{b}$ são perturbações modeladas por

$$\begin{bmatrix} \delta A & \delta \mathbf{b} \end{bmatrix} = M\Delta \begin{bmatrix} N_A & N_{\mathbf{b}} \end{bmatrix}, \quad \|\Delta\| \leq 1. \quad (\text{D.3})$$

O próximo lema é apresentado em [61] e em suas referências internas. É o lema fundamental

da teoria de filtragem robusta, onde é usada uma abordagem puramente determinística.

Lema D.1.1. *O problema de otimização (D.1)-(D.3) tem uma única solução $\widehat{\mathbf{x}}$ dada por*

$$\widehat{\mathbf{x}} = \left(\widehat{Q} + A^T \widehat{W} A \right)^{-1} \left(A^T \widehat{W} \mathbf{b} + \widehat{\lambda} N_A^T N_{\mathbf{b}} \right), \quad (\text{D.4})$$

sendo que as matrizes de ponderação modificadas $\widehat{Q}$ e $\widehat{W}$ são definidas por

$$\widehat{Q} := Q + \widehat{\lambda} N_A^T N_A, \quad (\text{D.5})$$

$$\widehat{W} := W + W M \left(\widehat{\lambda} I - M^T W M \right)^\dagger M^T W, \quad (\text{D.6})$$

e $\widehat{\lambda}$ é um parâmetro escalar não negativo obtido através do seguinte problema de otimização

$$\widehat{\lambda} := \arg \min_{\lambda \geq \|M^T W M\|} \Gamma(\lambda), \quad (\text{D.7})$$

sendo que

$$\Gamma(\lambda) := \|\mathbf{x}(\lambda)\|_Q^2 + \lambda \|N_A \mathbf{x}(\lambda) - N_{\mathbf{b}}\|^2 + \|A \mathbf{x}(\lambda) - \mathbf{b}\|_{W(\lambda)}^2, \quad (\text{D.8})$$

e as funções auxiliares são definidas por

$$\mathbf{x}(\lambda) := \left[Q(\lambda) + A^T W(\lambda) A \right]^{-1} \left[A^T W(\lambda) \mathbf{b} + \lambda N_A^T N_{\mathbf{b}} \right],$$

$$Q(\lambda) := Q + \lambda N_A^T N_A,$$

$$W(\lambda) := W + W M \left(\lambda I - M^T W M \right)^\dagger M^T W.$$

◇

É conhecido que há um mínimo global para a função $\Gamma(\lambda)$ em (D.7), veja [60], o qual deve ser minimizado para a obtenção da solução ótima do problema de otimização (D.1). No entanto, $\Gamma(\lambda)$ depende de $\mathbf{x}$ o qual é uma variável desconhecida. Não sendo uma tarefa simples realizar esta minimização em tempo real. Antes de apresentarmos a solução deste problema, serão apresentadas no próximo lema soluções equivalentes para este problema de otimização robusta.

Lema D.1.2. *As seguintes setenças são equivalentes*

$$(i) \quad \widehat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \max_{\{\delta A, \delta \mathbf{b}\}} f(\mathbf{x}), \quad (\text{D.9})$$

sendo δA e $\delta \mathbf{b}$ dadas por (D.3).

$$(ii) \quad \hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \left\{ \left(\begin{bmatrix} I \\ A \\ N_A \end{bmatrix} \mathbf{x} - \begin{bmatrix} 0 \\ \mathbf{b} \\ N_b \end{bmatrix} \right)^T \begin{bmatrix} Q & 0 & 0 \\ 0 & \widehat{W} & 0 \\ 0 & 0 & \widehat{\lambda} I \end{bmatrix} \left(\begin{bmatrix} \cdot \end{bmatrix} \right) \right\}. \quad (D.10)$$

(iii) $(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = (\hat{\mathbf{x}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$ é solução do sistema

$$\begin{bmatrix} Q & 0 & 0 & I & 0 & 0 & 0 \\ 0 & \widehat{W} & 0 & 0 & I & 0 & 0 \\ 0 & 0 & \widehat{\lambda} I & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & 0 & I \\ 0 & I & 0 & 0 & 0 & 0 & A \\ 0 & 0 & I & 0 & 0 & 0 & N_A \\ 0 & 0 & 0 & I & A^T & N_A^T & 0 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{b} \\ N_b \\ 0 \end{bmatrix}. \quad (D.11)$$

Se $(Q + A^T \widehat{W} A + \widehat{\lambda} N_A^T N_A)$ é invertível, tem-se que $\hat{\mathbf{x}}$ é a única solução de (ii). Além disso,

$$\begin{aligned} (Q + A^T \widehat{W} A + \widehat{\lambda} N_A^T N_A) &= - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \\ &\times \begin{bmatrix} Q & 0 & 0 & I & 0 & 0 & 0 \\ 0 & \widehat{W} & 0 & 0 & I & 0 & 0 \\ 0 & 0 & \widehat{\lambda} I & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & 0 & I \\ 0 & I & 0 & 0 & 0 & 0 & A \\ 0 & 0 & I & 0 & 0 & 0 & N_A \\ 0 & 0 & 0 & I & A^T & N_A^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I \end{bmatrix}. \end{aligned} \quad (D.12)$$

Prova: Veja em [5]. ◇

Lema D.1.3. *Seja $\widehat{\lambda}_k$ em (D.6) tal que $(\widehat{\lambda}_k I - M_k^T W_k M_k)$ é invertível. Então*

$$\widehat{W}_k := (W_k^{-1} - \lambda^{-1} M_k M_k^T)^{-1},$$

e a matriz do sistema (D.11) é equivalente a

$$\mathcal{T}\sigma = \mathcal{B} \quad (\text{D.13})$$

sendo

$$\mathcal{T} := \begin{bmatrix} -\mathcal{Q}_k & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & W_k^{-1} & 0 & 0 & 0 & 0 & M_k & A_k \\ 0 & 0 & 0 & \widehat{\lambda}_k^{-1}I & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & \widehat{\lambda}_k^{-1}I & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & N_{A_k} \\ 0 & 0 & 0 & I & 0 & I & 0 & 0 & 0 \\ 0 & 0 & M_k^T & 0 & I & 0 & 0 & 0 & 0 \\ 0 & I & A_k^T & 0 & 0 & N_{A_k}^T & 0 & 0 & 0 \end{bmatrix}, \sigma := \begin{bmatrix} \lambda_1 \\ \gamma_1 \\ \gamma_2 \\ e \\ d \\ \gamma_3 \\ \lambda_3 \\ c \\ x \end{bmatrix} \text{ e } \mathcal{B} := \begin{bmatrix} 0 \\ 0 \\ b_k \\ 0 \\ 0 \\ N_{b_k} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{D.14})$$

Prova: Veja em [5]. ◇

O próximo lema combina o Lema D.1.1 com os resultados apresentados em [2] e [45]. Eles fornecem expressões para soluções ótimas de problemas de otimização com restrição. Sua prova pode ser obtida seguindo os resultados fornecidos em [2].

Lema D.1.4. *Seja $V \in \mathbb{R}^{n \times n}$ definida positiva e $G \in \mathbb{R}^{k \times n}$. Considere o problema de minimização restrito*

$$\widehat{x} = \arg \min_x x^T V x, \quad (\text{D.15})$$

sujeito a

$$Gx = u \quad (\text{D.16})$$

sendo $u \in \mathbb{R}^{k \times 1}$ e $x \in \mathbb{R}^{n \times 1}$. Este problema pode ser transformado em um problema de minimização irrestrito considerando uma variável auxiliar μ

$$\widehat{x}(\mu) = \arg \min_x (\mathcal{G}x - \mathcal{B})^T \mathcal{V}(\mu) (\mathcal{G}x - \mathcal{B}) \quad (\text{D.17})$$

sendo

$$\mathcal{G} = \begin{bmatrix} I \\ G \end{bmatrix}, \quad \mathcal{V}(\mu) = \begin{bmatrix} V & 0 \\ 0 & \mu I \end{bmatrix} \quad e \quad \mathcal{B} = \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix},$$

$$\mu > 0.$$

Então, $\lim_{\mu \rightarrow \infty} \hat{\mathbf{x}}(\mu)$ sempre existe e é igual a

$$\lim_{\mu \rightarrow \infty} \hat{\mathbf{x}}(\mu) = \hat{\mathbf{x}}^0 \quad (\text{D.18})$$

sendo $\hat{\mathbf{x}}^0$ solução de (D.15)-(D.16) e $\hat{\mathbf{x}}(\mu)$ dado por

$$\hat{\mathbf{x}}(\mu) = \begin{bmatrix} 0 \\ I \end{bmatrix}^T \begin{bmatrix} \mathcal{V}(\mu)^{-1} & \mathcal{G} \\ \mathcal{G}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{B} \\ 0 \end{bmatrix}. \quad (\text{D.19})$$

Portanto,

$$\lim_{\mu \rightarrow \infty} (\mathcal{G}\hat{\mathbf{x}}(\mu) - \mathcal{B})^T \mathcal{V}(\mu) (\mathcal{G}\hat{\mathbf{x}}(\mu) - \mathcal{B}) = (\hat{\mathbf{x}}^0)^T V \hat{\mathbf{x}}^0.$$

◇

Observação D.1.1. Uma propriedade útil deste lema, o qual será usado para ajustar $\hat{\lambda}$ em (D.7), é que o termo quadrático

$$(G\hat{\mathbf{x}}(\mu) - \mathbf{u})^T \mu I (G\hat{\mathbf{x}}(\mu) - \mathbf{u}) \quad (\text{D.20})$$

tende para zero quando $\mu \rightarrow \infty$.

Prova do Teorema 2.3.1: Aplicando o Lema D.1.2, segue que o problema de otimização (2.34)

possui $\mathbf{x}$ como solução, por sua vez, é dada pela resolução do seguinte sistema linear

$$\begin{bmatrix} -\mathcal{Q}_k & 0 & 0 & I & 0 & 0 & 0 \\ 0 & -\widehat{W}_k & 0 & 0 & I & 0 & 0 \\ 0 & 0 & -\widehat{\lambda}_k I & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & 0 & I \\ 0 & I & 0 & 0 & 0 & 0 & A_k \\ 0 & 0 & I & 0 & 0 & 0 & N_{A_k} \\ 0 & 0 & 0 & I & A_k^T & N_{A_k}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_3 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{b}_k \\ N_{\mathbf{b}_k} \\ 0 \end{bmatrix}, \quad (\text{D.21})$$

sendo

$$\begin{aligned} \mathbf{x} &:= \begin{bmatrix} \widehat{\mathbf{x}}_1 \\ \widehat{\mathbf{x}}_2 \end{bmatrix}, \quad \widehat{\mathbf{x}}_1 = \begin{bmatrix} \widehat{\mathbf{x}}_k - \widehat{\mathbf{x}}_{k|k-1} \\ \widehat{\mathbf{w}}_k \\ \widehat{\mathbf{v}}_k \end{bmatrix}, \quad \widehat{\mathbf{x}}_2 = \begin{bmatrix} \widehat{\mathbf{x}}_{k+1|k} \end{bmatrix}, \quad A_k := \begin{bmatrix} A_{1,k} & A_{2,k} \end{bmatrix}, \\ A_{1,k} &:= \begin{bmatrix} F_k & \bar{G}_k \\ H_k & K_k \end{bmatrix}, \quad \bar{G}_k := \begin{bmatrix} G_k^T \\ 0 \end{bmatrix}^T, \quad K_k := \begin{bmatrix} 0 \\ I \end{bmatrix}^T, \quad A_{2,k} := \begin{bmatrix} -I \\ 0 \end{bmatrix}, \\ N_{A_k} &:= \begin{bmatrix} N_{A_{1,k}} & N_{A_{2,k}} \end{bmatrix}, \quad N_{A_{1,k}} := \begin{bmatrix} N_{F_k} & N_{\bar{G}_k} \\ N_{H_k} & 0 \end{bmatrix}, \quad N_{\bar{G}_k} := \begin{bmatrix} N_{G_k}^T \\ 0 \end{bmatrix}^T, \\ N_{A_{2,k}} &:= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathcal{Q}_k := \begin{bmatrix} \mathcal{Q}_{1,k} & 0 \\ 0 & \mathcal{Q}_2 \end{bmatrix}, \quad \mathcal{Q}_{1,k} := \begin{bmatrix} P_{k|k-1}^{-1} & 0 & 0 \\ 0 & Q_k^{-1} & 0 \\ 0 & 0 & R_k^{-1} \end{bmatrix}, \quad \text{e } \mathcal{Q}_2 := \begin{bmatrix} 0 \end{bmatrix}. \end{aligned} \quad (\text{D.22})$$

A expressão para $\widehat{W}_k$ é obtida através do Lema D.1.1

$$\widehat{W}_k := \left(\mu^{-1} I - \widehat{\lambda}_k^{-1} M_k M_k^T \right)^{-1}, \quad (\text{D.23})$$

desde que $(\mu^{-1} I - \widehat{\lambda}_k^{-1} M_k M_k^T)$ tenha inversa. O parâmetro $\widehat{\lambda}_k$ é obtido por meio das expressões D.7-D.8 do Lema D.1.1, ou seja, através da minimização da função $\Gamma(\lambda)$, quando

$$\widehat{\lambda}_k \geq \|M_k^T \mu I M_k\|, \quad (\text{D.24})$$

para $\mu > 0$ fixado. A Equação (D.24) pode ser reescrita como (2.36).

De forma equivalente, podemos escrever (D.21) como:

$$-\mathcal{Q}_k \boldsymbol{\lambda}_1 + \boldsymbol{\gamma}_1 = 0, \quad (\text{D.25})$$

$$-\widehat{W}_k \boldsymbol{\lambda}_2 + \boldsymbol{\gamma}_2 = 0, \quad (\text{D.26})$$

$$-\widehat{\lambda}_k I \boldsymbol{\lambda}_3 + \boldsymbol{\gamma}_3 = 0, \quad (\text{D.27})$$

$$\boldsymbol{\lambda}_1 + \mathbf{x} = 0, \quad (\text{D.28})$$

$$\boldsymbol{\lambda}_2 + A_k \mathbf{x} = \mathbf{b}_k, \quad (\text{D.29})$$

$$\boldsymbol{\lambda}_3 + N_{A_k} \mathbf{x} = N_{\mathbf{b}_k}, \quad (\text{D.30})$$

$$\boldsymbol{\gamma}_1 + A_k^T \boldsymbol{\gamma}_2 + N_{A_k}^T \boldsymbol{\gamma}_3 = 0, \quad (\text{D.31})$$

Através da Equação (D.26),

$$-\widehat{W}_k \boldsymbol{\lambda}_2 + \boldsymbol{\gamma}_2 = 0 \Rightarrow \boldsymbol{\gamma}_2 = \widehat{W}_k \boldsymbol{\lambda}_2 \Rightarrow \widehat{W}_k^{-1} \boldsymbol{\gamma}_2 = \boldsymbol{\lambda}_2. \quad (\text{D.32})$$

Já, pela Equação (D.29), segue que $\boldsymbol{\lambda}_2 + A_k \mathbf{x} = \mathbf{b}_k$. Dessa forma, $\widehat{W}_k^{-1} \boldsymbol{\gamma}_2 + A_k \mathbf{x} = \mathbf{b}_k$, que escrito em forma matricial torna-se

$$\begin{bmatrix} -\mathcal{Q}_k & 0 & I & 0 & 0 & 0 \\ 0 & -\widehat{\lambda}_k I & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & \widehat{W}_k^{-1} & 0 & A_k \\ 0 & I & 0 & 0 & 0 & N_{A_k} \\ 0 & 0 & I & A_k^T & N_{A_k}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_3 \\ \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \\ \boldsymbol{\gamma}_3 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{b}_k \\ N_{\mathbf{b}_k} \\ 0 \end{bmatrix}. \quad (\text{D.33})$$

Observe que neste primeiro procedimento da prova, $\boldsymbol{\lambda}_2$ foi retirado do vetor pré multiplicado. Como, por hipótese, $(\widehat{\lambda}_k I - M_k^T W_k M_k)$ é inversível, abrindo a expressão para $\widehat{W}_k^{-1}$ tem-se

$$\begin{bmatrix} -\mathcal{Q}_k & 0 & I & 0 & 0 & 0 \\ 0 & -\widehat{\lambda}_k I & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & \mu^{-1} I - \widehat{\lambda}_k^{-1} M_k M_k^T & 0 & A_k \\ 0 & I & 0 & 0 & 0 & N_{A_k} \\ 0 & 0 & I & A_k^T & N_{A_k}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_3 \\ \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \\ \boldsymbol{\gamma}_3 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{b}_k \\ N_{\mathbf{b}_k} \\ 0 \end{bmatrix}. \quad (\text{D.34})$$

Através da quinta linha de (D.34), segue a equação $(\mu^{-1}I - \hat{\lambda}_k^{-1}M_kM_k^T)\gamma_2 + A_k\mathbf{x} = \mathbf{b}_k$, ou seja, $\mu^{-1}I\gamma_2 - M_k(\hat{\lambda}_k^{-1}I)M_k^T\gamma_2 + A_kx = \mathbf{b}_k$.

Adotando a definição $(\hat{\lambda}_k^{-1}I)M_k^T\gamma_2 := -\mathbf{c}$, tem-se $M_k^T\gamma_2 + \hat{\lambda}_k\mathbf{c} = 0 \Rightarrow \mu^{-1}I\gamma_2 + M_k\mathbf{c} + A_k\mathbf{x} = \mathbf{b}_k$ que escrito em forma matricial, torna-se

$$\begin{bmatrix} -\mathcal{Q}_k & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & -\hat{\lambda}_k I & 0 & 0 & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & \mu^{-1}I & M_k & 0 & 0 & A_k \\ 0 & 0 & 0 & M_k^T & \hat{\lambda}_k I & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & N_{A_k} \\ 0 & 0 & I & A_k^T & 0 & N_{A_k}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_3 \\ \gamma_1 \\ \gamma_2 \\ \mathbf{c} \\ \gamma_3 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{b}_k \\ 0 \\ N_{\mathbf{b}_k} \\ 0 \end{bmatrix}. \quad (\text{D.35})$$

Neste segundo procedimento foi inserido $\mathbf{c}$ no vetor multiplicado, visando abrir a expressão de $\widehat{W}_k$.

Através da quinta linha de (D.35) tem-se $M_k^T\gamma_2 + \hat{\lambda}_k\mathbf{c} = 0$. Definindo agora $\hat{\lambda}_k\mathbf{c} := \mathbf{d}$ segue que $\mathbf{c} = \hat{\lambda}_k^{-1}\mathbf{d} \Rightarrow \mathbf{c} - \hat{\lambda}_k^{-1}\mathbf{d} = 0$. Além disso, $M_k^T\gamma_2 + \hat{\lambda}_k\mathbf{c} = 0 \Rightarrow M_k^T\gamma_2 + \mathbf{d} = 0$, que escrito em forma matricial é dado por

$$\begin{bmatrix} -\mathcal{Q}_k & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & -\hat{\lambda}_k I & 0 & 0 & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & \mu^{-1}I & M_k & 0 & 0 & A_k \\ 0 & 0 & 0 & M_k^T & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & -\hat{\lambda}_k^{-1}I & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & N_{A_k} \\ 0 & 0 & I & A_k^T & 0 & 0 & N_{A_k}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_3 \\ \gamma_1 \\ \gamma_2 \\ \mathbf{c} \\ \mathbf{d} \\ \gamma_3 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{b}_k \\ 0 \\ 0 \\ N_{\mathbf{b}_k} \\ 0 \end{bmatrix}. \quad (\text{D.36})$$

Veja que neste procedimento foi colocado $\mathbf{d}$ no vetor multiplicado, objetivando aparecer $\hat{\lambda}_k^{-1}$ no lugar de $\hat{\lambda}_k$.

Para finalizar, tem-se através da segunda linha de (D.36) que $-\hat{\lambda}_k I\lambda_3 + \gamma_3 = 0$.

Definindo $\mathbf{e} := -\hat{\lambda}_k I \boldsymbol{\lambda}_3 \Rightarrow \hat{\lambda}_k^{-1} \mathbf{e} + \boldsymbol{\lambda}_3 = 0$. Assim $\mathbf{e} + \boldsymbol{\gamma}_3 = 0$, ou seja,

$$\begin{bmatrix} -\mathcal{Q}_k & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & \mu^{-1}I & 0 & 0 & 0 & 0 & M_k & A_k & \\ 0 & 0 & 0 & \hat{\lambda}_k^{-1} & 0 & 0 & I & 0 & 0 & \\ 0 & 0 & 0 & 0 & -\hat{\lambda}_k^{-1} & 0 & 0 & I & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & N_{A_k} & \\ 0 & 0 & 0 & I & 0 & I & 0 & 0 & 0 & \\ 0 & 0 & M_k^T & 0 & I & 0 & 0 & 0 & 0 & \\ 0 & 0 & A_k^T & 0 & 0 & N_{A_k}^T & 0 & 0 & 0 & \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \\ \mathbf{e} \\ \mathbf{d} \\ \boldsymbol{\gamma}_3 \\ \boldsymbol{\lambda}_3 \\ \mathbf{c} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{b}_k \\ 0 \\ 0 \\ N_{\mathbf{b}_k} \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (\text{D.37})$$

Reescrevendo (D.37) em função das variáveis internas de (D.22), tem-se

$$\begin{bmatrix} -\mathcal{Q}_{1,k} & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & \mu^{-1}I & 0 & 0 & 0 & 0 & M_k & A_{1,k} & A_{2,k} & \\ 0 & 0 & 0 & 0 & 0 & \hat{\lambda}_k^{-1}I & 0 & 0 & I & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & -\hat{\lambda}_k^{-1}I & 0 & 0 & I & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & N_{A_{1,k}} & N_{A_{2,k}} & \\ 0 & 0 & 0 & 0 & 0 & I & 0 & I & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & M_k^T & 0 & I & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & I & 0 & A_{1,k}^T & 0 & 0 & N_{A_{1,k}} & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & I & A_{2,k}^T & 0 & 0 & N_{A_{2,k}} & 0 & 0 & 0 & 0 & \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1^1 \\ \boldsymbol{\lambda}_1^2 \\ \boldsymbol{\gamma}_1^1 \\ \boldsymbol{\gamma}_1^2 \\ \boldsymbol{\gamma}_2 \\ \mathbf{e} \\ \mathbf{d} \\ \boldsymbol{\gamma}_3 \\ \boldsymbol{\lambda}_3 \\ \mathbf{c} \\ \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{b}_k \\ 0 \\ 0 \\ N_{\mathbf{b}_k} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (\text{D.38})$$

sendo

$$\boldsymbol{\lambda}_1 = \begin{bmatrix} \boldsymbol{\lambda}_1^1 \\ \boldsymbol{\lambda}_1^2 \end{bmatrix}, \quad \boldsymbol{\gamma}_1 = \begin{bmatrix} \boldsymbol{\gamma}_1^1 \\ \boldsymbol{\gamma}_1^2 \end{bmatrix}. \quad (\text{D.39})$$

Da primeira linha de (D.38) tem-se $-\mathcal{Q}_{1,k} \boldsymbol{\lambda}_1^1 + \boldsymbol{\gamma}_1^1 = 0$. Definindo a variável auxiliar $\mathbf{g} := -\mathcal{Q}_{1,k} \boldsymbol{\lambda}_1^1$, segue que

$$\mathcal{Q}_{1,k}^{-1} \mathbf{g} + \boldsymbol{\lambda}_1^1 = 0, \text{ e} \quad (\text{D.40})$$

$$\mathbf{g} = -\boldsymbol{\gamma}_1^1. \quad (\text{D.41})$$

Substituindo a terceira linha de (D.38), $\boldsymbol{\lambda}_1^1 + \widehat{\mathbf{x}}_1 = 0$, e (D.41) em (D.40), tem-se

$$\mathcal{Q}_{1,k}^{-1} \boldsymbol{\lambda}_1^1 + \widehat{\mathbf{x}}_1 = 0. \quad (\text{D.42})$$

Substituindo a segunda linha de (D.38), $\gamma_1^2 = 0$, na ultima linha de (D.38), $\gamma_1^2 I + A_2^T \gamma_2 + N_{A,2}^T \gamma_3 = 0$, obtém-se

$$A_2^T \gamma_2 + N_{A,2}^T \gamma_3 = 0. \quad (\text{D.43})$$

Reescrevendo (D.38) utilizando (D.42) e (D.43), tem-se

$$\left[\begin{array}{cccccccccc} \mathcal{Q}_{1,k}^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & \mu^{-1} I & 0 & 0 & 0 & 0 & M_k & A_{1,k} & A_{2,k} \\ 0 & 0 & \widehat{\lambda}_k^{-1} I & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & -\widehat{\lambda}_k^{-1} I & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & N_{A_{1,k}} & N_{A_{2,k}} \\ 0 & 0 & I & 0 & I & 0 & 0 & 0 & 0 \\ 0 & M_k^T & 0 & I & 0 & 0 & 0 & 0 & 0 \\ I & A_{1,k}^T & 0 & 0 & N_{A_{1,k}}^T & 0 & 0 & 0 & 0 \\ 0 & A_{2,k}^T & 0 & 0 & N_{A_{2,k}}^T & 0 & 0 & 0 & 0 \end{array} \right] \left[\begin{array}{c} \gamma_1^1 \\ \gamma_2 \\ \mathbf{e} \\ \mathbf{d} \\ \gamma_3 \\ \boldsymbol{\lambda}_3 \\ \mathbf{c} \\ \widehat{\mathbf{x}}_1 \\ \widehat{\mathbf{x}}_2 \end{array} \right] = N_{\mathbf{b}_k} \left[\begin{array}{c} 0 \\ \mathbf{b}_k \\ 0 \\ 0 \\ N_{\mathbf{b}_k} \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right]. \quad (\text{D.44})$$

Observe que neste procedimento da prova foram retirados $\boldsymbol{\lambda}_1^1$ e $\boldsymbol{\lambda}_1^2$ do vetor pré multiplicado.

Utilizando as variáveis definidas em (D.22) em (D.44) a expressão (2.37) é obtida para $\widehat{\mathbf{x}}_{k|k}$.

Utilizando os mesmos cálculos algébricos feitos para $\widehat{\mathbf{x}}_{k|k}$, obtém-se a expressão (2.38) para $P_{k+1|k}$.

Agora, desejamos provar a Equação (2.39). Para isso, a partir de (D.37), tem-se

$$\mathcal{Q}_1 \boldsymbol{\lambda}_1 + I \boldsymbol{\gamma}_1 = 0 \Rightarrow I \boldsymbol{\gamma}_1 = \mathcal{Q}_1 \boldsymbol{\lambda}_1 \quad (\text{D.45})$$

$$I \boldsymbol{\lambda}_1 + I \mathbf{x} = 0 \Rightarrow I \boldsymbol{\lambda}_1 = -I \mathbf{x}, \quad (\text{D.46})$$

$$\mu^{-1} I \boldsymbol{\gamma}_2 + M_k \mathbf{c} + A_k \mathbf{x} = \mathbf{b}_k, \quad (\text{D.47})$$

$$\widehat{\lambda}_k^{-1} I \mathbf{e} + I \boldsymbol{\lambda}_3 = 0, \quad (\text{D.48})$$

$$-\widehat{\lambda}_k^{-1} I \mathbf{d} + I \mathbf{c} = 0, \quad (\text{D.49})$$

$$I \boldsymbol{\lambda}_3 + N_{A_k} \mathbf{x} = N_{\mathbf{b}_k} \Rightarrow I \boldsymbol{\lambda}_3 = N_{\mathbf{b}_k} - N_{A_k} \mathbf{x}, \quad (\text{D.50})$$

$$I \mathbf{e} + I \boldsymbol{\gamma}_3 = 0 \Rightarrow I \mathbf{e} = -I \boldsymbol{\gamma}_3, \quad (\text{D.51})$$

$$M_k^T \boldsymbol{\gamma}_2 + I \mathbf{d} = 0 \Rightarrow I \mathbf{d} = -M_k^T \boldsymbol{\gamma}_2, \quad (\text{D.52})$$

$$I \boldsymbol{\gamma}_1 + A_k^T \boldsymbol{\gamma}_2 + N_{A_k}^T \boldsymbol{\gamma}_3 = 0. \quad (\text{D.53})$$

Substituindo (D.46) em (D.45), tem-se

$$\boldsymbol{\gamma}_1 = -\mathcal{Q}_k \mathbf{x}. \quad (\text{D.54})$$

Substituindo (D.52) em (D.49), tem-se

$$\mathbf{c} = -\lambda_k^{-1} I M_k^T \boldsymbol{\gamma}_2. \quad (\text{D.55})$$

Substituindo (D.55) em (D.47), tem-se

$$W_k^{-1} \boldsymbol{\gamma}_2 - M_k \widehat{\lambda}_k^{-1} I M_k^T \boldsymbol{\gamma}_2 = \mathbf{b}_k - A_k \mathbf{x} \quad (\text{D.56})$$

$$(W_k^{-1} - M_k \widehat{\lambda}_k^{-1} I M_k^T) \boldsymbol{\gamma}_2 = \mathbf{b}_k - A_k \mathbf{x} \quad (\text{D.57})$$

$$\boldsymbol{\gamma}_2 = (W_k^{-1} - M_k \widehat{\lambda}_k^{-1} I M_k^T)^{-1} (\mathbf{b}_k - A_k \mathbf{x}) \quad (\text{D.58})$$

$$\boldsymbol{\gamma}_2 = \widehat{W}_k (\mathbf{b}_k - A_k \mathbf{x}). \quad (\text{D.59})$$

Substituindo (D.50) e (D.51) em (D.48), tem-se

$$-\widehat{\lambda}_k^{-1} I \boldsymbol{\gamma}_3 + N_{\mathbf{b}_k} - N_{A_k} \mathbf{x} = 0 \quad (\text{D.60})$$

$$\boldsymbol{\gamma}_3 = \widehat{\lambda}_k I (N_{\mathbf{b}_k} - N_{A_k} \mathbf{x}). \quad (\text{D.61})$$

Substituindo (D.54), (D.59) e (D.61) em (D.53), tem-se

$$-\mathcal{Q}_k \mathbf{x} + A_k^T \widehat{W}_k (\mathbf{b}_k - A_k \mathbf{x}) + N_{A_k}^T \widehat{\lambda}_k I (N_{\mathbf{b}_k} - N_{A_k} \mathbf{x}) = 0 \quad (\text{D.62})$$

$$-(\mathcal{Q}_k + A_k^T \widehat{W}_k A_k + N_{A_k}^T \widehat{\lambda}_k I N_{A_k}) \mathbf{x} + A_k^T \widehat{W}_k \mathbf{b}_k + N_{A_k}^T \widehat{\lambda}_k I N_{\mathbf{b}_k} = 0. \quad (\text{D.63})$$

Segue abaixo a expansão dos termos de (D.63).

Expandindo o termo $\widehat{W}_k = (\mu^{-1}I - \widehat{\lambda}^{-1}M_k M_k^T)^{-1}$, tem-se

$$\begin{aligned} \widehat{W}_k &= (\mu^{-1}I - \widehat{\lambda}^{-1}M_k M_k^T)^{-1} \\ &= \left(\begin{bmatrix} \mu^{-1}I & 0 \\ 0 & \mu^{-1}I \end{bmatrix} - \widehat{\lambda}^{-1} \begin{bmatrix} M_{1,k} & 0 \\ 0 & M_{2,1} \end{bmatrix} \begin{bmatrix} M_{1,k}^T & 0 \\ 0 & M_{2,1}^T \end{bmatrix} \right)^{-1} \\ &= \left(\begin{bmatrix} \mu^{-1}I & 0 \\ 0 & \mu^{-1}I \end{bmatrix} - \begin{bmatrix} \widehat{\lambda}^{-1}M_{1,k}M_{1,k}^T & 0 \\ 0 & \widehat{\lambda}^{-1}M_{2,1}M_{2,1}^T \end{bmatrix} \right)^{-1} \\ &= \left(\begin{bmatrix} \mu^{-1}I - \widehat{\lambda}^{-1}M_{1,k}M_{1,k}^T & 0 \\ 0 & \mu^{-1}I - \widehat{\lambda}^{-1}M_{2,1}M_{2,1}^T \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} (\mu^{-1}I - \widehat{\lambda}^{-1}M_{1,k}M_{1,k}^T)^{-1} & 0 \\ 0 & (\mu^{-1}I - \widehat{\lambda}^{-1}M_{2,k}M_{2,k}^T)^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \widehat{W}_{11,k} & 0 \\ 0 & \widehat{W}_{22,k} \end{bmatrix}. \end{aligned} \quad (\text{D.64})$$

$$\begin{aligned} A_k^T \widehat{W}_k A_k &= \begin{bmatrix} F_k^T & H_k^T \\ \bar{G}_k^T & K_k^T \\ -I & 0 \end{bmatrix} \begin{bmatrix} \widehat{W}_{11,k} & 0 \\ 0 & \widehat{W}_{22,k} \end{bmatrix} \begin{bmatrix} F_k & \bar{G}_k & -I \\ H_k & K_k & 0 \end{bmatrix} \\ &= \begin{bmatrix} F_k^T & H_k^T \\ \bar{G}_k^T & K_k^T \\ -I & 0 \end{bmatrix} \begin{bmatrix} \widehat{W}_{11,k}F_k & \widehat{W}_{11,k}\bar{G}_k & -\widehat{W}_{11,k} \\ \widehat{W}_{22,k}H_k & \widehat{W}_{22,k}K_k & 0 \end{bmatrix} \\ &= \begin{bmatrix} F_k^T \widehat{W}_{11,k}F_k + H_k^T \widehat{W}_{22,k}H_k & F_k^T \widehat{W}_{11,k}\bar{G}_k + H_k^T \widehat{W}_{22,k}K_k & -F_k^T \widehat{W}_{11,k} \\ \bar{G}_k^T \widehat{W}_{11,k}F_k + K_k^T \widehat{W}_{22,k}H_k & \bar{G}_k^T \widehat{W}_{11,k}\bar{G}_k + K_k^T \widehat{W}_{22,k}K_k & -\bar{G}_k^T \widehat{W}_{11,k} \\ -\widehat{W}_{11,k}F_k & -\widehat{W}_{11,k}\bar{G}_k & \widehat{W}_{11,k} \end{bmatrix} \end{aligned} \quad (\text{D.65})$$

$$\begin{aligned}
N_{A_k}^T \hat{\lambda}_k I N_{A_k} &= \begin{bmatrix} N_{F_k}^T & N_{H_k}^T \\ N_{\bar{G}_k}^T & N_{K_k}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\lambda}_k & 0 \\ 0 & \hat{\lambda}_k \end{bmatrix} \begin{bmatrix} N_{F_k} & N_{\bar{G}_k} & 0 \\ N_{H_k} & N_{K_k} & 0 \end{bmatrix} \\
&= \begin{bmatrix} N_{F_k}^T & N_{H_k}^T \\ N_{\bar{G}_k}^T & N_{K_k}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\lambda}_k N_{F_k} & \widehat{W}_{11,k} N_{\bar{G}_k} & 0 \\ \hat{\lambda}_k N_{H_k} & \hat{\lambda}_k N_{K_k} & 0 \end{bmatrix} \\
&= \begin{bmatrix} N_{F_k}^T \hat{\lambda}_k N_{F_k} + N_{H_k}^T \hat{\lambda}_k N_{H_k} & N_{F_k}^T \hat{\lambda}_k N_{\bar{G}_k} + N_{H_k}^T \hat{\lambda}_k N_{K_k} & 0 \\ N_{\bar{G}_k}^T \hat{\lambda}_k N_{F_k} + N_{K_k}^T \hat{\lambda}_k N_{H_k} & N_{\bar{G}_k}^T \hat{\lambda}_k N_{\bar{G}_k} + N_{K_k}^T \hat{\lambda}_k N_{K_k} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (D.66)
\end{aligned}$$

$$\begin{aligned}
A_k^T \widehat{W}_k \mathbf{b}_k &= \begin{bmatrix} F_k^T & H_k^T \\ \bar{G}_k^T & K_k^T \\ -I & 0 \end{bmatrix} \begin{bmatrix} \widehat{W}_{11,k} & 0 \\ 0 & \widehat{W}_{22,k} \end{bmatrix} \begin{bmatrix} -F_k \widehat{\mathbf{x}}_{k|k-1} \\ \mathbf{z}_k - H_k \widehat{\mathbf{x}}_{k|k-1} \end{bmatrix} \\
&= \begin{bmatrix} F_k^T & H_k^T \\ \bar{G}_k^T & K_k^T \\ -I & 0 \end{bmatrix} \begin{bmatrix} -\widehat{W}_{11,k} F_k \widehat{\mathbf{x}}_{k|k-1} \\ \widehat{W}_{22,k} (\mathbf{z}_k - H_k \widehat{\mathbf{x}}_{k|k-1}) \end{bmatrix} \\
&= \begin{bmatrix} -F_k^T \widehat{W}_{11,k} F_k \widehat{\mathbf{x}}_{k|k-1} + H_k^T \widehat{W}_{22,k} (\mathbf{z}_k - H_k \widehat{\mathbf{x}}_{k|k-1}) \\ -\bar{G}_k^T \widehat{W}_{11,k} F_k \widehat{\mathbf{x}}_{k|k-1} + K_k^T \widehat{W}_{22,k} (\mathbf{z}_k - H_k \widehat{\mathbf{x}}_{k|k-1}) \\ \widehat{W}_{11,k} F_k \widehat{\mathbf{x}}_{k|k-1} \end{bmatrix} \quad (D.67)
\end{aligned}$$

$$\begin{aligned}
N_{A_k}^T \hat{\lambda}_k I N_{\mathbf{b}_k} &= \begin{bmatrix} N_{F_k}^T & N_{H_k}^T \\ N_{\bar{G}_k}^T & N_{K_k}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\lambda}_k & 0 \\ 0 & \hat{\lambda}_k \end{bmatrix} \begin{bmatrix} -N_{F_k} \hat{\mathbf{x}}_{k|k-1} \\ -N_{H_k} \hat{\mathbf{x}}_{k|k-1} \end{bmatrix} \\
&= \begin{bmatrix} N_{F_k}^T & N_{H_k}^T \\ N_{\bar{G}_k}^T & N_{K_k}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\hat{\lambda}_k N_{F_k} \hat{\mathbf{x}}_{k|k-1} \\ -\hat{\lambda}_k N_{H_k} \hat{\mathbf{x}}_{k|k-1} \end{bmatrix} \\
&= \begin{bmatrix} -N_{F_k}^T \hat{\lambda}_k N_{F_k} \hat{\mathbf{x}}_{k|k-1} - N_{H_k}^T \hat{\lambda}_k N_{H_k} \hat{\mathbf{x}}_{k|k-1} \\ -N_{\bar{G}_k}^T \hat{\lambda}_k N_{F_k} \hat{\mathbf{x}}_{k|k-1} - N_{K_k}^T \hat{\lambda}_k N_{H_k} \hat{\mathbf{x}}_{k|k-1} \\ 0 \end{bmatrix}. \tag{D.68}
\end{aligned}$$

Da terceira linha de (D.63) temos

$$- \left(-\widehat{W}_{11,k} F_k (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}) - \widehat{W}_{11,k} \bar{G}_k \boldsymbol{\nu}_k + \widehat{W}_{11,k} \hat{\mathbf{x}}_{k+1|k} \right) + \widehat{W}_{11,k} F_k \hat{\mathbf{x}}_{k|k-1} = 0. \tag{D.69}$$

Desenvolvendo a Equação (D.69) acima, tem-se

$$\begin{aligned}
\widehat{W}_{11,k} F_k (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}) + \widehat{W}_{11,k} \bar{G}_k \hat{\boldsymbol{\nu}}_k - \widehat{W}_{11,k} \hat{\mathbf{x}}_{k+1|k} + \widehat{W}_{11,k} F_k \hat{\mathbf{x}}_{k|k-1} &= 0, \\
\widehat{W}_{11,k} F_k \hat{\mathbf{x}}_{k|k} + \widehat{W}_{11,k} \bar{G}_k \hat{\boldsymbol{\nu}}_k - \widehat{W}_{11,k} \hat{\mathbf{x}}_{k+1|k} &= 0, \\
\widehat{W}_{11,k} \hat{\mathbf{x}}_{k+1|k} &= \widehat{W}_{11,k} F_k \hat{\mathbf{x}}_{k|k} + \widehat{W}_{11,k} \bar{G}_k \hat{\boldsymbol{\nu}}_k, \\
\hat{\mathbf{x}}_{k+1|k} &= \widehat{W}_{11,k}^{-1} \widehat{W}_{11,k} F_k \hat{\mathbf{x}}_{k|k} + \widehat{W}_{11,k}^{-1} \widehat{W}_{11,k} \bar{G}_k \hat{\boldsymbol{\nu}}_k, \\
\hat{\mathbf{x}}_{k+1|k} &= F_k \hat{\mathbf{x}}_{k|k} + \bar{G}_k \hat{\boldsymbol{\nu}}_k,
\end{aligned}$$

Substituindo $\bar{G}_k$ definido em (D.22), obtém-se a expressão (2.39).

APÊNDICE E

Código embarcado

O código embarcado foi desenvolvido em *Dynamic C*. O código principal para o desenvolvimento do algoritmos de filtragem apresentados no Capítulo 2 é mostrado abaixo, e as principais funções são descritas na sequência.

```
// Configuration of the Wi-Fi
#define TCPCONFIG 5 #define _PRIMARY_STATIC_IP "192.168.0.110"
#define _PRIMARY_NETMASK "255.255.255.0"
#define MY_GATEWAY "10.10.6.1"
#define MY_NAMESERVER "10.10.6.1"
#define IFC_WIFI_SSID "LASI"
#define IFC_WIFI_ROAM_ENABLE 1
#define IFC_WIFI_ROAM_BEACON_MISS 20
#define IFC_WIFI_MODE IFPARAM_WIFI_INFRASTRUCTURE
#define IFC_WIFI_REGION IFPARAM_WIFI_REGION_AMERICAS
#define IFC_WIFI_ENCRYPTION IFPARAM_WIFI_ENCR_WEP
#define IFC_WIFI_WEP_KEYNUM 0
#define IFC_WIFI_WEP_KEY0_HEXSTR "1234567890"
```

```
#memmap xmem

#define BUFFER_LEN 32
#define MSG_LEN 1024 //in Hz
#define Min_NMEA 40

// Libraries
#include "dcrtcp.lib"
#include "uav_config.lib"
#include "servos_pololu.lib"
#include "IMU.LIB"
#include "gps_mod.LIB"
#include "kalman_filter.lib"
#include "libmat.lib"
#include "matrices_a.lib"
#include "matrices_p.lib"

// Variables
tcp_Socket sock;
GPSPosition pos, pos_ant, pos_IMU;
NEDFrame vel, vel_ant, vel_IMU;
GPSDop dop;
struct tm ntime;
char sentence[MSG_LEN], *p_sen;
char buffer[BUFFER_LEN];
char *motor_number, *position, *position_end;
unsigned char msg_IMU[128];
unsigned char msg_send[90];
unsigned long time, time0;
int bytes;
int cont;
int nb;
```

```
int checksum;
int IMU_OK;
int GPS_OK, DOP_OK, TM_OK;
int initIMU_OK;
int initGPS_OK;
int restart_initGPS;
int restart_initIMU;
int mem;
float dlambda, dphi;

main() {
// *****
initIMU_OK = 0;
initGPS_OK = 0;
GPS_OK = 0;
DOP_OK = 0;
TM_OK = 0;
restart_initGPS = 0;
restart_initIMU = 0;
mem = 1;
// *****
GPS_init(&pos, &pos_ant, &vel, &vel_ant, &dop, &ntime);
// *****
BoardInit(); // Configuration of the Rabbit
adj_IMU(&IMU); // Calibration of the IMU
reset_servos(); // Servos init
// *****
sock_init_or_exit(1);
tcp_listen(&sock, 1234, 0, 0, NULL, 0);
// Update attitude weighting matrices
update_wm_a(&wm_a);
// *****
// Update position weighting matrices
```

```
update_wm_p(&wm_p);
// *****
time = MS_TIMER;
while(1) {
time = MS_TIMER;
// *****
costate {
if (!sock_established(&sock)) {
tcp_listen(&sock, 1234, 0, 0, NULL, 0);
while(!sock_established(&sock)){
tcp_tick(NULL);
waitfor(DelayMs(20));
}
}
}
// *****
costate {
//waitfor(IMU_OK = get_IMU(msg_IMU, &IMU));
IMU_OK = get_IMU(msg_IMU, &IMU);
waitfor(DelayMs(40));
}
// *****
costate {
nb = serFread(sentence,1023,1);
if (nb) {
// *****
//Saving previous data
if (GPS_OK) {
pos_ant = pos;
}
GPS_OK = 0;
DOP_OK = 0;
TM_OK = 0;
```

```

// *****
sentence[nb] = '\0';
p_sen = sentence;
while ((&sentence[nb] - p_sen) > Min_NMEA) {
p_sen = _n_strchr(p_sen, '$');
if(p_sen == NULL)
break;
//Getting new data
if (gps_get_pos(&pos,p_sen) == 0) {
GPS_OK = 1;
break;
}
if (gps_get_dop(&dop, p_sen) == 0) {
DOP_OK = 1;
}
if (gps_get_utc(&ntime, p_sen)==0) { TM_OK = 1;
}
p_sen++;
}
// *****
if (GPS_OK) {
llh_2_ned(&vel, &pos, &pos_ant);
}
else {
pos = pos_ant;
}
// *****
}
waitfor(DelayMs(80));
}
// *****
costate {
if ((MS_TIMER - time < 10000) & IMU_OK || restart_initIMU) {

```

```
// Initial conditions
m [0] = IMU.scaled.mag.x;
m [1] = IMU.scaled.mag.y;
m [2] = IMU.scaled.mag.z;
eangles.psi = headingangle (m);
eangles.theta = asin(IMU.scaled.acel.x); // pitch angle
eangles.phi = asin(IMU.scaled.acel.y); // roll angle
euler2quat_m(&eangles, ss_a.x);
ss_a.x[4] = bg[0];
ss_a.x[5] = bg[1];
ss_a.x[6] = bg[2];
update_wm_a(&wm_a);
initIMU_OK = 1;
restart_initIMU = 0;
}
if (initIMU_OK & IMU_OK) {
// Magnetometer
m [0] = IMU.scaled.mag.x;
m [1] = IMU.scaled.mag.y;
m [2] = IMU.scaled.mag.z;
// Angular rate
omega_g[0] = IMU.scaled.rate.x*PI/180;
omega_g[1] = IMU.scaled.rate.y*PI/180;
omega_g[2] = IMU.scaled.rate.z*PI/180;
// Accelerometers
a[0] = IMU.scaled.acel.x*9.81 - ba[0];
a[1] = -IMU.scaled.acel.y*9.81 - ba[1];
a[2] = -IMU.scaled.acel.z*9.81 - ba[2];
// Observation
ss_a.z[0] = a[0];
ss_a.z[1] = a[1];
ss_a.z[2] = a[2];
ss_a.z[3] = headingangle(m);
```

```

// Estimated angular rate
less_m(omega_g,naxes, 1, bg, naxes, 1, omega_hat, naxes, 1);

// Update attitude matrices
update_ss_a(omega_hat, &ss_a);

// Predictor ekf_p_a(&wm_a, &ss_a);
erkfsayed_p(&wm_p, &ss_p, &wm_a, &ss_a, 0);

// Update H matrix with estimated state
update_H_a(ss_a.H, ss_a.xp);

// Error of the observation
update_h_a(ss_a.xp,ss_a.h);

less_m(ss_a.z,nobs_a, 1, ss_a.h, nobs_a, 1, ss_a.deltaz, nobs_a, 1);
ss_a.deltaz[3] = subtr_ang(ss_a.z[3], ss_a.h[3]);

// Update ekf_u_a(&wm_a, &ss_a);
erkfsayed_u(&wm_p, &ss_p, &wm_a, &ss_a, 0);

// Update of the gyros bias
bg[0] = ss_a.x[4];
bg[1] = ss_a.x[5];
bg[2] = ss_a.x[6];

// Quaternion for euler angles
quat2euler(ss_a.x, &eangles);
}

waitfor(DelayMs(30));
}

// *****
costate {
if (GPS_OK && (pos.num_SVs >= 6) && (initGPS_OK == 0) || restart_initGPS) {
// Initial conditions for position
// LLA position
ss_p.x[0] = pos.lat_degrees*_pi/180;
ss_p.x[1] = pos.lon_degrees*_pi/180;
ss_p.x[2] = pos.altitude;

// NED velocity
ss_p.x[3] = vel.n;

```

```
ss_p.x[4] = vel.e;
ss_p.x[5] = vel.d;
// Acelerometers bias
ss_p.x[6] = ba[0];
ss_p.x[7] = ba[1];
ss_p.x[8] = ba[2];
//eye_m(1,wm_p.P,nstates_p, nstates_p);
update_wm_p(&wm_p);
pos_ant = pos;
vel_ant = vel;
initGPS_OK = 1;
restart_initGPS = 0;
}

// *****
if (initGPS_OK && GPS_OK) {
    dlambda = fabs((pos.lat_degrees - pos_ant.lat_degrees));
    dphi = fabs((pos.lon_degrees - pos_ant.lon_degrees));
    if ((dlambda > 0.5) || (dphi > 0.5)) {
        GPS_OK = 0;
        pos = pos_ant;
        vel = vel_ant;
    }
}

// *****
if (!GPS_OK) {
    if (mem) {
        pos_IMU = pos_ant;
        vel_IMU = vel_ant;
        mem = 0;
    }
    GPS_IMU(a, ss_a.x, &pos_IMU, &vel_IMU, &pos, &vel);
    pos_IMU = pos;
    vel_IMU = vel;
```

```

}
else {
mem = 1;
}

// *****

if (initGPS_OK) {
// Update position matrices
update_ss_p(&ss_p, &pos, &vel, ss_a.x);
// Predictor ekf_p_p(&wm_p, &ss_p);
erkfsayed_p(&wm_p, &ss_p, &wm_a, &ss_a, 1);
// Observation error
update_h_p(ss_p.xp, ss_p.h);
less_m(ss_p.z, nobsp, 1, ss_p.h, nobsp, 1, ss_p.deltaz, nobsp, 1);
ss_p.deltaz[0] = subtr_ang(ss_p.z[0], ss_p.h[0]);
ss_p.deltaz[1] = subtr_ang(ss_p.z[1], ss_p.h[1]);
// Update ekf_u_p(&wm_p, &ss_p);
erkfsayed_u(&wm_p, &ss_p, &wm_a, &ss_a, 1);
// Update acelerometers bias
ba[0] = ss_p.x[6];
ba[1] = ss_p.x[7];
ba[2] = ss_p.x[8];
}
waitfor(DelayMs(30));
}

// printf("time_kf_p:  %d a \n", (MS_TIMER - time));
// time = MS_TIMER;

// *****

costate{
msg_send[0] = 'U';
msg_send[1] = 'U';
:
msg_send[87] = (0x000000FF & ((long int) (ss_p.x[5]*1e7) ) ); // v_d
msg_send[86] = (0x0000FF00 & ((long int) (ss_p.x[5]*1e7) ) ) » 8;

```

```
msg_send[85] = (0x00FF0000 & ((long int) (ss_p.x[5]*1e7) ) ) » 16;
msg_send[84] = (0xFF000000 & ((long int) (ss_p.x[5]*1e7) ) ) » 24;

// checksum
checksum = msg_send[2];
for (cont = 3; cont<88; cont++) {
checksum += msg_send[cont];
}
msg_send[89] = (0x00FF & checksum) ;
msg_send[88] = (0xFF00 & checksum) » 8;

if (sock_established(&sock)) {
tcp_tick(NULL);
sock_fastwrite( &sock, msg_send, sizeof(msg_send));
}
waitfor(DelayMs(40));
}

// *****
costate {
if (sock_established(&sock)) {
tcp_tick(NULL);
bytes = sock_fastread(&sock, buffer, BUFFER_LEN-1);
if (bytes > 0) {
buffer[bytes] = '\0';
if(!_n_strncmp(buffer,"set_position8",13)) {
motor_number = _n_strchr(buffer,'(');
position = _n_strchr(buffer,',');
position_end = _n_strchr(buffer,')');
if ((motor_number != NULL) & (position != NULL) & (position_end != NULL)) {
*position = '\0';
*position_end = '\0';
motor_number++;
position ++;
}
```

```

set_position8(atoi(motor_number),atoi(position));
}
}
if(!_n_strncmp(buffer,"restart_initGPS",13))
restart_initGPS = 1;
if(!_n_strncmp(buffer,"restart_initIMU",13))
restart_initIMU = 1;
}
}
waitfor(DelayMs(20));
}
}
}

```

◇

As principais funções desenvolvidas para o programa acima, são mostradas abaixo.

gps_get_pos <gps.lib>

SINTAXE: int gps_get_pos(GPSPosition *newpos, char *sentence);

DESCRIÇÃO: analisa uma sentença para extrair dados de posição. Esta função é capaz de analisar qualquer dos seguintes formatos de sentença GPS: GGA, GLL, RMC.

PARÂMETRO 1: newpos - uma estrutura GPSPosition a ser preenchida. PARÂMETRO 2: sentence - uma string contendo uma linha de dados GPS no formato NMEA-0183.

VALOR DE RETORNO: 0 - sucesso

-1 - erro de análise

-2 - sentença inválida

gps_get_dop <gps.lib>

SINTAXE: int gps_get_dop(GPSDop *newdop, char *sentence);

DESCRIÇÃO: analisa uma sentença para extrair dados DOP (Dilution of Precision, em inglês).

Esta função é capaz de analisar qualquer dos seguintes formatos de sentença GPS: GSA

PARÂMETRO 1: newdop - uma estrutura GPSDop a ser preenchida. PARÂMETRO 2: sentence - uma string contendo uma linha de dados GPS no formato NMEA-0183.

VALOR DE RETORNO: 0 - sucesso

-1 - erro de análise

-2 - sentença inválida

llh_2_ned <ned_frame.lib>

SINTAXE: int llh_2_ned(NEDFrame *vel, GPSPositon *pos, LLH *p_ant);

DESCRIÇÃO: função para converter coordenadas NED (North, East and Down, em inglês) a partir de coordenadas LLH (Latitude, Longitude and Height, em inglês).

PARÂMETRO 1: vel - estrutura NEDFrame a ser preenchida.

PARÂMETRO 2: pos - estrutura GPSPosition contendo a posição atual do sistema, em coordenadas LLH.

PARÂMETRO 3: p_ant - estrutura contendo a posição anterior do sistema, em LLH.

set_position8 <servos_pololu.lib>

SINTAXE: void set_position8(int motor_number, int position);

DESCRIÇÃO: analisa uma sentença para extrair dados DOP (Dilution of Precision, em inglês).

Esta função é capaz de analisar qualquer dos seguintes formatos de sentença GPS: GSA

PARÂMETRO 1: newdop - uma estrutura GPSDop a ser preenchida PARÂMETRO 2: sentence - uma string contendo uma linha de dados GPS no formato NMEA-0183.

VALOR DE RETORNO: 0 - sucesso

-1 - erro de análise

-2 - sentença inválida

eye_m <libmat.lib>

SINTAXE: void eye_m(double value, double *p, int nr, int nc);

DESCRIÇÃO: preenche a diagonal de uma matriz de tamanho nr x nc com um valor fornecido,

enquanto as outras posições assumem o valor zero.

PARÂMETRO 1: value - valor do tipo double que será alocado na diagonal da matriz.

PARÂMETRO 2: p - matriz do tipo double cuja diagonal será preenchida.

PARÂMETRO 3: nr - número de linhas da matriz p.

PARÂMETRO 4: nc - número de colunas da matriz p.

equal_m <libmat.lib>

SINTAXE: int equal_m(double *pA, int nrA, int ncA, double *pB, int nrB, int ncB);

DESCRIÇÃO: operação de atribuição matricial.

PARÂMETRO 1: pA - matriz cujos valores serão recebidos.

PARÂMETRO 2: nrA - número de linhas da matriz pA.

PARÂMETRO 3: ncA - número de colunas da matriz pA.

PARÂMETRO 4: pB - matriz cujos valores serão fornecidos.

PARÂMETRO 5: nrB - número de linhas da matriz pB.

PARÂMETRO 6: ncB - número de colunas da matriz pB.

sum_m <libmat.lib>

SINTAXE: int sum_m(double *pA, int nrA, int ncA, double *pB, int nrB, int ncB, double *pR, int nrresult, int ncresult);

DESCRIÇÃO: operação de adição matricial.

PARÂMETRO 1: pA - matriz que será parcela da soma.

PARÂMETRO 2: nrA - número de linhas da matriz pA.

PARÂMETRO 3: ncA - número de colunas da matriz pA.

PARÂMETRO 4: pB - matriz que será parcela da soma.

PARÂMETRO 5: nrB - número de linhas da matriz pB.

PARÂMETRO 6: ncB - número de colunas da matriz pB.

PARÂMETRO 7: pR - matriz que receberá o resultado da soma.

PARÂMETRO 8: nrresult - número de linhas da matriz pR.

PARÂMETRO 9: ncresult - número de colunas da matriz pR.

less_m <libmat.lib>

SINTAXE: `int less_m(double *pA, int nrA, int ncA, double *pB, int nrB, int ncB, double *pR, int nrresult, int ncresult);`

DESCRIÇÃO: operação de subtração matricial.

PARÂMETRO 1: pA - matriz que será minuendo.

PARÂMETRO 2: nrA - número de linhas da matriz pA.

PARÂMETRO 3: ncA - número de colunas da matriz pA.

PARÂMETRO 4: pB - matriz que será subtraendo.

PARÂMETRO 5: nrB - número de linhas da matriz pB.

PARÂMETRO 6: ncB - número de colunas da matriz pB.

PARÂMETRO 7: pR - matriz que receberá o resultado da diferença.

PARÂMETRO 8: nrresult - número de linhas da matriz pR.

PARÂMETRO 9: ncresult - número de colunas da matriz pR.

`times_m <libmat.lib>`

SINTAXE: `int times_m(double *pA, int nrA, int ncA, double *pB, int nrB, int ncB, double *pR, int nrresult, int ncresult);`

DESCRIÇÃO: operação de produto matricial.

PARÂMETRO 1: pA - matriz que será multiplicando.

PARÂMETRO 2: nrA - número de linhas da matriz pA.

PARÂMETRO 3: ncA - número de colunas da matriz pA.

PARÂMETRO 4: pB - matriz que será multiplicador.

PARÂMETRO 5: nrB - número de linhas da matriz pB.

PARÂMETRO 6: ncB - número de colunas da matriz pB.

PARÂMETRO 7: pR - matriz que receberá o resultado do produto.

PARÂMETRO 8: nrresult - número de linhas da matriz pR.

PARÂMETRO 9: ncresult - número de colunas da matriz pR.

`timesc_m <libmat.lib>`

SINTAXE: `int timesc_m(double value, double *pA, int nrA, int ncA, double *pR, int nrresult, int ncresult);`

DESCRIÇÃO: operação de produto entre uma constante e uma matriz.

PARÂMETRO 1: value - constante do tipo double que multiplicará matriz.

PARÂMETRO 2: pA - matriz que será multiplicada.

PARÂMETRO 3: nrA - número de linhas da matriz pA.

PARÂMETRO 4: ncA - número de colunas da matriz pA.

PARÂMETRO 5: pR - matriz que receberá o resultado do produto.

PARÂMETRO 6: nrresult - número de linhas da matriz pR.

PARÂMETRO 7: ncresult - número de colunas da matriz pR.

inv_m <libmat.lib>

SINTAXE: short int inv_m(double *pA, int nrA, int ncA, double *pR, int nrresult, int ncresult);

DESCRIÇÃO: operação de inversão de uma matriz.

PARÂMETRO 1: pA - matriz que será invertida.

PARÂMETRO 2: nrA - número de linhas da matriz pA.

PARÂMETRO 3: ncA - número de colunas da matriz pA.

PARÂMETRO 4: pR - matriz que receberá o resultado da inversão.

PARÂMETRO 5: nrresult - número de linhas da matriz pR.

PARÂMETRO 6: ncresult - número de colunas da matriz pR.

euler2quat_m <libmat.lib>

SINTAXE: void euler2quat_m(_eangles *eangles, double *q);

DESCRIÇÃO: operação de transformação de ângulos de Euler em quatérnios.

PARÂMETRO 1: eangles - vetor contendo os ângulos de Euler.

PARÂMETRO 2: q - vetor que receberá os quatérnios.

subtr_ang <libmat.lib>

SINTAXE: double subtr_ang(double ang1, double ang2);

DESCRIÇÃO: operação de subtração de ângulos.

PARÂMETRO 1: ang1 - variável do tipo double que será minuendo e receberá o resultado da subtração.

PARÂMETRO 2: ang2 - variável do tipo double que será o subtraendo.

adj_IMU <IMU.lib>

SINTAXE: void adj_IMU(data_IMU *IMU);

DESCRIÇÃO: grava os pesos para calibração do IMU.

PARÂMETRO 1: IMU - estrutura do tipo data_IMU.

get_IMU <IMU.lib>

SINTAXE: int get_IMU (unsigned char *msg_IMU, data_IMU *IMU);

DESCRIÇÃO: obter dados do tipo IMU.

PARÂMETRO 1: msg_IMU - string para preencher a sentença IMU contendo dados IMU.

PARÂMETRO 2: IMU - estrutura do tipo data_IMU a ser preenchida.

VALOR DE RETORNO: 1 - sucesso

0 - falha

get_IMU <IMU.lib>

SINTAXE: int get_IMU (unsigned char *msg_IMU, data_IMU *IMU);

DESCRIÇÃO: obter dados do tipo IMU.

PARÂMETRO 1: msg_IMU - string para preencher a sentença IMU contendo dados IMU.

PARÂMETRO 2: IMU - estrutura do tipo data_IMU a ser preenchida.

VALOR DE RETORNO: 1 - sucesso

0 - falha

headingangle <IMU.lib>

SINTAXE: double headingangle(double *m);

DESCRIÇÃO: cálculo do ângulo de direção.

PARÂMETRO 1: m - medidas do magnetômetro (x, y, z).

VALOR DE RETORNO: angle - ângulo de direção.

uBinToInt <IMU.lib>

SINTAXE: unsigned int uBinToInt(unsigned char* p, int n, char LSB);

DESCRIÇÃO: transformação de binário não sinalizado para inteiro.

PARÂMETRO 1: p - string não sinalizado.

PARÂMETRO 2: n - número de caracteres.

PARÂMETRO 3: LSB - bit menos significativo.

VALOR DE RETORNO: value - número convertido.

GPS_IMU <IMU.lib>

SINTAXE: void GPS_IMU(double *a, double *q, GPSPosition *p1_ant, NEDFrame *ned_ant, GPSPosition *p1, NEDFrame *ned);

DESCRIÇÃO: fornece coordenadas GPS baseado em IMU.

PARÂMETRO 1: a - string não sinalizado.

PARÂMETRO 2: q - número de caracteres.

PARÂMETRO 3: p1_ant - estrutura GPSPosition contendo os dados obtidos a partir do GPS em um instante anterior.

PARÂMETRO 4: ned_ant - estrutura NEDFrame contendo dados de velocidades de um instante anterior.

PARÂMETRO 5: p1 - estrutura GPSPosition contendo os dados obtidos a partir do GPS no instante atual.

PARÂMETRO 6: ned - estrutura NEDFrame contendo dados de velocidades de um instante atual.

ekf_p_a <kalman_filter.lib>

SINTAXE: void ekf_p_a(_wm_a *wm, _ss_a *ss);

DESCRIÇÃO: etapa preditora do filtro de Kalman extendido para Atitude.

PARÂMETRO 1: wm - estrutura do tipo _wm_a contendo as matrizes de pesos.

PARÂMETRO 2: ss - estrutura do tipo _ss_a contendo as matrizes de espaço de estado.

ekf_u_a <kalman_filter.lib>

SINTAXE: void ekf_u_a(_wm_a *wm, _ss_a *ss);

DESCRIÇÃO: etapa de atualização do filtro de Kalman extendido para Atitude.

PARÂMETRO 1: wm - estrutura do tipo _wm_a contendo as matrizes de pesos.

PARÂMETRO 2: ss - estrutura do tipo `_ss_a` contendo as matrizes de espaço de estado.

`ekf_p_p` <kalman_filter.lib>

SINTAXE: `void ekf_p_p(_wm_p *wm, _ss_p *ss);`

DESCRIÇÃO: etapa preditora do filtro de Kalman estendido para Posição.

PARÂMETRO 1: wm - estrutura do tipo `_wm_a` contendo as matrizes de pesos.

PARÂMETRO 2: ss - estrutura do tipo `_ss_a` contendo as matrizes de espaço de estado.

`ekf_u_p` <kalman_filter.lib>

SINTAXE: `void ekf_u_a(_wm_a *wm, _ss_a *ss);`

DESCRIÇÃO: etapa de atualização do filtro de Kalman estendido para Posição.

PARÂMETRO 1: wm - estrutura do tipo `_wm_a` contendo as matrizes de pesos.

PARÂMETRO 2: ss - estrutura do tipo `_ss_a` contendo as matrizes de espaço de estado.

`erkfsayed_p` <kalman_filter.lib>

SINTAXE: `void erkfsayed_p(_wm_p *wm_p, _ss_p *ss_p, _wm_a *wm_a, _ss_a *ss_a, int option);`

DESCRIÇÃO: etapa de predição do filtro de Kalman robusto estendido.

PARÂMETRO 1: wm_p - estrutura do tipo `_wm_a` contendo as matrizes de pesos para Posição.

PARÂMETRO 2: ss_p - estrutura do tipo `_ss_a` contendo as matrizes de espaço de estado para Posição.

PARÂMETRO 3: wm_a - estrutura do tipo `_wm_a` contendo as matrizes de pesos para Atitude.

PARÂMETRO 4: ss_a - estrutura do tipo `_ss_a` contendo as matrizes de espaço de estado para Atitude.

PARÂMETRO 5: option - inteiro para selecionar qual bloco será submetido à filtragem.

`erkfsayed_u` <kalman_filter.lib>

SINTAXE: `void erkfsayed_u(_wm_p *wm_p, _ss_p *ss_p, _wm_a *wm_a, _ss_a *ss_a,`

int option);

DESCRIÇÃO: etapa de atualização do filtro de Kalman robusto estendido.

PARÂMETRO 1: `wm_p` - estrutura do tipo `_wm_a` contendo as matrizes de pesos para Posição.

PARÂMETRO 2: `ss_p` - estrutura do tipo `_ss_a` contendo as matrizes de espaço de estado para Posição.

PARÂMETRO 3: `wm_a` - estrutura do tipo `_wm_a` contendo as matrizes de pesos para Atitude.

PARÂMETRO 4: `ss_a` - estrutura do tipo `_ss_a` contendo as matrizes de espaço de estado para Atitude.

PARÂMETRO 5: `option` - inteiro para selecionar qual bloco será submetido à filtragem.

`updateOmega <matrices_a.lib>`

SINTAXE: `void updateOmega (double *pomega, int nro, int nco, double *pOmega, int nrO, int ncO);`

DESCRIÇÃO: atualiza a matriz Omega.

PARÂMETRO 1: `pomega` - vetor com medidas do giroscópio.

PARÂMETRO 2: `nro` - número de linhas de `pomega`.

PARÂMETRO 3: `nco` - número de colunas de `pomega`.

PARÂMETRO 4: `pOmega` - matriz de multiplicação.

PARÂMETRO 5: `nrO` - número de linhas de `pOmega`.

PARÂMETRO 6: `ncO` - número de colunas de `pOmega`.

`update_q <matrices_a.lib>`

SINTAXE: `void update_q (double *pomega, int nro, int nco, double *q, int nrq, int ncq);`

DESCRIÇÃO: atualização

PARÂMETRO 1: `pomega` - vetor com medidas do giroscópio.

PARÂMETRO 2: `nro` - número de linhas de `pomega`.

PARÂMETRO 3: `nco` - número de colunas de `pomega`.

PARÂMETRO 4: `q` - matriz `q`.

PARÂMETRO 5: `nrq` - número de linhas de `q`.

PARÂMETRO 6: ncq - número de colunas de q.

update_rot <matrices_a.lib>

SINTAXE: void update_rot(double *q, double *pRot, int nrRot, int ncRot);

DESCRIÇÃO: atualização da matriz de rotação.

PARÂMETRO 1: q - vetor de quatérnios.

PARÂMETRO 2: pRot - matriz de rotação.

PARÂMETRO 3: nrRot - número de linhas de pRot.

PARÂMETRO 4: ncRot - número de colunas de pRot.

update_h_a <matrices_a.lib>

SINTAXE: void update_h_a(double *q, double *z);

DESCRIÇÃO: atualização do erro de observação para atitude.

PARÂMETRO 1: q - vetor de quatérnios.

PARÂMETRO 2: z - saída da função, $z = h(x)$.

update_ss_a <matrices_a.lib>

SINTAXE: void update_ss_a(double *omega, _ss_a *ss_a);

DESCRIÇÃO: atualização das matrizes de estado da atitude.

PARÂMETRO 1: omega - vetor de medidas do giroscópio.

PARÂMETRO 2: ss_a - estrutura do tipo _ss_a contendo as matrizes de estado.

update_H_a <matrices_a.lib>

SINTAXE: void update_H_a(double *H, double *q);

DESCRIÇÃO: atualização da matriz de observação H da atitude com a estimativa de x.

PARÂMETRO 1: H - matriz de observação H.

PARÂMETRO 2: q - estimativa de x.

update_wm_a <matrices_a.lib>

SINTAXE: void update_wm_a(_wm_a *wm_a);

DESCRIÇÃO: atualização das matrizes de peso de atitude.

PARÂMETRO 1: `wm_a` - estrutura do tipo `_wm_a` contendo as matrizes de peso a serem atualizadas.

`update_h_p` <matrices_p.lib>

SINTAXE: `void update_h_p(double *xp, double *h);`

DESCRIÇÃO: atualização do erro de observação para posição.

PARÂMETRO 1: `xp` - estimativa de `x`.

PARÂMETRO 2: `h` - vetor de observação.

`update_ss_p` <matrices_p.lib>

SINTAXE: `void update_ss_p(_ss_p *ss_p, GPSPosition *pos, NEDFrame *v, double *q);`

DESCRIÇÃO: atualização das matrizes de estado para posição.

PARÂMETRO 1: `ss_p` - estrutura do tipo `_ss_p` contendo as matrizes de estado.

PARÂMETRO 2: `pos` - estrutura do tipo `GPSPosition` contendo os dados obtidos do GPS.

PARÂMETRO 3: `v` - estrutura do tipo `NEDFrame` contendo os valores de velocidade calculados a partir dos dados obtidos do GPS.

`update_wm_p` <matrices_a.lib>

SINTAXE: `void update_wm_p(_wm_p *wm_p);`

DESCRIÇÃO: atualização das matrizes de peso para posição.

PARÂMETRO 1: `wm_p` - estrutura do tipo `_wm_p` contendo as matrizes de peso a serem atualizadas.

O código apresentado nesta seção está funcionando de maneira adequada com filtro para estimativa a posição, velocidade e atitude e enviando comandos para a placa de servos.

ANEXO A

Equipamentos

A.1 Helimodelo TREX 450 XL

Para a composição do robô helicóptero foi adquirido o helimodelo TREX 450 XL, mostrado na Figura A.1, que será utilizado como estrutura para o robô helicóptero. Segue abaixo as especificações do helimodelo:

- Comprimento 650 mm
- Altura 230 mm
- Diâmetro do rotor principal 680/700 mm
- Diâmetro do rotor da calda 150 mm
- Peso sem o sistema elétrico 330/355g
- Peso com o sistema elétrico 620~670g
- Motor Brushless RCM-BL430L 3550KV
- Controlador de velocidade eletrônico de 25 A
- Bateria de Litio Polimero DC 11.1 V 1300 mAh

- Receptor Futaba PCM 1024
- Transmissor Futaba 7CAP/7CHP

(a)

(b)

Figura A.1: Helimodelo TREX 450 XL (a) e Trans. Futaba 7CAP/7CHP (b) (figuras retiradas do manual do fabricante).

Maiores detalhes do Helimodelo TREX 450 XL podem ser vistos no manual do fabricante. E detalhes da composição do robô helicóptero podem ser vistos na Seção 4.1.

A.2 RCM5400W

O módulo RabbitCore RCM5400W que será usado no piloto automático é mostrado na Figura A.2. O módulo RabbitCore RCM5400W usa funcionalidades Wi-Fi /802.11b/g dos microprocessadores Rabbit® 5000 que possibilitam o desenvolvimento de sistemas de embarcados de baixo custo, baixo consumo, com controle wireless embarcado e com soluções de comunicações. Entre suas características o Rabbit® 5000 inclui *hardware* DMA, velocidade clock superior a 100 MHz, conexões I/O compartilhadas com seis portas seriais e quatro níveis de funções de pinos alternados que incluem fase variável PWM, I/O auxiliares, decodificadores de quadratura, e captura de entrada. Possui códigos existentes que ajudam a reduzir o tamanho do código e melhoram a velocidade de processamento, um módulo de núcleo rápido, eficiente, o qual é a solução ideal para aplicações *wireless* embarcadas.

O kit de desenvolvimento do RCM5400W tem o essencial para projetar um sistema wireless baseado em microprocessador, e inclui o sistema de desenvolvimento de *software* *Dynamic C*. Esse kit de desenvolvimento contém uma placa protótipo que permite avaliar o módulo RCM5400W e criar o protótipo do circuito que faz a interface com o módulo RCM5400W. Isso permite escrever e testar softwares para esses módulos.

Figura A.2: RCM5400W (figura retirada do manual do fabricante).

O RCM5400W tem o microprocessador Rabbit 5000 que opera em 73.73 MHz, *RAM* estática, memórias *flash*, três *clocks* (oscilador principal, oscilador Wi-Fi, e *timekeeping*), e o circuito necessário para o *reset* e administração da bateria de *backup* do tempo real interno do Rabbit 5000 e da *RAM* estática. Possui 50 pinos que trazem o barramento de conexões do Rabbit 5000, portas paralelas e portas seriais.

O módulo RCM5400W recebe alimentação de 3.3 V das placas mães onde estão montados, e podem fazer interface com muitos dispositivos CMOS compatíveis através da placa mãe.

As principais características do módulo RCM5400W são

- Microprocessador Rabbit 5000 @ 73.73 MHz
- 512K de dados SRAM
- 512K de memória Flash
- 1M de memória serial Flash
- Conectividade Wi-Fi 802.11b/g padrão, ISM 2.4 GHz
- 35 portas digitais I/O configuráveis para quatro funções diferentes
- 6 portas seriais de alta velocidade, CMOS compatível
- Possui relógio de tempo real

- Supervisor *Watchdog*
- 10 temporizadores de 8-bit, 1 de 10-bit, e 1 de 16-bit
- 4 canais PWM
- Alimentação e I/O de 3.3 V
- Pequeno tamanho: $1.84 \times 2.85 \times 0.55$ (47 mm $\times$ 72 mm $\times$ 14 mm)

A série RCM5400W é programada sobre uma porta USB de um computador padrão através do cabo de programação que vem com o kit de desenvolvimento. Outros detalhes do RCM5400W podem ser vistos no manual do usuário do RabbitCore RCM5400W.

Maiores detalhes da integração do Rabbit 5400W no piloto automático podem ser vistos na Seção 4.1.

A.3 IMU 6DOF v4

A unidade medida inercial 6DOF v4 da Figura A.3 possui acelerômetros, giroscópios e magnetômetros nos três eixos. A frequência e cada canal independente pode ser selecionado pelo usuário. O dispositivo pode comunicar-se em ASCII ou em formato binário por *Bluetooth* ou em nível TTL.

Figura A.3: IMU 6DOF v4 (figura retirada do manual do fabricante)

A IMU 6-DOF v4 é composta pelos seguintes sensores:

1. Três acelerômetros Freescale MMA7260Q com sensibilidade configuráveis em 1.5g, 2g, 4g ou 6g
2. Dois giroscópios InvenSense IDG300 graus/segundos

3. Sensores magnéticos HMC1052L e HMC1051Z

As especificações elétricas da IMU 6-DOF v4 são:

- Tensão de entrada: 4.2 V para 7 V DC
- Tensão TX e RX de 3.3 V
- Frequência de resposta
 - Sensores magnéticos: 312 Hz
 - Giroscópios IDG300: 120 Hz
 - Acelerômetros MMA7260Q
 - * 350 Hz, eixos X e Y
 - * 150 Hz, eixo Z.

Maiores detalhes da IMU 6-DOF v4 podem ser vistos nas figuras A.4.a e A.4.b.

Figura A.4: Esquemático da placa controladora da IMU 6DOF v4 (a) e esquemático da placa de sensores da IMU 6DOF v4 (b) (esquemático retirado do manual do fabricante).

A Figura A.4.a mostra a placa de controladora da IMU 6DOF v4, cujos componentes identificados são

1. Indicador de alimentação LED
2. Chave de alimentação

3. Indicador de conexão LED
4. LED de estado de três cores
5. Conector de alimentação
6. Porta de alimentação para a placa do sensor, chaveada da placa do controlador
7. Conexões dos sensores magnéticos
8. Conexões dos acelerômetros
9. Conexões dos giroscópios
10. Porta de programação e conexão serial TTL.

E a Figura A.4.b mostra a placa de sensores da IMU 6DOF v4, cujos componentes identificados são

1. Porta de alimentação
2. Conexões dos sensores magnéticos, mais conexões *ser/reset*
3. Conexões dos sensores de aceleração, mais conexões *range set*
4. Conexões dos giroscópios
5. Porta do giroscópio montado na vertical
6. Indicador de alimentação LED.

A IMU 6-DOF v4 pode ser configurada via terminal. Para usar a IMU 6-DOF v4 de um terminal, basta configurar o terminal para 115200 *baud rate*, *hardware flow control*, 8 *data bits*, um *stop bit* e *no parity*. Iniciando o terminal com o *Bluetooth* ou com a serial, basta pressionar a barra de espaço e o menu de configuração aparecerá na tela, veja Figura A.5 do menu de configuração.

Pressionando “ 1 ” a lista de canais ativos aparece na tela, veja Figura A.6 da lista de canais ativos.

A composição dos dados das medidas ativas da IMU 6-DOF v4 é composta por um “ A ” (ASCII 65) no início e um “ Z ” (ASCII 90) no final. Cada canal é informado conforme a sequência abaixo

```

6DOF v4 setup, version 1.0

1) View/edit active channel
   list
2) Change output mode,
   currently binary
3) Set Auto run mode,
   currently off
4) Set accelerometer
   sensitivity, currently 1.5 g
5) Set output frequency,
   currently 100
6) Change output port,
   currently Bluetooth
9) Save settings and run unit

```

Figura A.5: Menu de configuração da IMU 6-DOF v4 (figura retirada do manual do fabricante).

```

1) Magneto X = on
2) Magneto Y = on
3) Magneto Z = on
4) Accel X = on
5) Accel Y = on
6) Accel Z = on
7) Pitch = on
8) Roll = on
9) Yaw = on

Press the number of the channel you wish to
change, or press x to exit.

```

Figura A.6: Lista de canais ativos da IMU 6-DOF v4 (figura retirada do manual do fabricante).

1. Contador
2. Magnetômetro x
3. Magnetômetro y
4. Magnetômetro z
5. Acelerômetro x
6. Acelerômetro y
7. Acelerômetro z
8. Giroscópio x
9. Giroscópio y
10. Giroscópio z

O contador é composto por dois *bytes* que vem como MSB-LSB, e irá variar de 0 para 32767. Se algum dos outros canais estiver selecionado como inativo, o dado é omitido da saída de dados e o dado subsequente sobe e é informado na sequência.

No modo binário, cada canal ativo é informado com 2 *bytes*: MSB e LSB, nesta sequência, e eles estarão sempre entre 0 e 1023 por causa do analógico digital de 10-*bit*. O tamanho da estrutura de dados será de 4 *bytes* (“A”, “Z” e o contador) mais dois *bytes* de cada medida ativa. Com todos os canais ativos a estrutura de dados terá o tamanho de 22 *bytes*.

Outros detalhes da IMU podem ser vistos no manual do fabricante. E a integração da IMU no piloto automático pode ser vista na Seção 4.1.

A.4 GPS

A Figura A.7 mostra o receptor GPS da *San Jose Navigation* que será usado no piloto automático. Segue abaixo sua especificação.

Figura A.7: Receptor GPS da *San Jose Navigation* 32 canais, 5 Hz com antena (figuras retiradas do manual do fabricante).

As definições dos pinos do GPS da Figura A.7 são as seguintes

1. Tensão de entrada 3.3~5 V DC $\pm$ 10%

2. Terra
3. Transmissor da porta serial 1, saída TTL, 0~2.8 V (deixar aberto se não for usado)
4. Receptor da porta serial 1, entrada TTL, 0~2.8 V (deixar aberto se não for usado)
5. Transmissor da porta serial 2, saída TTL, 0~2.8 V (deixar aberto se não for usado)
6. Reservado (não utilizar e deixar aberto)
7. Pulso de tempo, pode ser usado para ligar e desligar LED, saída TTL, 0~2.8 V
8. Tensão de entrada da bateria de *backup* 2~5 V DC $\pm$ 10%

As especificações do receptor GPS da *San Jose Navigation* são

- Taxa de atualização de 1 ~ 5 Hz
- Baud rate de 4800 a 115200 bps
- Suporte a DGPS, WAAS, EGNOS, MSAS
- Arquitetura do receptor de 32 canais paralelos
- -158dBm de sensibilidade
- 3.3 m de precisão de posição
- 2.6 m de precisão de posição com DGPS
- 0.1 Knot RMS de precisão de velocidade
- Alimentação de 3.3 a 5 V
- Potência de consumo de 59/42/33 mA
- 2 portas seriais NMEA
- Pino para bateria de backup

O protocolo usado pelo GPS da *San Jose Navigation* é o NMEA. O protocolo NMEA informa os dados no formato ASCII. Este é um formato padrão para aplicações com GPS. Deste módulo pode-se obter 7 mensagens do padrão NMEA que são

Protocolo NMEA	Descrição
GGA	Sistema de posicionamento global de dados fixo
GSA	GNSS DOP e Satélites ativos
GSV	GNSS Satélites à vista
RMC	Informação de navegação mínima recomendada
VTG	Curso sobre a terra e velocidade terrestre
GLL	Posição geográfica - Latitude/Longitude
ZDA	Tempo e Data

Essas mensagens são obtidas através da porta TX1 a uma taxa de 4800 bps (*default*). Quando mais de 4 mensagens são escolhidas, um *baud rate* maior que 4800 bps é necessário.

Maiores detalhes sobre o GPS e o protocolo NMEA podem ser vistos no manual do fabricante. E a integração do GPS no piloto automático do robô helicóptero pode ser vista na Seção 4.1.

A.5 Placa de controle de servos motores

O controlador de servo motor serial Pololu mostrado na Figura A.8 é uma solução compacta para controlar oito servos motores. A posição angular de cada servo motor pode ser controlada independentemente.

Figura A.8: *Layout* da placa do controlador de servos motores serial Pololu (figura retirada do manual do fabricante).

As especificações do controlador de servos motores serial Pololu são

- 8 portas de servos motores
- Resolução de 0.5 us (aproximadamente 0.05 graus)
- Faixa de 250-2750 us
- Alimentação lógica de 5-16 VDC
- Tensão dos dados 0-5 VDC
- Taxa do pulso de 50 Hz
- Serial baud rate de 1200-38400 (auto detectável)
- Corrente média de 5 mA

Para comunicar-se com o controlador de servos motores, envia-se uma sequência de 5 ou 6 *bytes*. O primeiro *byte* é um valor de sincronização que deve estar sempre em 0x80(128). O segundo é o número do dispositivo Pololu, o qual é 0x01 para o controlador de 8 servos. O *byte* 3 são comandos para o controlador dos servos, que são discutidos abaixo. O *byte* 4 é o servo motor para o qual o comando deve ser aplicado. Os *bytes* 5 e 6 são os valores de dados para o comando enviado.

Os comandos para o controlador de servos motores são os seguintes:

- Comando 1: *Set Speed*(1 *byte* de dado). Este comando possibilita configurar as velocidades dos servos motores. Se a velocidade é configurada em 0 (*default*), o pulso de saída instantaneamente mudará para a configuração de posição. Se a velocidade é diferente de zero, o pulso muda gradualmente da posição antiga para a nova. Com velocidade de 1, o tamanho de pulso muda de 50 microsegundos para 1 segundo; a velocidade máxima de 6.35 ms por segundos é alcançada com uma configuração de velocidade de 127.
- Comando 2: *Set Position*, 7-bit (1 *byte* de dado). Este comando é útil para comunicações velozes pois somente 5 *bytes* são enviados para configurar uma posição. Configurar a posição de um servo motor irá automaticamente ligar os servos motores.
- Comando 3: *Set Position*, 8-bit(2 *bytes* de dados). Este comando é como a versão de 7-bit, exceto que dois *bytes* de dados devem ser enviados para transferir os 8 *bits*. O *bit* 0 contém o *bit* mais significativo, e os *bits* baixos do dado 2 contém os setes *bits* menores. (O *bit* 7 nos *bytes* do dado devem ser 0). Configurando uma posição do servo motor irá automaticamente ligar os servos motores.

- Comando 4: *Set Position, Absolute* (2 bytes de dados). Este comando permite o controle direto da posição interna dos servos motores. Neutro, faixa e configurações de direções não afetam este comando. Contém 2 dados, o dado 2 contém os 7 bits menores e o dado 1 os bits maiores. A faixa de valores válidos está entre 500 a 5500. Configurando uma posição do servo motor irá automaticamente ligar os servos motores.
- Comando 5: *Set Neutral* (2 bytes de dados). A configuração do neutro aplica-se somente para os comandos de 7 e 8 bits. O valor de neutro configura o meio da faixa, e corresponde ao 7-bit de valor de posição 63.5 ou ao 8-bit de valor de posição 127.5. O valor de neutro é uma posição absoluta como a do comando 4, e configurando a posição do neutro moverá o servo motor para aquela posição. O valor *default* é 3000. Pode ser útil mudar o neutro se forem mudados os servos motores e se for necessário calibrar o sistema, ou para ajustar as ligações mecânicas para o tamanho certo.

Maiores detalhes sobre a placa de servos motores Pololu podem ser vistos no manual do fabricante. A integração da placa de servos motores no piloto automático do robô helicóptero pode ser vista na Seção 4.1.