

UNIVERSIDADE FEDERAL DE SÃO CARLOS– UFSCAR  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA– CCET  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA– DEE  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA– PPGEE

**Leandro José Evilásio Campos**

**Arquitetura Systolic Array Baseada em  
FPGA para o Filtro de Kalman  
Robusto Aplicado a um Sistema Aéreo**

São Carlos  
2023



**Leandro José Evilásio Campos**

**Arquitetura Systolic Array Baseada em  
FPGA para o Filtro de Kalman  
Robusto Aplicado a um Sistema Aéreo**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Sistemas Elétricos e Eletrônicos

Orientador: Roberto Santos Inoue

São Carlos

2023





**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Engenharia Elétrica

---

**Folha de Aprovação**

---

Defesa de Dissertação de Mestrado do candidato Leandro José Evilásio Campos, realizada em 28/02/2023.

**Comissão Julgadora:**

Prof. Dr. Roberto Santos Inoue (UFSCar)

Prof. Dr. Glauco Augusto de Paula Caurin (USP)

Prof. Dr. João Vitor de Carvalho Fontes (UFSCar)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia Elétrica.



*Este trabalho é dedicado aos curiosos perseverantes que compreendem verdadeiramente o sentido de viver como missão de solicitude científica e contribuição póstera, ainda que anônimos.*





---

# Agradecimentos

---

Agradeço imensamente ao professor Roberto Santos Inoue pela orientação, competência no trabalho, ensinamentos transmitidos e principalmente por ser fonte de inspiração pessoal e profissional ao longo deste estudo. Agradeço também ao professor Ricardo Menotti pelos ensinamentos técnicos e exemplos de vida nos momentos mais árduos que passei nessa longa jornada de mestrado.

Agradeço ao professor Marco Henrique Terra pelo apoio e intermédio à concessão de bolsa junto ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Agradeço a UFSCar e a USP por oferecerem toda estrutura necessária neste estudo.

Agradeço aos docentes do PPGEE da UFSCar pelos ensinamentos passados. Agradeço também a todos colegas e professores da UFSJ e UFMG, em especial ao professor Edgar Campos Furtado, que participaram da minha formação acadêmica.

Agradeço aos colegas de LARIS ( *Laboratory of Autonomous Robots and Intelligent Systems*), Robson, Gustavo, Eduardo, Paulo, João e Neto, e aos colegas do CRob (Centro de Robótica da USP), Kenny, João, Igor, Paulo, Guilherme, Raphael e Kaio, pelos bons momentos que passamos juntos no convívio diário.

Agradeço aos professores que compuseram a banca de avaliação deste trabalho, professor Glauco Augusto de Paula Caurin e professor João Vitor de Carvalho Fontes.

Agradeço aos meus pais, Trindade e Gizélia, que mesmo com a ausência de formação escolar completa, me apoiaram durante toda minha trajetória nos estudos. Aos meus irmãos, Lara, Luigi e Lizandra, que também sempre me apoiaram.

Agradeço à minha esposa, Brenda Gomes Campos, pelo amparo, amor, carinho, amizade, cumplicidade, lealdade, respeito, atenção, cuidado e companheirismo demonstrados todos os dias desde que a aconheci. Agradeço ao meu filho, Isaac Campos, por ser a razão para não ceder aos desafios enfrentados neste trabalho. Ao escrever essas palavras lembro do seu virtuoso sorriso, o qual não consigo descrever.

Finalmente, agradeço à Deus pela vida, por me capacitar intelectualmente e me fortalecer para concluir mais esta etapa da minha vida.



*“To simply dismiss the concept of God being unscientific is to violate the very  
objectivity of science itself.”  
(Wernher von Braun)*



---

# Resumo

---

Esta dissertação apresenta a proposta de implementação de um acelerador de *hardware*, com arquitetura paralela baseada em *systolic array*, para construção do sistema de navegação em um veículo aéreo. A estimativa de atitude, posição e velocidade do drone é estabelecida por um filtro de Kalman robusto, devido às possíveis incertezas nos parâmetros dos sensores. A estrutura é assistida por dados de uma unidade de medição inercial, informações de atitude e posição de uma solução SLAM visual monocular e referências de posição e velocidade obtidas via GPS. Além disso, os modos de operação do sistema de navegação são definidos de acordo com a disponibilidade e confiabilidade dos sensores. Assim, a flexibilização da operação do veículo aéreo em ambientes distintos dependerá da multiplicidade e qualidade das fontes de informação para estimar os estados do sistema de navegação aérea. A principal contribuição deste estudo é a possibilidade da construção de um sistema de estimativa de estados com resultados precisos e em tempo real, necessidades fundamentais em robótica aérea. Para isso, a solução do Filtro de Kalman Robusto é realizada em um FPGA. A estratégia para o cálculo é baseada na fatoração matricial utilizando rotação de Givens e, em sequência, empregando o algoritmo de substituição reversa. Assim, a análise das configurações em uma abordagem heterogênea de *software/hardware* pode fornecer possíveis melhorias no desempenho computacional do sistema de navegação aérea.

**Palavras-chave:** robótica aérea, rotação de Givens, computação paralela, estimativa robusta de estados.



---

# Abstract

---

This dissertation presents a proposed implementation of a *hardware* accelerator, with parallel architecture based on *systolic array*, for building the navigation system in an aerial vehicle. The estimate of the attitude, position, and speed of the drone is established by a robust Kalman filter due to possible uncertainties in the parameters of the sensors. The structure is assisted by data from an inertial measurement unit, attitude, and position information from a monocular visual SLAM solution, and position and velocity references obtained via GPS. Furthermore, the operating modes of the navigation system are defined according to the availability and reliability of the sensors. Thus, the flexibility of the air vehicle operation in different environments will depend on the multiplicity and quality of the information sources to estimate the states of the air navigation system. The main contribution of this study is the possibility of building a state estimation system with accurate and real-time results, which is a fundamental need in aerial robotics. For this, the Robust Kalman Filter solution is performed in an FPGA. The calculation strategy is based on matrix factorization using Givens rotation and, in sequence, using the reverse substitution algorithm. Thus, the analysis of configurations in a heterogeneous *software/hardware* approach can provide possible improvements in the computational performance of the air navigation system.

**Keywords:** aerial robotics, Givens rotation, parallel computing, robust state estimation.





---

# Lista de Figuras

---

Figura 1 – Sequência de leitura recomendada conforme organização do texto. . . .	30
Figura 2 – KF na estrutura preditora-corretora. Modificado de (NYSETH, 2017).	34
Figura 3 – Estimativas filtradas. . . . .	42
Figura 4 – Norma do erro das estimativas de estados. . . . .	43
Figura 5 – Estrutura do sistema de navegação aérea. . . . .	46
Figura 6 – Fluxo de dados do KF indireto para o sistema de navegação aérea. . .	48
Figura 7 – Configuração de eixos dos giroscópios. . . . .	49
Figura 8 – Configuração de eixos dos acelerômetros. . . . .	53
Figura 9 – (a) Sistemas de coordenadas dos sensores e sistemas de coordenadas local e global. Baseado em (INOUE et al., 2017). (b) Sistemas de coordenadas ECEF e ENU para a Terra. Baseado em (WANG; HUYNH; WILLIAMSON, 2013). . . . .	56
Figura 10 – Projeção da gravidade nos eixos do acelerômetro. . . . .	57
Figura 11 – Mapa da declinação do campo magnético da Terra (OCEAN; NOAA, 2020). . . . .	59
Figura 12 – Estrutura básica do FPGA. . . . .	72
Figura 13 – Placa de desenvolvimento Zybo Z7-20. . . . .	73
Figura 14 – <i>Systolic array</i> unidimensional. . . . .	74
Figura 15 – Estrutura <i>systolic array</i> para inversão da matrix $A$ . . . . .	75
Figura 16 – Elementos de processamento diagonal e interno, respectivamente. . . .	76
Figura 17 – Elemento de processamento de saída . . . . .	76
Figura 18 – Estrutura <i>systolic array</i> para triangularização da matriz $A$ . . . . .	77
Figura 19 – Estrutura de simulação do algoritmo do Quadro II. . . . .	82
Figura 20 – Fragmento do circuito desenvolvido no Simulink para o cálculo da solução de $Ax = b$ usando a estrutura de <i>systolic arrays</i> baseadas na rotação de Givens. . . . .	83
Figura 21 – Circuitos dos PEs construídos no Simulink. . . . .	84

Figura 22 – Diagrama do sensoriamento para voo de teste. . . . .	85
Figura 23 – UAV modelo DJI Matrice 600 Pro. . . . .	86
Figura 24 – IMU com um receptor GNSS integrado Xsens modelo MTI-G-710 acoplada no UAV. . . . .	86
Figura 25 – Arranjo dos frames do INS e magnetômetros. Nota: a origem do frame está localizada nos acelerômetros. . . . .	87
Figura 26 – Câmera estéreo Stereolabs modelo ZED Mini acoplada ao UAV. . . . .	87
Figura 27 – Kit desenvolvedor Jetson Xavier NX. . . . .	88
Figura 28 – Velocidade Angular. . . . .	91
Figura 29 – Aceleração Linear. . . . .	92
Figura 30 – Intensidade do Campo Magnético. . . . .	92
Figura 31 – Quantidade de ciclos da simulação para cada conjunto de sensores utilizados. . . . .	94
Figura 32 – Atitude de referência e estimada com filtro de Kalman padrão e robusto. . . . .	95
Figura 33 – Posição local de referência e estimada com filtro de Kalman robusto. . . . .	96
Figura 34 – Sequência para triangularização matricial de $A^{4 \times 3}$ . . . . .	112
Figura 35 – Sequência usada nas funções MATLAB para anulação dos elementos nas decomposições $QR$ , $QL$ , $RQ$ e $LQ$ . . . . .	132

---

# Lista de Tabelas

---

Tabela 1 – Algoritmo recursivo preditor de Kalman de único estágio. . . . .	35
Tabela 2 – Dimensão dos elementos que compõem a matriz do algoritmo do Quadro II. . . . .	41
Tabela 3 – Variáveis usadas na modelagem da atitude. . . . .	50
Tabela 4 – Variáveis usadas nas modelagens de velocidade e posição. . . . .	52
Tabela 5 – Componentes do campo magnético da Terra em São Carlos - SP. . . . .	59
Tabela 6 – Parâmetros do elipsoide que representa a Terra. . . . .	62
Tabela 7 – Recursos da Zybo z7-20. . . . .	73
Tabela 8 – Estados dos PEs diagonais da Figura 18. . . . .	78
Tabela 9 – Estados dos PEs internos da Figura 18. . . . .	78
Tabela 10 – Estados dos PEs de saída da Figura 18. . . . .	79
Tabela 11 – Parâmetros usados na simulação. . . . .	93
Tabela 12 – Erro absoluto médio dos estados de navegação. . . . .	95
Tabela 13 – Recursos do FPGA para PEs. . . . .	96
Tabela 14 – Rotação de Givens para decomposição $QR$ , $QL$ , $RQ$ e $LQ$ , respectivamente. . . . .	113



---

# Lista de Siglas e Abreviaturas

---

**ASIC** *Application Specific Integrated Circuit*

**CL** *Configurable Logic*

**CPU** *Central Processing Unit*

**DSP** *Digital Signal Processor*

**ECEF** *Earth-centered, Earth-fixed*

**ENU** *East-North-Up*

**EKF** *Extended Kalman Filter*

**FLOPs** *FLoating-point OPerations*

**GNSS** *Global Navigation Satellite System*

**GPS** *Global Positioning System*

**GPU** *Graphic Processing Unit*

**HDL** *Hardware Description Language*

**INS** *Inertial Navigation System*

**IP** *Intellectual Property*

**KF** *Kalman Filter*

**MAC** *Multiplier-ACcumulator*

**MEMS** *MicroElectroMechanical Systems*

**NED** *Nort-East-Down*

**NOAA** *National Oceanic and Atmospheric Administration*

**PE** *Processing Element*

**ROS** *Robot Operating System*

**SLAM** *Simultaneous Localization and Mapping*

**SoC** *System-on-Chip*

**SM** *Switch Matrix*

**UAVs** *Unmanned Aerial Vehicles*

**UKF** *Unscented Kalman Filter*

**VHDL** *Very high speed integrated circuits HDL*

**V-SLAM** *Visual SLAM*

---

# Sumário

---

<b>I</b>	<b>INTRODUÇÃO</b>	<b>23</b>
1	INTRODUÇÃO . . . . .	25
1.1	Motivação . . . . .	26
1.2	Objetivos . . . . .	28
1.3	Organização do Texto . . . . .	28
<b>II</b>	<b>ASPECTOS TEÓRICOS</b>	<b>31</b>
2	FILTRO DE KALMAN ROBUSTO . . . . .	33
2.1	Estimativa Robusta . . . . .	36
2.1.1	Incertezas no Modelo do Sistema . . . . .	37
2.1.2	Mínimos Quadrados Regularizados e Penalizados . . . . .	37
2.2	Exemplo Numérico . . . . .	42
3	NAVEGAÇÃO AÉREA . . . . .	45
3.1	Modelagem . . . . .	47
3.1.1	Giroscópio . . . . .	49
3.1.2	Modelo de Atitude . . . . .	50
3.1.3	Acelerômetro . . . . .	52
3.1.4	Modelos de Velocidade e Posição . . . . .	53
3.1.5	Modelos dos Biases . . . . .	55
3.2	Sistemas de Coordenadas . . . . .	56
3.3	Modelo de Medidas . . . . .	57
3.3.1	Acelerômetro e Magnetômetro para Atitude . . . . .	57
3.3.2	GPS para Velocidade e Posição . . . . .	62
3.3.3	Barômetro para Altitude . . . . .	62
3.3.4	SLAM para Atitude e Posição . . . . .	63

<b>3.4</b>	<b>Representação no Espaço de Estados</b> . . . . .	<b>63</b>
3.4.1	Equação de estados . . . . .	63
3.4.2	Equação de Medidas . . . . .	64
3.4.3	Discretização . . . . .	64
<b>3.5</b>	<b>Filtro de Kalman para Navegação</b> . . . . .	<b>64</b>
3.5.1	Propagação dos Estados . . . . .	65
3.5.2	Propagação da Covariância . . . . .	65
3.5.3	Predição das Medidas Sensoriais . . . . .	66
3.5.4	Atualização dos Estados . . . . .	66
3.5.5	Algoritmo . . . . .	67
<b>III</b>	<b>ASPECTOS METODOLÓGICOS</b>	<b>69</b>
<b>4</b>	<b>SYSTOLIC ARRAY</b> . . . . .	<b>71</b>
<b>4.1</b>	<b>FPGA</b> . . . . .	<b>72</b>
4.1.1	Dispositivos <i>System-on-Chip</i> . . . . .	73
<b>4.2</b>	<b>Systolic Array</b> . . . . .	<b>74</b>
4.2.1	Inversão Matricial . . . . .	74
<b>4.3</b>	<b>Exemplo Analítico</b> . . . . .	<b>77</b>
<b>5</b>	<b>SENSORIAMENTO, AQUISIÇÃO DE DADOS E SIMULAÇÃO</b> . . . . .	<b>81</b>
<b>5.1</b>	<b>Simulação</b> . . . . .	<b>81</b>
<b>5.2</b>	<b>Voo de Teste</b> . . . . .	<b>85</b>
5.2.1	Materiais . . . . .	86
5.2.2	Suporte ROS . . . . .	88
<b>IV</b>	<b>RESULTADOS E CONCLUSÃO</b>	<b>89</b>
<b>6</b>	<b>RESULTADOS</b> . . . . .	<b>91</b>
<b>7</b>	<b>CONCLUSÕES</b> . . . . .	<b>97</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>99</b>
	<b>APÊNDICES</b>	<b>107</b>
	<b>APÊNDICE A – ROTAÇÃO DE GIVENS</b> . . . . .	<b>109</b>
<b>A.1</b>	<b>Decomposição QR</b> . . . . .	<b>110</b>
A.1.1	Interpretação Geométrica . . . . .	111



A.1.2	Decomposições Similares . . . . .	112
A.2	Exemplo Analítico . . . . .	113
A.3	Rotação Rápida de Givens . . . . .	116
A.4	Leitura Adicional . . . . .	120
<b>APÊNDICE B – QUATÉRNIOS . . . . .</b>		<b>121</b>
B.1	Descrição de Orientação . . . . .	122
B.2	Álgebra dos Quatérnios . . . . .	122
B.2.1	Definições e Operações Aritméticas . . . . .	122
B.2.2	Representação Matricial . . . . .	125
B.2.3	Diferenciação no Tempo . . . . .	126
B.3	Operador Rotacional . . . . .	127
<b>ANEXOS</b>		<b>129</b>
<b>ANEXO A – REPOSITÓRIO GITHUB . . . . .</b>		<b>131</b>
A.1	Decomposições Matriciais . . . . .	131
A.2	Fast Givens Rotation . . . . .	132



# Parte I

## Introdução



---

# Capítulo 1

## Introdução

---

Popularmente utilizado para obter estimativas ótimas de estados não observáveis e pela fusão de dados sensoriais, o consolidado filtro de Kalman (KF, do inglês *Kalman Filter*) (KALMAN, 1960) e suas versões para sistemas não lineares, *Extended Kalman Filter* (EKF) (SMITH; SCHMIDT; MCGEE, 1962) e *Unscented Kalman Filter* (UKF) (JULIER; UHLMANN, 1997), são amplamente usados em sistemas de navegação aérea. O KF é um algoritmo ótimo que apresenta uma metodologia computacional recursiva para estimar valores de estado em um processo controlado a partir de medidas sensoriais, ao mesmo tempo que fornece uma estimativa de erro dos valores obtidos. O KF gera resultados mais próximos dos valores reais em comparação com medições isoladas, mesmo que os sinais contenham ruídos. De forma geral, no caso da navegação aérea, as variáveis de interesse são a atitude, a posição e a velocidade do veículo aéreo.

A obtenção dos estados via KF ocorre a partir do modelo dinâmico do sistema, representado em espaço de estados, com a adição de ruídos gaussianos. Porém, incertezas paramétricas no modelo podem fazer com que o KF gere resultados insatisfatórios. A permanência de fontes de incertezas, mesmo após a calibração dos sensores, viola a premissa central do KF, no qual o modelo do sistema é exato. Nesse sentido, a utilização de um método robusto para estimativa dos estados torna-se relevante ao invés do emprego de filtros ótimos convencionais. Uma versão robusta do KF foi proposta por (ISHIHARA et al., 2015), baseado-se no problema de mínimos quadrados regularizados (SAYED, 2001) e na função de penalidade (LUENBERGER; YE, 2015). Esse método realiza uma abordagem assumindo que existam incertezas em todas as matrizes de parâmetros do modelo. Essa é uma vantagem se comparado a outros KFs robustos cujas incertezas são modeladas apenas em algumas matrizes específicas de parâmetros de estado e saída.

Os modelos do sistema dinâmico usados no KF geralmente são provenientes de sensores

que podem, em determinadas circunstâncias, apresentar incertezas em seus parâmetros. Essas incertezas surgem de especificações de tolerância em parâmetros físicos, imprecisões na resolução do sinal e influências externas nas medições do sensor, principalmente temperatura ambiente.

Em sistemas de navegação aérea, diversas fontes de informação podem ser utilizadas para obtenção dos estados. A principal referência para sistemas de navegação usados em drones é derivada do *Inertial Navigation System* (INS). Esse dispositivo utiliza sensores de movimento (acelerômetros) e rotação (gisoscópios), juntamente com um computador que integra os dados fornecidos para calcular a posição, a orientação e a velocidade do veículo. É comum o uso de sensores auxiliares, como magnetômetros e barômetros para calcular os estados de navegação aérea. Dados oriundos de *Global Navigation Satellite System* (GNSS) são frequentemente empregados para obtenção da posição e velocidade de UAVs. Algumas técnicas de *Simultaneous Localization and Mapping* (SLAM) também têm sido usadas na navegação em tempo real. Em suma, o problema SLAM consiste na estimativa da trajetória percorrida pelo veículo, apresentada como um conjunto de poses, enquanto constrói simultaneamente um mapa do ambiente. Particularmente, o *Visual SLAM* (V-SLAM), fundamenta-se na estimativa da pose da câmera e na reconstrução do modelo 3D do local percorrido durante a navegação, a partir de um fluxo de imagens.

A integração de dados do INS, informações do *Global Positioning System* (GPS)<sup>1</sup> e sistemas SLAM na construção de sistemas de navegação aérea é um tema abundante no campo da filtragem de sinais (GREWAL; ANDREWS, 2010). Diversos trabalhos, desde o advento do KF na década de 1960, utilizam o KF para o desenvolvimento de sistemas de navegação em *Unmanned Aerial Vehicles* (UAVs) (BARCZYK; LYNCH, 2012; MOURIKIS; ROUMELIOTIS, 2007; CRASSIDIS, 2006). Entretanto, a qualidade e disponibilidade dos dados sensoriais são obstáculos para a implementação do KF em sistemas embarcados. Um veículo que opera em ambientes com perturbações eletromagnéticas poderá sofrer distorções nas saídas do magnetômetro. As imagens capturadas pela câmera poderão ter mudanças devido à iluminação local. No caso do GPS, o acesso ao sinal poderá ser afetado caso o veículo esteja em local fechado ou sob interferência material. Neste trabalho, os dados sensoriais usados para corrigir os estados propagados no INS usam critérios de seleção para não afetar as estimativas de navegação do UAV.

## 1.1 Motivação

As incertezas presentes nos modelos dinâmicos que descrevem os sensores usados na construção do sistema de navegação é um fator limitante para o desempenho de algoritmos de estimativas dos estados. Alguns trabalhos supracitados apresentam soluções

---

<sup>1</sup> O GPS (desenvolvida e controlada pela Estados Unidos) é o GNSS mais utilizado no mundo. Outros GNSSs comuns são GLONASS (Rússia), Galileo (Europa) e Beidou (China).

interessantes para este problema. Contudo, como observado no desenvolvimento desses algoritmos de filtragem robusta, o custo computacional poderá restringir a implementação em um projeto embarcado. A atualização dos estados pelo filtro proposto por (ISHIHARA et al., 2015) realiza o cálculo de inversão de uma matriz de grandes dimensões, exigindo grande esforço computacional. A solução desenvolvida por (YENGERA et al., 2018; NARASIMHAPPA et al., 2017) converte os cálculos em um sistema linear do tipo  $Ax = b$ , em que  $A$  e  $b$  contêm parâmetros associados ao algoritmo robusto e valores preditos no ciclo anterior, e  $x$  apresenta os resultados das estimativas na forma preditora e filtrada simultaneamente. Para solucionar o sistema linear, os autores utilizaram a decomposição  $QR$  por rotação de Givens (GIVENS, 1958). Após esse procedimento, apenas as variáveis de interesse, isto é, o vetor de estados e a matriz de covariância do sistema, são encontradas usando substituição regressiva.

Aceleradores de *hardware* têm sido extensivamente empregados para melhoria de desempenho em tarefas que exigem recursos computacionais excessivos, como processamento gráfico, execução de redes neurais e cálculos específicos, como a fatoração de matrizes (VÁZQUEZ-CASTILLO et al., 2020; MERCHANT et al., 2018). Embora soluções baseadas em *software* apresentem desenvolvimento mais rápido, simplicidade na correção de erros e menor custo de investimento, as estruturas computacionais fundamentadas em *hardware* possuem menor latência operacional e maior largura de banda (CARDOSO; COUTINHO; DINIZ, 2017; SOH; WU, 2016). Assim, determinadas tarefas de um processo podem ser transferidas para componentes de *hardware* especializados, com potencial de atingir maior eficiência comparadas à execução realizada apenas pela *Central Processing Unit* (CPU). Tecnologias emergentes como as *Graphic Processing Unit* (GPU)s de propósito geral, permitem o desenvolvimento de aceleradores de hardware para diferentes utilidades (ABDUL-KADHIM; LIEN; YEE, 2019; SUDA et al., 2016), pois contêm a flexibilidade de interação heterogênea *software/hardware* utilizando o mesmo padrão aberto de programação de alto desempenho, como a OpenCL (*Open Computing Language*). No caso dos FPGAs, a configuração é geralmente especificada usando alguma *Hardware Description Language* (HDL), sendo o Verilog e o *Very high speed integrated circuits HDL* (VHDL) as linguagens mais utilizadas.

Atualmente, aceleradores baseados em FPGAs adquiriram popularidade devido a sua maior eficiência energética comparada às GPUs e maior flexibilidade de configuração em relação aos *Application Specific Integrated Circuit* (ASIC)s, principalmente para aplicações relacionadas à inteligência artificial, como por exemplo em *Deep Learning* (SHAWAHNA; SAIT; EL-MALEH, 2018). Além disso, a tecnologia FPGA pode incorporar componentes de *hardware* paralelos e vários núcleos *Intellectual Property* (IP), devido ao grande número de malhas lógicas programáveis disponíveis no chip. Essa estrutura permite a elaboração de uma rede de unidades de processamento que estejam conectadas de forma regular. Um modo de operação interessante para implementação de estruturas computacionais parale-

las consiste no princípio de *systolic array*. Neste processo, cada unidade de processamento calcula independentemente um resultado parcial como uma função dos dados recebidos de unidades vizinhas, armazena o resultado e o transmite para outras células adjacentes. O uso de *systolic arrays* permite, por exemplo, a implementação paralela do algoritmo de decomposição  $QR$  via rotação de Givens para o propósito de solução de sistemas lineares (GENTLEMAN; KUNG, 1982).

A concepção de projetos mistos *software/hardware* apresenta as vantagens de ambas abordagens de implementação computacional. Nesse aspecto, o propósito da arquitetura do sistema de navegação aérea visa obter a versatilidade e simplicidade da programação em CPU com a eficiência e latência das estruturas FPGA.

## 1.2 Objetivos

A partir das considerações apresentadas anteriormente, o objetivo principal desta dissertação é acelerar o processo de estimativa de estados do sistema de navegação aérea baseado em um KF robusto. Para isso, uma estrutura de *systolic arrays* implementada em FPGA é utilizada. Ao invés do cálculo convencional de inversão matricial para obtenção dos estados e matriz de covariância do filtro, a aceleração do processo ocorrerá pela fatoração da matriz usando rotação de Givens. O resultado é usado na atualização do KF robusto.

Para alcançar o objetivo principal, os seguintes objetivos específicos foram definidos:

- ❑ Revisar os conceitos relacionados à filtragem robusta de sinais, em particular o filtro de Kalman robusto.
- ❑ Explorar as estratégias de implementação em *hardware* e interação com a CPU.
- ❑ Realizar simulações a fim de identificar possibilidades e limitações do sistema de navegação aérea.
- ❑ Verificar o desempenho do método de rotação de Givens com *systolic array* usado na solução robusta de estimativa de atitude, posição e velocidade.

## 1.3 Organização do Texto

A estrutura do texto está dividida em três partes. Na primeira parte, os Capítulos 2 e 3 abrangem os principais aspectos teóricos utilizados para desenvolvimento desta dissertação. A segunda parte, constituída pelos Capítulos 4 e 5, estabelece as metodologias de pesquisa e recursos utilizados para simulação e implementação do sistema de navegação aérea em *hardware*. A terceira parte do texto, composta pelo Capítulo 6 e 7, apresenta



os resultados, discussões e próximos passos da pesquisa. Os Apêndices A e B fornecem suporte e complementam o conteúdo da primeira parte desta dissertação. Por fim, o Anexo A traz os documentos suplementares desenvolvidos neste estudo, associado principalmente à Parte 2 deste documento. A seguir, os conteúdos dos capítulos deste texto são brevemente apresentados.

**Capítulo 2:** Neste capítulo é demonstrado um método de estimativa robusta baseada na solução de mínimos quadrados regularizados e penalizados. O resultado é apresentado em um algoritmo recursivo preditor-corretor de estimativas robustas. Para validação do algoritmo, foram desenvolvidas simulações numéricas.

**Capítulo 3:** A estrutura do sistema de navegação aérea é elaborada neste capítulo. As estimativas de posição, velocidade e atitude são obtidas a partir dos dados fornecidos por sensores inerciais, magnetômetro, GPS, barômetro e informações visuais. Os modelos de propagação dos estados e medidas dos sensores são desenvolvidas detalhadamente para posteriormente serem aplicadas à representação em espaço de estados usada no KF.

**Capítulo 5:** Neste capítulo são mostrados os meios de aquisição dos sinais sensoriais e avaliação da confiabilidade desses dados. A instrumentação para coleta dos dados sensoriais é feita utilizando o *Robot Operating System* (ROS). Voos de teste em um drone comercial, com uma trajetória pré-estabelecida, forneceram as informações para verificação dos métodos de estimativas robustas utilizados neste estudo.

**Capítulo 4:** Como principal contribuição desta pesquisa, a aceleração das estimativas dos estados de navegação é desenvolvida usando um FPGA. Neste capítulo são apresentadas as estratégias para implementação de *systolic arrays* e computação heterogênea *software/hardware*. Também é apresentado o desenvolvimento de um exemplo analítico para solução do sistema linear utilizando o conceito de *systolic arrays* baseado em rotação de Givens.

**Capítulo 6:** Os resultados práticos e simulações realizadas neste estudo são apresentados neste capítulo. Ainda são mostrados análises e discussões teóricas acerca dos resultados com base em estudos anteriores.

**Capítulo 7:** O último capítulo desta dissertação mostra a conclusão da pesquisa com análise técnica dos resultados obtidos e principais contribuições alcançadas. Também são indicadas as recomendações para continuidade da pesquisa em trabalhos futuros.

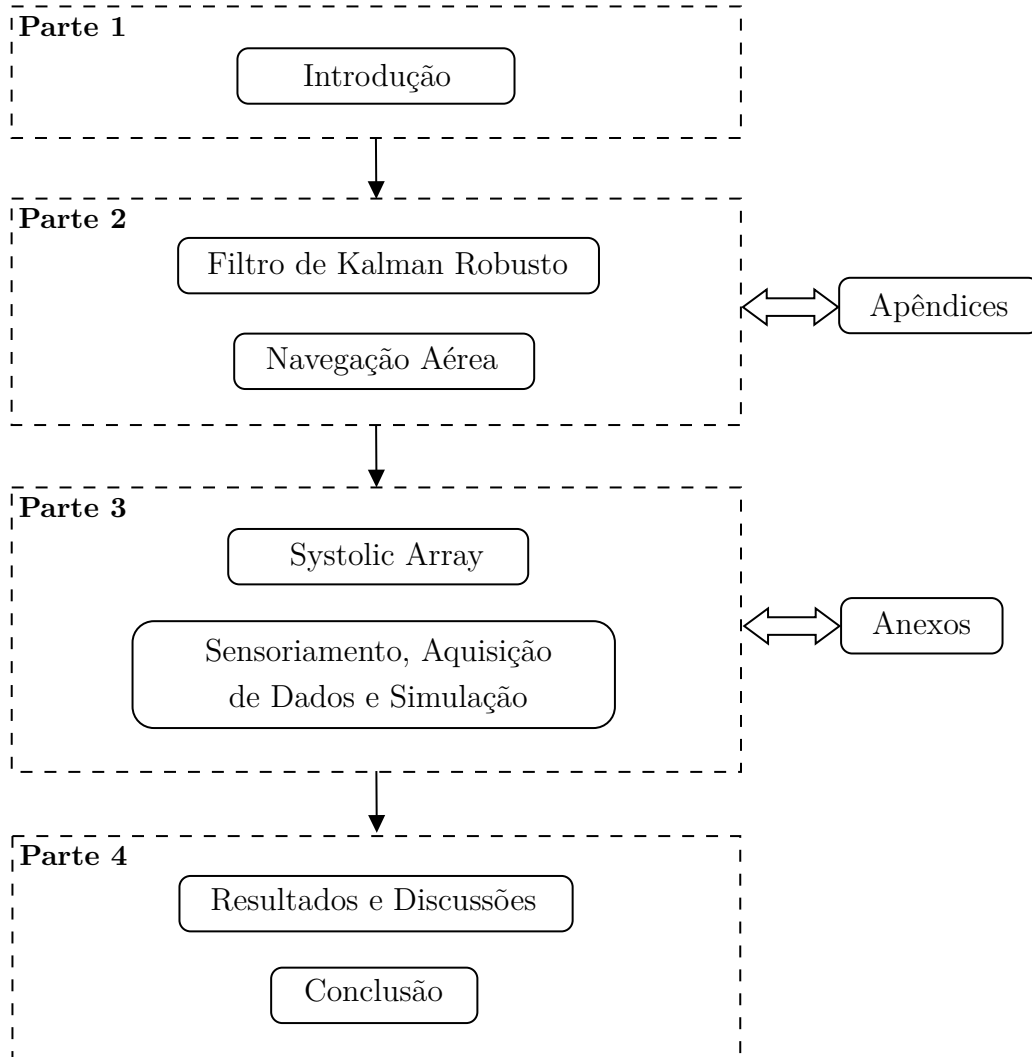
**Apêndice A:** Neste apêndice é desenvolvido detalhadamente o método de decomposição  $QR$  por rotação de Givens. A interpretação geométrica das decomposições correlatas,  $QL$ ,  $RQ$  e  $LQ$ , são discutidas com exemplos analíticos e simulações numéricas.

**Apêndice B:** Neste apêndice são exploradas as principais características da álgebra quaterniônica que servem de base para a modelagem do sistema de atitude usado no KF indireto.

**Anexo A:** Todo conteúdo de códigos numéricos e simulações desenvolvidos neste estudo

ficará disponível na plataforma de hospedagem GitHub. A descrição dos programas e simulações, assim como a organização das pastas e versões, são apresentadas neste anexo.

Figura 1 – Sequência de leitura recomendada conforme organização do texto.



# Parte II

## Aspectos Teóricos



---

## Capítulo 2

# Filtro de Kalman Robusto

---

Dispositivos eletrônicos atuais utilizam uma ampla variedade de fonte de dados provenientes de sensores, câmeras e demais recursos usados para coletar informações do meio externo. No entanto, é notório que os sinais obtidos quase sempre carregam consigo informações indesejadas, muitas vezes na forma de ruídos. Para isso, é bastante comum a aplicação de procedimentos computacionais ou o uso de circuitos eletrônicos, com o objetivo de atenuar a influência de ruídos e incertezas nos sinais de interesse. Em específico, alguns algoritmos utilizam dados estatísticos e o modelo matemático do sistema para obtenção da estimativa do sinal.

O *Kalman Filter* (KF) é um algoritmo recursivo usado para extrair as informações necessárias de um sinal contaminado com ruídos. O KF fornece estimativas de valores a partir da propagação dos estados, com base em uma equação dinâmica, e medidas observadas ao longo do tempo.

O KF é fundamentado em um sistema dinâmico descrito na forma de espaço de estados. Para um sistema dinâmico linear invariante no tempo, o modelo discreto é representado pela equação de estados e equação de saídas, nesta ordem, como

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathcal{F}\mathbf{x}_k + \mathcal{G}\mathbf{u}_k + \mathcal{H}\mathbf{w}_k \\ \mathbf{z}_k &= \mathcal{C}\mathbf{x}_k + \mathcal{D}\mathbf{v}_k\end{aligned}\quad k \geq 0, \quad (2.1)$$

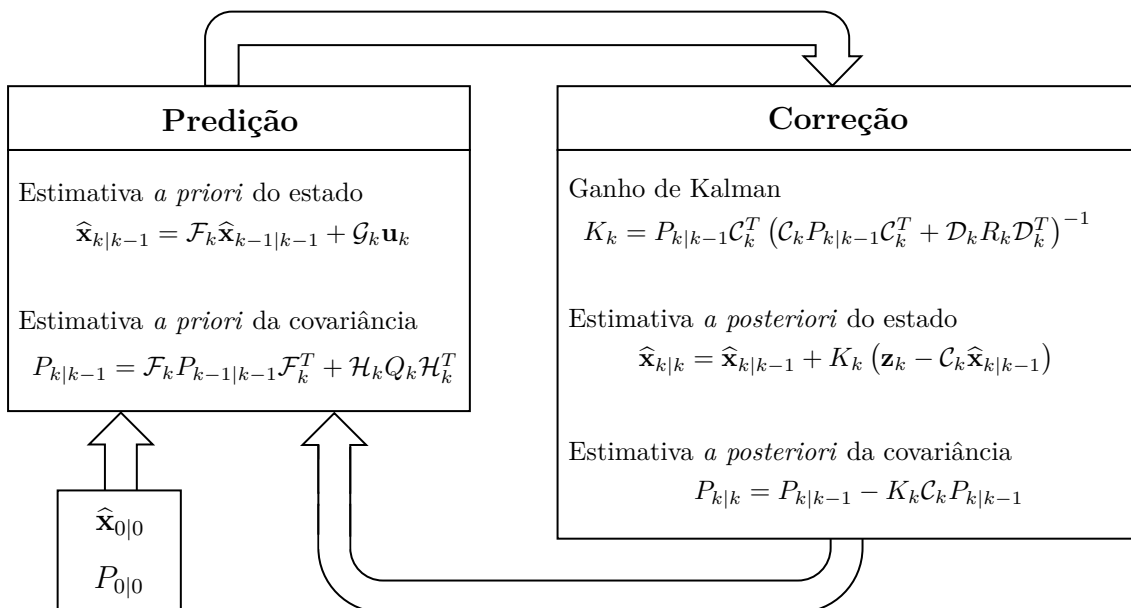
em que  $\mathbf{x}_k \in \mathbb{R}^n$  é o vetor de estado,  $\mathbf{u}_k \in \mathbb{R}^m$  é a entrada de controle do sistema e  $\mathbf{z}_k \in \mathbb{R}^p$  é o vetor de medidas. Os valores  $\mathbf{w} \in \mathbb{R}^{t_1}$  e  $\mathbf{v} \in \mathbb{R}^{t_2}$  são ruídos Gaussianos de processo e medição, respectivamente, que possuem média zero e satisfazem  $Q_k = \mathbb{E}\{\mathbf{w}_k \mathbf{w}_k^T\}$ ,  $R_k = \mathbb{E}\{\mathbf{v}_k \mathbf{v}_k^T\}$  e  $\mathbb{E}\{\mathbf{w}_k \mathbf{v}_k^T\} = 0$ . A solução da primeira equação de (2.1) fornece uma estimativa do estado no instante  $k + 1$ . Depois, com a disponibilidade da medida  $\mathbf{z}_k$ , a estimativa do estado pode ser otimizada, seguindo a minimização de  $\mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T (\mathbf{x}_k - \hat{\mathbf{x}}_k)\}$ .

**Observação 1** No contexto de filtragem, é comum o uso de algumas notações:

- O símbolo  $\hat{\square}$  é atribuído à variável estimada;
- Equação de medição ou observação se refere à equação de saída de (2.1);
- $\hat{\mathbf{x}}_{k+1|k}$  (*estimativa preditora do estado*) - Representa a estimativa de  $\mathbf{x}$  no tempo  $k + 1$  a partir do conjunto de observações fornecidas até o instante  $k$ ;
- $\hat{\mathbf{x}}_{k|k}$  (*estimativa filtrada do estado*) - Representa a estimativa de  $\mathbf{x}$  no tempo  $k$  a partir do conjunto de observações fornecidas até o instante  $k$ ;
- $\hat{\mathbf{x}}_{k-1|k}$ : (*estimativa suavizada do estado*) - Representa a estimativa de  $\mathbf{x}$  no tempo  $k - 1$  a partir do conjunto de observações fornecidas até o instante  $k$ .

Na maioria dos casos, o KF é desenvolvido em duas etapas: predição e correção. Alguns autores, como (AGUIRRE, 2004), utilizam as nomenclaturas “propagação” e “assimilação” para expressar as duas etapas anteriormente citadas, nesta ordem. Na etapa de predição, o estado estimado no passo anterior é utilizado para obter uma previsão do valor a partir da equação de movimento do processo. Esse valor é chamado estimativa *a priori* do estado. Em seguida, na etapa de correção ou atualização, a medida observada é combinada com a estimativa *a priori* do estado calculada anteriormente, usando uma média ponderada. O resultado é chamado estimativa *a posteriori* do estado. Os pesos da média ponderada dependem da matriz de covariância do erro de estimativa dos estados e das matrizes de covariâncias das variáveis aleatórias associadas ao processo e à medida. A estimativa resultante é mais próxima do valor real da grandeza, em comparação com o valor obtido por medições isoladas. Por esse motivo, o KF é muito útil para combinação de informações sensoriais. A Figura 2 apresenta a estrutura do KF, com as equações estabelecidas para cada etapa do algoritmo.

Figura 2 – KF na estrutura preditora-corretora. Modificado de (NYSETH, 2017).



A estrutura preditor-corretora do KF mostrada na Figura 2 pode ser modificada para calcular as estimativas de estados em um único estágio. Partindo do cálculo do ganho de Kalman  $K_k$ , com vetor de estado inicial  $\hat{\mathbf{x}}_{0|-1} = \bar{\mathbf{x}}_0$  e sua matriz de covariância  $P_{0|-1} = \Pi_0$ , as atualizações de  $\hat{\mathbf{x}}$  e  $P$  são efetuadas diretamente por duas equações. Pela substituição das equações de estimativas *a posteriori* nas equações de estimativas *a priori* da Figura 2, o algoritmo calcula a melhor estimativa de  $\mathbf{x}_{k+1}$  dado o conjunto de observações  $\mathbf{z}$  obtidas no instante  $k$ . Dessa forma, tanto a etapa de predição quanto a de correção são realizadas de uma só vez. Como  $\hat{\mathbf{x}}$  é calculado usando medidas da iteração anterior, este algoritmo é conhecido como preditor de Kalman. A Tabela 1 apresenta o algoritmo recursivo de estimativa preditor de Kalman.

Tabela 1 – Algoritmo recursivo preditor de Kalman de único estágio.

<p><b>Condições Iniciais:</b> <math>\hat{\mathbf{x}}_{0 -1} = \bar{\mathbf{x}}_0</math> e <math>P_{0 -1} = \Pi_0</math>.</p> <p><b>Passo 1:</b> Calcule <math>K_k</math> a partir de <math>P_{k k-1}</math> como segue:</p> $K_k = P_{k k-1} \mathcal{C}_k^T \left( \mathcal{C}_k P_{k k-1} \mathcal{C}_k^T + \mathcal{D}_k R_k \mathcal{D}_k^T \right)^{-1}$ <p><b>Passo 2:</b> Dado <math>\mathbf{z}_k</math>, atualize <math>\hat{\mathbf{x}}_{k+1 k}</math> e <math>P_{k+1 k}</math> como:</p> $\hat{\mathbf{x}}_{k+1 k} = \mathcal{F}_k \hat{\mathbf{x}}_{k k-1} + \mathcal{G}_k \mathbf{u}_k + \mathcal{F}_k K_k \left( \mathbf{z}_k - \mathcal{C}_k \hat{\mathbf{x}}_{k k-1} \right)$ $P_{k+1 k} = \mathcal{F}_k P_{k k-1} \mathcal{F}_k^T - \mathcal{F}_k K_k \mathcal{C}_k P_{k k-1} \mathcal{F}_k^T + \mathcal{H}_k Q_k \mathcal{H}_k^T$
--

O formato apresentado na Tabela 1 para o cálculo de estimativas de valores é muito útil para verificar qual a tendência dos estados no instante  $k$ , dadas  $k - 1$  medidas. Além do KF padrão que calcula as estimativas filtradas e do preditor de Kalman apresentado acima, é possível ainda calcular a estimativa suavizada que consiste na melhor estimativa de estados em  $k$ , com  $k + 1$  medidas fornecidas.

A equação de  $P_{k+1|k}$  apresentada na Tabela 1 é conhecida como Equação de Riccati discreta, em homenagem ao Conde Jacopo Francesco Riccati.<sup>1</sup> Essencialmente, a solução da Equação de Riccati é o tema central na teoria de filtragem e controle ótimo (ABOU-KANDIL et al., 2012).

A literatura referente ao KF é bastante extensa, com diversos trabalhos publicados desde os primeiros estudos relativos ao tema, na década de 1960. (MAYBECK, 1982) e (KAILATH; SAYED; HASSIBI, 2000) trazem extensas análises e conceitos sobre KF. Para um estágio acerca do KF, com estudos de caso e exemplos de código, (KIM, 2011) aborda o tema de forma prática e objetiva.

<sup>1</sup> matemático e jurista de Veneza com significativa contribuição nos estudos de hidráulica e teoria dos polinômios.

## 2.1 Estimativa Robusta

A estimativa ótima obtida via KF é alcançada sob a premissa de que o sistema é descrito por um modelo dinâmico exato. Entretanto, incertezas paramétricas contidas no modelo do sistema são comuns na prática. Atributos não modelados, parâmetros variantes no tempo ou imprecisões relacionadas ao processo de linearização podem gerar valores inexatos nas equações que descrevem o sistema dinâmico analisado. Caso algum parâmetro do modelo apresente incertezas, a eficiência do KF pode ser prejudicada severamente. Por esse motivo, diversos estudos realizados nas últimas décadas concentraram no desenvolvimento de técnicas de filtragem robusta para modelos com incertezas em seus parâmetros. As principais linhas de pesquisa sobre estimadores robustos envolvem abordagens com filtros  $\mathcal{H}_\infty$  (SIMON, 2006; PETERSEN; SAVKIN, 1999; NAGPAL; KHARGONEKAR, 1991), projetos de custo garantido (ZHAO; YANG, 2019; THEODOR; SHAKED, 1996; PETERSEN; MCFARLANE, 1994), e estratégias de mínimos quadrados regularizados robustos (ABOLHASANI; RAHMANI, 2018; SAYED; NASCIMENTO, 1999).

A abordagem com filtros  $\mathcal{H}_\infty$  tem por objetivo limitar o efeito das perturbações para o pior caso de influência, inclusive para incertezas no modelo do sistema, a partir na minimização da norma  $\mathcal{H}_\infty$ . Porém, esse estimador robusto não é recomendado para operações *on-line*, pois quando sua condição de existência falha, os resultados da filtragem podem divergir (SAYED, 2001).

Em projetos de filtragem de custo garantido, o estimador é projetado de forma que, para todas as incertezas possíveis, a variância do erro de estimativa possua um limite superior definido. A garantia da existência do estimador depende de alguns parâmetros de escala a cada iteração, o que pode prejudicar o desempenho do processo de filtragem.

Outra vertente para estimativa robusta é baseada na solução do problema de mínimos quadrados regularizados. Esses estimadores são projetados para minimizar a pior norma residual regularizada possível, sobre a classe de incertezas permissíveis no modelo (SAYED; NASCIMENTO, 1999). A vantagem desse estimador, comparado aos demais supracitados, está relacionada à sua operação recursiva. O filtro resultante realiza regularização de dados, o que torna sua aplicação *on-line* mais conveniente, pois as condições de existência do filtro são atendidas a cada iteração.

No trabalho desenvolvido por (ISHIHARA et al., 2015) foi proposto um filtro obtido pela combinação da técnica baseada na regularização e uma função de penalidade (LUENBERGER; YE et al., 1984). Para o cálculo das estimativas de estados, nenhum ajuste auxiliar nos parâmetros é necessário, apenas os valores conhecidos inicialmente. Ainda nesse estudo, o algoritmo de estimativa robusta foi sintetizado em uma estrutura unificada com matriz simétrica recursiva, o que permite que o cálculo, a cada iteração, seja realizado em uma única etapa. Contudo, essa operação envolve a inversão de uma matriz esparsa de grandes dimensões, o que pode ser um fator limitante para implementação embarcada do algoritmo devido ao custo computacional associado.



### 2.1.1 Incertezas no Modelo do Sistema

Para o modelo em espaço de estados de tempo discreto (2.1), as incertezas são adicionadas às matrizes de parâmetros como

$$\begin{aligned} \mathcal{F}_k &\triangleq F_k + \delta F_k; & \mathcal{G}_k &\triangleq G_k + \delta G_k; & \mathcal{H}_k &\triangleq H_k + \delta H_k; \\ \mathcal{C}_k &\triangleq C_k + \delta C_k; & \mathcal{D}_k &\triangleq D_k + \delta D_k. \end{aligned} \quad (2.2)$$

As matrizes  $F_k$ ,  $G_k$ ,  $H_k$ ,  $C_k$  e  $D_k$  são constituídas de parâmetros nominais do modelo linear discreto. As matrizes de incertezas  $\delta F$ ,  $\delta G$ ,  $\delta H$ ,  $\delta C$  e  $\delta D$  possuem dimensões compatíveis com as matrizes de parâmetros nominais e são modelados de acordo com a relação

$$\begin{aligned} [\delta F_k \quad \delta G_k \quad \delta H_k] &= \mathcal{M}_{1_k} \Delta_{1_k} [N_{F_k} \quad N_{G_k} \quad N_{H_k}] \\ [\delta C_k \quad \delta D_k] &= \mathcal{M}_{2_k} \Delta_{2_k} [N_{C_k} \quad N_{D_k}] \end{aligned} \quad \|\Delta_{(1,2)_k}\| \leq 1, \quad (2.3)$$

tal que  $\mathcal{M}_{1_k}$  e  $\mathcal{M}_{2_k}$  sejam matrizes não nulas,  $\Delta_{1_k}$  e  $\Delta_{2_k}$  fatores de contração arbitrários que podem assumir valores multidimensionais, e  $N_{(\cdot)_k}$  são matrizes conhecidas de dimensão compatível obtidas empiricamente para composição do modelo de incertezas. Em síntese, o modelo (2.3) surge de especificações de tolerância em parâmetros físicos.

Pela conjectura de disponibilidade da medida  $\mathbf{z}$  a todo instante  $k$ , a filtragem do sinal se restringe em obter a melhor estimativa do estado  $\hat{\mathbf{x}}_k$ , pois o estado  $\mathbf{x}_k$  não é totalmente observado. Além disso, o modelo do sistema está sujeito à incertezas paramétricas, o que afeta as estimativas de estados caso fosse utilizado um KF padrão. Em virtude dessas circunstâncias, o problema de estimativas robustas foi formulado a partir do critério de custo quadrático regularizado com um fator de penalidade, assim com os procedimentos apresentados por (ISHIHARA et al., 2015).

### 2.1.2 Mínimos Quadrados Regularizados e Penalizados

Sejam  $A \in \mathbb{R}^{n \times m}$  e  $b \in \mathbb{R}^n$  variáveis associadas a uma modelo matemático geral e  $x \in \mathbb{R}^m$  o vetor de incógnitas. O problema de mínimos quadrados regularizados propõe-se encontrar a melhor estimativa de  $x$  pela minimização de uma função custo. Essa minimização, geralmente é descrita por

$$\min_x \left\{ x^T \mathcal{Q}x + (Ax - b)^T \mathcal{W} (Ax - b) \right\}, \quad (2.4)$$

sendo  $x^T \mathcal{Q}x$  um termo de regularização e  $\mathcal{W}$  uma matriz de ponderação, com  $\mathcal{Q} > 0$  e  $\mathcal{W} \geq 0$ . Neste caso, a solução da estimativa  $\hat{x}$  em (2.4) é

$$\hat{x} = \left( \mathcal{Q} + A^T \mathcal{W} A \right)^{-1} A^T \mathcal{W} b. \quad (2.5)$$

**Observação 2** Para um vetor coluna  $v$  e uma matriz positiva definida  $M$ , geralmente são usadas as notações  $\|v\|^2$  e  $\|v\|_M^2$  para denotar a norma euclidiana e sua versão ponderada, as quais são algebricamente expressas por  $v^T v$  e  $v^T M v$ , respectivamente.

Entretanto, caso os parâmetros contidos em  $A$  e  $b$  apresentem incertezas, a solução (2.5) poderá ter o desempenho afetado. Para superar esse obstáculo, (SAYED; NASCIMENTO, 1999) desenvolveram uma versão robusta de mínimos quadrados regularizados, de forma que seja introduzido um vetor de perturbação na função custo como

$$\mathcal{J}(x, y) = x^T \mathcal{Q}x + \mathcal{R}(x, y), \quad (2.6)$$

na qual o custo residual  $\mathcal{R}(x, y)$  é definido por

$$\mathcal{R}(x, y) \triangleq (Ax - b + My)^T \mathcal{W} (Ax - b + My). \quad (2.7)$$

Em (2.7), a matriz  $\mathcal{M}$  restringe a incerteza  $y$  para certos espaços de alcance, tal como feito em (2.3). Neste caso, a solução para estimar  $\hat{x}$  é

$$\hat{x} = \min_x \max_{\|y\| \leq \phi(x)} \mathcal{J}(x, y), \quad (2.8)$$

de forma que  $\phi(x) = \|N_A x - N_b\|$  impõe um limite para  $y$ .

Para encontrar as estimativas de estados pela estrutura de modelo (2.1) com os parâmetros definidos em (2.2), é necessário realizar escolhas não triviais de  $\mathcal{Q}$ ,  $\mathcal{W}$  e  $\mathcal{M}$ , assim como para construção de  $\phi(x)$ . Aliás, como advertido por (SAYED, 2001), a solução deste problema requer alguns cuidados para evitar a introdução de múltiplos parâmetros de regularização. Para encontrar a estimativa  $\hat{x}$  de (2.8), o objetivo é solucionar a função custo  $\mathcal{J}(x, \delta)$  de (2.6), de modo que

$$\min_x \max_{\delta A, \delta b} \left\{ \|x\|_{\mathcal{Q}}^2 + \|(A + \delta A)x - (b + \delta b)\|_{\mathcal{W}}^2 \right\} \quad (2.9)$$

Em (2.9),  $\{\delta A\}$  denota a matriz de perturbação no espaço  $\mathbb{R}^{n \times m}$  referente à matriz nominal  $A$  e  $\{\delta b\}$  denota o vetor de perturbação no espaço  $\mathbb{R}^{n \times 1}$  referente ao vetor nominal  $b$ . Os elementos do conjunto  $\{\delta A, \delta b\}$  são definidos de forma que satisfazem

$$[\delta A \quad \delta b] = \mathcal{M} \Delta [N_A \quad N_b], \quad (2.10)$$

sendo que  $\Delta$  é uma contração arbitrária limitada, tal que  $\Delta \leq 1$ , e  $\{\mathcal{M}, N_A, N_b\}$  é o conjunto de grandezas conhecidas com dimensões apropriadas, assim como apresentado na relação (2.3).

O KF padrão apresentado na Figura 2 é derivado de um processo ótimo de minimização do erro quadrático do estado estimado. Assim, o problema da estimativa estocástica também pode ser desenvolvido em uma abordagem determinística baseada no problema de mínimos quadrados. Nesse sentido, os ruídos  $\mathbf{w}$  e  $\mathbf{v}$  do modelo (2.1) são tratados como erros de ajuste. Portanto, as variáveis aleatórias correspondentes aos ruídos de processo e medida devem ser minimizadas. Como os estados do sistema não são totalmente observados, a solução do problema consiste em obter a estimativa filtrada  $\hat{x}_k$  de  $x_k$  e a estimativa preditora  $\hat{x}_{k+1|k}$  de  $x_{k+1}$  a partir do estado estimado  $\hat{x}_{k|k-1}$ , das variáveis observáveis  $z_k$ , e das entradas  $u_k$ , em cada instante de tempo  $k$ .

No caso do modelo sujeito à incertezas em seus parâmetros, a função de custo deve ser minimizada sob a influência máxima das incertezas paramétricas. Desse modo, para o sistema dinâmico discreto com parâmetros incertos, o problema de mínimos quadrados regularizados apresenta a forma

$$\min_{\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{w}_k, \mathbf{v}_k} \max_{\delta_k} \|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\|_{P_{k|k-1}^{-1}}^2 + \|\mathbf{w}_k\|_{Q_k^{-1}}^2 + \|\mathbf{v}_k\|_{R_k^{-1}}^2, \quad (2.11)$$

sujeito à (2.1) e (2.3). Em (2.11),  $P_{k|k-1}^{-1} \succ 0$ ,  $Q_k^{-1} \succ 0$  e  $R_k^{-1} \succ 0$  são matrizes de ponderação associadas ao erro de estimativa  $x_k - \hat{x}_{k|k-1}$ , ao ruído de processo  $\mathbf{w}_k$  e ao ruído de medida  $\mathbf{v}_k$ , respectivamente.

**Observação 3** *Em modelos de filtros de processos estocásticos, as matrizes  $P$ ,  $Q$  e  $R$  representam as variâncias associadas ao erro de estimativa  $\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}$  e aos ruídos  $\mathbf{w}$  e  $\mathbf{v}$ , respectivamente. Entretanto, a composição determinística adotada em (2.11),  $P$ ,  $Q$  e  $R$  são componentes matriciais de ponderação.*

Para resolver o problema *min-max* restrito (2.11) é necessário transformá-lo em um problema irrestrito adequado. Dessa forma é introduzido um custo quadrático penalizado, como proposto por (ISHIHARA et al., 2015), no qual um parâmetro de penalidade  $\mu$  é incluído na função custo. Caso alguma restrição do problema *min-max* seja violada, a função será penalizada por  $\mu$ . Neste caso, a otimização resulta na melhor estimativa de  $\hat{\mathbf{x}}_k$  e  $\hat{\mathbf{x}}_{k+1|k}$  no cenário de influência máxima das incertezas paramétricas com a inclusão da penalidade na função custo, a partir da solução de

$$\min_{x_k} \max_{\delta_k} \{\mathcal{J}_k^\mu(x_k, \delta_k)\}, \quad (2.12)$$

sendo  $x_k \triangleq \{\mathbf{x}_k, \mathbf{w}_k, \mathbf{v}_k, \mathbf{x}_{k+1}\}$ ,  $\delta_k \triangleq \{\delta F_k, \delta B_k, \delta G_k, \delta C_k, \delta D_k\}$ , e função custo  $\mathcal{J}_k^\mu$  expressa por

$$\mathcal{J}_k^\mu \triangleq \|\mathbf{x}_k - \hat{\mathbf{x}}_{k+1}\|_{P_{k|k-1}^{-1}}^2 + \|\mathbf{w}_k\|_{Q_k^{-1}}^2 + \|\mathbf{v}_k\|_{R_k^{-1}}^2 + \mu \left( \|\mathbf{x}_{k+1} - \mathcal{F}_k \mathbf{x}_k - \mathcal{G}_k \mathbf{u}_k - \mathcal{H}_k \mathbf{w}_k\|^2 + \|\mathbf{z}_k - \mathcal{C}_k \mathbf{x}_k - \mathcal{D}_k \mathbf{v}_k\|^2 \right), \quad (2.13)$$

com  $P_{k|k-1}^{-1} \succ 0$ ,  $Q_k^{-1} \succ 0$ ,  $R_k^{-1} \succ 0$  e  $\mu > 0$ .

O problema (2.12-2.13) admite solução similar à (2.5), porém, com a introdução das incertezas paramétricas e penalidade  $\mu$ . A solução desse problema foi desenvolvido detalhadamente por (CERRI, 2013), de forma que as estimativas são organizadas em uma única operação matricial. (ROCHA; TERRA, 2021) também exploram a solução desse problema, porém a partir de expressões delineadas. Para  $\mu > 0$  fixo, a solução ótima para

(2.12) que resulta nas melhores estimativas de  $\hat{\mathbf{x}}_{k|k}$ ,  $\hat{\mathbf{x}}_{k+1|k}$ ,  $P_{k|k}$  e  $P_{k+1|k}$  é

$$\left[ \begin{array}{c} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{array} \right] \left[ \begin{array}{cc} P_{k|k}^\mu & * \\ * & P_{k+1|k}^\mu \end{array} \right]_\mu = \left[ \begin{array}{cc} 0 & 0 \\ 0 & 0 \\ \mathfrak{A} & 0 \\ 0 & I \end{array} \right]^T \left[ \begin{array}{cccc} \mathfrak{P}_{\mu,k} & 0 & I & 0 \\ 0 & \mathfrak{M}_{\mu,k} & \mathfrak{F}_k & -\mathfrak{L}_k \\ I & \mathfrak{F}_k^T & 0 & 0 \\ 0 & -\mathfrak{L}_k^T & 0 & 0 \end{array} \right]^{-1} \left[ \begin{array}{ccc} \mathfrak{X}_k & 0 & 0 \\ \mathfrak{G}_k & 0 & 0 \\ 0 & -\mathfrak{A}_k & 0 \\ 0 & 0 & -I \end{array} \right], \quad (2.14)$$

com os elementos matriciais

$$\mathfrak{P}_{\mu,k} \leftarrow \begin{bmatrix} Q_k & 0 & 0 \\ 0 & R_k & 0 \\ 0 & 0 & P_{\mu,k|k-1} \end{bmatrix}, \quad \mathfrak{X}_k \leftarrow \begin{bmatrix} 0 \\ 0 \\ \hat{\mathbf{x}}_{k|k-1} \end{bmatrix}, \quad \mathfrak{A}_k \leftarrow \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix},$$

$$\mathfrak{M}_{\mu,k} \leftarrow \begin{bmatrix} \Phi_{1,\mu,k} & 0 & 0 & 0 \\ 0 & \Phi_{2,\mu,k} & 0 & 0 \\ 0 & 0 & \hat{\lambda}_k^{-1} I & 0 \\ 0 & 0 & 0 & \hat{\lambda}_k^{-1} I \end{bmatrix}, \quad \mathfrak{F}_k \leftarrow \begin{bmatrix} H_k & 0 & F_k \\ 0 & D_k & C_k \\ E_{H_k} & 0 & E_{F_k} \\ 0 & E_{D_k} & E_{C_k} \end{bmatrix}, \quad \mathfrak{L}_k \leftarrow \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

sendo

$$\Phi_{1,\mu,k} \leftarrow \mu^{-1} I - \lambda_k^{-1} \mathcal{M}_{1,k} \mathcal{M}_{1,k}^T, \quad \Phi_{2,\mu,k} \leftarrow \mu^{-1} I - \lambda_k^{-1} \mathcal{M}_{2,k} \mathcal{M}_{2,k}^T.$$

Na atribuição acima,  $\lambda_k$  é um parâmetro escalar não negativo obtido pela solução ótima desenvolvida por (SAYED, 2001). O resultado é a aproximação

$$\lambda_k = (1 + \xi) \|\mathcal{M}_k^T \mu I \mathcal{M}_k\|,$$

em que  $\xi > 0$  e  $\mathcal{M}$  é uma matriz diagonal preenchida como

$$\mathcal{M}_k \leftarrow \begin{bmatrix} \mathcal{M}_{1k} & 0 \\ 0 & \mathcal{M}_{2k} \end{bmatrix}.$$

A condição de existência do filtro de Kalman robusto é satisfeita caso a matriz

$$\begin{bmatrix} H_k & 0 & F_k & -I \\ 0 & D_k & C_k & 0 \\ N_{H_k} & 0 & N_{F_k} & 0 \\ 0 & N_{D_k} & N_{C_k} & 0 \end{bmatrix}$$

apresente posto linha completo. Além disso, ao aplicar a função de penalidade para  $\mu \rightarrow \infty$ , a solução do problema irrestrito (2.9) e (2.10) fica próxima da solução ótima do problema restrito original (2.11). É importante notar que o parâmetro de penalidade é uma medida de robustez que deve ser escolhida de acordo com a influência das incertezas no modelo do sistema dinâmico.

A estrutura sintética do algoritmo recursivo robusto apresentado nesta Seção é destacada no Quadro II. O desenvolvimento completo, assim como estudos de estabilidade,

## Quadro II - Algoritmo recursivo preditor-corretor para estimativas robustas.

**Modelo com Incertezas:** Considere (2.1) com  $\Pi_0 \succ 0$ ,  $Q_k \succ 0$ , e  $R_k \succ 0$ .

**Condições Iniciais:**  $\hat{\mathbf{x}}_{0|-1} = 0$  e  $P_{0|-1} = \Pi_0$ .

**Para  $k \geq 0$ :** Dado  $\mathbf{z}_k$ , atualize  $\{\hat{\mathbf{x}}_{k|k}, \hat{\mathbf{x}}_{k+1|k}, P_{k|k}$  e  $P_{k+1|k}\}$  a partir de  $\{\hat{\mathbf{x}}_{k|k-1}$  e  $P_{k|k-1}\}$  como segue:

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} & P_{k|k} & * \\ \hat{\mathbf{x}}_{k+1|k} & * & P_{k+1|k} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\hat{\mathbf{x}}_{k|k-1}^* & 0 & 0 \\ -G_k \mathbf{u}_k & 0 & 0 \\ \mathbf{z}_k & 0 & 0 \\ -N_{G_k} \mathbf{u}_k & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -I_n & 0 \\ 0 & 0 & -I_n \end{bmatrix},$$

$$\begin{bmatrix} Q_k & 0 & 0 & 0 & 0 & 0 & 0 & I_{r_1} & 0 & 0 & 0 \\ 0 & R_k & 0 & 0 & 0 & 0 & 0 & 0 & I_{r_2} & 0 & 0 \\ 0 & 0 & P_{k|k-1} & 0 & 0 & 0 & 0 & 0 & 0 & I_n & 0 \\ 0 & 0 & 0 & \Phi_{1,k} & 0 & 0 & 0 & H_k & 0 & F_k & -I_n \\ 0 & 0 & 0 & 0 & \Phi_{2,k} & 0 & 0 & 0 & D_k & C_k & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{\lambda}_k^{-1} I_{t_1} & 0 & N_{H_k} & 0 & N_{F_k} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \hat{\lambda}_k^{-1} I_{t_2} & 0 & N_{D_k} & N_{C_k} & 0 \\ I_{r_1} & 0 & 0 & H_k^T & 0 & N_{H_k}^T & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{r_2} & 0 & 0 & D_k^T & 0 & N_{D_k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & I_n & F_k^T & C_k^T & N_{F_k}^T & N_{C_k}^T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I_n & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

sendo

$$\hat{\lambda}_k = (1 + \xi) \mu \left\| \text{blkdiag}^\dagger \left( M_{1,k}^T M_{1,k}, M_{2,k}^T M_{2,k} \right) \right\|,$$

$$\Phi_{1,k} = \mu^{-1} I_n - \hat{\lambda}_k^{-1} M_{1,k} M_{1,k}^T \quad \text{e} \quad \Phi_{2,k} = \mu^{-1} I_p - \hat{\lambda}_k^{-1} M_{2,k} M_{2,k}^T.$$

\*  $\hat{\mathbf{x}}_{k|k-1}^*$  denota a estimativa preditora em  $k-1$ .

† retorna a matriz diagonal em bloco, alinhando as matrizes de entrada  $M_{1,k}^T M_{1,k}$  e  $M_{2,k}^T M_{2,k}$ .

convergência e exemplos numéricos podem ser consultados em (ROCHA; TERRA, 2021), (ISHIHARA et al., 2015) e (ISHIHARA; TERRA; BIANCO, 2010).

A dimensão da matriz simétrica presente no Quadro II é definida com base nos elementos que compõem (2.14). Essa informação é relevante, pois, na aplicação das técnicas de *systolic array* para solução do sistema linear, a construção da malha de elementos de processamento dependerá do tamanho exato da matriz. As dimensões das linhas e colunas da matriz são mostradas na Tabela 2. Nela,  $n$  representa a quantidade de variáveis de estado do modelo,  $r_1$  e  $r_2$  são o total de elementos nos ruídos  $\mathbf{w}_k$  e  $\mathbf{v}_k$ , respectivamente,  $t_1$  e  $t_2$  estão associadas às linhas de  $[N_{F_k} \ N_{G_k} \ N_{H_k}]$  e  $[N_{C_k} \ N_{D_k}]$ , nesta ordem, e  $p$  é a quantidade de medidas em  $\mathbf{z}_k$ .

Tabela 2 – Dimensão dos elementos que compõem a matriz do algoritmo do Quadro II.

Linha/ Coluna	1	2	3	4	5	6	7	8	9	10	11
Dimensão	$r_1$	$r_2$	$n$	$n$	$p$	$t_1$	$t_2$	$r_1$	$r_2$	$n$	$n$

## 2.2 Exemplo Numérico

**Exemplo 2.2.1** Considere um sistema dinâmico linear descrito como (2.1), com parâmetros nominais constantes e elementos de incerteza definidos por

$$F = \begin{bmatrix} 0.92 & 0.10 \\ 0.20 & 0.75 \end{bmatrix}; \quad G = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \quad H = \begin{bmatrix} 0.20 & 0.10 \\ 0.10 & 0.01 \end{bmatrix}; \quad C = [0 \ 1]; \quad D = 0.01;$$

$$N_F = [0.22 \ -0.02]; \quad N_G = 0; \quad N_H = [0.25 \ 0.61]; \quad N_C = [1.01 \ 0.13]; \quad N_D = 2.31.$$

Para o mesmo sistema, as matrizes de ponderação usadas no modelo de incertezas (2.3) são indicadas por

$$\mathcal{M}_1 = [1 \ 1]^T \quad \mathcal{M}_2 = 0.1.$$

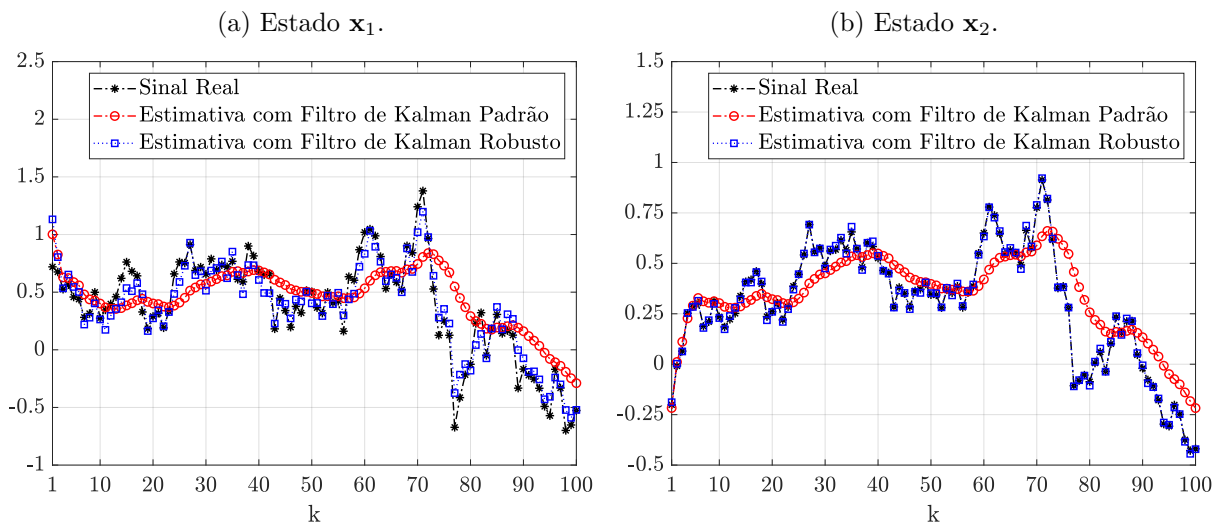
### Solução

Com base nos parâmetros sugeridos, foi desenvolvido um código em ambiente MATLAB® utilizando o algoritmo recursivo apresentado no Quadro II. Inicialmente, sinais foram gerados a partir dos parâmetros nominais do sistema com ruídos associados de variâncias unitárias. A simulação consiste em comparar o KF padrão, apresentado na Figura 2, com as estimativas robustas filtradas baseadas no algoritmo do Quadro II. Foram consideradas as condições iniciais:  $\Pi_0 = 100I_2$ ,  $Q = 2I_2$  e  $R = 1$ . Os fatores de contração  $\Delta_{(1,2)_k}$  foram atualizados apenas no início da simulação. Com propósito de investigar o comportamento inicial dos algoritmos, foram definidos os estados iniciais dos sinais  $\mathbf{x}_1 = 1$  e  $\mathbf{x}_2 = -1$ .

### Resultados

Para entrada de controle  $\mathbf{u}_k = 0$ , em um horizonte  $N = 100$ , as estimativas dos estados foram:

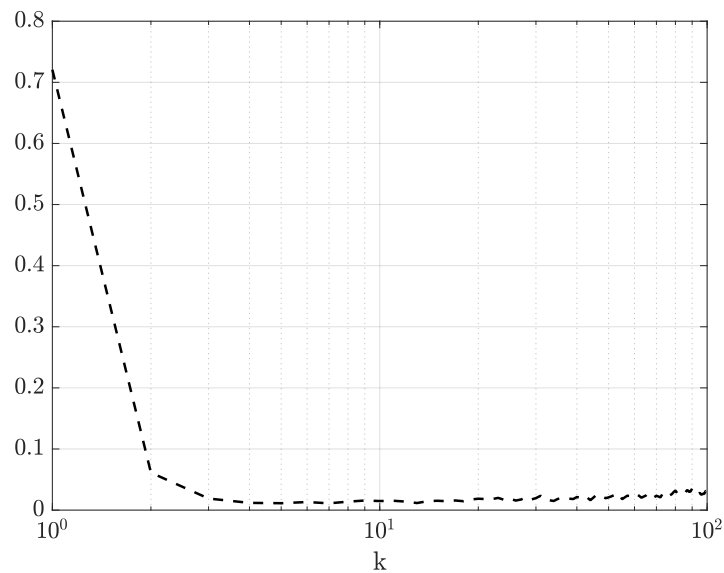
Figura 3 – Estimativas filtradas.



Com base nos resultados apresentados na Figura 3, é possível notar o desempenho superior do KF robusto comparado ao KF padrão. Entretanto, uma abordagem formal para indicar o desempenho do filtro robusto pode ser feita a partir do erro das estimativas. Uma boa aproximação do erro pode ser alcançada usando o método de Monte Carlo. Para este exemplo foram realizadas  $L = 5000$  simulações, tal que, para cada instante  $k$  o quadrado da norma do erro de estimativa seja acumulado. Ao final, as médias dos valores acumulados são computadas. Em síntese,

$$\mathbb{E} \left\{ \|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}\|^2 \right\} \approx \frac{1}{L} \sum_{\ell=1}^L \|\mathbf{x}_k^{(\ell)} - \hat{\mathbf{x}}_{k|k}^{(\ell)}\|^2. \quad (2.15)$$

Figura 4 – Norma do erro das estimativas de estados.







---

## Capítulo 3

# Navegação Aérea

---

O conceito de navegação possui atribuições abrangentes dependendo da sua aplicação, não dispondo de uma definição exata para o termo. O dicionário de Oxford (OED, 2021) define navegação como a “habilidade ou o processo de planejar uma rota para um veículo e levá-lo até lá”. (LAWRENCE, 2012) refere-se à navegação como o processo de seguir um caminho em um mapa através de latitudes e longitudes pré-determinadas. Já (FARRELL, 2008) divide o conceito de navegação em duas categorias:

- planejamento e execução das manobras necessárias para se mover entre os locais desejados;
- determinação precisa dos estados de navegação do veículo.

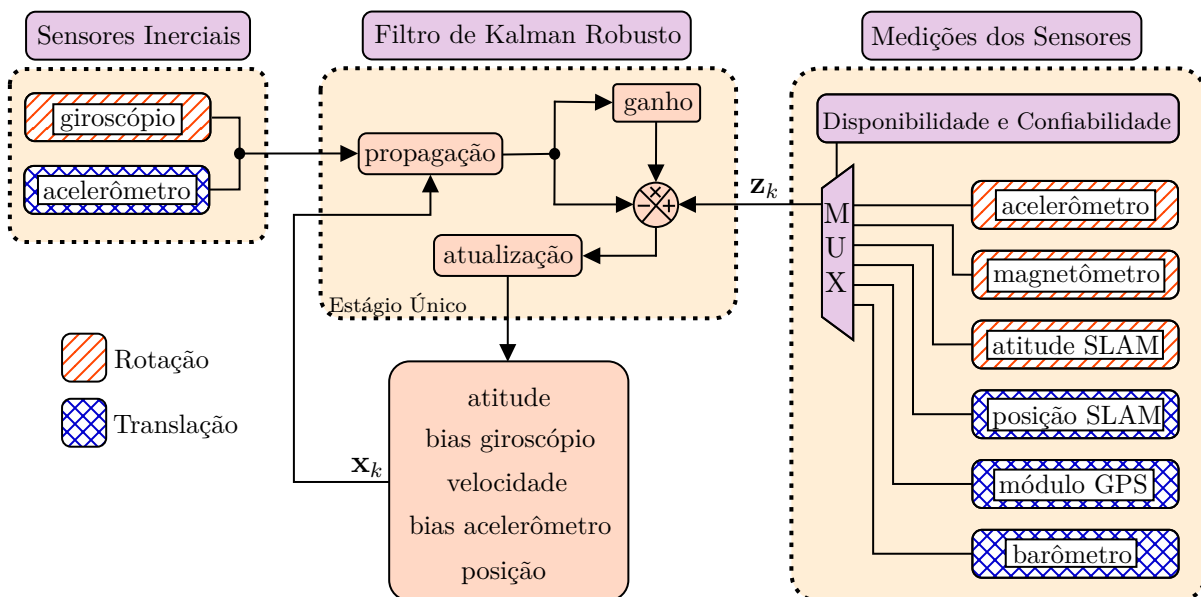
Para a primeira perspectiva, existem diferentes meios de viabilizá-la, tais como: pilotagem manual: respaldada no reconhecimento visual e percepção das ações do veículo; *dead reckoning*: baseado no conhecimento prévio do ambiente, trajetória e sua origem; navegação via rádio: guiada por fontes de radiofrequência como sinal de referência; inercial: sustentada nas informações de velocidade, posição e orientação em tempo real do veículo em relação a um referencial inercial (WELLENHOF; LEGAT; WIESER, 2003).

Para a proposta desta dissertação, foi adotada a segunda definição referente à navegação. Ou seja, o objetivo do sistema de navegação aérea é realizar estimativas dos estados de velocidade, posição e atitude do veículo em tempo real, à medida que se movimenta ao longo de uma trajetória. Simultaneamente, os dados estimados poderão ser utilizados para o controle automático do veículo aéreo, planejamento de rotas, localização e mapeamento do ambiente. Portanto, para se alcançar a primeira definição de navegação descrita por (FARRELL, 2008), necessariamente no caso deste estudo, a segunda definição deverá ser desenvolvida.

Com o propósito de obter resultados rápidos e precisos, a adoção de um sistema automático de navegação aérea será capaz de gerar resultados melhores, se comparados ao processo manual. Dentre os meios historicamente utilizados para essa finalidade, o Filtro de Kalman (KF), discutido no Capítulo 2, é a forma mais consolidada, sendo utilizada inclusive em projetos de baixo custo (NGO et al., 2017; SHIN; EL-SHEIMY, 2004). As fontes de dados mais comuns para construção do sistema automático de navegação aérea são baseadas em sensores inerciais, magnetômetros, GPS e dados visuais, assim como adotado nesta pesquisa.

A organização do sistema de navegação aérea é mostrada na Figura 5. A velocidade angular e aceleração linear, fornecidas pelos giroscópios e acelerômetros, respectivamente, são usados no modelo de propagação dos estados de navegação. Para correção dos estados estimados *a priori*, as medidas fornecidas pelos outros sensores são selecionadas de acordo com a disponibilidade e confiabilidade dos sinais. A escolha dos sinais sensoriais para atualização dos estados de navegação é discutida na Seção 3.3 deste Capítulo.

Figura 5 – Estrutura do sistema de navegação aérea.



O desenvolvimento do KF robusto para estimativa dos estados de navegação segue o conceito construtivo de filtragem apresentado no Quadro II. Dadas as medidas observadas, os estados corrigidos são calculados a partir das estimativas preditoras da iteração anterior. Ao mesmo tempo, a predição dos estados e da matriz de covariância são computadas para uso na próxima iteração. Ao fim de cada processo, o vetor de estado completo é atualizado com as estimativas de erro filtradas.

## 3.1 Modelagem

A construção de um sistema automático de navegação baseado na estimativa de estados, como o da Figura 5, se desenvolve sob algumas etapas de projeto. A primeira delas consiste em encontrar o modelo dinâmico do sistema. No caso do KF, é estruturado, essencialmente, um modelo de propagação dos estados e outro modelo de medição. Assim, as informações dos sensores são fundidas e estados não observáveis podem ser estimados.

Existem dois modos de realizar estimativas usando o KF. A primeira delas, conhecida como KF direto ou KF de estado completo, é baseada na estimativa do vetor de estado na sua forma padrão. A segunda maneira, conhecida como KF indireto ou KF de estado de erro, realiza a estimativa do vetor de erro dos estados e, em seguida, o resultado é somado ao estado acumulado. (LIU et al., 2019) fazem uma análise comparativa entre estimativas de estado de erro e de estado total, com integração IMU/GNSS baseada em KF.

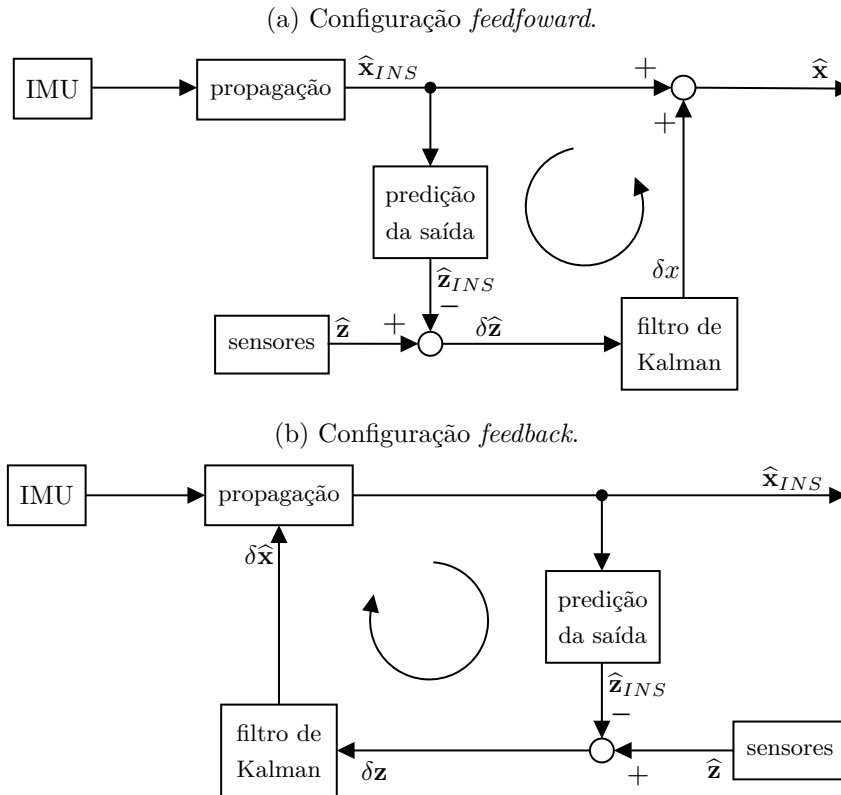
Em formulações de KF de estado de erro, o valor verdadeiro do estado é definido como a soma do estado estimado e o erro

$$y = \hat{y} + \delta y. \quad (3.1)$$

A principal vantagem da perspectiva indireta é que o sistema está sempre operando próximo à origem e, portanto, longe de possíveis singularidades que possam ocasionar perdas de grau de liberdade no sistema. Além disso, fornece uma garantia de que a validade de linearização é mantida em todos os momentos (MADYASTHA et al., 2011). Outra questão interessante do modelo de estado de erro é que os estados estimados são sempre valores pequenos, para uma frequência de amostragem suficiente. Assim, os produtos entre pequenos valores podem ser desconsiderados na modelagem, sem grandes perdas.

O KF indireto pode ser representado na configuração *feedforward* ou *feedback*. A Figura 6 mostra as duas abordagens para implementação do KF baseado no erro de estado. Na configuração *feedforward*, os estados obtidos via IMU são propagados e os resultados são usados na predição de saída  $\hat{\mathbf{z}}$ . O erro entre as medidas observadas nos outros sensores e a predição de saída é aplicado ao KF que faz a estimativa do estado de erro  $\delta \mathbf{x}$ . Ao final, o erro de estado é somado ao estado propagado do sistema de navegação inercial (INS)  $\hat{\mathbf{x}}_{INS}$  gerando a estimativa corrigida com compensação *feedforward*. Na configuração *feedback* do KF indireto, as estimativas de erro são usadas para atualizar as estimativas do INS diretamente, para que o crescimento do erro dos resultados do INS. Para que essa configuração seja praticável, a frequência do *loop* tem que ser maior que o cálculo de propagação dos estados via IMU. A saída estimada  $\hat{\mathbf{x}}$  é obtida regulando o erro  $\delta \hat{\mathbf{x}}$ , de modo que  $\hat{\mathbf{x}}_{INS} \rightarrow \hat{\mathbf{x}}$  à medida que  $\delta \hat{\mathbf{x}} \rightarrow 0$ . Esse ajuste da saída a partir do sinal de entrada é similar a um sistema de controle por realimentação, por isso a designação *feedback* para essa configuração do KF indireto. Para este estudo foi utilizado a configuração *feedforward*.

Figura 6 – Fluxo de dados do KF indireto para o sistema de navegação aérea.



A modelagem dos sensores inerciais resulta em equações diferenciais de primeira ordem que descrevem a dinâmica dos estados de navegação. Essas equações são incorporadas à representação por espaço de estados de forma que possam ser introduzidos na estrutura do algoritmo do KF, sem grandes alterações. Para construção do modelo de atitude, posição, velocidade e biases dos sensores inerciais, foram pesquisadas diversas fontes de referência na literatura relacionada. Dentre os trabalhos que abordam o assunto com maior detalhamento, destacam-se as produções de (SOLA, 2017) e (TRAWNY; ROUMELIOTIS, 2005). Entretanto, essas obras apresentam uma ligeira diferença na convenção para representação do quatérnio de atitude. A primeira utiliza a notação de Hamilton, em que a parte escalar é o primeiro elemento do quatérnio. Já a segunda obra citada, utiliza a notação em que a parte escalar se encontra no último elemento do quatérnio. Essa troca no posicionamento dos elementos do quatérnio impacta diretamente em todas as operações algébricas, inclusive na interpretação geométrica de rotação. Devido ao suporte em MATLAB®, ROS e algumas bibliotecas em linguagem C que empregam a convenção de Hamilton, a modelagem desenvolvida neste trabalho utiliza essa convenção para descrição do quatérnio de atitude. Por esse motivo, o desenvolvimento subsequente dos modelos é principalmente respaldado por (SOLA, 2017). Além disso, (TRAWNY; ROUMELIOTIS, 2005) apresentam apenas as estimativas de atitude e bias do giroscópio, enquanto que (SOLA, 2017) realiza estimativas de atitude, velocidade, posição, vetor de gravidade e biases do giroscópio e acelerômetro.

O objetivo da etapa de modelagem é obter as equações que descrevem o sistema de navegação. Para isso, o acelerômetro irá fornecer dados para estimar a velocidade e posição do veículo. Já o giroscópio irá fornecer dados para estimativa da atitude do veículo. O vetor de estados de erro, associado aos estados do sistema de navegação da Figura 5 através da relação (3.1), é

$$\delta \mathbf{x} = \begin{bmatrix} \delta \boldsymbol{\theta} \\ \delta \mathbf{b}_g \\ \mathcal{G} \delta \mathbf{v} \\ \delta \mathbf{b}_a \\ \mathcal{G} \delta \mathbf{p} \end{bmatrix}. \quad (3.2)$$

Uma questão importante relacionada aos modelos dinâmicos para o KF de estado de erro desenvolvidos para este estudo é que a dimensão do estado completo do sistema não acompanha a dimensão do vetor de estimativas definido em (3.2). A estimativa do erro de atitude realizada pelo KF é descrita por um vetor em  $\mathbb{R}^3$ . Já o quatérnio de atitude, atualizado a cada iteração do algoritmo, é constituído por elementos no espaço vetorial quadridimensional sobre os números reais. Essa relação fica evidente para representação de pequenas rotações por um quatérnio unitário, com representação algébrica do operador rotacional apresentada em (B.4).

### 3.1.1 Giroscópio

O giroscópio é um dispositivo que fornece a velocidade angular baseando-se no princípio da inércia. Para o modelo de atitude, o conteúdo relevante refere-se apenas à dinâmica do sensor. No caso mais simples, desconsiderando alguns fatores de influência nas medidas, como vibração, temperatura e histerese, o sinal fornecido pelo giroscópio pode ser descrito por

$$\mathbf{y}_g(t) = \boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{w}_g(t), \quad (3.3)$$

com as variáveis descritas na Tabela 3. As medidas  $\mathbf{y}_g$  são vetores que representam a velocidade de rotação em cada eixo ortogonal no espaço  $\mathbb{R}^3$ . A Figura 7 mostra o sentido da velocidade angular mensurada pelo giroscópio, seguindo a regra da mão direita.

Figura 7 – Configuração de eixos dos giroscópios.

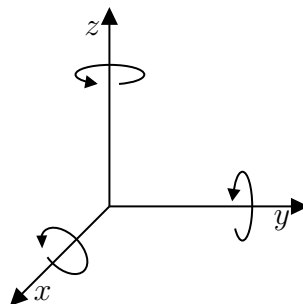


Tabela 3 – Variáveis usadas na modelagem da atitude.

Variável Verdadeira	Descrição	Variável Estimada	Erro de Estado	Espaço Vetorial
$\bar{q}$	quatérnio unitário de atitude	$\hat{q}$	$\delta\bar{q} = \hat{q}^{-1} \otimes \bar{q}$	$\mathbb{R}^4$
$\mathbf{b}_g$	bias do giroscópio	$\hat{\mathbf{b}}_g$	$\delta\mathbf{b}_g = \mathbf{b}_g - \hat{\mathbf{b}}_g$	$\mathbb{R}^3$
$\boldsymbol{\omega}$	velocidade angular	$\hat{\boldsymbol{\omega}}$	$\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$	$\mathbb{R}^3$
$q_\omega$	quatérnio da velocidade angular	$\hat{q}_\omega$	$\delta q_\omega = q_\omega - \hat{q}_\omega$	$\mathbb{R}^4$
$\delta\boldsymbol{\theta}$	vetor de ângulos de rotação	–	–	$\mathbb{R}^3$
$\mathbf{y}_g$	sinal medido do giroscópio	–	–	$\mathbb{R}^3$
$\mathbf{w}_g$	ruído Gaussiano do giroscópio	–	–	$\mathbb{R}^3$
$\mathbf{w}_{\mathbf{b}_g}$	ruído Gaussiano do bias do giroscópio	–	–	$\mathbb{R}^3$

### 3.1.2 Modelo de Atitude

O desenvolvimento completo do modelo dinâmico de atitude para o sistema de navegação pode envolver muitas variáveis, ampliando muito a complexidade do sistema. Para o caso deste estudo, a modelagem de atitude baseada no estado do erro é derivada a partir do modelo simplificado de medida do giroscópio (3.3). O estado de erro de atitude utiliza como base algébrica os quatérnios, beneficiando-se principalmente da interpolação linear de rotações e prevenção de possíveis singularidades que podem ocorrer com o uso de ângulos de Euler na representação da atitude como o *Gimbal Lock*<sup>1</sup>. O Apêndice B apresenta uma análise detalhada sobre a álgebra quaterniônica e suas propriedades.

A partir das medidas fornecidas pelo giroscópio fixado no veículo, detalhada em (3.3), a velocidade angular do corpo no espaço é descrita por

$$\boldsymbol{\omega}(t) = \mathbf{y}_g(t) - \mathbf{b}_g(t) - \mathbf{w}_g(t). \quad (3.4)$$

Os valores de  $\boldsymbol{\omega}(t)$  são usados na equação diferencial que descreve a dinâmica da atitude do veículo. A Seção B.2.3 mostra o desenvolvimento do cálculo da diferenciação no tempo do quatérnio que representa a atitude entre os dois frames. A taxa de movimento rotacional do frame do corpo  $\{\mathcal{B}\}$  em relação ao frame global  $\{\mathcal{G}\}$  é calculada pela derivada em relação ao tempo do quatérnio correspondente, indicada como

$$\dot{\bar{q}}(t) = \frac{1}{2}\bar{q}(t) \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}(t) \end{bmatrix} = \frac{1}{2}\bar{q}(t) \otimes q_\omega(t). \quad (3.5)$$

Em (3.5),  $q_\omega$  representa uma aproximação das velocidades angulares descritas em um quatérnio com parte escalar nula e a parte vetorial preenchida pelo vetor  $\boldsymbol{\omega}$ . A conceituação formal dessa aproximação é apresentada na Seção B.2.3 deste documento.

<sup>1</sup> *Gimbal Lock*. é a perda de um grau de liberdade quando os eixos de um sistema em  $\mathbb{R}^3$  ficam alinhados, comprometendo severamente o mecanismo de navegação e controle. O incidente mais conhecido ocorreu na missão lunar Apollo 11.

Considerando o uso dos quatérnios para composição de pequenas rotações, o erro entre a atitude verdadeira  $\bar{q}(t)$  e sua estimativa  $\hat{q}(t)$  é aproximadamente o vetor de erro angular  $3 \times 1$ ,  $\delta\boldsymbol{\theta}(t)$ , definido pelo quatérnio de erro  $\delta\bar{q}(t)$  de acordo com a relação (B.17), tal que

$$\delta\bar{q}(t) = \hat{q}^{-1}(t) \otimes \bar{q}(t) \simeq \left[ 1 \quad \frac{1}{2} \delta\boldsymbol{\theta}^T(t) \right]^T, \quad (3.6)$$

sendo que  $\delta\bar{q}$  descreve uma pequena rotação do corpo, de forma que a atitude estimada se alinhe com a atitude verdadeira. Conseqüentemente, a derivada temporal da atitude verdadeira, calculada em (3.5), também pode ser desenvolvida em termos de  $\hat{q}$  e  $\delta\bar{q}$  como

$$\begin{aligned} \frac{1}{2} \bar{q}(t) \otimes \boldsymbol{q}_\omega(t) &= \frac{d}{dt} (\hat{q}(t) \otimes \delta\bar{q}(t)); \\ \frac{1}{2} \hat{q}(t) \otimes \delta\bar{q}(t) \otimes \boldsymbol{q}_\omega(t) &= \dot{\hat{q}}(t) \otimes \delta\bar{q}(t) + \hat{q}(t) \otimes \dot{\delta\bar{q}}(t); \\ \frac{1}{2} \hat{q}(t) \otimes \delta\bar{q}(t) \otimes \boldsymbol{q}_\omega(t) &= \frac{1}{2} \hat{q}(t) \otimes \hat{\boldsymbol{q}}_\omega(t) \otimes \delta\bar{q}(t) + \hat{q}(t) \otimes \dot{\delta\bar{q}}(t). \end{aligned}$$

Realizando uma multiplicação à esquerda por  $\hat{q}(t)^{-1}$  em ambos os lados da equação e isolando o termo  $\dot{\delta\bar{q}}(t)$  é obtido

$$\dot{\delta\bar{q}}(t) = \frac{\delta\bar{q}(t) \otimes \boldsymbol{q}_\omega(t) - \hat{\boldsymbol{q}}_\omega(t) \otimes \delta\bar{q}(t)}{2}.$$

Como  $\delta\bar{q}(t)$  é assintoticamente igual a  $\left[ 1 \quad \frac{1}{2} \delta\boldsymbol{\theta}^T(t) \right]^T$  para pequenas rotações, de acordo com (3.6), a derivada do quatérnio de erro torna-se

$$\begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}}(t) \end{bmatrix} = \delta\bar{q}(t) \otimes \boldsymbol{q}_\omega(t) - \hat{\boldsymbol{q}}_\omega(t) \otimes \delta\bar{q}(t). \quad (3.7)$$

A notação matricial para multiplicação de quatérnios à esquerda e à direita, descritas em (B.23) e (B.24) respectivamente, permite reescrever a equação diferencial como

$$\begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}}(t) \end{bmatrix} = \left[ \begin{array}{c|c} 0 & -\boldsymbol{\omega}(t) \\ \boldsymbol{\omega}(t) & -[\boldsymbol{\omega}(t) \times] \end{array} \right] \begin{bmatrix} \delta\bar{q}_0(t) \\ \delta\bar{\boldsymbol{q}}(t) \end{bmatrix} - \left[ \begin{array}{c|c} 0 & -\hat{\boldsymbol{\omega}}(t) \\ \hat{\boldsymbol{\omega}}(t) & [\hat{\boldsymbol{\omega}}(t) \times] \end{array} \right] \begin{bmatrix} \delta\bar{q}_0(t) \\ \delta\bar{\boldsymbol{q}}(t) \end{bmatrix}. \quad (3.8)$$

A relação (3.1) pode ser usada para associar  $\boldsymbol{\omega}(t)$  e  $\hat{\boldsymbol{\omega}}(t)$ . Além disso,  $\delta\bar{q}$  pode ser convertido para  $\delta\boldsymbol{\omega}$  segundo (3.6). Assim, a equação (3.8) torna-se

$$\begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}}(t) \end{bmatrix} = \left[ \begin{array}{c|c} 0 & -[\delta\boldsymbol{\omega}(t)]^T \\ \delta\boldsymbol{\omega}(t) & -[2\hat{\boldsymbol{\omega}}(t) + \delta\boldsymbol{\omega}(t)] \times \end{array} \right] \begin{bmatrix} 1 \\ \delta\boldsymbol{\theta}(t)/2 \end{bmatrix},$$

que resulta nas igualdades:

$$\begin{cases} 0 &= -[\delta\boldsymbol{\omega}(t)]^T \delta\boldsymbol{\theta}(t)/2, \\ \dot{\delta\boldsymbol{\theta}}(t) &= \delta\boldsymbol{\omega}(t) - [\hat{\boldsymbol{\omega}}(t) \times] \delta\boldsymbol{\theta}(t) - [\delta\boldsymbol{\omega}(t) \times] \delta\boldsymbol{\theta}(t)/2. \end{cases}$$

O primeiro termo é um valor infinitesimal já esperado, conforme estabelecido em (3.7). O segundo termo do sistema de equações possui um produto vetorial entre  $\delta\boldsymbol{\omega}(t)$  e  $\delta\boldsymbol{\theta}(t)/2$

que também resulta em um valor infinitesimal pelos mesmos motivos anteriormente argumentados. Desta forma, a variação no tempo do erro de atitude é definida como

$$\delta\dot{\boldsymbol{\theta}}(t) = \delta\boldsymbol{\omega}(t) - \left[ \hat{\boldsymbol{\omega}}(t) \times \right] \delta\boldsymbol{\theta}(t).$$

Sendo  $\mathbf{w}_g$  um sinal aleatório atribuído à leitura do giroscópio descrita em (3.3), tal que  $\mathbb{E}\{\mathbf{w}_a\} = 0$ , a estimativa da velocidade angular não o incluiria. Portanto, foram consideradas as seguintes definições para separar os grandes e pequenos sinais de (3.4):

$$\begin{aligned} \boldsymbol{\omega}(t) &\triangleq \hat{\boldsymbol{\omega}}(t) + \delta\boldsymbol{\omega}(t), \\ \hat{\boldsymbol{\omega}}(t) &\triangleq \mathbf{y}_g(t) - \hat{\mathbf{b}}_g(t), \\ \delta\boldsymbol{\omega}(t) &\triangleq -\delta\mathbf{b}_g(t) - \mathbf{w}_g(t). \end{aligned} \quad (3.9)$$

Assim, considerando as definições estabelecidas em (3.9), em termos das medidas do giroscópio e estados, a equação que estabelece a dinâmica do erro de atitude torna-se

$$\delta\dot{\boldsymbol{\theta}}(t) = - \left[ \left[ \mathbf{y}_g(t) - \hat{\mathbf{b}}_g(t) \right] \times \right] \delta\boldsymbol{\theta}(t) - \delta\mathbf{b}_g(t) - \mathbf{w}_g(t) \quad (3.10)$$

### 3.1.3 Acelerômetro

O modelo do acelerômetro em tempo contínuo é descrito como

$$\mathbf{y}_a(t) = {}^{\mathcal{G}}\mathbf{R}^T \left[ {}^{\mathcal{G}}\mathbf{a}(t) + {}^{\mathcal{G}}\mathbf{g}_e \right] + \mathbf{b}_a(t) + \mathbf{w}_a(t), \quad (3.11)$$

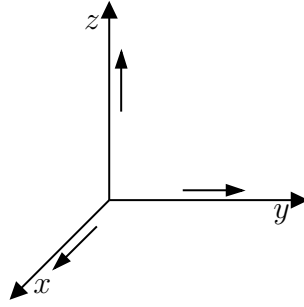
com variáveis descritas na Tabela 4. O uso de  ${}^{\mathcal{G}}\mathbf{R}^T$  é possível devido à condição de ortogonalidade que esta componente abrange, facilitando os cálculos decorrentes. O sentido das acelerações lineares medidas em cada eixo em  $\mathbb{R}^3$  pelo acelerômetro é mostrado na Figura 8.

Tabela 4 – Variáveis usadas nas modelagens de velocidade e posição.

Variável	Descrição	Variável Estimada	Erro de Estado	Espaço Vetorial
${}^{\mathcal{G}}\mathbf{v}$	velocidade linear do corpo com relação à $\{\mathcal{G}\}$	${}^{\mathcal{G}}\hat{\mathbf{v}}$	${}^{\mathcal{G}}\delta\mathbf{v} = {}^{\mathcal{G}}\mathbf{v} - {}^{\mathcal{G}}\hat{\mathbf{v}}$	$\mathbb{R}^3$
${}^{\mathcal{G}}\mathbf{R}$	matriz de rotação de $\{\mathcal{B}\}$ para $\{\mathcal{G}\}$	${}^{\mathcal{G}}\hat{\mathbf{R}}$	${}^{\mathcal{G}}\delta\mathbf{R} = {}^{\mathcal{G}}\hat{\mathbf{R}}^{-1}{}^{\mathcal{G}}\mathbf{R}$	$\mathbb{R}^{3 \times 3}$
$\mathbf{b}_a$	bias do acelerômetro	$\hat{\mathbf{b}}_a$	$\delta\mathbf{b}_a = \mathbf{b}_a - \hat{\mathbf{b}}_a$	$\mathbb{R}^3$
${}^{\mathcal{G}}\mathbf{a}$	aceleração linear do corpo com relação à $\{\mathcal{G}\}$	${}^{\mathcal{G}}\hat{\mathbf{a}}$	${}^{\mathcal{G}}\delta\mathbf{a} = {}^{\mathcal{G}}\mathbf{a} - {}^{\mathcal{G}}\hat{\mathbf{a}}$	$\mathbb{R}^3$
${}^{\mathcal{G}}\mathbf{g}_e$	vetor de gravidade da Terra no frame $\{\mathcal{G}\}$	–	–	$\mathbb{R}^3$
$\mathbf{y}_a$	sinal medido do acelerômetro	–	–	$\mathbb{R}^3$
$\mathbf{w}_a$	ruído Gaussiano do acelerômetro	–	–	$\mathbb{R}^3$
$\mathbf{w}_{b_a}$	ruído Gaussiano do bias do acelerômetro	–	–	$\mathbb{R}^3$



Figura 8 – Configuração de eixos dos acelerômetros.



### 3.1.4 Modelos de Velocidade e Posição

A aceleração linear é uma informação significativa para qualquer sistema de navegação, utilizada essencialmente para os cálculos das velocidade e posição lineares do veículo, por meio da integração temporal do sinal. Os modelos cinemáticos translacionais são desenvolvidos a partir do modelo do acelerômetro, (3.11), resultando em

$${}^{\mathcal{G}}\mathbf{a}(t) = {}^{\mathcal{G}}\mathbf{R} [\mathbf{y}_a(t) - \mathbf{b}_a(t) - \mathbf{w}_a(t)] - {}^{\mathcal{G}}\mathbf{g}_e. \quad (3.12)$$

Como  $\boldsymbol{\theta}(t)$  e  $\mathbf{b}_a(t)$  são estados não observáveis do sistema, e portanto valores estimados, e  $\mathbf{w}_a$  é um sinal aleatório, tal que  $\mathbb{E}\{\mathbf{w}_a\} = 0$ , a estimativa da aceleração linear do veículo em relação ao frame global torna-se

$${}^{\mathcal{G}}\hat{\mathbf{a}}(t) = {}^{\mathcal{G}}\widehat{\mathbf{R}} [\mathbf{y}_a(t) - \hat{\mathbf{b}}_a(t)] - {}^{\mathcal{G}}\mathbf{g}_e. \quad (3.13)$$

O erro da matriz de rotação estimada,  ${}^{\mathcal{G}}\delta\mathbf{R}$ , é detalhado algebricamente na Tabela 4. Baseando-se na descrição da matriz de rotação por quatérnios, (B.25), juntamente com a aproximação de atitude, (3.6), a matriz de rotação em função de  $\delta\boldsymbol{\theta}$  pode ser calculada como

$${}^{\mathcal{G}}\delta\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -\delta\theta_z & \delta\theta_y \\ \delta\theta_z & 0 & -\delta\theta_x \\ -\delta\theta_y & \delta\theta_x & 0 \end{bmatrix} - \begin{bmatrix} \delta\theta_x^2 & \delta\theta_x\delta\theta_y & \delta\theta_x\delta\theta_z \\ \delta\theta_y\delta\theta_x & \delta\theta_y^2 & \delta\theta_y\delta\theta_z \\ \delta\theta_z\delta\theta_x & \delta\theta_z\delta\theta_y & \delta\theta_z^2 \end{bmatrix}. \quad (3.14)$$

Desconsiderando o termo quadrático de (3.14), a aproximação da matriz de rotação, em termos de elementos lineares, é descrita por

$${}^{\mathcal{G}}\delta\mathbf{R} \simeq \mathbf{I} + [\delta\boldsymbol{\theta} \times], \quad (3.15)$$

em que  $[\delta\boldsymbol{\theta} \times]$  é a matriz antissimétrica de produto vetorial (B.13), dado o vetor de erro da estimativa da atitude,  $\delta\boldsymbol{\theta}$ . Assim, a partir da relação entre o valor verdadeiro e estimado, a matriz de rotação do frame  $\{\mathcal{B}\}$  para o frame  $\{\mathcal{G}\}$ , pode ser reescrita como

$${}^{\mathcal{G}}\mathbf{R} = {}^{\mathcal{G}}\widehat{\mathbf{R}} (\mathbf{I} + [\delta\boldsymbol{\theta} \times]).$$

Deste modo, a aceleração linear em (3.12) é descrita em termos dos estados estimados e do erro de estado por

$${}^{\mathcal{G}}\mathbf{a}(t) = {}^{\mathcal{G}}\widehat{\mathbf{R}}(\mathbf{I} + [\delta\boldsymbol{\theta}(t)\times]) [{}^{\mathcal{B}}\widehat{\mathbf{a}}(t) + {}^{\mathcal{B}}\delta\mathbf{a}(t)] - {}^{\mathcal{G}}\mathbf{g}_e,$$

sendo válidas as seguintes definições, congêneres à (3.9):

$$\begin{aligned} {}^{\mathcal{B}}\mathbf{a}(t) &\triangleq {}^{\mathcal{B}}\widehat{\mathbf{a}}(t) + {}^{\mathcal{B}}\delta\mathbf{a}(t), \\ {}^{\mathcal{B}}\widehat{\mathbf{a}}(t) &\triangleq \mathbf{y}_a(t) - \widehat{\mathbf{b}}_a(t), \\ {}^{\mathcal{B}}\delta\mathbf{a}(t) &\triangleq -\delta\mathbf{b}_a(t) - \mathbf{w}_a(t). \end{aligned} \quad (3.16)$$

Observe que o índice  $\mathcal{B}$  em (3.16) designa a aceleração linear no corpo, ainda sem a transformação espacial para o referencial global,  $\{\mathcal{G}\}$ , como feito em (3.12) e (3.13).

A aceleração linear do corpo em relação ao frame global é igual a taxa de variação da sua velocidade linear no tempo. Logo, a derivada temporal da velocidade do corpo, em termos de estados estimados e do erro relativo, é desenvolvida como

$$\begin{aligned} {}^{\mathcal{G}}\dot{\mathbf{v}}(t) &= {}^{\mathcal{G}}\mathbf{a}(t); \\ {}^{\mathcal{G}}\dot{\widehat{\mathbf{v}}}(t) + {}^{\mathcal{G}}\dot{\delta\mathbf{v}}(t) &= {}^{\mathcal{G}}\widehat{\mathbf{R}}(\mathbf{I} + [\delta\boldsymbol{\theta}(t)\times]) [{}^{\mathcal{B}}\widehat{\mathbf{a}}(t) + {}^{\mathcal{B}}\delta\mathbf{a}(t)] - {}^{\mathcal{G}}\mathbf{g}_e; \\ \cancel{{}^{\mathcal{G}}\widehat{\mathbf{R}}{}^{\mathcal{B}}\widehat{\mathbf{a}}(t)} - \cancel{{}^{\mathcal{G}}\mathbf{g}_e} + {}^{\mathcal{G}}\dot{\delta\mathbf{v}}(t) &= \cancel{{}^{\mathcal{G}}\widehat{\mathbf{R}}{}^{\mathcal{B}}\widehat{\mathbf{a}}(t)} + \cancel{{}^{\mathcal{G}}\widehat{\mathbf{R}}{}^{\mathcal{B}}\delta\mathbf{a}(t)} + \dots \\ &\quad \cancel{{}^{\mathcal{G}}\widehat{\mathbf{R}}[\delta\boldsymbol{\theta}(t)\times]}{}^{\mathcal{B}}\widehat{\mathbf{a}}(t) + \cancel{{}^{\mathcal{G}}\widehat{\mathbf{R}}[\delta\boldsymbol{\theta}(t)\times]}{}^{\mathcal{B}}\delta\mathbf{a}(t) - \cancel{{}^{\mathcal{G}}\mathbf{g}_e}. \end{aligned} \quad (3.17)$$

O termo de (3.17) resultante do produto entre  $[\delta\boldsymbol{\theta}(t)\times]$  e  $\delta\mathbf{a}_{\mathcal{B}}(t)$ , neste caso, é desconsiderado devido à influência pouco significativa sobre o resultado final do cálculo. Ou seja,

$${}^{\mathcal{G}}\dot{\delta\mathbf{v}}(t) = {}^{\mathcal{G}}\widehat{\mathbf{R}} [{}^{\mathcal{B}}\delta\mathbf{a}(t) + [\delta\boldsymbol{\theta}(t)\times]{}^{\mathcal{B}}\mathbf{a}(t)]. \quad (3.18)$$

A fim de viabilizar o cálculo de  ${}^{\mathcal{G}}\dot{\delta\mathbf{v}}(t)$  como um produto à esquerda de  $\delta\boldsymbol{\theta}(t)$ , a ordem do produto vetorial  $[\delta\boldsymbol{\theta}(t)\times]\mathbf{a}_{\mathcal{B}}(t)$  em (3.18) é alterada, resultando em

$${}^{\mathcal{G}}\dot{\delta\mathbf{v}}(t) = {}^{\mathcal{G}}\widehat{\mathbf{R}} [{}^{\mathcal{B}}\delta\mathbf{a}(t) - [{}^{\mathcal{B}}\mathbf{a}(t)\times]\delta\boldsymbol{\theta}(t)].$$

Verifique que o sinal do produto foi alterado, devido à troca dos fatores do produto vetorial. Assim, pelas definições apresentadas em (3.16), a equação diferencial do erro da velocidade linear do veículo torna-se

$${}^{\mathcal{G}}\dot{\delta\mathbf{v}}(t) = -{}^{\mathcal{G}}\widehat{\mathbf{R}} \left[ \left[ \mathbf{y}_a(t) - \widehat{\mathbf{b}}_a(t) \right] \times \right] \delta\boldsymbol{\theta}(t) - {}^{\mathcal{G}}\widehat{\mathbf{R}}\delta\mathbf{b}_a(t) - {}^{\mathcal{G}}\widehat{\mathbf{R}}\mathbf{w}_a(t). \quad (3.19)$$

Como a atitude do veículo é baseada em quatérnio, a matriz de rotação  ${}^{\mathcal{G}}\widehat{\mathbf{R}}$  terá que estar, preferencialmente, na base quaterniônica. Essa representação é descrita em (B.26). Portanto, a dinâmica do erro da velocidade linear do veículo é expressa por:

$$\boxed{{}^{\mathcal{G}}\dot{\delta\mathbf{v}}(t) = -{}^{\mathcal{G}}\widehat{\mathbf{R}}(\widehat{\mathbf{q}}(t)) \left[ \left[ \mathbf{y}_a(t) - \widehat{\mathbf{b}}_a(t) \right] \times \right] \delta\boldsymbol{\theta}(t) - {}^{\mathcal{G}}\widehat{\mathbf{R}}(\widehat{\mathbf{q}}(t))\delta\mathbf{b}_a(t) - {}^{\mathcal{G}}\widehat{\mathbf{R}}(\widehat{\mathbf{q}}(t))\mathbf{w}_a(t)} \quad (3.20)$$

Diferente da modelagem apresentada para descrever a dinâmica do erro de velocidade linear, a equação diferencial da posição do corpo no espaço não precisa de ações algébricas extensivas. Como a velocidade é taxa de variação da posição com o tempo, e toda abordagem de alinhamento dos frames  $\{\mathcal{B}\}$  e  $\{\mathcal{G}\}$  já foi tratada anteriormente, a dinâmica do erro de posição linear é expressa por:

$$\boxed{{}^{\mathcal{G}}\dot{\delta\mathbf{p}}(t) = {}^{\mathcal{G}}\delta\mathbf{v}(t)} \quad (3.21)$$

### 3.1.5 Modelos dos Biases

O bias é um componente de erro associado ao sinal fornecido por algum instrumento, o qual, no caso dos sensores inerciais, apresenta flutuações em baixa frequência que variam no tempo de acordo com uma constante estabelecida. Em sensores com tecnologia *MicroElectroMechanical Systems* (MEMS), essas flutuações estão relacionadas principalmente com a variação da temperatura ambiente.

Usualmente, alguns modelos de processos estocásticos são empregados para caracterizar a tendência do bias que afetam os sinais dos sensores inerciais. Dentre as formas mais utilizadas estão o processo de Gauss-Markov de primeira ordem, o processo *random-walk* e os baseados em modelos auto-regressivos. Análises sobre a aplicação dos modelos estocásticos em sensores de navegação inercial são apresentadas por (QUINCHIA et al., 2013; EL-DIASTY; PAGIATAKIS, 2008). Neste trabalho, o modelo usado para propagação dos estados de navegação é fundamentado no trabalho desenvolvido por (INOUE et al., 2017). Nele, foi utilizado o processo de Gauss-Markov de primeira ordem para descrever a dinâmica do estado de erro do bias, tanto do giroscópio, quanto do acelerômetro.

O processo de Gauss-Markov possui uma descrição matemática relativamente simples e, devido a dinâmica de primeira ordem considerada para este caso, é naturalmente inserida no modelo do sistema de navegação descrito no espaço de estados. O modelo contínuo para o processo estocástico do bias dos sensores inerciais é descrito por

$$\dot{y} = -\frac{1}{\tau}y + \mathbf{w}_y, \quad (3.22)$$

sendo  $y$  um processo aleatório de média zero,  $\tau$  o tempo de correlação do processo de Gauss-Markov, e  $\mathbf{w}_y$  um ruído branco de média zero e variância  $\sigma^2$ . Portanto, os modelos de bias do giroscópio e do acelerômetro, propriamente respaldadas em (3.22) para estado de erro, são expressas respectivamente por

$$\boxed{\delta\dot{\mathbf{b}}_g(t) = \Lambda_g\delta\mathbf{b}_g(t) + \mathbf{w}_{b_g}(t)} \quad \text{e} \quad \boxed{\delta\dot{\mathbf{b}}_a(t) = \Lambda_a\delta\mathbf{b}_a(t) + \mathbf{w}_{b_a}(t)}, \quad (3.23)$$

tal que  $\Lambda_g$  e  $\Lambda_a$  são matrizes do tempo de correlação do processo de Gauss-Markov associadas ao giroscópios e acelerômetros, respectivamente.

## 3.2 Sistemas de Coordenadas

Um dos tópicos mais importantes na navegação é a definição do sistema de coordenadas adotado. Embora existam diferentes modos de representar a posição e orientação de um corpo no espaço em aplicações reais, para o desenvolvimento do sistema de navegação aérea serão utilizadas apenas os sistemas de coordenadas que facilitarão a modelagem e integração da IMU, GPS e SLAM visual. Em virtude da aplicação local de abrangência do drone, foram utilizadas coordenadas de plano tangente local. O frame global inercial utiliza a referência *Nort-East-Down* (NED), enquanto o frame dos sensores acoplados no drone utiliza a referência *East-North-Up* (ENU), em razão da disposição dos dispositivos para obtenção de dados. A Figura 9 apresenta os frames local  $\{\mathcal{L}\}$  e global  $\{\mathcal{G}\}$  desta aplicação e a sua relação com o sistema de coordenadas geocêntrico.

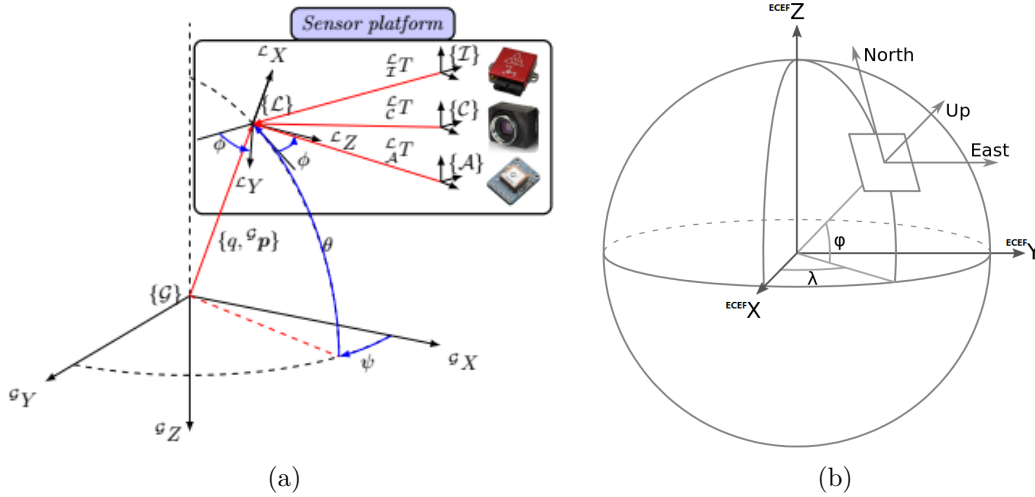


Figura 9 – (a) Sistemas de coordenadas dos sensores e sistemas de coordenadas local e global. Baseado em (INOUE et al., 2017). (b) Sistemas de coordenadas ECEF e ENU para a Terra. Baseado em (WANG; HUYNH; WILLIAMSON, 2013).

As transformações espaciais dos frames do módulo GPS, IMU e câmera são constantes e portanto não afetam os resultados de medidas, pois no processo de calibração essas transformações já são consideradas. A conversão ENU/NED é utilizada simultaneamente no algoritmo de navegação para posicionar as variáveis em um mesmo sistema de referência. O cálculo de conversão é simples, pois envolve apenas a troca de eixos e inserção do valor negativo em  $Z$ . A relação ENU/NED é estabelecida como:

$$\begin{matrix} NED \\ ENU \end{matrix} \mathbf{R} = \begin{matrix} ENU \\ NED \end{matrix} \mathbf{R} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.24)$$

Para obtenção dos dados de velocidade e posição via GPS, é necessário ainda realizar a conversão da coordenada geodésica para *Earth-centered, Earth-fixed* (ECEF) e a conversão de ECEF para NED. Essas conversões são apresentadas com detalhes na Subseção 3.3.2.

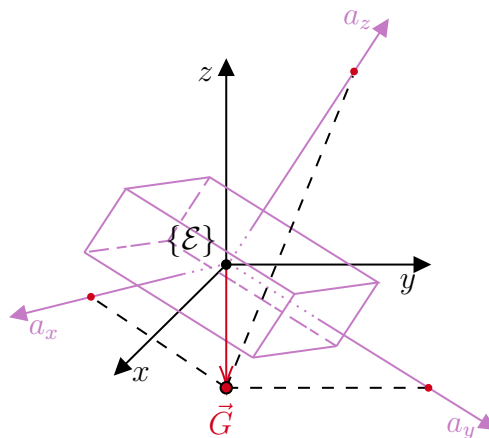
### 3.3 Modelo de Medidas

A estrutura do modelo de medidas observadas em um KF pode assumir múltiplas formas, dependendo dos sensores disponíveis. Dado o tipo de sistema, linear ou não, a modelagem é desenvolvida com objetivo de ser precisa e evitar possíveis falhas, como o *Gimbal Lock*. Muitas vezes, as medidas utilizadas na etapa de correção do KF são derivadas de um algoritmo auxiliar que realiza a fusão de um conjunto de sensores e entrega apenas os dados de interesse para a filtragem. Esse é o caso dos métodos TRIAD (BLACK, 1964) e QUEST (SHUSTER; OH, 1981) para estimativa de atitude. Nos casos em que as informações sensoriais precisam preferencialmente estar desacopladas, pode-se optar pelo modelo de medição independente, como é desenvolvido em FARREL e INOUE. Neste trabalho é utilizada a última abordagem, em razão da disponibilidade e qualidade das informações sensoriais. Nas próximas Subseções são desenvolvidos os modelos de medição para cada sensor apresentado na estrutura de navegação aérea da Figura 5.

#### 3.3.1 Acelerômetro e Magnetômetro para Atitude

Quando o veículo apresenta aceleração linear nula, é possível calcular sua orientação espacial a partir das projeções do campo gravitacional nos acelerômetros ortogonalmente dispostos, assim como apresentado na Figura 10. Entretanto, caso o veículo realize uma rotação em torno no eixo  $z$  do frame  $\mathcal{E}$ , não seria possível calcular a orientação completa apenas com as medidas do acelerômetro, pois para qualquer rotação em  $z$  a projeção anterior se manteria. Para isso, pode-se utilizar outra referência direcional para complementar o cálculo da atitude. Um sensor amplamente utilizado em aplicações de navegação aérea para complementar as informações do acelerômetro é o magnetômetro. O magnetômetro é um instrumento que mede a direção e intensidade de um campo magnético em um determinado local. Em locais com pouca ou nenhuma interferência eletromagnética é possível usar o campo magnético da Terra para orientação.

Figura 10 – Projeção da gravidade nos eixos do acelerômetro.



O desenvolvimento dos cálculos de roll ( $\phi$ ), pitch ( $\theta$ ) e yaw ( $\psi$ ) a partir das medições fornecidas pelo acelerômetro e magnetômetro são baseados na nota de aplicação AN4248, disponível em (OZYAGCILAR, 2012). As leituras dos sensores após três rotações na sequência  $R_z(\psi) - R_y(\theta) - R_x(\phi)$ , são descritas pelas equações:

$$\mathbf{a}_m = R_x(\phi)R_y(\theta)R_z(\psi)G_e = R_x(\phi)R_y(\theta)R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ \mathbf{g}_e \end{bmatrix}, \quad (3.25)$$

$$\mathbf{m}_m = R_x(\phi)R_y(\theta)R_z(\psi)B_e = R_x(\phi)R_y(\theta)R_z(\psi)\mathbf{B}_e \begin{bmatrix} \cos(\alpha) \\ 0 \\ \sin(\alpha) \end{bmatrix}, \quad (3.26)$$

em que  $\mathbf{g}_e$  é o módulo do campo gravitacional terrestre e  $\mathbf{B}_e$  é o módulo do campo magnético local da Terra. O ângulo  $\alpha$  se refere à inclinação direcional do campo magnético da Terra e seu valor varia de acordo com a localização geodésica do veículo. As matrizes de rotação usadas em 3.25 e 3.26 são:

$$\begin{aligned} R_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}, \\ R_y(\theta) &= \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \\ R_z(\psi) &= \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3.27)$$

A matriz  $R_z(\psi)$  em 3.25 não modifica o resultado da equação. Assim, a projeção de  $\mathbf{g}_e$  nos eixos de  $\mathbf{a}_m$  é descrita por

$$\begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ \sin(\phi)\sin(\theta) & \cos(\phi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta) & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \mathbf{g}_e \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (3.28)$$

A solução, independente da magnitude do campo gravitacional local, é:

$$\boxed{\phi = \tan^{-1}\left(\frac{a_y}{a_z}\right) \quad \text{e} \quad \theta = \tan^{-1}\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right)} \quad (3.29)$$

A partir dos cálculos de  $\phi$  e  $\theta$ , e utilizando as medidas do magnetômetro, é desenvolvido o cálculo do ângulo de guinada  $\psi$ . Entretanto, é importante destacar que a direção do campo magnético terrestre não coincide com o norte verdadeiro utilizado nas coordenadas

*NED* e *ENU*. Para isso, é necessário realizar uma correção angular. Esse ajuste angular é executado pela rotação sequencial nos eixos  $z$  e  $y$  do magnetômetro, de forma que

$$\begin{bmatrix} \mathbf{B}_e \\ 0 \\ 0 \end{bmatrix} = \mathbf{R}_y^T(\alpha)\mathbf{R}_z^T(\beta)\mathbf{m}_m. \quad (3.30)$$

O ângulo  $\alpha$  ajusta a inclinação no horizonte e o ângulo  $\beta$  ajusta a declinação do campo magnético. Os valores de  $\alpha$  e  $\beta$  variam de acordo com a latitude e longitude em que a medição é realizada. O mapa apresentado na Figura 11 mostra as curvas de declinação angular de acordo com a localização na Terra. O site<sup>2</sup> da *National Oceanic and Atmospheric Administration* (NOAA) fornece os ângulos de declinação e inclinação do campo magnético local e componentes NED, a partir da posição geodésica e altitude. Por exemplo, para a cidade de São Carlos ( $22^\circ 0' 51'' S$ ;  $47^\circ 53' 26'' W$ ; Altitude:  $854m$ ), as informações do campo magnético em setembro de 2022 são resumidas na Tabela 5.

Figura 11 – Mapa da declinação do campo magnético da Terra (OCEAN; NOAA, 2020).

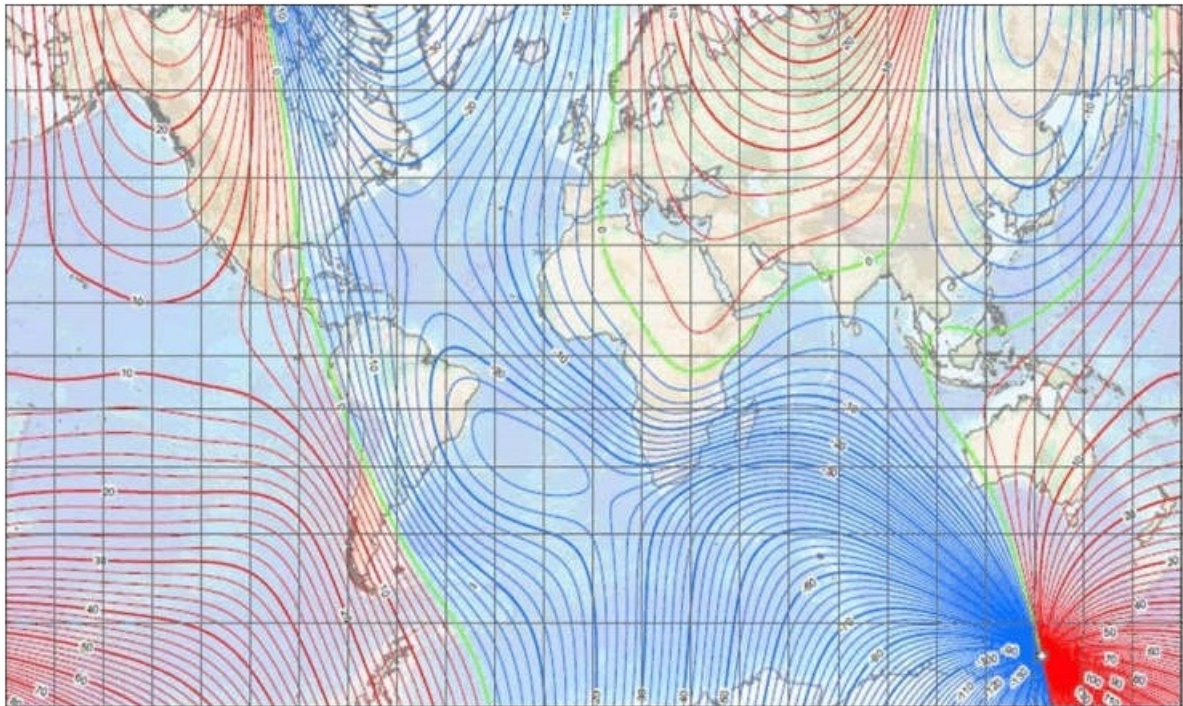


Tabela 5 – Componentes do campo magnético da Terra em São Carlos - SP.

Declinação ( $\overleftarrow{-} \mid \overrightarrow{+}$ )	Inclinação ( $\uparrow + \mid - \downarrow$ )	Componente Norte	Componente Leste	Componente Vertical	Total
$-21^\circ 27' 30''$	$36^\circ 54' 50''$	$17,007mT$	$-6,685mT$	$13,727mT$	$22,855mT$

<sup>2</sup> <<https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml>>

Por meio da equação 3.26 e dos resultados de  $\phi$  e  $\theta$  calculados anteriormente,

$$\mathbf{B}_e \begin{bmatrix} \cos(\psi)\cos(\alpha) \\ -\text{sen}(\psi)\cos(\alpha) \\ \text{sen}(\alpha) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \text{sen}(\theta) \\ 0 & 1 & 0 \\ -\text{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \text{sen}(\phi) \end{bmatrix} (\mathbf{m}_m - \mathbf{V}). \quad (3.31)$$

Em 3.31,  $\mathbf{V}$  se refere ao vetor de *Hard-Iron*. Detalhes para o cálculo de  $\mathbf{V}$ , podem ser consultados na nota de aplicação 4246 (OZYAGCILAR, 2013). No caso deste estudo, o dispositivo utilizado para obtenção dos dados de intensidade do campo magnético já exclui a influência de  $\mathbf{V}$ , pois o magnetômetro passa por um processo prévio de calibração. Esse processo é apresentado no Capítulo 5 deste texto. Portanto, as componentes do campo magnético relacionadas 3.26 tornam-se

$$\mathbf{B}_e \begin{bmatrix} \cos(\psi)\cos(\alpha) \\ -\text{sen}(\psi)\cos(\alpha) \\ \text{sen}(\alpha) \end{bmatrix} = \begin{bmatrix} \mathbf{m}_{m_x}\cos(\theta) + \mathbf{m}_{m_y}\text{sen}(\theta)\text{sen}(\phi) + \mathbf{m}_{m_z}\text{sen}(\theta)\cos(\phi) \\ \mathbf{m}_{m_y}\cos(\phi) - \mathbf{m}_{m_z}\text{sen}(\phi) \\ -\mathbf{m}_{m_x}\text{sen}(\theta) + \mathbf{m}_{m_y}\cos(\theta)\text{sen}(\phi) + \mathbf{m}_{m_z}\cos(\theta)\cos(\phi) \end{bmatrix} \quad (3.32)$$

Assim, pela divisão das componentes  $x$  e  $y$  de 3.32, o ângulo de guinada  $\psi$  é calculado como

$$\psi_m = \tan^{-1} \left( \frac{\mathbf{m}_{m_z}\text{sen}(\phi) - \mathbf{m}_{m_y}\cos(\phi)}{\mathbf{m}_{m_x}\cos(\theta) + \mathbf{m}_{m_x}\text{sen}(\theta)\text{sen}(\phi) + \mathbf{m}_{m_x}\text{sen}(\theta)\cos(\phi)} \right) \quad (3.33)$$

Observe que o ângulo de inclinação  $\alpha$  não interfere no cálculo de  $\psi_m$ . Entretanto, a declinação magnética  $\beta$  deve ser considerada no cálculo pois a equação 3.33 está referente ao norte magnético da Terra. Ou seja:

$$\psi = \psi_m - \beta \quad (3.34)$$

Com o emprego da correção angular de campo magnético, o modelo de medição do magnetômetro é descrito como

$$y_m(t) = {}^G_B \mathbf{R}^T R_z(\beta) R_y(\alpha) {}^G \mathbf{m}_e + \mathbf{v}_m(t). \quad (3.35)$$

A estimativa da intensidade do campo magnético considera o ruído de medida  $\mathbf{v}_m(t)$  um sinal aleatório, tal que  $\mathbb{E}\{\mathbf{v}_m\} = 0$ . Assim, a estimativa das medidas do magnetômetro torna-se

$$\hat{y}_m(t) = {}^G_B \widehat{\mathbf{R}}^T R_z(\beta) R_y(\alpha) {}^G \mathbf{m}_e. \quad (3.36)$$

Similar à definição do estado de erro (3.1), o erro de medição do magnetômetro é calculado a partir de (3.35) e (3.36) como

$$\begin{aligned} \tilde{z}_m(t) &= y_m(t) - \hat{y}_m(t), \\ \tilde{z}_m(t) &= \left( {}^G_B \mathbf{R}^T R_z(\beta) R_y(\alpha) {}^G \mathbf{m}_e + \mathbf{v}_m(t) \right) - \left( {}^G_B \widehat{\mathbf{R}}^T R_z(\beta) R_y(\alpha) {}^G \mathbf{m}_e \right), \\ \tilde{z}_m(t) &= \left( {}^G_B \mathbf{R}^T - {}^G_B \widehat{\mathbf{R}}^T \right) R_z(\beta) R_y(\alpha) {}^G \mathbf{m}_e + \mathbf{v}_m(t). \end{aligned} \quad (3.37)$$



Pelas relações apresentadas na Tabela (4) e Equação (3.15), é possível realizar as seguintes manipulações algébricas em (3.37)

$$\begin{aligned}\tilde{z}_m &= \left[ \left( \frac{\mathcal{G}}{\mathcal{B}} \delta \mathbf{R}^T - I_3 \right) \frac{\mathcal{G}}{\mathcal{B}} \widehat{\mathbf{R}}^T \right] R_z(\beta) R_y(\alpha) \mathcal{G} \mathbf{m}_e + \mathbf{v}_m(t), \\ \tilde{z}_m &= -[\delta\theta \times] \frac{\mathcal{G}}{\mathcal{B}} \widehat{\mathbf{R}}^T R_z(\beta) R_y(\alpha) \mathcal{G} \mathbf{m}_e + \mathbf{v}_m(t).\end{aligned}$$

A matriz de rotação  $\mathbf{R}$  estruturada em termos do quatérnio de atitude propagado, é descrita em (B.26). Devido à propriedade anticomutativa do produto vetorial, o erro de medição do magnetômetro em termos do erro de atitude é definido por

$$\tilde{z}_m = \left[ \left[ \frac{\mathcal{G}}{\mathcal{B}} \mathbf{R}^T(\widehat{\mathbf{q}}(t)) \mathbf{R}_z(\beta) \mathbf{R}_y(\alpha) \mathcal{G} \mathbf{m}_e \right] \times \right] \delta\boldsymbol{\theta}(t) + \mathbf{v}_m(t) \quad (3.38)$$

Semelhante ao processo de modelagem do magnetômetro, o modelo de medição do acelerômetro em tempo contínuo e seu o modelo de estimativa são expressos por

$$\begin{aligned}y_G(t) &= \frac{\mathcal{G}}{\mathcal{B}} \mathbf{R}^{T\mathcal{G}} \mathbf{g}_e + \mathbf{b}_a + \mathbf{v}_a(t), \\ \hat{y}_G(t) &= \frac{\mathcal{G}}{\mathcal{B}} \widehat{\mathbf{R}}^{T\mathcal{G}} \mathbf{g}_e + \hat{\mathbf{b}}_a.\end{aligned} \quad (3.39)$$

O subscrito  $G$  em (3.39) indica que apenas a projeção do campo gravitacional no frame do acelerômetro será usada para indicar a orientação espacial do veículo. Pelo mesmo processo algébrico de (3.37)

$$\begin{aligned}\tilde{z}_G(t) &= y_G(t) - \hat{y}_G(t), \\ \tilde{z}_G(t) &= \left( \frac{\mathcal{G}}{\mathcal{B}} \mathbf{R}^{T\mathcal{G}} \mathbf{g}_e + \mathbf{b}_a(t) + \mathbf{v}_a(t) \right) - \left( \frac{\mathcal{G}}{\mathcal{B}} \widehat{\mathbf{R}}^{T\mathcal{G}} \mathbf{g}_e + \hat{\mathbf{b}}_a(t) \right), \\ \tilde{z}_G(t) &= \left( \frac{\mathcal{G}}{\mathcal{B}} \mathbf{R}^T - \frac{\mathcal{G}}{\mathcal{B}} \widehat{\mathbf{R}}^T \right) \mathcal{G} \mathbf{g}_e + \delta \mathbf{b}_a(t) + \mathbf{v}_a(t), \\ \tilde{z}_G(t) &= -[\delta\theta \times] \frac{\mathcal{G}}{\mathcal{B}} \widehat{\mathbf{R}}^{T\mathcal{G}} \mathbf{g}_e + \delta \mathbf{b}_a(t) + \mathbf{v}_a(t).\end{aligned}$$

Com o produto do erro de atitude  $\delta\theta$  à direita do vetor de estimativas da projeção de  $\mathcal{G} \mathbf{g}_e$ , o erro de medição do acelerômetro se torna

$$\tilde{z}_a = \left[ \left[ \frac{\mathcal{G}}{\mathcal{B}} \mathbf{R}^T(\widehat{\mathbf{q}}(t)) \mathcal{G} \mathbf{g}_e \right] \times \right] \delta\boldsymbol{\theta}(t) + \delta \mathbf{b}_a(t) + \mathbf{v}_a(t) \quad (3.40)$$

As Equações (3.38) e (3.40) são aplicadas diretamente na construção da matriz de medidas usada no KF. Entretanto, as informações sensoriais nem sempre estão disponíveis e a confiabilidade dos sinais pode ser um fator de risco para estimar os estados de navegação. Por exemplo, quando o UAV está em aceleração cinemática, as medições do acelerômetro são tendenciosas e não são práticas para calcular a atitude. O mesmo é válido para interferência eletromagnética que influencia os dados do magnetômetro. Para isso, foram definidos os seguintes critérios para o uso desses sensores:

$$\left| \left( \frac{\|\mathbf{a}\|}{\mathcal{G} \mathbf{g}_e} - 1 \right) \right| < \rho_{acc} \quad \text{and} \quad \left| \left( \frac{\|\mathbf{m}\|}{\mathcal{G} \mathbf{m}_e} - 1 \right) \right| < \rho_{mag}, \quad (3.41)$$

sendo  $\rho$  um valor limiar de confiabilidade do sinal.

### 3.3.2 GPS para Velocidade e Posição

O Sistema de Posicionamento Global (GPS) utiliza o Sistema Geodésico Mundial 1984 (WGS 84) para localização de um ponto próximo à superfície da Terra. O WGS define o sistema de coordenadas utilizado pelo GPS, considerando a Terra como um elipsoide. As informações do WGS são fundamentais para conversão de dados do GPS para uma coordenada local, como NED ou ENU. A Tabela 6 apresenta os valores que caracterizam o elipsoide da Terra.

Tabela 6 – Parâmetros do elipsoide que representa a Terra.

Semieixo Maior	Semieixo Menor	Achatamento Inverso	Excentricidade
6.378.137m	6.356.752,314m	298,257	0,082

O cálculo para obtenção de  $\hat{\mathbf{z}}_p^{INS}$  é apresentado na Seção 3.5 deste Capítulo. A partir da relação (3.1), o resíduo da posição via GPS é determinado como

$$\begin{aligned}\tilde{\mathbf{z}}_p &= \mathbf{z}_p^{GPS} - \hat{\mathbf{z}}_p^{INS}, \\ \tilde{\mathbf{z}}_p &= \left( {}^G\mathbf{p}(t) + \mathbf{v}_p(t) \right) - \left( {}^G\mathbf{p}(t) - {}^G\delta\mathbf{p}(t) \right), \\ \boxed{\tilde{\mathbf{z}}_p^{GPS} = {}^G\delta\mathbf{p}(t) + \mathbf{v}_p(t)} &\end{aligned} \quad (3.42)$$

O desenvolvimento do cálculo do resíduo da velocidade via GPS é similar ao apresentado acima. Logo,

$$\boxed{\tilde{\mathbf{z}}_v^{GPS} = {}^G\delta\mathbf{v}(t) + \mathbf{v}_v(t)} \quad (3.43)$$

### 3.3.3 Barômetro para Altitude

As informações de posição e velocidade obtidas via GPS no eixo z apresentam imprecisão inferior aos eixos x e y. Além disso, a frequência de fornecimento dos dados é um fator que influencia na filtragem para estimativas de posição e velocidade. Por esses motivos, a utilização do barômetro é interessante para aplicação no sistema de navegação aérea. O resíduo de medição do barômetro é calculada fazendo:

$$\begin{aligned}\tilde{\mathbf{z}}_{p_z} &= y_{p_z}^{BAR} - \hat{y}_{p_z}^{INS} \\ \tilde{\mathbf{z}}_{p_z} &= \left( {}^G\mathbf{p}_z(t) + \mathbf{v}_{p_z}(t) \right) - \left( {}^G\mathbf{p}_z(t) - {}^G\delta\mathbf{p}_z(t) \right) \\ \boxed{\tilde{\mathbf{z}}_{p_z}^{BAR} = {}^G\delta\mathbf{p}_z(t) + \mathbf{v}_{p_z}(t)} &\end{aligned} \quad (3.44)$$

Pelos mesmos motivos:

$$\boxed{\tilde{\mathbf{z}}_{v_z}^{BAR} = {}^G\delta\mathbf{v}_z(t) + \mathbf{v}_{v_z}(t)} \quad (3.45)$$

### 3.3.4 SLAM para Atitude e Posição

Para o cálculo da matriz de observação  $H_k$ , é necessário que as medidas possam se relacionar com os estados estimados. Neste sentido, as medidas de atitude do SLAM, que são disponibilizadas pelo quaternião de atitude, deverão ser convertidas em um vetor de ângulos de rotação. Assim, o resíduo da atitude é calculado fazendo:

$$\tilde{\mathbf{z}}_{\delta q} = \hat{\mathbf{q}}^{-1}(t) \otimes \bar{\mathbf{q}}^{SLAM}(t).$$

Depois, o quaternião do erro de atitude  $\tilde{\mathbf{z}}_{\delta q}$  é convertido em  $\tilde{\mathbf{z}}_{\delta\theta}$  usando a aproximação (3.6). Portanto,

$$\boxed{\tilde{\mathbf{z}}_{\theta}^{SLAM} = \delta\boldsymbol{\theta}(t) + \mathbf{v}_{\theta}(t)} \quad (3.46)$$

Agora, o resíduo de medição da atitude por Visual-SLAM poderá ser utilizado para calcular a atualização do estado de erro do KF.

As medidas de posição via Visual-SLAM são fornecidas como deslocamento local em metros. Desse modo, a equação do resíduo de posição a partir dos dados do Visual-SLAM tem o mesmo desenvolvimento de (3.42). Portanto:

$$\boxed{\tilde{\mathbf{z}}_p^{SLAM} = {}^G\delta\mathbf{p}(t) + \mathbf{v}_p(t)} \quad (3.47)$$

## 3.4 Representação no Espaço de Estados

### 3.4.1 Equação de estados

De acordo com as Equações (3.10), (3.20), (3.21) e (3.23), o modelo de propagação dos estados de erro (3.2) em tempo contínuo é estruturado como

$$\dot{\delta\mathbf{x}}(t) = F(t)\delta\mathbf{x}(t) + H(t)\mathbf{w}(t), \quad (3.48)$$

com as matrizes  $F(t)$  e  $H(t)$ :

$$F(t) = \begin{bmatrix} \left[ -[y_g(t) - \hat{b}_g(t)] \times \right] & -I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \Lambda_g & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ -{}^G\mathbf{R}(\hat{\mathbf{q}}(t)) \left[ [y_a(t) - \hat{b}_a(t)] \times \right] & 0_{3 \times 3} & 0_{3 \times 3} & -{}^G\mathbf{R}(\hat{\mathbf{q}}(t)) & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \Lambda_a & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad (3.49)$$

$$H(t) = \begin{bmatrix} -I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad (3.50)$$

e o vetor de estados de erro  $\delta\mathbf{x}$  definido em (3.2). Os ruídos das leituras dos sensores inerciais e dos biases associados à (3.48) por  $H(t)$ , são indicados pelo vetor  $\mathbf{w} = [\mathbf{w}_g^T \quad \mathbf{w}_{b_g}^T \quad \mathbf{w}_a^T \quad \mathbf{w}_{b_a}^T]^T$ .

### 3.4.2 Equação de Medidas

A equação de medidas para o KF dependerá do conjunto de sensores com dados disponíveis. Como o barômetro pode ser utilizado individualmente ou como complemento dos dados do GPS para estimativas de posição e velocidade no eixo z, 32 combinações de sensores entre as Equações (3.38), (3.40), (3.42–3.47) são aplicadas no cálculo do resíduo de medidas no KF.

### 3.4.3 Discretização

A implementação do KF em um sistema embarcado ou para fins de simulação exige que o sistema dinâmico esteja representado no tempo discreto. Os métodos de integração são indicados de acordo com a precisão exigida e capacidade computacional para implementação numérica. Em alguns casos, será possível usar soluções exatas de forma fechada. Em outros casos, pode ser empregada integração numérica de vários graus de precisão. Para mais detalhes, consulte (SOLA, 2017), (MIDDLETON; GOODWIN, 1990) e (CHEN, 1984). Neste estudo, as matrizes em tempo contínuo (3.49), (3.50) da equação dinâmica do sistema e a matriz  $C(t)$  da equação de medidas foram aproximadas considerando um tempo de amostragem  $\Delta t$ . Assim,

$$\mathcal{F}_k \simeq I + F\Delta t, \quad \mathcal{H}_k \simeq H\Delta t^{\frac{1}{2}} \quad \text{e} \quad \mathcal{C}_k = \frac{\partial \mathbf{z}(\delta\mathbf{x})}{\partial \delta\mathbf{x}}. \quad (3.51)$$

No caso de  $\mathcal{H}_k$ , a computação de  $\sqrt{\Delta t}$  pode não ser necessária. A matriz discreta dos ruídos de processo  $\mathcal{H}_k$  é utilizada na propagação da covariância no KF, como será evidenciada na Subseção 3.5.2. O cálculo de  $P_{k+1|k}$  envolve a operação  $\mathcal{H}_k Q_k \mathcal{H}_k^T$ , não havendo perdas caso seja substituída por  $H Q_k H^T \Delta t$ .

## 3.5 Filtro de Kalman para Navegação

O desenvolvimento matemático reproduzido até a Seção anterior deste Capítulo focou na descrição e modelagem dos sensores para aplicação no KF. Entretanto, pela estrutura de filtragem baseada no estado de erro, é preciso analisar e entender os outros passos do algoritmo de estimativas de estados. A Figura 6a apresenta de maneira mais nítida essa estrutura. Os estados estimados pelo INS são propagados e usados para atualização do modelo dinâmico do sistema. Os mesmos valores também são aplicados nos cálculos de predição das medidas sensoriais. A diferença entre as medidas observadas e as medidas preditas são aplicadas ao KF, gerando os estados de erro. Ao final, os valores filtrados são

somados ao estado propagado anteriormente. Nas próximas Subseções são apresentadas as equações de propagação e atualização usadas para estimar os estados de navegação aérea.

### 3.5.1 Propagação dos Estados

Os biases propagados assumem os mesmos valores do ciclo anterior:

$$\begin{aligned}\widehat{\mathbf{b}}_{\omega,k+1|k} &= \widehat{\mathbf{b}}_{\omega,k|k}, \\ \widehat{\mathbf{b}}_{\mathbf{a},k+1|k} &= \widehat{\mathbf{b}}_{\mathbf{a},k|k}.\end{aligned}\quad (3.52)$$

A partir das equações dos biases atualizados, as medições da IMU são corrigidas como:

$$\widehat{\boldsymbol{\omega}}_{k+1|k} = \boldsymbol{\omega}_{k+1} - \widehat{\mathbf{b}}_{\omega,k+1|k}, \quad (3.53)$$

$$\widehat{\mathbf{a}}_{k+1|k} = \widehat{\mathbf{a}}_{k+1} - \widehat{\mathbf{b}}_{\mathbf{a},k+1|k}. \quad (3.54)$$

Conforme demonstrado em (SOLA, 2017), a integração de ordem zero pela convenção quaterniônica de Hamilton tem a forma:

$$\widehat{\mathbf{q}}_{k+1} = \left( I_4 \cos\Theta + \Omega(\widehat{\boldsymbol{\omega}}_{k+1|k}) \frac{\sin\Theta}{\|\boldsymbol{\omega}\|} \right) \otimes \widehat{\mathbf{q}}_k, \quad (3.55)$$

sendo

$$\Theta = \|\widehat{\boldsymbol{\omega}}_{k+1|k}\| \Delta t / 2 \quad \text{e} \quad \Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -[\boldsymbol{\omega} \times] \end{bmatrix}.$$

A velocidade e a posição propagadas são obtidas por:

$${}^{\mathcal{G}}\dot{\widehat{\mathbf{v}}}_{k+1|k} = {}^{\mathcal{G}}\widehat{R} \widehat{\mathbf{a}}_{k+1|k} + {}^{\mathcal{G}}\mathbf{g}_e, \quad (3.56)$$

$${}^{\mathcal{G}}\widehat{\mathbf{v}}_{k+1|k} = {}^{\mathcal{G}}\widehat{\mathbf{v}}_{k|k} + {}^{\mathcal{G}}\dot{\widehat{\mathbf{v}}}_{k+1|k} \Delta t, \quad (3.57)$$

$${}^{\mathcal{G}}\widehat{\mathbf{p}}_{k+1|k} = {}^{\mathcal{G}}\widehat{\mathbf{p}}_{k|k} + {}^{\mathcal{G}}\widehat{\mathbf{v}}_{k+1|k} \Delta t + \frac{{}^{\mathcal{G}}\dot{\widehat{\mathbf{v}}}_{k+1|k} \Delta t^2}{2}, \quad (3.58)$$

em que  ${}^{\mathcal{G}}\mathbf{g}_e$  é o vetor de gravidade  ${}^{\mathcal{G}}\mathbf{g}_e = [0 \quad 0 \quad |\mathbf{g}_e|]^T$ .

### 3.5.2 Propagação da Covariância

A estimativa *a priori* da matriz de covariância  $P$  é calculada assim como foi apresentada na Figura 2. Ou seja:

$$P_{k+1,k} = \mathcal{F}_k P_{k|k} \mathcal{F}_k^T + \mathcal{H}_k Q_K \mathcal{H}_k^T \quad (3.59)$$

### 3.5.3 Predição das Medidas Sensoriais

O resíduo de medidas  $\delta \mathbf{z}_k$  é calculado usando o mesmo princípio (3.1) que define o erro de estado. Os valores fornecidos pelos sensores auxiliares são confrontados com os dados propagados do INS para a computação de  $\delta \mathbf{x}_k$ . Os valores de  $\hat{\mathbf{z}}$  são apresentados nas equações a seguir:

$$\hat{\mathbf{z}}_{\theta}^{MAG} = {}^{\mathcal{G}}\mathbf{R}(\hat{q}) R_z(\beta) R_y(\alpha) \mathbf{m}_e \quad (3.60)$$

$$\hat{\mathbf{z}}_{\theta}^{ACC} = -{}^{\mathcal{G}}\mathbf{R}(\hat{q}) \mathbf{g}_e \quad (3.61)$$

$$\hat{\mathbf{z}}_{\mathbf{v}}^{GPS} = {}^{\mathcal{G}}\hat{\mathbf{v}}_{k+1|k} \quad (3.62)$$

$$\hat{\mathbf{z}}_{\mathbf{v}_z}^{BAR} = {}^{\mathcal{G}}\hat{\mathbf{v}}_{z,k+1|k} \quad (3.63)$$

$$\hat{\mathbf{z}}_{\mathbf{p}}^{GPS} = {}^{\mathcal{G}}\hat{\mathbf{p}}_{k+1|k} \quad (3.64)$$

$$\hat{\mathbf{z}}_{\mathbf{p}_z}^{BAR} = {}^{\mathcal{G}}\hat{\mathbf{p}}_{z,k+1|k} \quad (3.65)$$

$$\hat{\mathbf{z}}_{\theta}^{SLAM} = [0 \ 0 \ 0]^T \quad (3.66)$$

$$\hat{\mathbf{z}}_{\mathbf{p}}^{SLAM} = {}^{\mathcal{G}}\hat{\mathbf{p}}_{k+1|k} \quad (3.67)$$

Algumas observações importantes acerca das equações acima: 1) O valor negativo em (3.61) é pelo fato do vetor  $\mathbf{g}_e$  estar na coordenada NED e as medidas da IMU estarem na coordenada ENU. O mesmo não é válido para as medidas do magnetômetro, que são fornecidas na coordenada NED. 2) As estimativas (3.62–3.65) e (3.67) são obtidas diretamente a partir das Equações (3.57) e (3.58). 3) Excepcionalmente para o caso da estimativa do erro de atitude via SLAM, o resíduo  $\hat{\mathbf{z}}_{\theta}^{SLAM}$  é obtido por (3.46). Por isso (3.66) é nulo.

### 3.5.4 Atualização dos Estados

A última etapa para o cálculo das estimativas dos estados de navegação aérea é a atualização do estado completo. Aqui, entende-se como estado completo as estimativas de atitude, velocidade, posição e biases representados pelo vetor de estados

$$\mathbf{x} = \begin{bmatrix} q \\ \mathbf{b}_g \\ {}^{\mathcal{G}}\mathbf{v} \\ \mathbf{b}_a \\ {}^{\mathcal{G}}\mathbf{p} \end{bmatrix}. \quad (3.68)$$

O processo de filtragem realiza as estimativas do estado de erro  $\delta \mathbf{x} \in \mathbb{R}^{15 \times 1}$ , como mostrado em (3.2). Entretanto, a atitude em  $\mathbf{x}$  é representada pelo quatérnio  $q$ , que ocasiona o aumento de uma dimensão, tal que  $\mathbf{x} \in \mathbb{R}^{16 \times 1}$ . Os motivos dessa escolha foram abordados na Subseção 3.1.2.

Pela aproximação estabelecida em (3.6), a estimativa atualizada do quatérnio de atitude é

$$\hat{q}_{k+1|k+1} = \delta \hat{q}_{k+1|k+1} \otimes \hat{q}_{k+1|k}, \quad (3.69)$$

sendo

$$\delta \hat{q}_{k+1|k+1} = \begin{cases} \begin{bmatrix} \sqrt{1 - \|\delta\boldsymbol{\theta}\|^2} \\ \delta\boldsymbol{\theta} \end{bmatrix}, & \text{se } \|\delta\boldsymbol{\theta}\|^2 \leq 1 \\ \frac{1}{\sqrt{1 + \|\delta\boldsymbol{\theta}\|^2}} \begin{bmatrix} 1 \\ \delta\boldsymbol{\theta} \end{bmatrix}, & \text{se } \|\delta\boldsymbol{\theta}\|^2 > 1 \end{cases}$$

Os demais estados de navegação podem ser atualizados diretamente com a soma do estado propagado anteriormente e o estado de erro obtido na filtragem:

$$\begin{aligned} \hat{\mathbf{b}}_{\omega,k+1|k+1} &= \hat{\mathbf{b}}_{\omega,k+1|k} + \delta \hat{\mathbf{b}}_{\omega}, \\ \mathcal{G} \hat{\mathbf{v}}_{k+1|k+1} &= \mathcal{G} \hat{\mathbf{v}}_{k+1|k} + \mathcal{G} \delta \hat{\mathbf{v}}, \\ \hat{\mathbf{b}}_{\mathbf{a},k+1|k+1} &= \hat{\mathbf{b}}_{\mathbf{a},k+1|k} + \delta \hat{\mathbf{b}}_{\mathbf{a}}, \\ \mathcal{G} \hat{\mathbf{p}}_{k+1|k+1} &= \mathcal{G} \hat{\mathbf{p}}_{k+1|k} + \mathcal{G} \delta \hat{\mathbf{p}}. \end{aligned} \quad (3.70)$$

### 3.5.5 Algoritmo

O desenvolvimento matemático expresso neste Capítulo abrangeu todas as etapas do processo para elaboração do sistema de navegação aérea baseado no KF indireto. Para sintetizar esse conteúdo, o Algoritmo 1 apresenta as operações utilizadas em cada ciclo do processo. As simulações implementadas neste estudo basearam-se na seguinte rotina:

---

#### Algoritmo 1 Estimativa de estados de navegação

---

1: Resete a estimativa de erro de estado  $\delta \hat{\mathbf{x}}_{k|k}$ ;

#### Propagação

2: Atualize as medições da IMU usando (3.53) e (3.54);

3: Propague o quatérnio de atitude usando (3.55);

4: Propague a velocidade e a posição usando (3.56), (3.57) e (3.58);

5: Atualize as matrizes  $\mathcal{F}_k$  e  $\mathcal{H}_k$  da representação em espaço de estados, de acordo com as equações (3.51), (3.49) e (3.50);

6: Propague a matriz de covariância  $P_{k+1|k}$  usando (3.59);

#### Filtragem

7: Determine o modo de observação, de acordo com a disponibilidade e restrições definidas em (3.41);

8: A partir das medições observadas, atualize a matriz  $\mathcal{C}_k$  and  $R_k$ ;

9: Calcule os resíduos de medição  $\delta \mathbf{z}$  pela diferença das medidas  $\mathbf{z}$  e os estados propagados (3.60–3.67), dependendo dos resultados da etapa 7;

10: Aplique a etapa de atualização da filtragem de Kalman para estimar  $P_{k+1|k+1}$  e  $\delta \hat{\mathbf{x}}$  a partir do KF padrão (Quadro I) ou KF robusto (Quadro II);

#### Atualização do Estado Completo $\hat{\mathbf{x}}_{k+1|k+1}$

11: Atualize o quatérnio de atitude de acordo com (3.69);

12: Atualize os biases, velocidade e posição de acordo com (3.70).

---





## Parte III

# Aspectos Metodológicos



---

# Capítulo 4

## Systolic Array

---

No âmbito de sistemas embarcados, a tecnologia baseada em microcontroladores ainda é a principal escolha computacional para implementação em equipamentos de baixo custo, como sistema de iluminação, eletrodomésticos e alguns equipamentos industriais. Entretanto, para aplicações embarcadas de alto desempenho computacional, a utilização de tecnologias alternativas, como *Digital Signal Processor* (DSP)s, FPGAs e GPUs de uso geral, tem se tornado uma boa opção para aplicações em tempo real. Segundo estudo realizado por (ASPENCORE, 2019), 24% das empresas do setor de sistemas embarcados utilizam FPGA como ferramenta de projetos. Já no caso das GPUs de uso geral, o rápido crescimento do poder de processamento em comparação com as CPUs, tornou a computação acelerada por GPU uma opção viável para muitas aplicações (ÖFVERSTEDT; LINDBLAD; SLADOJE, 2021).

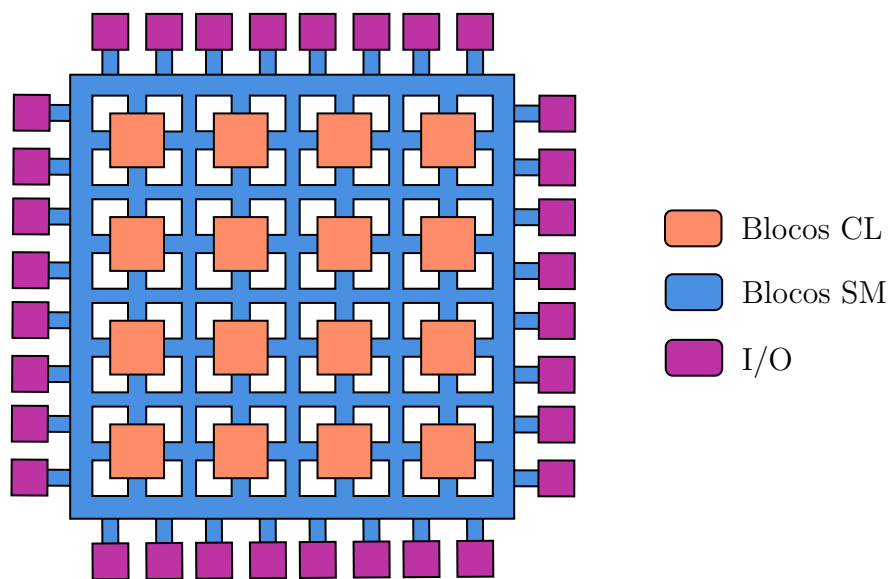
Abordagens heterogêneas envolvendo as plataformas computacionais supracitadas tem motivado uma quantidade significativa de pesquisas sobre técnicas de interação computacional (ZENG; PRASANNA, 2020; GUZMÁN et al., 2019). O uso em conjunto e controlado permite o sistema embarcado usufruir o melhor que cada dispositivo pode disponibilizar. Pra citar algumas vantagens dessa abordagem, a concepção de projetos heterogêneos para sistemas embarcados poderá abranger: a versatilidade de programação das CPUs, o alto desempenho energético dos FPGAs, o poder de processamento gráfico das GPUs e a combinação ideal entre desempenho e consumo energético das ASICs.

Em geral, o objetivo do uso de diferentes instrumentos computacionais é reparticionar o sistema para atribuir funções especializadas a cada um deles. Essa estratégia permite melhorar o desempenho de sistemas com alta demanda computacional, mas, ao mesmo tempo, requer que o uso dos recursos de hardware seja feito de forma organizada e mais simples possível (ANDRADE; CRNKOVIC, 2018).

## 4.1 FPGA

Dentre os dispositivos apresentados na introdução deste capítulo, o FPGA se destaca pela flexibilidade de ajuste à funções específicas, mediante reconfiguração estática ou dinâmica, juntamente com a capacidade de processamento paralelo. Especificamente, um FPGA é um dispositivo semicondutor baseado em uma matriz de blocos lógicos configuráveis conectados por meio de interconexões programáveis (XILINX, 2021). A Figura 12 apresenta a estrutura básica de um FPGA. Seus componentes são: blocos *Configurable Logic* (CL), blocos de entrada e saída (I/O) e *Switch Matrix* (SM).

Figura 12 – Estrutura básica do FPGA.



**Blocos CL:** Malha lógica disposta na forma bidimensional, formando módulos funcionais de circuitos, a partir de conjuntos de *flip-flops* com lógica combinacional. Pode ser configurada para executar funções combinatórias de diferentes níveis de complexidade.

**Blocos SM:** Trilhas de conexão entre os blocos I/O e os blocos lógicos configuráveis, além da interconexão de blocos CL.

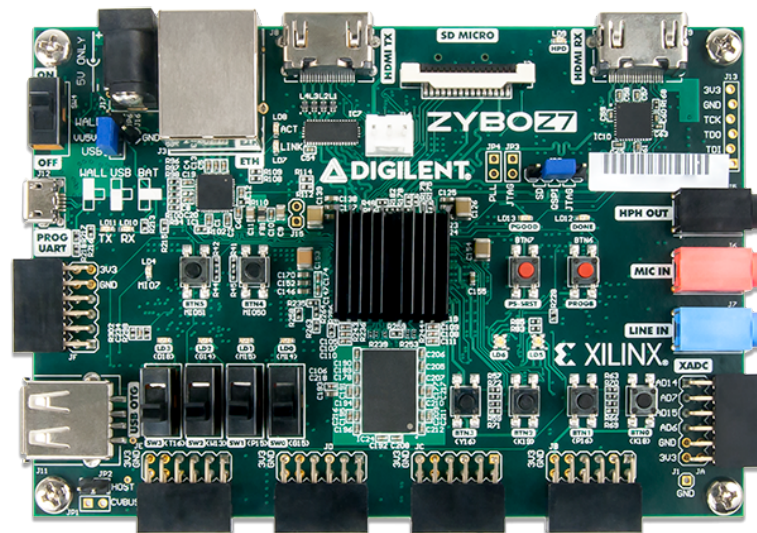
**Blocos I/O:** São pontos onde são realizadas as conexões dos dados de entrada como os resultados dos blocos lógicos configuráveis.

A implementação em FPGA é desenvolvida por Linguagens de Descrição de Hardware (HDLs) para representar a estrutura e comportamento de circuitos eletrônicos digitais. Dentre as linguagens mais utilizadas se destacam o VHDL e o Verilog. É possível também construir circuitos por meio do desenvolvimento de diagramas esquemáticos, utilizando ferramentas gráficas assistidas por computador. A arquitetura flexível desenvolvida em HDL permite a elaboração de sistemas complexos, com aplicações em diversos campos. As mais comuns abrangem os setores aeroespacial, defesa, sistemas monetários distribuídos, computação de alto desempenho e instrumentos médicos e científicos.

### 4.1.1 Dispositivos *System-on-Chip*

Atualmente, muitos fabricantes oferecem dispositivos *System-on-Chip* (SoC) programáveis, os quais geralmente combinam CPU, FPGA e interfaces periféricas. A popularização deste tipo de ferramenta de desenvolvimento está ligada diretamente à difusão de projetos “*do it yourself*” e redução dos custos de fabricação do dispositivos. Por exemplo, a fabricante Xilinx oferece a família *Zynq* de SoCs que combinam processadores ARM com FPGA. Neste trabalho, o desenvolvimento do sistema de navegação utilizou a placa de desenvolvimento *Digilent Zybo z7-20*, mostrada na Figura 13, com detalhes técnicos apresentados na Tabela 7. Esse dispositivo baseia-se em um circuito digital fundamentado na família Xilinx *Zynq<sup>TM</sup>-7000*. O objetivo do dispositivo voltado para sistemas embarcados é a distribuição de recursos, com processamentos de alta carga sendo executados em *hardware* FPGA, e tarefas que exigem maior flexibilidade sendo executadas via software pelo processador.

Figura 13 – Placa de desenvolvimento Zybo Z7-20.



Fonte: <<https://digilent.com/>>.

Tabela 7 – Recursos da Zybo z7-20.

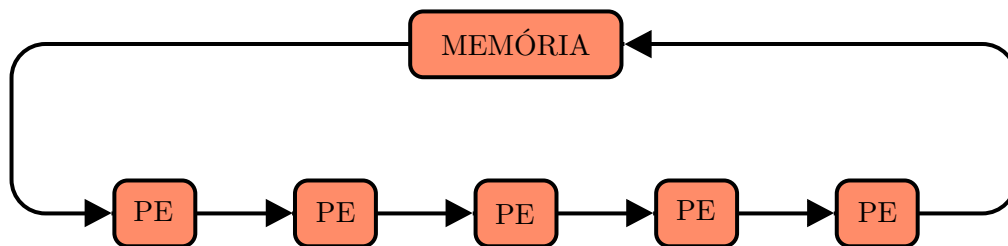
FPGA	série 7 Xilinx	FPGA I/O	40
Processador	dual-core Cortex-A9	Frequência	667Mhz
LUTs	53.200	Flip-Flops	106.400
Memória	1GB DDR3 32 bits	Flash	Quad-SPI de 16 MB
Fonte	5V ou USB	Potência	12, 5Watts
Interfaces	6 <i>push-buttons</i> ; 4 <i>switches</i> ; 7 <i>LEDs</i>	Áudio	fone estéreo; entrada de linha; microfone
Expansão e Conexão	6 portas Pmod; Ethernet; USB	Vídeo	Portas HDMI entrada/saída

## 4.2 Systolic Array

Proposto por (KUNG; LEISERSON, 1979), o *systolic array* é definido como um conjunto de *Processing Element* (PE)s, normalmente organizados em uma rede bidimensional, que opera em etapas de cálculo e armazenamento com a vizinhança (KUNG, 1982; LU, Abril de 2017). O processo compreende ações de *Multiplier-ACcumulator* (MAC) para executar operações aritméticas, convolução, integração, interpolação, entre outras.

A denominação de *systolic array* deriva em analogia ao sistema circulatório de um ser vivo, em que o sangue flui ritmicamente através de um coração biológico (NULL; LOBUR, 2014). Nesse caso, o coração equivale a memória e a rede de artérias e veias equivale a matriz de processadores e conexões. Assim, os elementos de dados não são usados apenas quando são inseridos, mas também reutilizados conforme se movem pela rede de PEs, como exemplificado na Figura 14, em que é apresentado o processamento em *systolic array* unidimensional.

Figura 14 – *Systolic array* unidimensional.



A complexidade de implementação de circuitos baseados em *systolic array* está vinculada à diversos fatores relacionados, como o conjunto de operações necessárias no projeto, o perfil dos dados e características de *hardware*. Apesar dessa abordagem viabilizar ao projetista a execução embarcada de diversos métodos numéricos, o procedimento exige uma extensa análise acerca de como as MACs irão gerar resultado e conhecimento de implementação em *hardware*. Alguns métodos implementação *systolic array* para soluções de álgebra linear podem ser consultados em (BAYOUMI; LING, 1999; H.; LANG, 1992). No caso deste trabalho, o objetivo é acelerar o processo de inversão matricial para solução de um KF robusto. Desta forma, a inversão matricial baseada em *systolic arrays* será apresentada na próxima Subseção.

### 4.2.1 Inversão Matricial

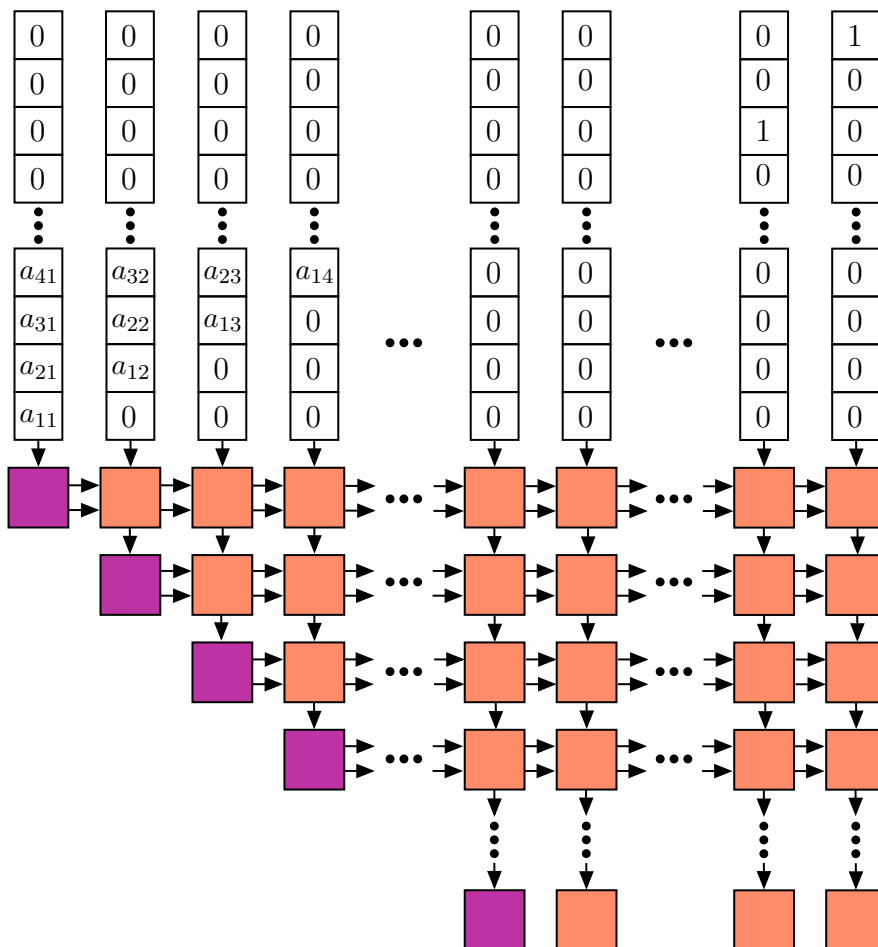
O processo de inversão matricial utilizando a estrutura *systolic array* é baseada na decomposição  $QR$  da matriz a ser invertida. Um dos métodos utilizados pra obtenção dos fatores  $QR$  de uma matriz arbitrária envolve transformações no espaço  $\mathbb{R}^{n \times n}$  da matriz, de forma que os elementos abaixo da diagonal principal sejam anulados. Esse processo é conhecido como rotação de Givens, assunto tratado no Apêndice A desta dissertação.

Adicionando um grau de liberdade no sistema apresentado na Figura 14, a estratégia de fatoração consiste em anular os elementos abaixo da diagonal principal da matriz em um processo paralelo e sincronizado. Para isso, dois tipos de PEs são utilizados:

1. **PE Diagonal:** É responsável por gerar os senos e cossenos que transformam os elementos da linha correspondente. Este PE recebe apenas elementos referentes à coluna compatível e, utilizando o seu estado interno, calcula a norma entre esses valores. O resultado é usado na atualização do estado interno  $S_m$  e cálculo dos senos e cossenos que serão transmitidos à PE Interna adjacente. Antes do início do processo, é atribuído o valor  $S_m = 0$  ao estado interno.
2. **PE Interno:** É responsável por atualizar o estado interno  $S_{nm}$  e conduzir os elementos da coluna correspondente à PE inferior subsequente. Também propaga os valores dos senos e cossenos para a PE à sua direita, sem alterar o valor original. Assim como na PE diagonal, é atribuído o valor  $S_{nm} = 0$  ao estado interno antes do início do processo.

A estrutura *systolic array* que resulta na decomposição da matriz  $A$  em fatores  $R$  e  $Q^{-1}$  é apresentada na Figura 15. O circuito dos PEs é mostrado na Figura 16.

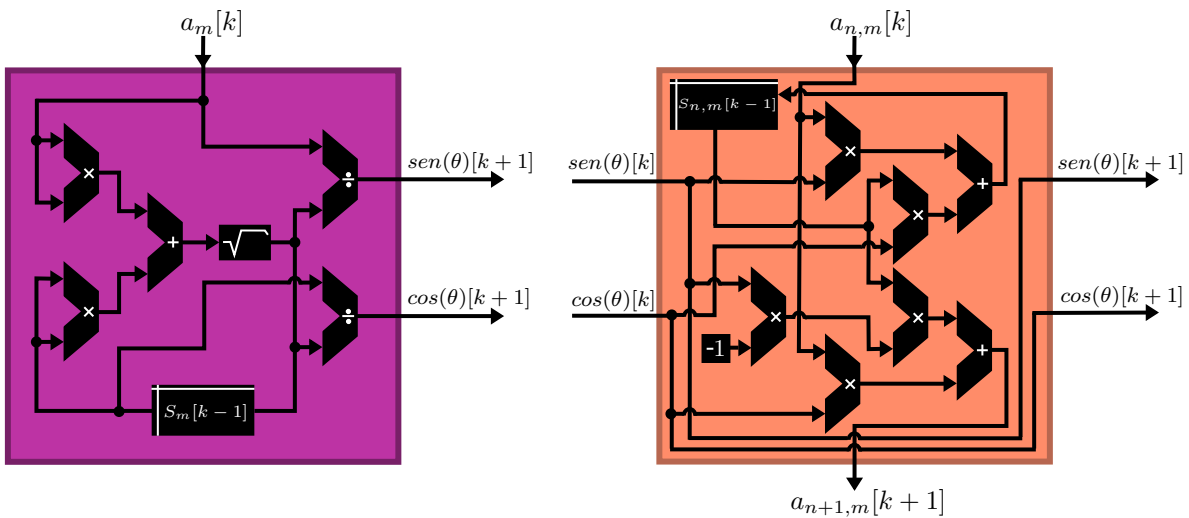
Figura 15 – Estrutura *systolic array* para inversão da matrix  $A$ .



A Figura 15 mostra a disposição dos elementos na malha baseada em *systolic array*. O posicionamento adequado é fundamental para o funcionamento do processo, pois a computação paralela depende da sincronia perfeita entre todos os elementos.

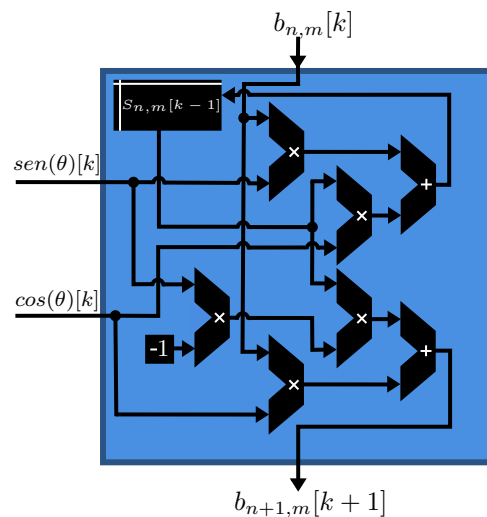
Para obtenção de  $A^{-1}$ , a inversa de  $R$  ainda precisa ser calculada e o resultado multiplicado por  $Q^{-1}$ . No entanto, o objetivo para o emprego de *systolic array* neste trabalho é a solução de um sistema linear na forma  $Ax = b$  e, para fins didáticos, a aplicação de uma substituição reversa ao final do processo pode ser mais intuitiva neste caso.

Figura 16 – Elementos de processamento diagonal e interno, respectivamente.



No caso da solução de  $Ax = b$ , é mais interessante que o resultado gerado no processo seja a matriz  $R$  e o vetor  $bQ^{-1}$ . Assim, a organização à direita da Figura 15 recebe o vetor de saídas  $b$  do sistema linear. Para se adequar a esse arranjo, um novo PE é utilizado. O PE de saídas, mostrado na Figura 17, tem o mesmo circuito do PE interno, porém sem transmissão dos senos e cossenos para a direita.

Figura 17 – Elemento de processamento de saída





### 4.3 Exemplo Analítico

**Exemplo 4.3.1** Considere a solução linear  $x = A^{-1}b$ , tal que  $A$  e  $b$  sejam

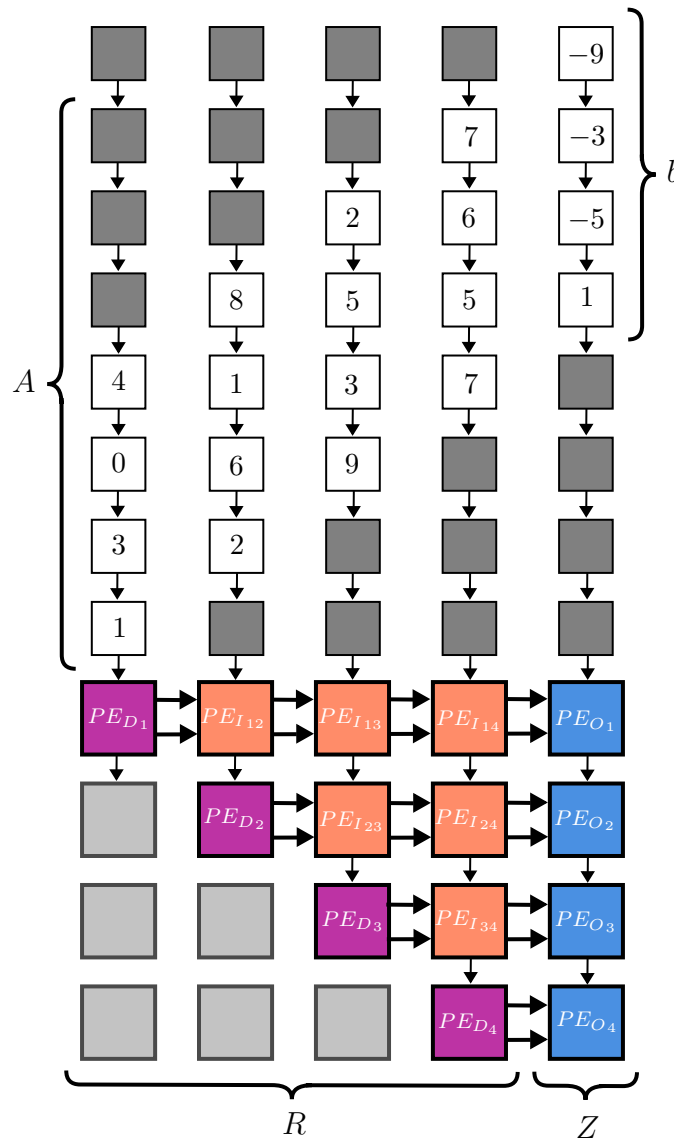
$$A = \begin{bmatrix} 1 & 2 & 9 & 7 \\ 3 & 6 & 3 & 5 \\ 0 & 1 & 5 & 6 \\ 4 & 8 & 2 & 7 \end{bmatrix} \quad e \quad b = \begin{bmatrix} 1 \\ -5 \\ -3 \\ -9 \end{bmatrix}.$$

A equação linear possui o mesmo padrão do algoritmo recursivo da Tabela ???. Desse modo, o procedimento usado neste exemplo, também pode ser aplicado ao referido algoritmo.

#### Solução

O sistema linear pode ser reescrito na forma direta  $Ax = b$ , de modo que seja possível usar a estrutura 15 baseada em systolic array para triangularizar a matriz  $A$ .

Figura 18 – Estrutura *systolic array* para triangularização da matriz  $A$ .



A atualização dos estados de cada PE ocorre simultaneamente com o movimento descendente dos elementos de  $A$  e  $b$  mostrado na Figura 18. Ao final de 11 etapas, os estados se estabelecem na forma  $Rx = z$ , sendo que  $R$  é a matriz triangular superior e  $z$  é um vetor proveniente de  $b$ . As tabelas 8, 9 e 10 mostram os estados das PEs em  $k$ .

Tabela 8 – Estados dos PEs diagonais da Figura 18.

$k$	$PE_{D_1}$			$PE_{D_2}$			$PE_{D_3}$			$PE_{D_4}$		
	$S_m$	$sen(\theta)$	$cos(\theta)$	$S_m$	$sen(\theta)$	$cos(\theta)$	$S_m$	$sen(\theta)$	$cos(\theta)$	$S_m$	$sen(\theta)$	$cos(\theta)$
0	–	–	–	–	–	–	–	–	–	–	–	–
1	1	1	0	–	–	–	–	–	–	–	–	–
2	3,16	0,95	0,32	–	–	–	–	–	–	–	–	–
3	3,16	0	1	0	0*	1*	–	–	–	–	–	–
4	5,10	0,78	0,62	0	0*	1*	–	–	–	–	–	–
5	5,10	0	1	1	1	0	0	0*	1*	–	–	–
6	5,10	0	1	1	0	1	7,59	–1	0	–	–	–
7	5,10	0	1	1	0	1	7,59	0	1	0	0*	1*
8	5,10	0	1	1	0	1	8,25	–0,39	0,92	0	0*	1*
9	5,10	0	1	1	0	1	8,25	0	1	0	0*	1*
10	5,10	0	1	1	0	1	8,25	0	1	0,95	1	0
11	5,10	0	1	1	0	1	8,25	0	1	0,95	0	1

Tabela 9 – Estados dos PEs internos da Figura 18.

$k$	$PE_{I_{12}}$		$PE_{I_{13}}$		$PE_{I_{14}}$		$PE_{I_{23}}$		$PE_{I_{24}}$		$PE_{I_{34}}$	
	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$
0	–	–	–	–	–	–	–	–	–	–	–	–
1	–	–	–	–	–	–	–	–	–	–	–	–
2	2	0	–	–	–	–	–	–	–	–	–	–
3	6,33	0	9	0	–	–	–	–	–	–	–	–
4	6,33	1	5,69	–7,59	7	0	0	0	–	–	–	–
5	10,20	0	5,69	5	6,96	–5,06	0	–7,59	0	0	–	–
6	10,20	0	5,10	–3,23	6,96	6	5	0	0	–5,06	0	0
7	10,20	0	5,10	0	9,81	–1,12	5	–3,23	6	0	0	0
8	10,20	0	5,10	0	9,81	0	5	0	6	–1,12	0	0
9	10,20	0	5,10	0	9,81	0	5	0	6	0	5,09	0,96
10	10,20	0	5,10	0	9,81	0	5	0	6	0	5,09	0
11	10,20	0	5,10	0	9,81	0	5	0	6	0	5,09	0

\*Quando o cálculo de senos e cossenos apresentam, em ambos, a operação  $\frac{0}{0}$ , é atribuído os valores  $cos(\theta) = 1$  e  $sen(\theta) = 0$  para execução da *systolic array*. Geometricamente isso significa que não ocorre transformação espacial na matriz  $A$ .

Tabela 10 – Estados dos PEs de saída da Figura 18.

$k$	$PEO_1$		$PEO_2$		$PEO_3$		$PEO_4$	
	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$	$S_m$	$a_{nm}$
0	–	–	–	–	–	–	–	–
1	–	–	–	–	–	–	–	–
2	–	–	–	–	–	–	–	–
3	–	–	–	–	–	–	–	–
4	–	–	–	–	–	–	–	–
5	1	0	–	–	–	–	–	–
6	–4, 43	–2, 53	0	0	–	–	–	–
7	–4, 43	–3	0	–2, 53	0	0	–	–
8	–9, 81	–2, 11	–3	0	2, 53	0	0	–
9	–9, 81	0	–3	–2, 11	2, 53	0	0	–
10	–9, 81	0	–3	0	3, 15	–0, 95	0	–
11	–9, 81	0	–3	0	3, 15	0	–0, 95	–

Assim, o sistema linear se torna um problema de substituição reversa

$$\begin{bmatrix} \frac{26}{\sqrt{26}} & \frac{52}{\sqrt{26}} & \frac{26}{\sqrt{26}} & \frac{50}{\sqrt{26}} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & \frac{34}{\sqrt{17}} & \frac{21}{\sqrt{17}} \\ 0 & 0 & 0 & \frac{20}{\sqrt{442}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -\frac{50}{\sqrt{26}} \\ -3 \\ \frac{13}{\sqrt{17}} \\ -\frac{20}{\sqrt{442}} \end{bmatrix},$$

com a solução  $x$ :

$$x = \begin{bmatrix} 3 \\ -2 \\ 1 \\ -1 \end{bmatrix}.$$

**Observação:** Neste exemplo foi mostrado o desenvolvimento algébrico para solução do sistema linear na forma  $x = A^{-1}b$ , baseando-se na estrutura de systolic arrays com os PEs realizando a triangularização matricial via rotação de Givens. O exemplo 2.1 do Apêndice A realiza a decomposição QR da mesma matriz  $A$  deste exemplo. Verifique a matriz  $R$  resultante de ambos exemplos são iguais. Como a matriz  $Q$  do exemplo A.2.1 é ortogonal, o vetor  $z$  deste exemplo é equivalente ao cálculo

$$z = \begin{bmatrix} \frac{1}{\sqrt{26}} & 0 & \frac{4}{\sqrt{17}} & \frac{3}{\sqrt{442}} \\ \frac{3}{\sqrt{26}} & 0 & 0 & -\frac{17}{\sqrt{442}} \\ 0 & 1 & 0 & 0 \\ \frac{4}{\sqrt{26}} & 0 & -\frac{1}{\sqrt{17}} & \frac{12}{\sqrt{442}} \end{bmatrix}^T \begin{bmatrix} 1 \\ -2 \\ -3 \\ -9 \end{bmatrix}$$



---

## Capítulo 5

# Sensoriamento, Aquisição de Dados e Simulação

---

Nos Capítulos anteriores foram apresentados os contextos da pesquisa, no que se referem às revisões teóricas que fundamentaram o escopo deste estudo e trabalhos relacionados. Para validação da proposta de aceleração computacional do KF robusto aplicado ao sistema de navegação aérea, neste Capítulo são apresentadas as metodologias da pesquisa, as quais envolvem simulações e testes de campo.

Primeiramente, foram realizadas simulações computacionais em ambiente Matlab/Simulink do sistema de navegação aérea. Para isso, foram usados registros de voo do piloto automático de um UAV obtido em Zürich-Suíça. O conjunto de dados é um exemplo do aplicativo Matlab Flight Log Analyzer (MATHWORKS, ). A definição dos parâmetros  $\mu$  e  $\xi$  associados à função de penalidade do KF robusto, assim como as incertezas paramétricas nos modelos do sistema, foram ajustadas de acordo com a resposta obtida e o *ground truth* fornecido pelo banco de dados.

A segunda parte da metodologia deste estudo consiste na aplicação do mesma simulação da primeira parte, porém, com dados coletados em campo. Como não existe uma forma compatível de comparação dos resultados obtidos, os parâmetros do KF robusto usados na primeira parte dos testes foram mantidos.

### 5.1 Simulação

A Figura 19 apresenta a organização da simulação. Seguindo a sequência da rotina mostrada no Algoritmo 1, A etapa 10 é calculada pelo circuito desenvolvido no Simulink. As demais etapas do algoritmo são calculadas no Script Matlab. A disposição dessa

simulação caracteriza o conjunto CPU-FPGA para a computação embarcada do sistema de navegação aérea. A Figura 20 mostra um fragmento do circuito desenvolvido em ambiente Simulink.

Figura 19 – Estrutura de simulação do algoritmo do Quadro II.

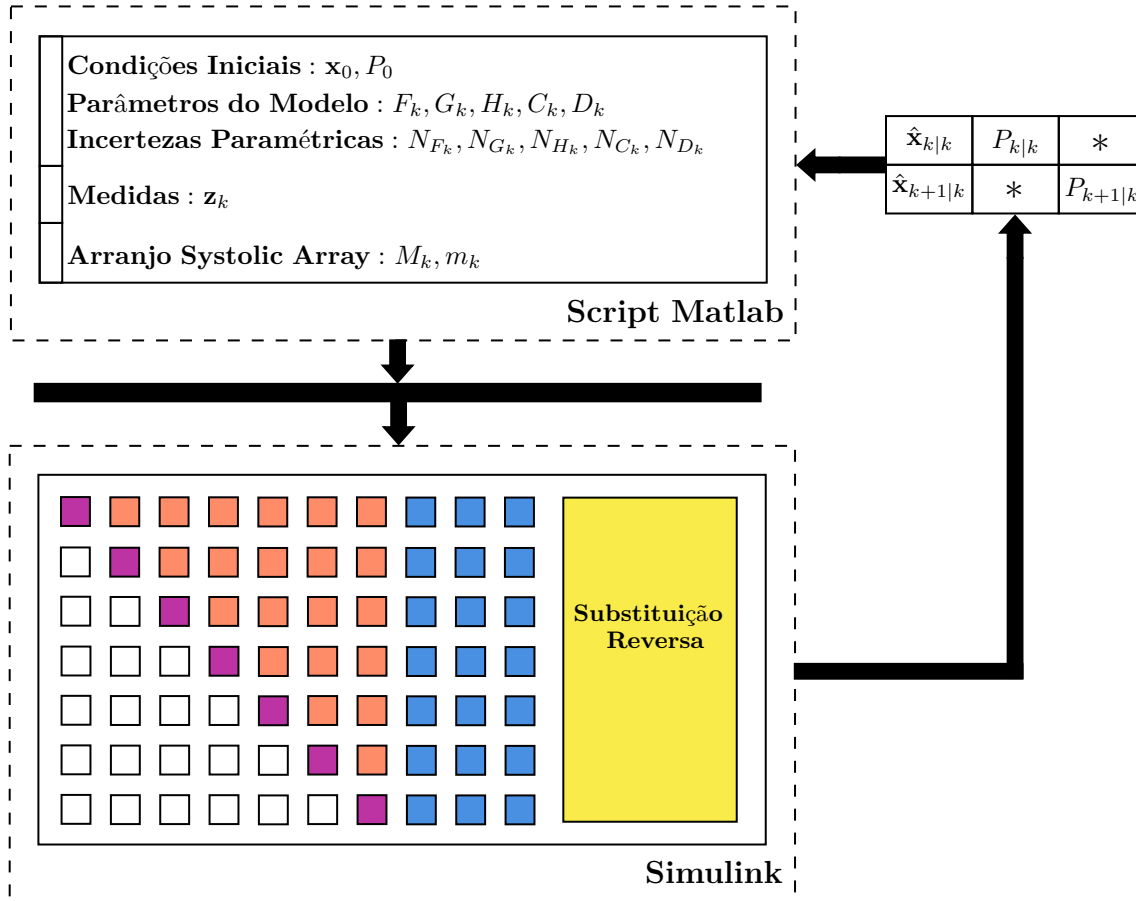
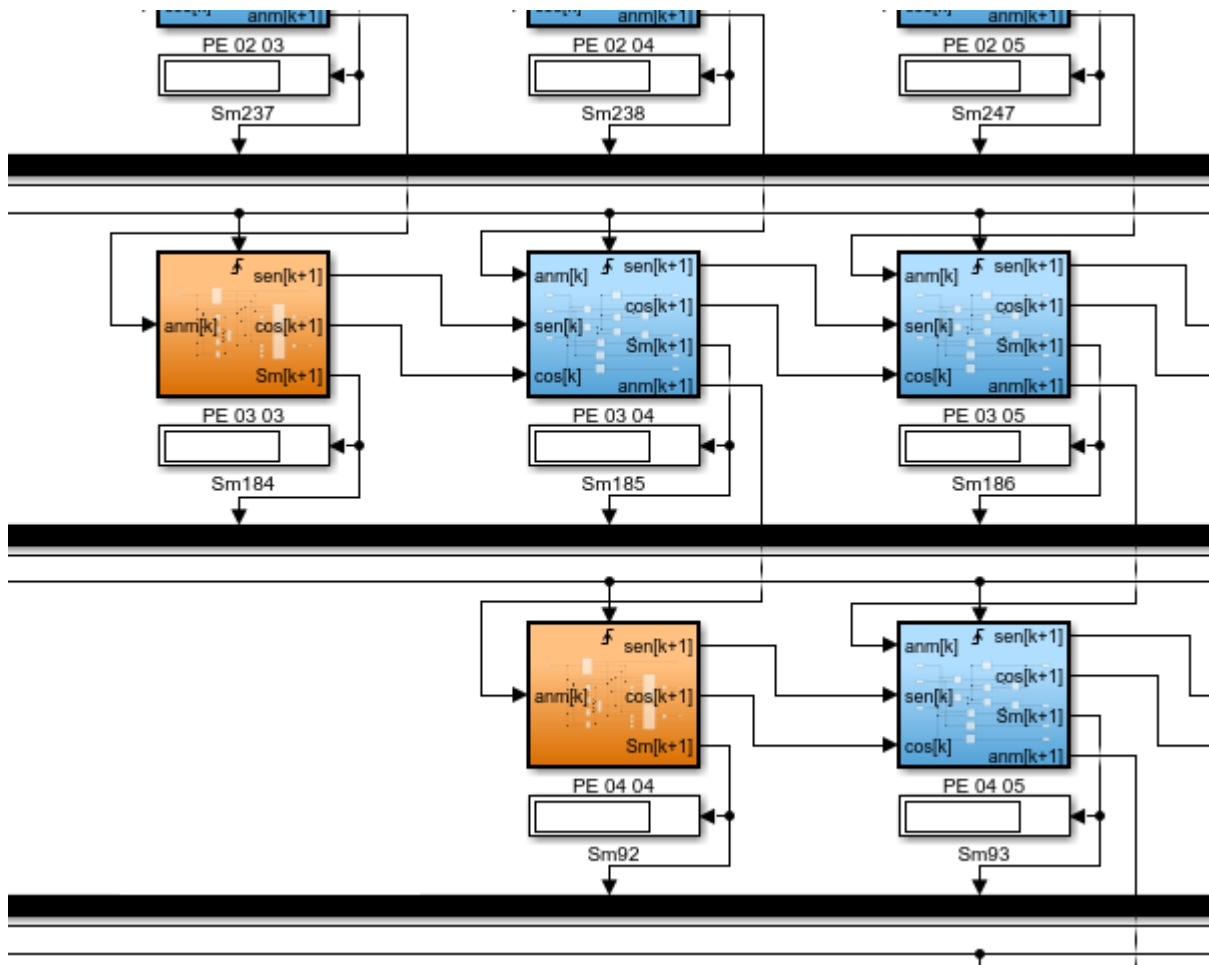
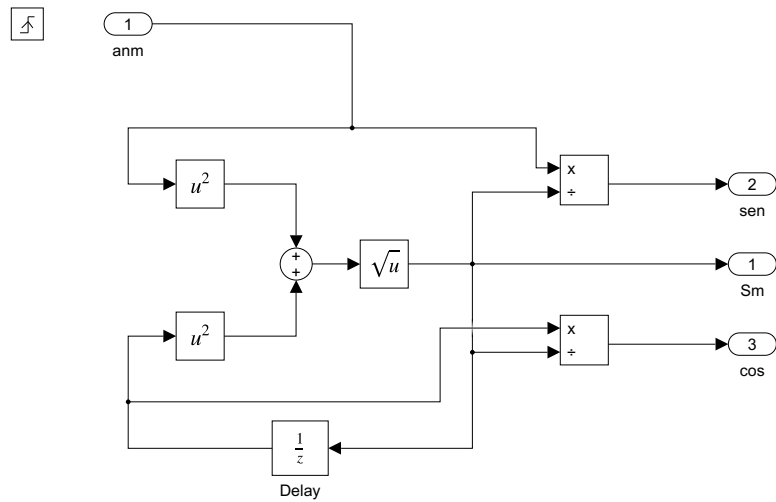
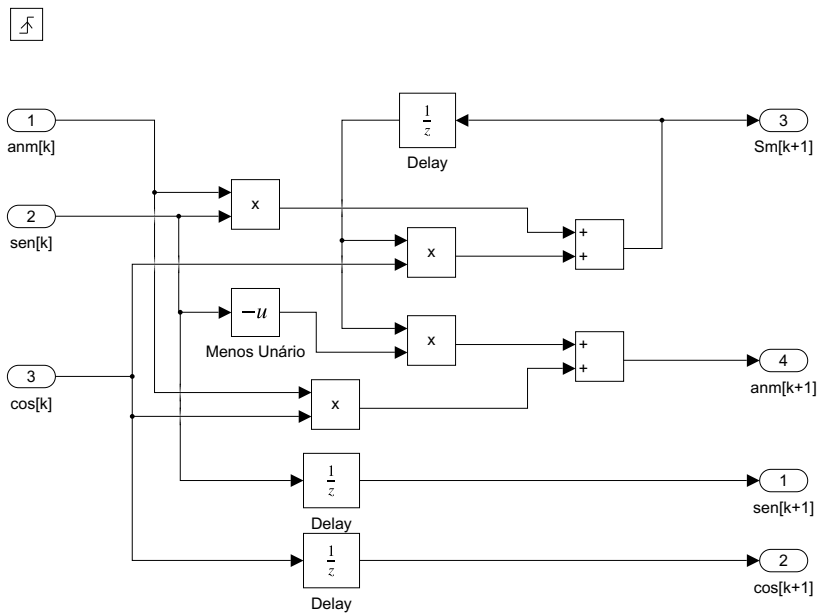


Figura 20 – Fragmento do circuito desenvolvido no Simulink para o cálculo da solução de  $Ax = b$  usando a estrutura de *systolic arrays* baseadas na rotação de Givens.

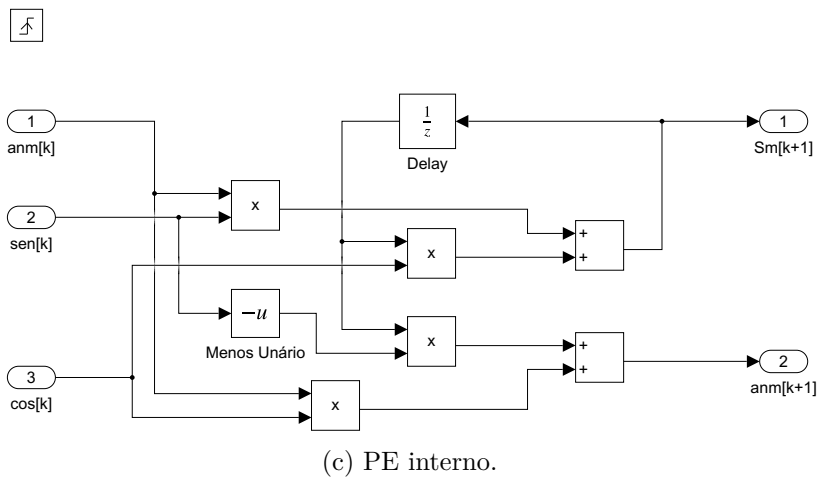




(a) PE diagonal.



(b) PE interno.



(c) PE interno.

Figura 21 – Circuitos dos PEs construídos no Simulink.

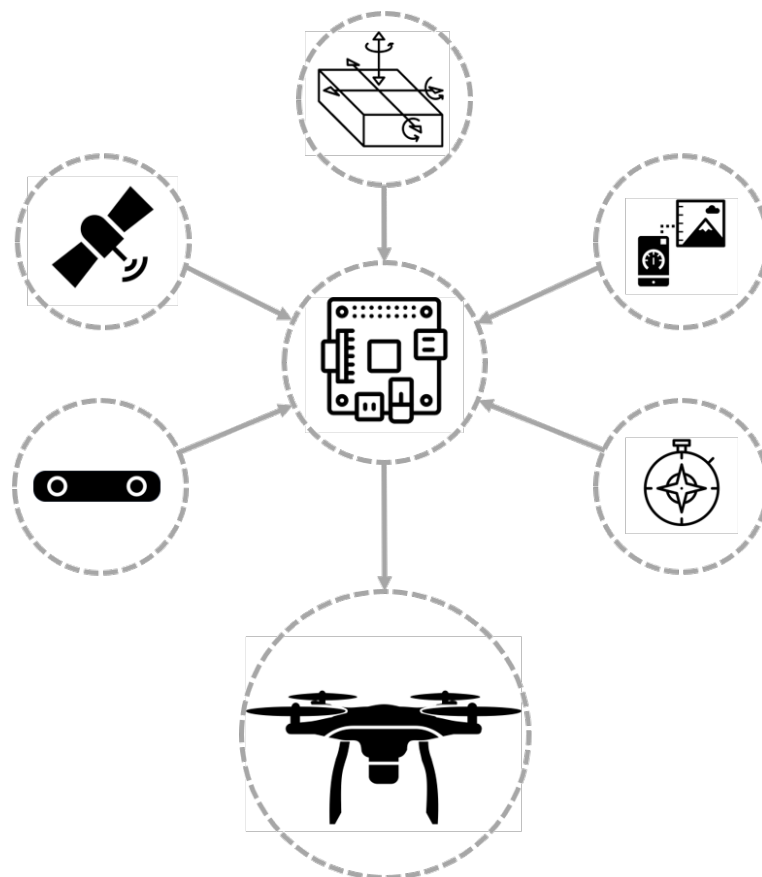


## 5.2 Voo de Teste

Stereo Labs. 2018. Vive Pro AR (SRWorks) vs. ZED Mini. Stereo Labs. Retrieved 2020-12-22 from <https://www.stereolabs.com/blog/vive-pro-ar-zed-mini/> (CROCKETT et al., 2014)

A Figura 22 mostra o diagrama de sensoriamento para o voo de teste. Nesse teste foram empregues dois conjuntos de sensores: o módulo Xsens MTi-G-710 que integra um dispositivo de INS com magnetômetro e GNSS, e a câmera estéreo StereoLabs modelo ZED mini. Os sensores foram conectados ao kit de desenvolvimento Nvidia Jetson Xavier NX via conexão USB.

Figura 22 – Diagrama do sensoriamento para voo de teste.



### 5.2.1 Materiais

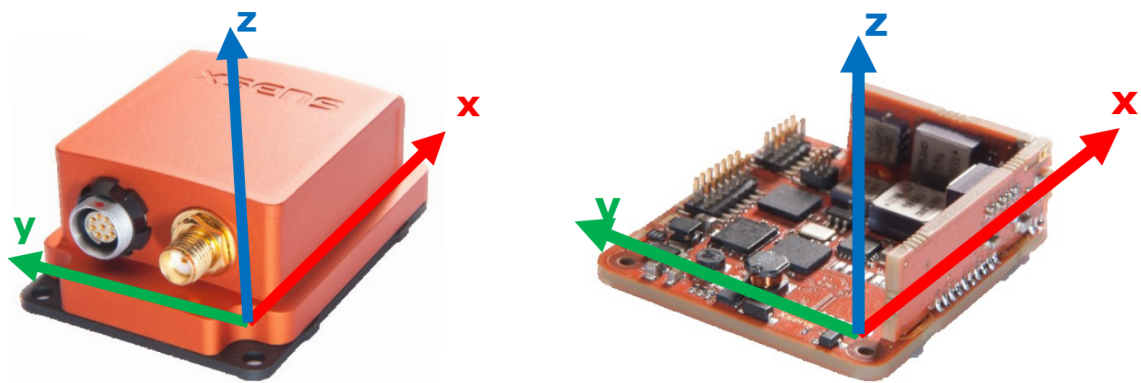
Os equipamentos utilizados no voo de teste são descritos a seguir.

Figura 23 – UAV modelo DJI Matrice 600 Pro.



Figura 24 – IMU com um receptor GNSS integrado Xsens modelo MTI-G-710 acoplada no UAV.





(a) Sistema de coordenadas do invólucro do dispositivo.

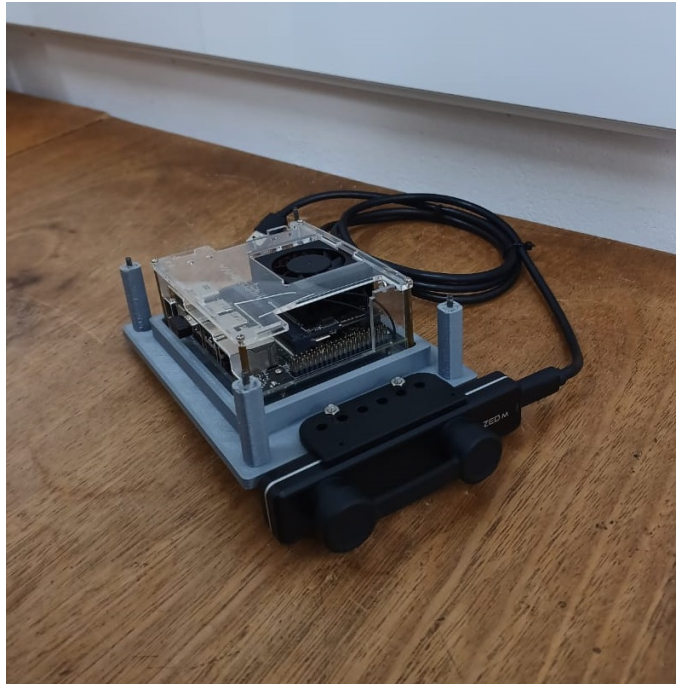
(b) Sistema de coordenadas do circuito interno.

Figura 25 – Arranjo dos frames do INS e magnetômetros. Nota: a origem do frame está localizada nos acelerômetros.

Figura 26 – Câmera estéreo Stereolabs modelo ZED Mini acoplada ao UAV.



Figura 27 – Kit desenvolvedor Jetson Xavier NX.



### 5.2.2 Suporte ROS

(Stanford Artificial Intelligence Laboratory et al., )

Segundo o próprio desenvolvedor, o ROS<sup>1</sup> é um conjunto de bibliotecas de software e ferramentas que ajudam a criar aplicativos de robô de código aberto. Para coleta de dados foram utilizados os seguintes tópicos dos pacotes associados à Xsens MTI-G-710 e à câmera estéreo StereoLabs ZED mini.

Xsens:

- /filter/positionlla
- /filter/quaternion
- /filter/velocity
- /imu/acceleration
- /imu/angular\_velocity
- /imu/mag
- /imu/pressure
- /imu/temperature
- /tf

ZED Mini:

- /zedm/zed\_node/odom
- /zedm/zed\_node/pose

---

<sup>1</sup> <https://www.ros.org/>

## Parte IV

# Resultados e Conclusão



---

## Capítulo 6

## Resultados

---

O conjunto de dados do aplicativo Matlab Flight Log Analyzer (MATHWORKS, ) com os dados de Zürich são mostrados a seguir.

Figura 28 – Velocidade Angular.

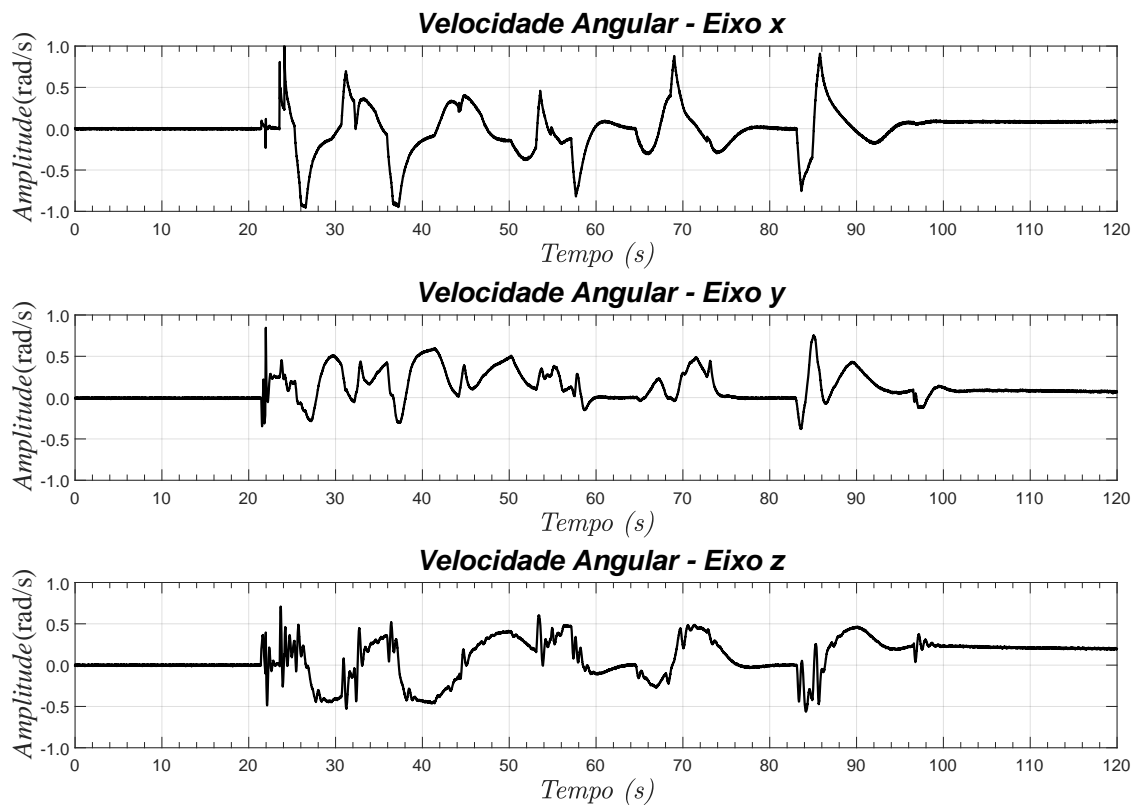


Figura 29 – Aceleração Linear.

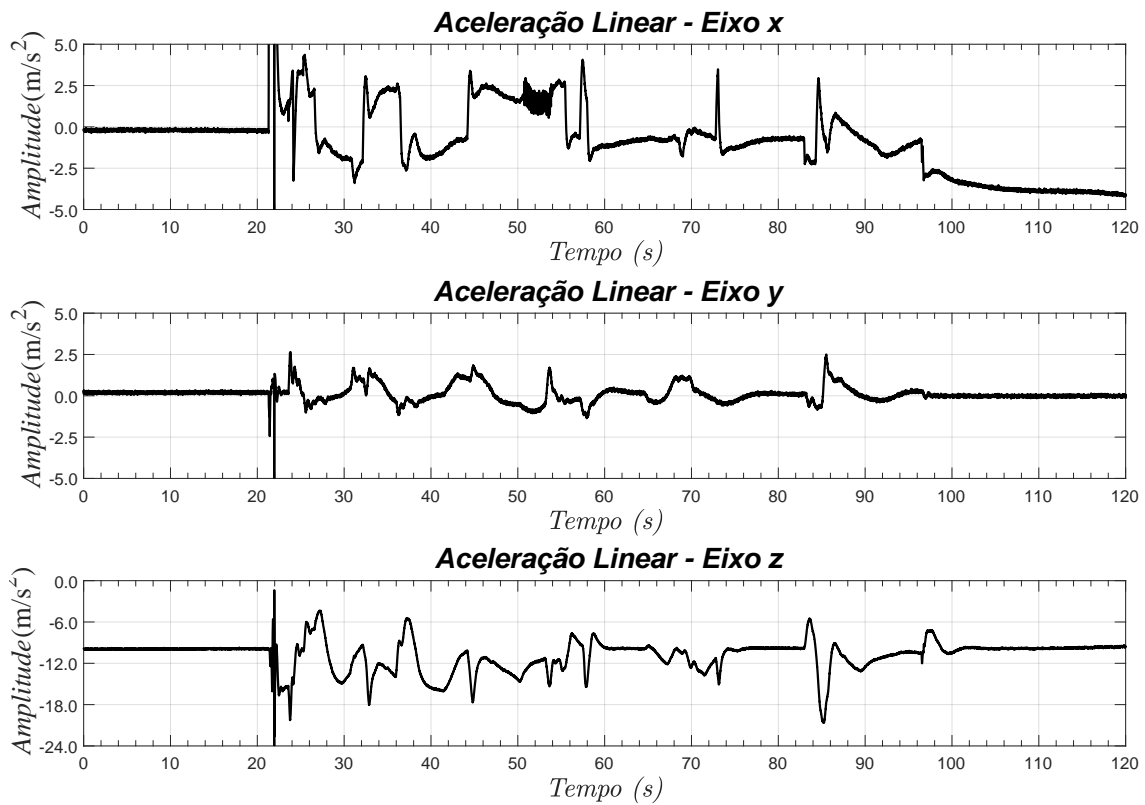
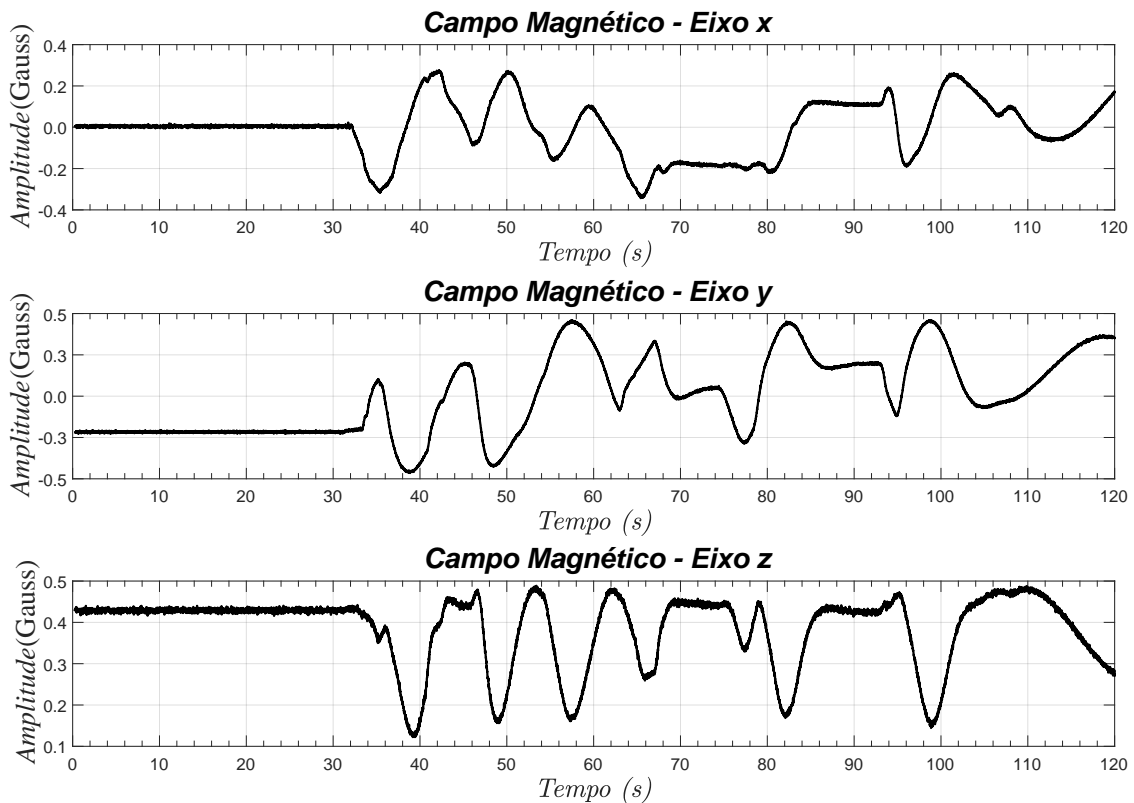


Figura 30 – Intensidade do Campo Magnético.



Os parâmetros KF robustos usados para esta simulação estão contidos na Tabela 11. A filtragem ocorre a uma frequência de  $250Hz$ , de acordo com dados da IMU.



Tabela 11 – Parâmetros usados na simulação.

Parâmetro	Valor	Parâmetro	Valor
$\Xi_0$	$\mathbf{diag}_n(0.01)$	$\xi$	1.7
$\Pi_0$	$0_{n \times 1}$	$\mu$	55000
$\alpha^*$	$64.24^\circ$	$\Delta\boldsymbol{\omega}$	$0.48 \text{ rad/s}$
$\beta^*$	$-2.56^\circ$	$\Delta\mathbf{a}$	$0.02 \text{ m/s}^2$
$\rho_{acc}$	$0.05 \text{ m/s}^2$	$\Delta\Lambda_{\boldsymbol{\omega}}$	$0.02 \text{ s}^{-1}$
$\rho_{mag}$	$0.08 \text{ Gauss}$	$\Delta\Lambda_{\mathbf{a}}$	$0.01 \text{ s}^{-1}$

\*Os valores de inclinação e declinação magnética em Zürich foram obtidos no site: <<https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml>>

A matriz de incerteza  $\Delta F_k$ , apresentada em (2.3), assume a estrutura da matriz (6.1) para que  $M_1$  e  $E_F$  apresentem valores condizentes com o algoritmo mostrado no Quadro II.

$$M_1 E_F = \begin{bmatrix} -[\Delta\boldsymbol{\omega} \times] & 0 & 0 & 0 & 0 \\ 0 & \mathbf{diag}(\Delta\Lambda_{b_{\boldsymbol{\omega}}}) & 0 & 0 & 0 \\ -\hat{R}[\Delta\mathbf{a} \times] & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{diag}(\Delta\Lambda_{b_{\mathbf{a}}}) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.1)$$

Em (6.1),  $\Delta\boldsymbol{\omega}$  e  $\Delta\mathbf{a}$  representam as incertezas das medições do giroscópio e do acelerômetro, respectivamente.  $\Delta\Lambda_{b_{\boldsymbol{\omega}}}$  e  $\Delta\Lambda_{b_{\mathbf{a}}}$  representam as incertezas de bias da velocidade angular e da aceleração linear, respectivamente.

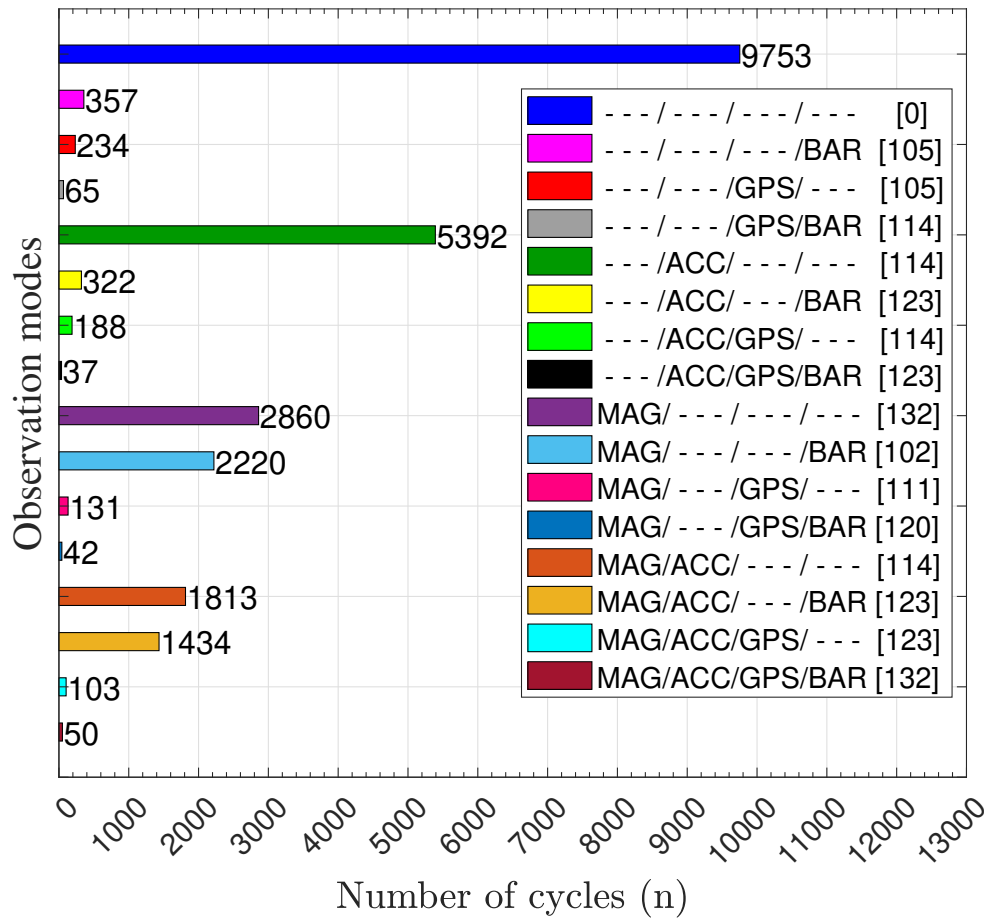
A dimensão do sistema linear mostrado no Quadro II depende diretamente da informação sensorial disponível durante a leitura. Para a simulação, a incerteza do modelo foi aplicada apenas à matriz de estado  $F_k$ . Portanto, a linha e a coluna contendo  $\hat{\lambda}^{-1}I_{t_2}$  foram excluídas. A equação que determina a dimensão do sistema linear é

$$\dim = 4n + 2(r_1 + r_2) + p + t_1 + t_2,$$

sendo  $n$  a dimensão do vetor de estado;  $r_1$  e  $r_2$  são as quantidades de ruído de processo e medição, respectivamente;  $p$  é o número de medições;  $t_1$  e  $t_2$  são os números das colunas de  $M_1$  e  $M_2$ .

Como o custo computacional está diretamente ligado a dimensão do sistema, realizamos uma investigação sobre os conjuntos de sensores utilizados em cada ciclo. A Fig. 31 apresenta os resultados desta análise. Foi verificado que em quase 40% dos ciclos, nenhuma filtragem é realizada, apenas propagação de estado. Quando ocorre a filtragem, verifica-se a dimensão máxima do sistema linear Box II quando é utilizado o conjunto de sensores MAG/GPS/ACC/BAR ou MAG/GPS/ACC.

Figura 31 – Quantidade de ciclos da simulação para cada conjunto de sensores utilizados.



A Tabela 12 mostra um estudo comparativo da precisão do KF e do KF robusto. Inicialmente, testamos valores de  $\mu$ ,  $\xi$ ,  $\Delta\omega$  e  $\Delta\Lambda_\omega$  que minimizassem o erro de estimativa de atitude. A partir dos valores fixos, ajustamos os demais parâmetros do KF robusto  $\Delta\mathbf{a}$  e  $\Delta\Lambda_{\mathbf{a}}$ . É possível notar a melhora nas estimativas de atitude usando o KF robusto. Entretanto, algumas estimativas de posição e velocidade apresentaram resultados piores, ainda que, em números absolutos, essa piora represente uma diferença de milímetros. Isso ocorre porque os mesmos parâmetros  $\mu$  e  $\xi$  são usados no KF robusto para estimar todos os estados simultaneamente.

O desempenho da estimativa de atitude usando o RKF, em comparação com o KF, é comprovado visualmente pela Fig. 32 durante os 100s de simulação. Quanto à estimativa de posição, é impossível notar visualmente distinções entre os resultados robustos do KF e a posição real, conforme mostrado na trajetória do UAV no plano Nordeste da Fig. 33.

Também foi investigado os recursos necessários para implementar a solução numérica baseada em matriz sistólica em um FPGA. Foram Testadas duas configurações para o formato de representação digital, usando ponto flutuante e ponto fixo. Para escolher o tamanho do ponto fixo, foi verificado o valor máximo absoluto entre os elementos do sistema linear e a precisão exigida pelo algoritmo. A conclusão foi de que uma operação

Tabela 12 – Erro absoluto médio dos estados de navegação.

Estado	Filtro de Kalman	Filtro de Kalman Robusto	Comparação Percentual
$\phi$	2.3260°	2.0167°	-13.3%
$\theta$	2.0625°	1.6718°	-18.9%
$\psi$	1.2908°	1.2377°	-4.1%
$v_x$	0.2574m/s	0.2446m/s	-5.0%
$v_y$	0.2031m/s	0.2131m/s	+4.9%
$v_z$	0.8177m/s	0.8042m/s	-1.7%
$x$	0.7072m	0.7089m	+0.2%
$y$	0.5212m	0.5252m	+0.8%
$z$	0.2650m	0.2691m	+1.5%

Figura 32 – Atitude de referência e estimada com filtro de Kalman padrão e robusto.

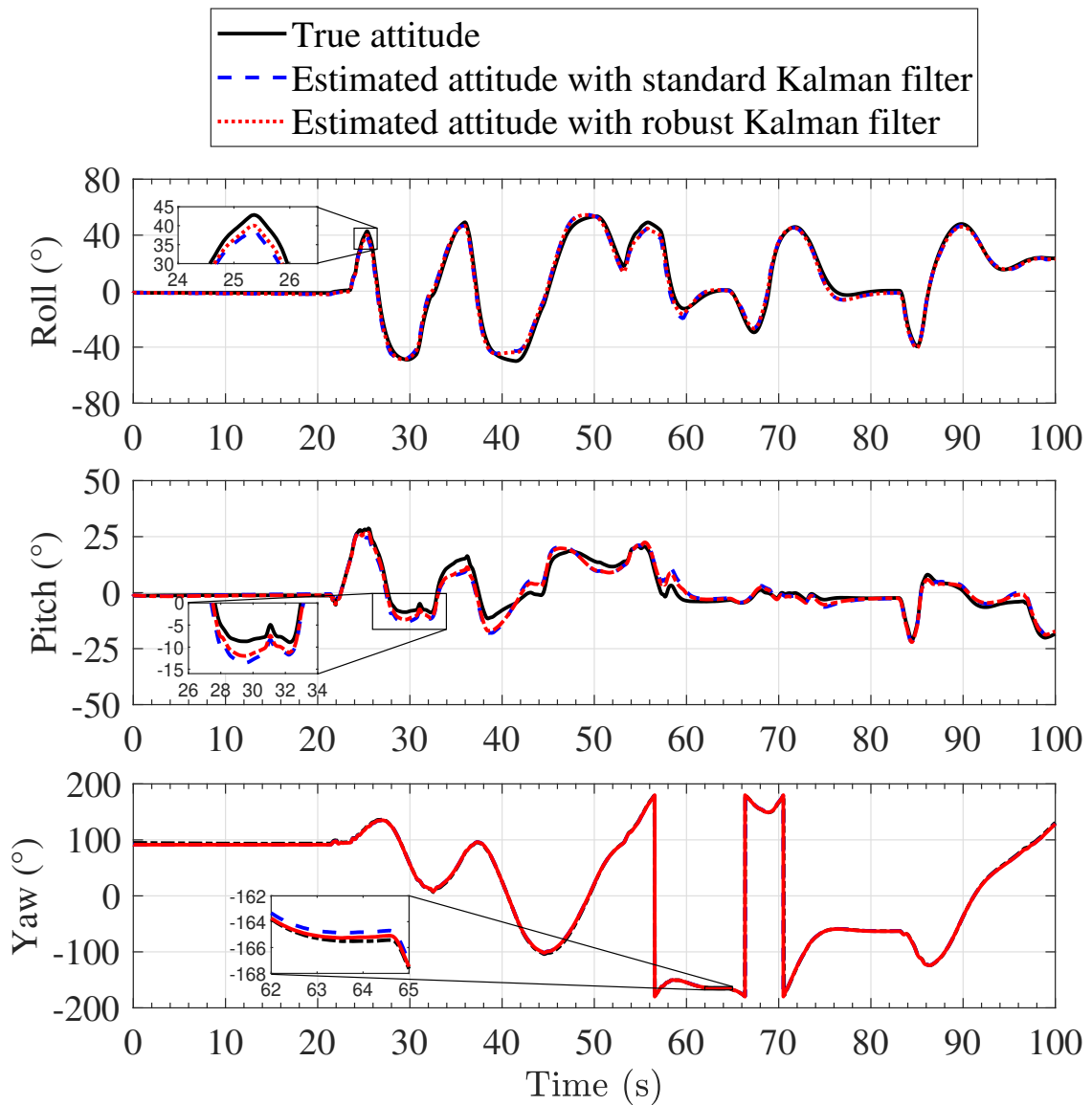
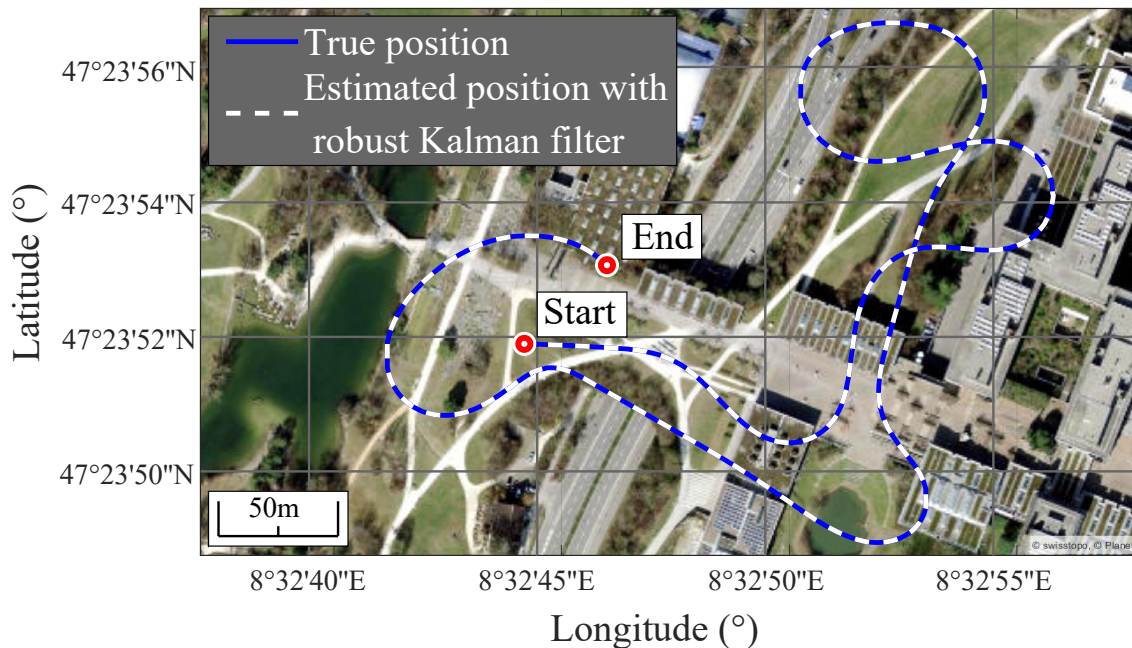


Figura 33 – Posição local de referência e estimada com filtro de Kalman robusto.



segura exigiria 1 bit para o sinal, 19 bits para a parte inteira e 12 bits para a parte fracionária. A Tabela 13 informa as funcionalidades necessárias para implementar PEs. Foi desenvolvido o circuito e gerado o código Verilog utilizando as ferramentas Simulink e o software Xilinx Vivado. No Simulink, foi desenvolvido o circuito PE e depois gerado o código Verilog com o auxílio dos aplicativos HDL Code e HDL Verifier. Em seguida, os circuitos PE foram sintetizados no Xilinx Vivado, gerando o relatório de recursos do FPGA.

Tabela 13 – Recursos do FPGA para PEs.

Formato	PE	LUTs	FFs	BRAM	DSPs
Ponto Flutuante	PE Diagonal	4608	6232	0	2
	PE Interno	2683	3714	0	4
	PE de Saída	2682	3585	0	4
Ponto Fixo	PE Diagonal	4736	1075	0	71
	PE Interna	208	103	0	16
	PE de Saída	203	32	0	16

---

## Capítulo 7

### Conclusões

---

Neste trabalho, foi apresentado um circuito baseado em systolic arrays para resolver o sistema linear extensivo utilizado pelo filtro de Kalman robusto para estimar estados de navegação aérea. Para confirmar a estrutura de cálculo, simulações do filtro de Kalman padrão e da versão robusta foram realizadas usando dados reais. O estudo aqui apresentado mostra o potencial de aplicação em sistemas embarcados de arquitetura híbrida software/hardware, nos quais etapas computacionalmente críticas podem ser paralelizadas. Neste contexto, pesquisas futuras podem ser exploradas para a construção de um sistema de controle de fluxo de dados e memória para implementar o circuito proposto em um FPGA. Também é possível reescrever o sistema de equações para a solução robusta de Kalman, de forma que apenas o vetor de estado e a covariância filtrada sejam usados. Neste trabalho, os valores preditos pelo filtro de Kalman robusto não são utilizados. Nos trabalhos de referência deste estudo, esses valores são utilizados no próximo ciclo de filtragem recursiva robusta.



---

# Referências

---

- ABDUL-KADHIM, A. A.; LIEN, F.-S.; YEE, E. Gpgpu implementation of a lattice boltzmann methodology for particle transport and deposition in complex flow. **International Journal of Numerical Methods for Heat & Fluid Flow**, Emerald Publishing Limited, 2019.
- ABOLHASANI, M.; RAHMANI, M. Robust kalman filtering for discrete-time time-varying systems with stochastic and norm-bounded uncertainties. **Journal of Dynamic Systems, Measurement, and Control**, American Society of Mechanical Engineers Digital Collection, v. 140, n. 3, 2018.
- ABOU-KANDIL, H. et al. **Matrix Riccati equations in control and systems theory**. [S.l.]: Birkhäuser, 2012.
- AGUIRRE, L. A. **Introdução à identificação de sistemas—Técnicas lineares e não-lineares aplicadas a sistemas reais**. [S.l.]: Editora UFMG, 2004.
- ANDRADE, H.; CRNKOVIC, I. A review on software architectures for heterogeneous platforms. In: IEEE. **2018 25th Asia-Pacific Software Engineering Conference (APSEC)**. [S.l.], 2018. p. 209–218.
- APOLINÁRIO, J. A.; RAUTMANN, R. **QRD-RLS Adaptive Filtering**. [S.l.]: Springer, 2009. ISBN 978-0-387-09734-3.
- ASPENCORE. **2019 Embedded Markets Study - Integrating IoT and Advanced Technology Designs, Application Development & Processing Environments**. [S.l.], 2019.
- BARCZYK, M.; LYNCH, A. F. Invariant observer design for a helicopter uav aided inertial navigation system. **IEEE Transactions on Control Systems Technology**, IEEE, v. 21, n. 3, p. 791–806, 2012.
- BAYOUMI, M. A.; LING, N. **Specification and verification of systolic arrays**. [S.l.]: World Scientific, 1999.
- BEILINA, L.; KARCHEVSKII, E.; KARCHEVSKII, M. **Numerical Linear Algebra: Theory and Applications**. [S.l.]: Springer, 2017. ISBN 978-3-319-57304-5.
- BJÖRCK, Å. **Numerical Methods in Matrix Computations**. [S.l.]: Springer, 2015. v. 59. ISBN 978-3-319-05088-1.

- BLACK, H. D. A passive system for determining the attitude of a satellite. **AIAA journal**, v. 2, n. 7, p. 1350–1351, 1964.
- CARDOSO, J. M. P.; COUTINHO, J. G. de F.; DINIZ, P. C. **Embedded Computing for High Performance: Efficient Mapping of Computations Using Customization, Code Transformations and Compilation**. [S.l.]: Morgan Kaufmann, 2017.
- CERRI, J. P. **Controle e filtragem para sistemas lineares discretos incertos sujeitos a saltos Markovianos**. Tese (Doutorado) — Universidade de São Paulo, 2013.
- CHEN, C.-T. **Linear system theory and design**. [S.l.]: Saunders college publishing, 1984.
- CRASSIDIS, J. L. Sigma-point kalman filtering for integrated gps and inertial navigation. **IEEE Transactions on Aerospace and Electronic Systems**, IEEE, v. 42, n. 2, p. 750–756, 2006.
- CROCKETT, L. H. et al. **The Zynq book: embedded processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 all programmable SoC**. [S.l.]: Strathclyde Academic Media, 2014.
- DATTA, B. N. **Numerical Methods for Linear Control Systems**. [S.l.]: Academic Press, 2004. v. 1. ISBN 978-0-12-203590-6.
- \_\_\_\_\_. **Numerical Linear Algebra and Applications**. [S.l.]: SIAM-Society for Industrial and Applied Mathematic, 2010. ISBN 978-0-898716-85-6.
- EL-DIASTY, M.; PAGIATAKIS, S. Calibration and stochastic modelling of inertial navigation sensor errors. **Journal of Global Positioning Systems**, v. 7, n. 2, p. 170–182, 2008.
- FARRELL, J. **Aided navigation: GPS with high rate sensors**. [S.l.]: McGraw-Hill, Inc., 2008.
- FORD, W. **Numerical Linear Algebra with Applications: Using MATLAB**. [S.l.]: Academic Press, 2014. ISBN 978-0-12-394435-1.
- FRANCIS, J. G. The QR Transformation a Unitary Analogue to the LR Transformation — Part 1. **The Computer Journal**, Oxford University Press, v. 4, n. 3, p. 265–271, 1961.
- \_\_\_\_\_. The QR Transformation — Part 2. **The Computer Journal**, Oxford University Press, v. 4, n. 4, p. 332–345, 1962.
- GENTLEMAN, W. M. Least Squares Computations By Givens Transformations Without Square Roots. **IMA Journal of Applied Mathematics**, Oxford University Press, v. 12, n. 3, p. 329–336, 1973.
- GENTLEMAN, W. M.; KUNG, H. Matrix triangularization by systolic arrays. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Real-time signal processing IV**. [S.l.], 1982. v. 298, p. 19–26.



- GIVENS, W. Computation of Plain Unitary Rotations Transforming a General Matrix to Triangular Form. **Journal of the Society for Industrial and Applied Mathematics**, SIAM, v. 6, n. 1, p. 26–50, 1958.
- GOLUB, G. H.; LOAN, C. F. V. **Matrix Computations**. 4. ed. [S.l.]: Johns Hopkins University Press, 2013. ISBN 978-1-421-40794-4.
- GREWAL, M. S.; ANDREWS, A. P. Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. **IEEE Control Systems Magazine**, IEEE, v. 30, n. 3, p. 69–78, 2010.
- GUZMÁN, M. A. D. et al. Cooperative cpu, gpu, and fpga heterogeneous execution with enginecl. **The Journal of Supercomputing**, Springer, v. 75, n. 3, p. 1732–1746, 2019.
- H., M. H. M.; LANG, T. **Matrix computations on systolic-type arrays**. [S.l.]: Springer, 1992.
- HAMMARLING, S. A Note on Modifications to The Givens Plane Rotation. **IMA Journal of Applied Mathematics**, Oxford University Press, v. 13, n. 2, p. 215–218, 1974.
- INOUE, R. S. et al. Markovian Jump Linear Systems-Based Filtering for Visual and GPS Aided Inertial Navigation System. In: IEEE. **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2017. p. 4083–4089.
- IPSEN, I. C. F. **Numerical Matrix Analysis: Linear Systems and Least Squares**. [S.l.]: SIAM, 2009. ISBN 978-0-89871-676-4.
- ISHIHARA, J. Y. et al. Optimal robust filtering for systems subject to uncertainties. **Automatica**, Elsevier, v. 52, p. 111–117, 2015.
- ISHIHARA, J. Y.; TERRA, M. H.; BIANCO, A. F. Recursive linear estimation for general discrete-time descriptor systems. **Automatica**, Elsevier, v. 46, n. 4, p. 761–766, 2010.
- JULIER, S. J.; UHLMANN, J. K. New extension of the kalman filter to nonlinear systems. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Signal processing, sensor fusion, and target recognition VI**. [S.l.], 1997. v. 3068, p. 182–193.
- KAILATH, T.; SAYED, A. H.; HASSIBI, B. **Linear Estimation**. [S.l.]: Prentice Hall, 2000. ISBN 9780130224644.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. 1960.
- KIM, P. **Kalman filter for beginners: with MATLAB examples**. [S.l.]: CreateSpace, 2011.
- KONTOGHIORGHES, E. **Parallel Algorithms for Linear Models: Numerical Methods and Estimation Problems**. [S.l.]: Springer Science & Business Media, 2000. v. 15. ISBN 978-1-4615-4571-2.
- KUBLANOVSKAYA, V. N. On Some Algorithms for the Solution of the Complete Eigenvalue Problem. **USSR Computational Mathematics and Mathematical Physics**, Elsevier, v. 1, n. 3, p. 637–657, 1962.

- KUNG, H.; LEISERSON, C. E. Systolic arrays (for vlsi). In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. **Sparse Matrix Proceedings 1978**. [S.l.], 1979. v. 1, p. 256–282.
- KUNG, H.-T. Why systolic architectures? **Computer**, IEEE, n. 1, p. 37–46, 1982.
- LAWRENCE, A. **Modern inertial technology: navigation, guidance, and control**. [S.l.]: Springer Science & Business Media, 2012.
- LIU, W. et al. Comparative analysis between error-state and full-state error estimation for kf-based imu/gnss integration against imu faults. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 22, p. 4912, 2019.
- LU, C.-P. **Should We All Embrace Systolic Arrays? On Scalability of the Matrix Multiply Unit in Google’s TPU**. Abril de 2017. <<https://medium.com/@CPLu/should-we-all-embrace-systolic-array-df3830f193dc>>.
- LUENBERGER, D. G.; YE, Y. Linear and nonlinear programming. Springer Publishing Company, Incorporated, 2015.
- LUENBERGER, D. G.; YE, Y. et al. **Linear and nonlinear programming**. [S.l.]: Springer, 1984. v. 2.
- MADYASTHA, V. et al. Extended kalman filter vs. error state kalman filter for aircraft attitude estimation. In: **AIAA Guidance, Navigation, and Control Conference**. [S.l.: s.n.], 2011. p. 6615.
- MATHWORKS. **Flight Log Analyzer 2020 - Analyze UAV Autopilot Flight Logs. Import, analysis, and plotting of telemetry logs**. Disponível em: <<https://www.mathworks.com/help/uav/ref/flightloganalyzer-app.html>>.
- MAYBECK, P. S. **Stochastic models, estimation, and control**. [S.l.]: Academic press, 1982.
- MERCHANT, F. et al. Efficient Realization of Givens Rotation Through Algorithm-Architecture Co-Design for Acceleration of QR Factorization. **arXiv preprint arXiv:1803.05320**, 2018.
- MIDDLETON, R. H.; GOODWIN, G. C. **Digital control and estimation: a unified approach**. [S.l.]: Prentice Hall Professional Technical Reference, 1990.
- MONTICELLI, A. **State Estimation in Electric Power Systems: A Generalized Approach**. [S.l.]: Springer Science & Business Media, 2012. ISBN 978-1-4615-4999-4.
- MOURIKIS, A. I.; ROUMELIOTIS, S. I. A multi-state constraint kalman filter for vision-aided inertial navigation. In: IEEE. **Proceedings 2007 IEEE international conference on robotics and automation**. [S.l.], 2007. p. 3565–3572.
- NAGPAL, K. M.; KHARGONEKAR, P. P. Filtering and smoothing in an  $\mathcal{H}_\infty$  - setting. **IEEE Transactions on Automatic Control**, IEEE, v. 36, n. 2, p. 152–166, 1991.
- NARASIMHAPPA, M. et al. Fpga Implementation of robust array kalman filter based on givens rotation. 2017.

NGO, H.-Q.-T. et al. Experimental comparison of complementary filter and kalman filter design for low-cost sensor in quadcopter. In: **IEEE. 2017 International Conference on System Science and Engineering (ICSSE)**. [S.l.], 2017. p. 488–493.

NULL, L.; LOBUR, J. **The essentials of computer organization and architecture**. [S.l.]: Jones & Bartlett Publishers, 2014.

NYSETH, M. **How NaviSuite’s Kalman filter ensures precise subsea positioning**. 2017. Disponível em: <<https://www.eiva.com/about/eiva-log/how-navisuites-kalman-filter-ensures-precise-subsea-positioning>>.

OCEAN, N. C. for E. I. N.; NOAA, A. A. **Maps of Magnetic Elements from the WMM2020**. 2020. Disponível em: <<https://ngdc.noaa.gov/geomag/WMM/image.shtml>>.

OED, O. **navigation, n**. Oxford University Press, 2021. Disponível em: <<https://www.oxfordlearnersdictionaries.com/definition/english/navigation>>.

ÖFVERSTEDT, J.; LINDBLAD, J.; SLADOJE, N. Fast computation of mutual information in the frequency domain with applications to global multimodal image alignment. **arXiv preprint arXiv:2106.14699**, 2021.

OZYAGCILAR, T. Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors. sl: Freescale semiconductor inc. **Application Note AN4248, Rev**, v. 3, 2012.

\_\_\_\_\_. Calibrating an ecompass in the presence of hard-and soft-iron interference. **Application Note AN4246, Rev**, v. 4, 2013.

PETERSEN, I. R.; MCFARLANE, D. C. Optimal guaranteed cost control and filtering for uncertain linear systems. **IEEE Transactions on Automatic Control**, IEEE, v. 39, n. 9, p. 1971–1977, 1994.

PETERSEN, I. R.; SAVKIN, A. V. **Robust Kalman filtering for signals and systems with large uncertainties**. [S.l.]: Springer Science & Business Media, 1999.

QUINCHIA, A. G. et al. A comparison between different error modeling of mems applied to gps/ins integrated systems. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 13, n. 8, p. 9549–9588, 2013.

ROCHA, K. D.; TERRA, M. H. Robust Kalman filter for systems subject to parametric uncertainties. **Systems & Control Letters**, Elsevier, v. 157, p. 105034, 2021.

SAYED, A. H. A framework for state-space estimation with uncertain models. **IEEE Transactions on Automatic Control**, IEEE, v. 46, n. 7, p. 998–1013, 2001.

SAYED, A. H.; NASCIMENTO, V. H. Design criteria for uncertain models with structured and unstructured uncertainties. In: **Robustness in identification and control**. [S.l.]: Springer, 1999. p. 159–173.

SHAWAHNA, A.; SAIT, S. M.; EL-MALEH, A. Fpga-based accelerators of deep learning networks for learning and classification: A review. **IEEE Access**, IEEE, v. 7, p. 7823–7859, 2018.

- SHIN, E.-H.; EL-SHEIMY, N. An unscented kalman filter for in-motion alignment of low-cost imus. In: **IEEE. PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No. 04CH37556)**. [S.l.], 2004. p. 273–279.
- SHUSTER, M. D.; OH, S. D. Three-axis attitude determination from vector observations. **Journal of guidance and Control**, v. 4, n. 1, p. 70–77, 1981.
- SIMON, D. Optimal state estimation: Kalman,  $\mathcal{H}$  infinity, and nonlinear approaches. hoboken. **NJ: John Wiley and Sons**, v. 10, p. 0470045345, 2006.
- SMITH, G. L.; SCHMIDT, S. F.; MCGEE, L. A. **Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle**. [S.l.]: National Aeronautics and Space Administration, 1962.
- SOH, J.; WU, X. An FPGA-Based Unscented Kalman Filter for System-On-Chip Applications. **IEEE Transactions on Circuits and Systems II: Express Briefs**, IEEE, v. 64, n. 4, p. 447–451, 2016.
- SOLA, J. Quaternion kinematics for the error-state kalman filter. **arXiv preprint arXiv:1711.02508**, 2017.
- Stanford Artificial Intelligence Laboratory et al. **Robotic Operating System**. Disponível em: <<https://www.ros.org>>.
- SUDA, N. et al. Throughput-optimized opencl-based fpga accelerator for large-scale convolutional neural networks. In: **Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays**. [S.l.: s.n.], 2016. p. 16–25.
- THEODOR, Y.; SHAKED, U. Robust discrete-time minimum-variance filtering. **IEEE Transactions on Signal Processing**, IEEE, v. 44, n. 2, p. 181–189, 1996.
- TRAWNY, N.; ROUMELIOTIS, S. I. Indirect kalman filter for 3d attitude estimation. **University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep**, Citeseer, v. 2, p. 2005, 2005.
- VÁZQUEZ-CASTILLO, J. et al. Fpga-based hardware matrix inversion architecture using hybrid piecewise polynomial approximation systolic cells. **Electronics**, Multidisciplinary Digital Publishing Institute, v. 9, n. 1, p. 182, 2020.
- WANG, Y.; HUYNH, G.; WILLIAMSON, C. Integration of google maps/earth with microscale meteorology models and data visualization. **Computers & Geosciences**, Elsevier, v. 61, p. 23–31, 2013.
- WELLENHOF, B. H.; LEGAT, K.; WIESER, M. **Navigation: Principles of Positioning and Guidance**. [S.l.]: Springer Science & Business Media, 2003.
- XILINX, I. **O que é um FPGA?** 2021. Disponível em: <<https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>>.
- YENGERA, G. et al. Computation of Extended Robust Kalman Filter for Real-Time Attitude and Position Estimation. **arXiv preprint arXiv:1801.04386**, 2018.

ZENG, H.; PRASANNA, V. Graphact: Accelerating gcn training on cpu-fpga heterogeneous platforms. In: **Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays**. [S.l.: s.n.], 2020. p. 255–265.

ZHAO, Y.; YANG, C. Information fusion robust guaranteed cost kalman estimators with uncertain noise variances and missing measurements. **International Journal of Systems Science**, Taylor & Francis, v. 50, n. 15, p. 2853–2869, 2019.



# Apêndices





---

# APÊNDICE A

## Rotação de Givens

---



J. Wallace Givens Jr.  
(1910 - 1993)

*Epônimo do processo de rotação planar para fatoração matricial, James Wallace Givens Jr foi pioneiro no desenvolvimento de soluções numéricas para ciência da computação, com trabalhos expressivos publicados nas décadas de 1940 e 1950. Matemático de formação, foi professor na Universidade do Tennessee, Wayne State University e na Universidade Northwestern. Mais tarde, exerceu o cargo de cientista sênior no Argonne National Laboratory. Sua obra mais notável foi justamente a proposta do cálculo de rotações unitárias planas que transformam uma matriz geral na forma triangular. Atualmente, esse método é conhecido como rotação de Givens.*

A decomposição matricial é um procedimento matemático que aplica reformas estruturais em uma matriz para reescrevê-la com determinada forma canônica. Em geral, a decomposição resulta em fatores de matrizes, e por isso, muitas vezes esse recurso é chamado de fatoração matricial.

O processo de decomposição de matrizes é amplamente utilizado em soluções numéricas e estudos demonstrativos de resultados teóricos, como, por exemplo, na resolução de sistemas de equações lineares, cálculos de inversão e determinante matriciais, ajuste de mínimos quadrados e obtenção de valores característicos em matrizes de grandes dimensões. A conveniência computacional e a viabilidade analítica justificam o emprego

desse mecanismo algébrico nas soluções de problemas matemáticos e de engenharia em geral, sendo que, para cada classe particular de problemas seja pertinente a utilização de um tipo específico de decomposição. Os principais modelos de decomposição matricial incluem a decomposição  $LU$ , a decomposição  $SVD$ , a decomposição de Cholesky e a decomposição  $QR$ . Esse último, fornece uma modo eficaz para solução de sistemas de equações do tipo  $Ax = b$ , pois, sendo  $Q$  uma matriz ortogonal e  $R$  uma matriz triangular superior, a operação  $Rx = Q^T b$  torna-se uma tarefa menos complexa.

## A.1 Decomposição QR

A fatoração de uma matriz retangular  $A \in \mathbb{R}^{m \times n}$  em um produto de  $Q \in \mathbb{R}^{m \times m}$  e  $R \in \mathbb{R}^{m \times n}$ , tal que  $Q$  seja uma matriz ortogonal e  $R$  uma matriz triangular superior, torna-se:

$$A = QR. \quad (\text{A.1})$$

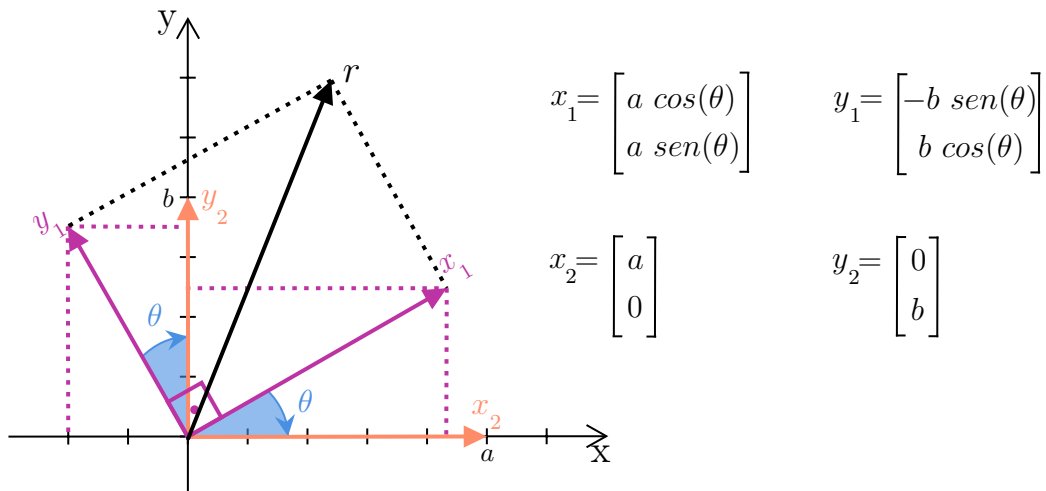
Essa decomposição surgiu como recurso algébrico para solução de grandes sistemas de equações lineares simultâneas. Com isso, diversos problemas envolvidos passaram a ter viabilidade numérica, tais como o cálculo de autovalores e autovetores com o algoritmo  $QR$ , veja (FRANCIS, 1961) (FRANCIS, 1962) e (KUBLANOVSKAYA, 1962), problemas de mínimos quadrados e filtragem de processos estocásticos (APOLINÁRIO; RAUTMANN, 2009).

Alguns métodos consolidados para execução da decomposição  $QR$  são descritos na literatura básica de álgebra linear. O processo de Gram-Schmidt gera um conjunto de vetores ortonormais a partir dos vetores colunas de  $A$ . Embora tenha um aspecto algébrico mais intuitivo, esse processo é numericamente instável pela natureza de ortogonalização em ponto flutuante. A transformação linear de Householder descreve reflexões em relação a um plano ou hiperplano que contém a origem, por isso também é referida como reflexão de Householder. Essa ferramenta é numericamente estável devido a forma que introduz zeros em  $R$ . Entretanto, como cada reflexão altera completamente as matrizes  $Q$  e  $R$ , essa operação inclui uma largura de banda significativa. Outro instrumento para realização da decomposição  $QR$  é a rotação de Givens. Para algumas aplicações esse método torna-se mais atraente que os anteriores, pois apresenta maior estabilidade numérica comparada ao processo de Gram-Schmidt, além de dispor da capacidade de paralelizar o processo, algo limitante nas reflexões de Householder. Nesse método de fatoração, apenas duas linhas de  $R$  são afetadas em cada transformação matricial, o que torna esse algoritmo mais eficiente em termos de largura de banda e viabiliza a computação paralela. A desvantagem está na complexidade de implementação, pois apresenta certa dificuldade para estabelecer a ordenação de anulação dos elementos. Uma sequência de anulação possível é descrita no Anexo A deste documento, com implementação desenvolvida em MATLAB da decomposição  $QR$  baseada em rotações de Givens.

O processo de introdução de zeros em vetores a partir de rotações unitárias planares foi apresentado por (GIVENS, 1958) para converter matrizes no formato triangular. Esse procedimento também é conhecido por rotações de Jabobi, pois, segundo alguns autores, consulte (DATTA, 2010) e (BJÖRCK, 2015), o processo foi originalmente desenvolvido por Carl Gustav Jakob Jacobi para uso na solução do problema de autovalor simétrico em 1846. O número de anulações necessárias para triangularizar uma matriz dependerá da sua dimensão. Para a matriz retangular  $A$  de (A.1), a quantidade de etapas da decomposição  $QR$  será igual a  $\sum_{k=1}^{n-1} k$ , se  $m \leq n$  ou  $\sum_{k=m-n}^{m-1} k$ , se  $m > n$ . Para triangulação completa de  $A$ , o algoritmo de decomposição  $QR$  via rotação de Givens requer *FLloating-point Operations* (FLOPs) de uma ordem de magnitude igual a  $3n^2(m - n/3)$ .

### A.1.1 Interpretação Geométrica

O método proposto por Givens realiza a rotação do vetor  $r = [a \ b]^T$  ao longo de uma círculo de raio  $\|r\|$  por um ângulo  $\theta = \tan^{-1}(b/a)$ , na ordem para alinhá-lo com o vetor base  $[\|r\| \ 0]^T$ . Na projeção de  $r$  em  $x_1$  e  $y_1$ , o movimento de alinhamento com  $x_2$  e  $y_2$ , respectivamente, é traçado como:



A transformação para alinhamento dos quadros é exercida pela matriz de rotação  $G(\theta)$  em ambos eixos coordenados, com relação

$$\begin{cases} x_2 = G(\theta)x_1 \\ y_2 = G(\theta)y_1 \end{cases}, \quad \text{tal que} \quad G = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}.$$

Então, a solução desse sistema linear resulta na matriz de Givens  $G(\theta)$  de rotação planar:

$$G(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (\text{A.2})$$

O conceito iterativo para transformação da matriz  $A$  de (A.1) é expressa por

$$A_{n+1} = G_{jk}(\theta)A_n. \quad (\text{A.3})$$

De forma geral, a Matriz de Givens (A.2), usada para anular termos  $a = A(j, k) \mid j > k$  (elementos abaixo da diagonal principal de  $A$ ), apresenta a estrutura (A.4). O elemento  $a = A(i, k)$  pode ser qualquer componente de  $A$  presente na coluna  $k$ , desde que  $k \leq i < j$ .

$$G_{jk}(\theta) = \begin{bmatrix} 1 & \cdots & 0 & \overset{i\downarrow}{0} & 0 & \cdots & 0 & \overset{j\downarrow}{0} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ i \rightarrow & 0 & \cdots & 0 & \frac{a}{\sqrt{a^2+b^2}} & 0 & \cdots & 0 & \frac{b}{\sqrt{a^2+b^2}} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ j \rightarrow & 0 & \cdots & 0 & -\frac{b}{\sqrt{a^2+b^2}} & 0 & \cdots & 0 & \frac{a}{\sqrt{a^2+b^2}} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \quad (A.4)$$

A cadeia de operações em que se aplica (A.4) em (A.3) é nitidamente observada na Figura 34, na qual a sequência de rotações planares para encontrar a forma trapezoidal de uma matriz  $A_{4 \times 3}$  realiza o procedimento de anulação por colunas com movimento ascendente.

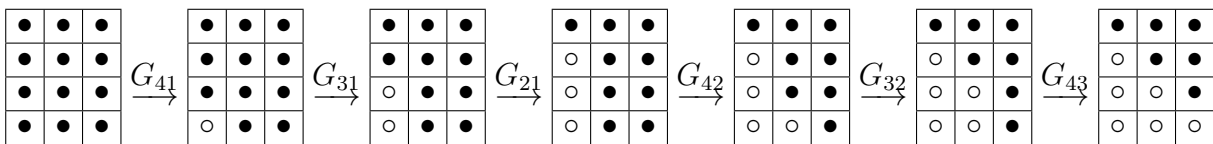


Figura 34 – Sequência para triangularização matricial de  $A^{4 \times 3}$ .

Valendo-se da ortogonalidade das Matrizes de Givens, o cálculo de  $Q$  para sequência de anulações mostradas na Figura 34 é

$$Q = [G_{41}G_{31}G_{21}G_{42}G_{32}G_{43}]^T. \quad (A.5)$$

### A.1.2 Decomposições Similares

Formas análogas à decomposição  $QR$  que geram fatores ortogonal e triangular podem ser úteis em algumas aplicações. A decomposição  $RQ$  tem como resultado o produto da matriz ortogonal  $Q$  à direita da matriz triangular superior  $R$ . Já as decomposições  $QL$  e  $LQ$  geram uma matriz triangular inferior  $L$  e outra ortogonal  $Q$ , com a ordem dos fatores contrárias. As interpretações algébricas e geométricas dessas decomposições são apresentadas na Tabela 14.

A fatoração  $LQ$  de  $A$  é essencialmente a mesma que a fatoração  $QR$  de  $A^T$ . Esse vínculo ocorre do mesmo modo entre a fatoração  $RQ$  e  $QL$ .

$$\begin{aligned} A_1 = L_1 Q_1 &\iff A_1^T = Q_1^T R_1 : R_1 \equiv L_1^T \\ A_2 = R_2 Q_2 &\iff A_2^T = Q_2^T L_2 : L_2 \equiv R_2^T \end{aligned} \tag{A.6}$$

Tabela 14 – Rotação de Givens para decomposição  $QR$ ,  $QL$ ,  $RQ$  e  $LQ$ , respectivamente.

	Triangular Superior	Triangular Inferior
Anulação Vertical	$\begin{bmatrix} r \\ 0 \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix}$ $c = \frac{a}{\ r\ } \quad s = \frac{b}{\ r\ }$ $\ r\  = \sqrt{a^2 + b^2}$	$\begin{bmatrix} 0 \\ \ell \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix}$ $c = \frac{b}{\ \ell\ } \quad s = \frac{a}{\ \ell\ }$ $\ \ell\  = \sqrt{a^2 + b^2}$
Anulação Horizontal	$\begin{bmatrix} 0 & r \end{bmatrix} = \begin{bmatrix} a & b \\ -s & c \end{bmatrix} \cdot \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ $c = \frac{b}{\ r\ } \quad s = \frac{a}{\ r\ }$ $\ r\  = \sqrt{a^2 + b^2}$	$\begin{bmatrix} \ell & 0 \end{bmatrix} = \begin{bmatrix} a & b \\ s & c \end{bmatrix} \cdot \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ $c = \frac{a}{\ \ell\ } \quad s = \frac{b}{\ \ell\ }$ $\ \ell\  = \sqrt{a^2 + b^2}$

## A.2 Exemplo Analítico

**Exemplo A.2.1** Considere a seguinte matriz não singular quadridimensional:

$$A = \begin{bmatrix} 1 & 2 & 9 & 7 \\ 3 & 6 & 3 & 5 \\ 0 & 1 & 5 & 6 \\ 4 & 8 & 2 & 7 \end{bmatrix}.$$

Neste exemplo, a decomposição  $QR$  de  $A$  por rotações de Givens ocorre descensionalmente, assim como realizado na estrutura de systolic array expressa no Apêndice B. Nesta circunstância,  $A(i, k)$  é fixado em  $i=k$  para anulação dos elementos abaixo da mesma coluna.

**Solução**

Utilizando a relação iterativa de (A.3), foram realizados os seguintes passos:

**Passo 1:** Eliminar  $A(2, 1)$ .  $\text{sen}(\theta) = \frac{3}{\sqrt{10}}$ ,  $\text{cos}(\theta) = \frac{1}{\sqrt{10}}$ .

$$G_{2,1} = \begin{bmatrix} \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} & 0 & 0 \\ -\frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_1 = G_{2,1}A = \begin{bmatrix} \frac{10}{\sqrt{10}} & \frac{20}{\sqrt{10}} & \frac{18}{\sqrt{10}} & \frac{22}{\sqrt{10}} \\ 0 & 0 & -\frac{24}{\sqrt{10}} & -\frac{16}{\sqrt{10}} \\ 0 & 1 & 5 & 6 \\ 4 & 8 & 2 & 7 \end{bmatrix}$$

**Passo 2:** Eliminar  $A(3, 1)$ .  $\text{sen}(\theta) = 0$ ,  $\text{cos}(\theta) = 1$ .

$$G_{3,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = G_{3,1}A_1 = \begin{bmatrix} \frac{10}{\sqrt{10}} & \frac{20}{\sqrt{10}} & \frac{18}{\sqrt{10}} & \frac{22}{\sqrt{10}} \\ 0 & 0 & -\frac{24}{\sqrt{10}} & -\frac{16}{\sqrt{10}} \\ 0 & 1 & 5 & 6 \\ 4 & 8 & 2 & 7 \end{bmatrix}$$

**Passo 3:** Eliminar  $A(4, 1)$ .  $\text{sen}(\theta) = \frac{4}{\sqrt{26}}$ ,  $\text{cos}(\theta) = \frac{\sqrt{10}}{\sqrt{26}}$ .

$$G_{4,1} = \begin{bmatrix} \frac{\sqrt{10}}{\sqrt{26}} & 0 & 0 & \frac{4}{\sqrt{26}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{4}{\sqrt{26}} & 0 & 0 & \frac{\sqrt{10}}{\sqrt{26}} \end{bmatrix} \quad A_3 = G_{4,1}A_2 = \begin{bmatrix} \frac{26}{\sqrt{26}} & \frac{52}{\sqrt{26}} & \frac{26}{\sqrt{26}} & \frac{50}{\sqrt{26}} \\ 0 & 0 & -\frac{24}{\sqrt{10}} & -\frac{16}{\sqrt{10}} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & -\frac{26}{\sqrt{65}} & -\frac{9}{\sqrt{65}} \end{bmatrix}$$

**Passo 4:** Eliminar  $A(3, 2)$ .  $\text{sen}(\theta) = 1$ ,  $\text{cos}(\theta) = 0$ .

$$G_{3,2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad A_4 = G_{3,2}A_3 = \begin{bmatrix} \frac{26}{\sqrt{26}} & \frac{52}{\sqrt{26}} & \frac{26}{\sqrt{26}} & \frac{50}{\sqrt{26}} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & \frac{24}{\sqrt{10}} & \frac{16}{\sqrt{10}} \\ 0 & 0 & -\frac{26}{\sqrt{65}} & -\frac{9}{\sqrt{65}} \end{bmatrix}$$

**Passo 5:** Eliminar  $A(4, 2)$ .  $\text{sen}(\theta) = 0$ ,  $\text{cos}(\theta) = 1$ .

Assim como ocorrido no Passo 2, o elemento  $A(4, 2)$  já é nulo. Portanto,  $G_{4,2}$  é uma matriz identidade e conseqüentemente  $A_5 = A_4$ .

**Passo 6:** Eliminar  $A(4, 3)$ .  $\text{sen}(\theta) = -\frac{\sqrt{13}}{\sqrt{85}}$ ,  $\text{cos}(\theta) = \frac{6\sqrt{2}}{\sqrt{85}}$ .

$$G_{4,3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{6\sqrt{2}}{\sqrt{85}} & -\frac{\sqrt{13}}{\sqrt{85}} \\ 0 & 0 & \frac{\sqrt{13}}{\sqrt{85}} & \frac{6\sqrt{2}}{\sqrt{85}} \end{bmatrix} \quad A_6 = G_{4,3}A_5 = \begin{bmatrix} \frac{26}{\sqrt{26}} & \frac{52}{\sqrt{26}} & \frac{26}{\sqrt{26}} & \frac{50}{\sqrt{26}} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & \frac{34}{\sqrt{17}} & \frac{21}{\sqrt{17}} \\ 0 & 0 & 0 & \frac{20}{\sqrt{442}} \end{bmatrix}$$

### **Resultados**

Para uma matriz quadrada de dimensão 4, a decomposição é gerada após 6 etapas. A matriz triangular superior  $A_6$ , originada pelo produto sequencial  $[G_{4,3}G_{4,2}G_{3,2}G_{4,1}G_{3,1}G_{2,1}A]$ , é equivalente à  $R$ . Logo, a transposição de  $[G_{4,3}G_{4,2}G_{3,2}G_{4,1}G_{3,1}G_{2,1}]$ , assim como (A.5) da Figura 34, resulta na matriz ortogonal  $Q$ . Neste exemplo:

$$Q = \begin{bmatrix} \frac{1}{\sqrt{26}} & 0 & \frac{4}{\sqrt{17}} & \frac{3}{\sqrt{442}} \\ \frac{3}{\sqrt{26}} & 0 & 0 & -\frac{17}{\sqrt{442}} \\ 0 & 1 & 0 & 0 \\ \frac{4}{\sqrt{26}} & 0 & -\frac{1}{\sqrt{17}} & \frac{12}{\sqrt{442}} \end{bmatrix} \quad e \quad R = \begin{bmatrix} \frac{26}{\sqrt{26}} & \frac{52}{\sqrt{26}} & \frac{26}{\sqrt{26}} & \frac{50}{\sqrt{26}} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & \frac{34}{\sqrt{17}} & \frac{21}{\sqrt{17}} \\ 0 & 0 & 0 & \frac{20}{\sqrt{442}} \end{bmatrix}.$$

**Exemplo A.2.2** Para a mesma matriz  $A$  do Exemplo A.2.1, as decomposições  $QL$ ,  $RQ$  e  $LQ$  são determinadas conforme as rotações apresentadas na Tabela 14.

### **Solução**

Foram elaboradas quatro funções numéricas em *MATLAB* que calculam as matrizes ortogonal e triangular das decomposições  $QL$ ,  $RQ$  e  $LQ$  de matrizes quadradas. A descrição completa dos códigos está disponível no Anexo A deste documento.

### **Resultados**

#### ***Decomposição QL:***

$$Q = \begin{bmatrix} 0,337 & 0,245 & 0,720 & 0,555 \\ -0,707 & 0,562 & -0,166 & 0,397 \\ -0,379 & -0,790 & 0,079 & 0,476 \\ 0,493 & 0,031 & -0,669 & 0,555 \end{bmatrix} \quad e \quad L = \begin{bmatrix} 0,190 & 0 & 0 & 0 \\ 2,054 & 3,319 & 0 & 0 \\ -2,454 & -4,829 & 5,039 & 0 \\ 3,965 & 8,406 & 9,675 & 12,610 \end{bmatrix}$$

#### ***Decomposição RQ:***

$$R = \begin{bmatrix} 0,487 & 2,645 & 8,418 & 7,544 \\ 0 & 1,206 & 0,920 & 8,758 \\ 0 & 0 & 5,910 & 5,203 \\ 0 & 0 & 0 & 11,533 \end{bmatrix} \quad e \quad Q = \begin{bmatrix} 0,864 & -0,499 & 0,012 & 0,073 \\ 0,202 & 0,274 & 0,699 & -0,629 \\ -0,305 & -0,441 & 0,693 & 0,481 \\ 0,347 & 0,694 & 0,173 & 0,607 \end{bmatrix}$$

#### ***Decomposição LQ:***

$$L = \begin{bmatrix} 11,619 & 0 & 0 & 0 \\ 6,627 & 5,923 & 0 & 0 \\ 7,660 & 0,040 & 1,823 & 0 \\ 7,488 & 8,674 & 1,260 & 0,319 \end{bmatrix} \quad e \quad Q = \begin{bmatrix} 0,086 & 0,172 & 0,775 & 0,603 \\ 0,410 & 0,820 & -0,360 & 0,170 \\ -0,371 & -0,193 & -0,504 & 0,756 \\ 0,829 & -0,510 & -0,128 & 0,191 \end{bmatrix}$$

A verificação dos resultados pode ser feita usando as relações (A.6).

### A.3 Rotação Rápida de Givens

Embora a Rotação de Givens seja muito conveniente para triangularização de matrizes, a forma original contém cálculos exaustivos que aumentam substancialmente o custo de sua implementação. Contudo, existe uma solução na qual ocorrem reduções significativas nos cálculos. A Rotação Rápida de Givens, proposta inicialmente por (GENTLEMAN, 1973) e generalizada por (HAMMARLING, 1974), emprega um pré-escalamento na matriz  $A$  para realizar as operações de transformação matricial sem a necessidade de calcular raízes quadradas. Além disso, as  $4n$  multiplicações e  $2n$  somas empregadas na transformação matricial, com a versão rápida se reduz a  $2n$  multiplicações e  $2n$  somas, pois a matriz  $G(\theta)$  também é reconstruída. A proposta é introduzir normas ponderadas na transformação juntamente com a decomposição da matriz de Givens.

Sejam  $A$  e  $G(\theta)$  redefinidos como

$$A = \sqrt{D}B \quad (\text{A.7})$$

$$G(\theta) = \begin{bmatrix} \cos(\theta) & \text{sen}(\theta) \\ -\text{sen}(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & \tau \\ -\tau & 1 \end{bmatrix}, \quad \text{tal que } \tau = \tan(\theta) \quad (\text{A.8})$$

A aplicação de (A.7) e (A.8) na transformação (A.3) resulta em um novo produto na forma  $\sqrt{\overset{\circ}{D}}\overset{\circ}{B}$ . A equivalência dos fatores fica explícita quando se aglutina os elementos em duas matrizes, tal como

$$\begin{aligned} \begin{bmatrix} \sqrt{\overset{\circ}{D}_i} & 0 \\ 0 & \sqrt{\overset{\circ}{D}_j} \end{bmatrix} \begin{bmatrix} \overset{\circ}{b}_{ik} \\ \overset{\circ}{b}_{jk} \end{bmatrix} &= \begin{bmatrix} \cos(\theta) & 0 \\ 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & \tau \\ -\tau & 1 \end{bmatrix} \begin{bmatrix} \sqrt{D_i} & 0 \\ 0 & \sqrt{D_j} \end{bmatrix} \begin{bmatrix} b_{ik} \\ b_{jk} \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta)\sqrt{D_i} & 0 \\ 0 & \cos(\theta)\sqrt{D_j} \end{bmatrix} \begin{bmatrix} 1 & \tau\sqrt{\frac{D_j}{D_i}} \\ -\tau\sqrt{\frac{D_i}{D_j}} & 1 \end{bmatrix} \begin{bmatrix} b_{ik} \\ b_{jk} \end{bmatrix}. \end{aligned}$$

Em  $k = i$ , para anular o elemento  $b_{ji}$ ,  $-\tau\sqrt{\frac{D_i}{D_j}}b_{ii} + b_{ji} = 0$ . Logo,

$$\tau = \left( \frac{b_{ji}}{b_{ii}} \right) \sqrt{\frac{D_j}{D_i}}. \quad (\text{A.9})$$

Ainda para que a relação seja satisfatória,

$$\sqrt{\overset{\circ}{D}_i} = \cos(\theta)\sqrt{D_i} \quad \text{e} \quad \sqrt{\overset{\circ}{D}_j} = \cos(\theta)\sqrt{D_j}. \quad (\text{A.10})$$

No caso geral, para  $k = 1, 2, 3 \dots n$ ,

$$\overset{\circ}{b}_{ik} = b_{ik} + \frac{b_{ji}}{b_{ii}} \frac{D_j}{D_i} b_{jk} \quad \text{e} \quad \overset{\circ}{b}_{jk} = b_{jk} - \frac{b_{ji}}{b_{ii}} b_{ik} \quad (\text{A.11})$$



Observe que para o cálculo de  $\mathring{b}_{ik}$  em (A.11), os quadrados de  $\sqrt{D_i}$  e  $\sqrt{D_j}$  são utilizados. Eles podem ser inicializados na forma quadrada e atualizados como tal sem inconsistência aritmética, pois para o instante inicial,  $\sqrt{D_0} = I$ . Portanto, (A.9) e (A.10) são atualizados como

$$\tau^2 = \frac{D_j b_{ji}}{D_i b_{ii}}, \quad \mathring{D}_i = \cos^2(\theta) D_i \quad \text{e} \quad \mathring{D}_j = \cos^2(\theta) D_j. \quad (\text{A.12})$$

A única consequência dessa manipulação matemática é a necessidade de calcular as raízes de  $D$  ao final do processo de triangularização de  $B$ . O valor de  $\cos^2(\theta)$  é identificado pelo emprego de (A.7) no cálculo do cosseno usado na decomposição  $QR$  mostrado na Tabela 14. Para tal, o quadrado do cosseno resulta em

$$\cos^2(\theta) = \frac{1}{1 + \tau^2} \quad (\text{A.13})$$

Caso o elemento  $b_{ii}$  for nulo e  $b_{ji} \neq 0$ , deverá ser feita uma comutação entre as linhas  $i$  e  $j$  de  $B$ . O significado geométrico dessa troca é uma rotação de  $\pm 90^\circ$ , dependendo do sinal de  $b_{ji}$ . Para este caso, a decomposição (A.8) deverá ser baseada no cálculo de senos.

$$\begin{bmatrix} \cos(\theta) & \text{sen}(\theta) \\ -\text{sen}(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} \text{sen}(\theta) & 0 \\ 0 & \text{sen}(\theta) \end{bmatrix} \begin{bmatrix} \kappa & 1 \\ -1 & \kappa \end{bmatrix}, \quad \text{tal que } \kappa = \text{cotan}(\theta) \quad (\text{A.14})$$

Usando a mesma estratégia de (A.9-A.13) em (A.14), a atualização dos elementos de  $D$  e  $B$  nas linhas  $i$  e  $j$  são definidas como

$$\begin{aligned} \text{sen}^2(\theta) &= \frac{1}{1 + \kappa^2}, \\ \mathring{D}_i &= \text{sen}^2(\theta) D_j \quad \text{e} \quad \mathring{D}_j = \text{sen}^2(\theta) D_i, \\ \mathring{b}_{ik} &= b_{jk} + \frac{b_{ii}}{b_{ji}} \frac{D_i}{D_j} b_{ik} \quad \text{e} \quad \mathring{b}_{jk} = \frac{b_{ii}}{b_{ji}} b_{jk} - b_{ik}. \end{aligned} \quad (\text{A.15})$$

Assim como o uso do seno para decompor a matriz de Givens é recomendada quando  $\cos(\theta) = 0$ , ou muito próximo disso por possibilidade de ocorrer *underflow* no cálculo, quando  $\theta \rightarrow 0$  ou  $\theta \rightarrow \pi$  radianos, não é possível usar a decomposição utilizando  $\text{seno}(\theta)$ . Deste modo, para indicar a transformação que deve ser utilizada, a seguinte relação é apresentada:

$$\begin{cases} -45^\circ \leq \theta \leq 45^\circ \text{ ou } 135^\circ \leq \theta \leq 225^\circ, (\text{transformação cosseno}) \\ -135^\circ < \theta < -45^\circ \text{ ou } 45^\circ < \theta < 135^\circ, (\text{transformação seno}) \end{cases} \quad (\text{A.16})$$

Ou seja, a fronteira angular entre a transformação seno e cosseno situa-se em  $\tan^2(\theta) = 1$ . Como constatado em (A.8) e (A.9), esse valor pode ser calculado por

$$\tan^2(\theta) = \tau^2 = \left( \frac{b_{ji}}{b_{ii}} \right)^2 \frac{D_j}{D_i}. \quad (\text{A.17})$$

Visto que  $\tau^2$  é exigido no cálculo de  $\cos^2(\theta)$ , e por consequência  $\overset{\circ}{D}_i$  e  $\overset{\circ}{D}_j$ , seu valor será computado obrigatoriamente. Aliás, conforme a relação estabelecida em (A.16), a decomposição da matriz de Givens dependerá do valor de  $\tau^2$ . Se  $\tau^2 \leq 1$ , a transformação cosseno será utilizada, caso contrário, será aplicada a transformação seno, com necessidade de alteração das variáveis

$$\sigma = \frac{1}{\sigma}, \quad \rho = \frac{1}{\rho} \quad \text{e} \quad \kappa^2 = \frac{1}{\tau^2}, \quad (\text{A.18})$$

sendo  $\sigma = b_{ji}/b_{ii}$ ,  $\rho = \sigma D_j/D_i$ .

Em alguns casos poderá ocorrer a situação em que nenhuma transformação seja permitida. Se  $b_{ii} = b_{ji} = 0$ , o resultado de (A.17) será indefinido. Neste contexto,  $D_n$  e  $B_n$  deverão ser mantidos.

A computação da Rotação Rápida de Givens para anulação do elemento  $b_{ji}$  é descrita no Algoritmo 2. A rotina apresentada por (MONTICELLI, 2012) assume que  $b_{ii} \neq 0$  em todas as etapas, o que deixa  $\tau^2$  definido em qualquer circunstância. A cada iteração, (A.18) é calculado para definição da transformada e atualização dos elementos de  $D$  e  $B$ .

---

#### Algoritmo 2 Rotação Rápida de Givens

---

```

1: rotina FASTGIVENS( $b_{ik}, b_{jk}, D_i, D_j$ )
2:    $\sigma \leftarrow b_{ji}/b_{ii}, \quad \rho \leftarrow \sigma D_j/D_i, \quad \tau^2 \leftarrow \sigma \rho$            ▷ Cálculo de  $\sigma, \rho$  e  $\tau^2$ 
3:   se  $\tau^2 \leq 1$ , então
4:      $c^2 \leftarrow 1/(1 + \tau^2)$                                                          ▷ Cálculo de  $\cos^2(\theta)$ 
5:      $D_i \leftarrow c^2 D_i, D_j \leftarrow c^2 D_j$                                        ▷ Cálculo de  $D_i$  e  $D_j$ 
6:     para ( $k = 1 : n$ ) faça
7:        $b_{ik} \leftarrow b_{ik} + \rho b_{jk}, \quad b_{jk} \leftarrow b_{jk} - \sigma b_{ik}$        ▷ Atualização de  $b_{ik}$  e  $b_{jk}$ 
8:     senão
9:        $\sigma \leftarrow 1/\sigma, \quad \rho \leftarrow 1/\rho, \quad \kappa^2 \leftarrow \sigma \rho$      ▷ Atualização de  $\sigma, \rho$  e  $\kappa^2$ 
10:       $s^2 \leftarrow 1/(1 + \kappa^2)$                                                        ▷ Cálculo de  $\sin^2(\theta)$ 
11:       $aux \leftarrow D_i, \quad D_i \leftarrow s^2 D_j, \quad D_j \leftarrow s^2 aux$        ▷ Cálculo de  $D_i$  e  $D_j$ 
12:      para ( $k = 1 : n$ ) faça
13:         $b_{ik} \leftarrow b_{jk} + \rho b_{ik}, \quad b_{jk} \leftarrow \sigma b_{jk} - b_{ik}$      ▷ Atualização de  $b_{ik}$  e  $b_{jk}$ 
14: retorna  $b_{ik}, b_{jk}, D_i, D_j$ 

```

---

O desenvolvimento numérico da decomposição  $QR$  baseada no Algoritmo 2 é apresentado no Anexo A deste documento.

**Exemplo A.3.1** *Seja  $A$  a matriz usada nos exemplos A.2.1 e A.2.2:*

$$A = \begin{bmatrix} 1 & 2 & 9 & 7 \\ 3 & 6 & 3 & 5 \\ 0 & 1 & 5 & 6 \\ 4 & 8 & 2 & 7 \end{bmatrix}.$$

*O procedimento para decomposição de  $A$  em fatores  $QR$  usando a Rotação Rápida de Givens segue a rotina do Algoritmo 2. A quantidade de etapas e o resultado final deverão coincidir com o Exemplo A.2.1.*

**Solução****Passo 1:** Eliminar  $B(2,1)$ .  $\sigma = 3$ ,  $\rho = 3$ ,  $\tau = 9$  (transformação seno).

$$\sigma = \frac{1}{3}, \rho = \frac{1}{3}, \tau^2 = \frac{1}{9}.$$

$$\text{sen}^2(\theta) = \frac{9}{10}, d_1 = \frac{9}{10}, d_2 = \frac{9}{10}.$$

$$D_1 = \begin{bmatrix} \frac{9}{10} & 0 & 0 & 0 \\ 0 & \frac{9}{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B_1 = \begin{bmatrix} \frac{10}{3} & \frac{20}{3} & 6 & \frac{22}{3} \\ 0 & 0 & -8 & -\frac{16}{3} \\ 0 & 1 & 5 & 6 \\ 4 & 8 & 2 & 7 \end{bmatrix}$$

**Passo 2:** Eliminar  $B(3,1)$ .  $\sigma = 0$ ,  $\rho = 0$ ,  $\tau^2 = 0$  (transformação cosseno).

$$\text{cos}^2(\theta) = 1, d_1 = \frac{9}{10}, d_3 = 1.$$

$$D_2 = \begin{bmatrix} \frac{9}{10} & 0 & 0 & 0 \\ 0 & \frac{9}{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B_2 = \begin{bmatrix} \frac{10}{3} & \frac{20}{3} & 6 & \frac{22}{3} \\ 0 & 0 & -8 & -\frac{16}{3} \\ 0 & 1 & 5 & 6 \\ 4 & 8 & 2 & 7 \end{bmatrix}$$

**Passo 3:** Eliminar  $B(4,1)$ .  $\sigma = \frac{6}{5}$ ,  $\rho = \frac{4}{3}$ ,  $\tau^2 = \frac{8}{5}$  (transformação seno).

$$\sigma = \frac{5}{6}, \rho = \frac{3}{4}, \tau^2 = \frac{5}{8}.$$

$$\text{sen}^2(\theta) = \frac{8}{13}, d_1 = \frac{8}{13}, d_4 = \frac{36}{65}.$$

$$D_3 = \begin{bmatrix} \frac{8}{13} & 0 & 0 & 0 \\ 0 & \frac{9}{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{36}{65} \end{bmatrix} \quad B_3 = \begin{bmatrix} \frac{13}{2} & 13 & \frac{13}{2} & \frac{25}{2} \\ 0 & 0 & -8 & -\frac{16}{3} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & -\frac{13}{3} & -\frac{3}{2} \end{bmatrix}$$

**Passo 4:** Eliminar  $B(3,2)$ .  $\sigma \rightarrow +\infty$ ,  $\rho \rightarrow +\infty$ ,  $\tau^2 \rightarrow +\infty$  (transformação seno).

$$\sigma = 0, \rho = 0, \tau^2 = 0.$$

$$\text{sen}^2(\theta) = 1, d_2 = 1, d_3 = \frac{9}{10}.$$

$$D_4 = \begin{bmatrix} \frac{8}{13} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{9}{10} & 0 \\ 0 & 0 & 0 & \frac{36}{65} \end{bmatrix} \quad B_4 = \begin{bmatrix} \frac{13}{2} & 13 & \frac{13}{2} & \frac{25}{2} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & 8 & \frac{16}{3} \\ 0 & 0 & -\frac{13}{3} & -\frac{3}{2} \end{bmatrix}$$

**Passo 5:** Eliminar  $B(4,2)$ .  $\sigma = 0$ ,  $\rho = 0$ ,  $\tau^2 = 0$  (transformação cosseno).

Neste caso, assim como no Passo 2, não há necessidade de anular o elemento. Portanto,

 $D_5 = D_4$  e  $B_5 = B_4$ .**Passo 6:** Eliminar  $B(4,3)$ .  $\sigma = -\frac{13}{24}$ ,  $\rho = -\frac{1}{3}$ ,  $\tau^2 = \frac{13}{72}$  (transformação cosseno).

$$\text{cos}^2(\theta) = \frac{72}{85}, d_3 = \frac{324}{425}, d_4 = \frac{2592}{5525}.$$

$$D_6 = \begin{bmatrix} \frac{8}{13} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{324}{425} & 0 \\ 0 & 0 & 0 & \frac{2592}{5525} \end{bmatrix} \quad B_6 = \begin{bmatrix} \frac{13}{2} & 13 & \frac{13}{2} & \frac{25}{2} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & \frac{85}{9} & \frac{35}{6} \\ 0 & 0 & 0 & \frac{25}{18} \end{bmatrix}$$

### Resultados

A triangularização da matriz  $B$  é gerada após 6 etapas. Para solucionar  $R$ , ainda é necessário calcular a raiz quadrada da matriz  $D_6$  e multiplicar o resultado por  $B_6$ .

Beneficiando-se da forma triangular de  $R$ , a matriz ortogonal  $Q$  pode ser calculada fazendo  $Q = AR^{-1}$ . Como a inversa de uma matriz triangular superior também possui essa característica, o cálculo dos elementos de  $R^{-1}$  é feito por substituição. Para este exemplo:

$$\begin{aligned} r_{11}^{-1} &= \frac{1}{r_{11}}; & r_{12}^{-1} &= -\frac{r_{12}r_{22}^{-1}}{r_{11}}; & r_{13}^{-1} &= -\frac{r_{12}r_{23}^{-1} + r_{13}r_{33}^{-1}}{r_{11}}; & r_{14}^{-1} &= -\frac{r_{12}r_{24}^{-1} + r_{13}r_{34}^{-1} + r_{14}r_{44}^{-1}}{r_{11}}; \\ r_{21}^{-1} &= 0; & r_{22}^{-1} &= \frac{1}{r_{22}}; & r_{23}^{-1} &= -\frac{r_{23}r_{33}^{-1}}{r_{22}}; & r_{24}^{-1} &= -\frac{r_{23}r_{34}^{-1} + r_{24}r_{44}^{-1}}{r_{22}}; \\ r_{31}^{-1} &= 0; & r_{32}^{-1} &= 0; & r_{33}^{-1} &= \frac{1}{r_{33}}; & r_{34}^{-1} &= -\frac{r_{34}r_{44}^{-1}}{r_{33}}; \\ r_{41}^{-1} &= 0; & r_{42}^{-1} &= 0; & r_{43}^{-1} &= 0; & r_{44}^{-1} &= \frac{1}{r_{44}}; \end{aligned}$$

Calculando  $Q = AR^{-1}$  e  $R = \sqrt{D_6}B_6$ , os fatores  $Q$  e  $R$  de  $A$  deste exemplo são:

$$Q = \begin{bmatrix} \frac{1}{\sqrt{26}} & 0 & \frac{4}{\sqrt{17}} & \frac{3}{\sqrt{442}} \\ \frac{3}{\sqrt{26}} & 0 & 0 & -\frac{17}{\sqrt{442}} \\ 0 & 1 & 0 & 0 \\ \frac{4}{\sqrt{26}} & 0 & -\frac{1}{\sqrt{17}} & \frac{12}{\sqrt{442}} \end{bmatrix} \quad e \quad R = \begin{bmatrix} \frac{26}{\sqrt{26}} & \frac{52}{\sqrt{26}} & \frac{26}{\sqrt{26}} & \frac{50}{\sqrt{26}} \\ 0 & 1 & 5 & 6 \\ 0 & 0 & \frac{34}{\sqrt{17}} & \frac{21}{\sqrt{17}} \\ 0 & 0 & 0 & \frac{20}{\sqrt{442}} \end{bmatrix}.$$

## A.4 Leitura Adicional

Fundamentos de álgebra linear são úteis para um número abundante de aplicações em todas as áreas científicas. Por isso, seria quase impossível citar toda literatura relacionada a este assunto. Restringindo para fontes relacionadas à solução de sistemas lineares e implementação numérica, pode-se complementar o conhecimento desenvolvido neste apêndice com a abordagem feita por (KONTOGHORGHES, 2000), (DATTA, 2004), (IPSEN, 2009) e (FORD, 2014). Este último mostra alguns desenvolvimentos numéricos usando MATLAB para gerar a matriz de Givens e realizar a decomposição  $QR$ .

Em algumas obras com abordagem específica para controle de sistemas lineares e estimativa de parâmetros foram identificados um rico conteúdo relacionado aos métodos de fatoração matricial apresentados neste apêndice. (KAILATH; SAYED; HASSIBI, 2000) mostram o desenvolvimento algébrico da rotação rápida de Givens de (GENTLEMAN, 1973). Já a versão modificada apresentada por (HAMMARLING, 1974), detalhada na Seção A.3, é devidamente explorada em um apêndice da obra de (MONTICELLI, 2012).

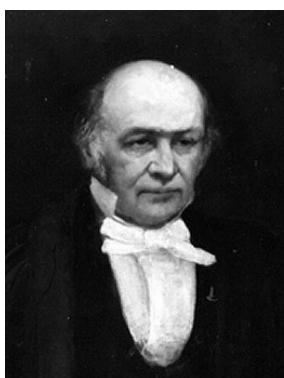
Alguns livros apresentam conteúdo mais abrangente, como por exemplo as obras edição única de (BEILINA; KARCHEVSKII; KARCHEVSKII, 2017) e (BJÖRCK, 2015). Para consulta de soluções numéricas com fundamentação matemática, a obra referência de (GOLUB; LOAN, 2013) fornece a apreciação de propriedades algébricas indispensáveis para este apêndice, além de uma extensiva cobertura de álgebra linear numérica.

---

## APÊNDICE B

### Quatérnios

---



William R. Hamilton  
(1805 - 1865)

*Sir William Rowan Hamilton descendeu de uma família de escritores e cientistas ilustres. Descrito como uma criança prodígio, ainda jovem aprendeu diversas línguas e se interessou pela matemática, tendo como base os escritos de Euclides, Clairaut, Newton, Laplace, Lagrange e principalmente a Geometria Analítica de Bartholomew Lloyd. Aos 18 anos, Hamilton ingressou no Trinity College, do qual se tornou professor de astronomia em 1827. Publicou diversos estudos no campo da matemática, física e astronomia. Em 1843, com um lampejo criativo enquanto caminhava em direção à Ponte Broughman em Dublin, Hamilton concebeu sua descoberta mais renomada, os quatérnios.*

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

$$\begin{aligned} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 &= -1 \\ \mathbf{ij} = -\mathbf{ji} &= \mathbf{k} \\ \mathbf{jk} = -\mathbf{kj} &= \mathbf{i} \\ \mathbf{ki} = -\mathbf{ik} &= \mathbf{j} \end{aligned} \quad (\text{B.2})$$

$$q = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \quad (\text{B.3})$$

## B.1 Descrição de Orientação

$$q = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{u} \operatorname{sen}\left(\frac{\theta}{2}\right) \end{bmatrix} \quad (\text{B.4})$$

## B.2 Álgebra dos Quatérnios

A operação de multiplicação entre dois quatérnios deve ser definida de forma que a relação fundamental (B.2) seja satisfeita. A não comutatividade do produto das bases vetoriais imaginárias reflete no produto dos quatérnios. Em geral, para  $q$  e  $p \in \mathbb{H}$ ,  $qp \neq pq$ . A operação de multiplicação entre dois quatérnios deve ser definida de forma que a relação fundamental (B.2) seja satisfeita. A não comutatividade do produto das bases vetoriais imaginárias reflete no produto dos quatérnios. Em geral, para  $q$  e  $p \in \mathbb{H}$ ,  $qp \neq pq$ . A operação de multiplicação entre dois quatérnios deve ser definida de forma que a relação fundamental (B.2) seja satisfeita. A não comutatividade do produto das bases vetoriais imaginárias reflete no produto dos quatérnios. Em geral, para  $q$  e  $p \in \mathbb{H}$ ,  $qp \neq pq$ . A operação de multiplicação entre dois quatérnios deve ser definida de forma que a relação fundamental (B.2) seja satisfeita. A operação de multiplicação entre dois quatérnios

### B.2.1 Definições e Operações Aritméticas

#### Igualdade de Quatérnios

Dois quatérnios são ditos iguais, se e somente se ambos possuírem exatamente os mesmo componentes escalar e vetorial.

$$q = p \quad \Leftrightarrow \quad q_0 = p_0 \quad \text{e} \quad \mathbf{q} = \mathbf{p}.$$

#### Adição e Subtração

A soma dos quatérnios  $q$  e  $p$  é definida pela adição dos componentes correspondentes, ou seja,

$$q + p = (q_0 + p_0) + (q_1 + p_1)\mathbf{i} + (q_2 + p_2)\mathbf{j} + (q_3 + p_3)\mathbf{k}.$$

O quatérnio oposto ou inverso aditivo é denotado por  $-q$ , o qual cada componente é oposto ao seu correspondente de  $q$ . Portanto, a operação de subtração ocorre pela adição do quatérnio oposto.

$$q - p = q + (-p).$$

Para ambas operações, são preservadas as propriedades associativa e comutativa dos cálculos baseados em números reais. Além disso, o conjunto dos quatérnios é fechado sob adição/subtração, isto é, a soma e a diminuição entre dois quatérnios resulta em outro quatérnio.

### Multiplicação

A operação de multiplicação entre dois quatérnios deve ser definida de forma que a relação fundamental (B.2) seja satisfeita. A não comutatividade do produto das bases vetoriais imaginárias reflete no produto dos quatérnios. Em geral, para  $q$  e  $p \in \mathbb{H}$ ,  $qp \neq pq$ .

Usando a regra distributiva para multiplicação algébrica geral, a tarefa de multiplicar  $q$  e  $p$ , nesta ordem, torna-se mais espontânea, embora trabalhosa.

$$\begin{aligned}
 q \otimes p &= (q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k})(p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}) \\
 &+ q_0p_0 + q_0p_1\mathbf{i} + q_0p_2\mathbf{j} + q_0p_3\mathbf{k} \\
 &+ q_1p_0\mathbf{i} - q_1p_1 + q_1p_2\mathbf{k} - q_1p_3\mathbf{j} \\
 &+ q_2p_0\mathbf{j} - q_2p_1\mathbf{k} - q_2p_2 + q_2p_3\mathbf{i} \\
 &+ q_3p_0\mathbf{k} + q_3p_1\mathbf{j} - q_3p_2\mathbf{i} - q_3p_3 \tag{B.5} \\
 &= \begin{bmatrix} q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3 \\ q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2 \\ q_0p_2 - q_1p_3 + q_2p_0 + q_3p_1 \\ q_0p_3 + q_1p_2 - q_2p_1 + q_3p_0 \end{bmatrix}
 \end{aligned}$$

No caso da multiplicação de um escalar por um quatérnio, o procedimento matemático é mais curto. Por exemplo, seja  $\alpha$  a parte real de um quatérnio  $q$ , tal que  $\text{Im } m(q) = 0$  e  $p$  um quatérnio qualquer. Aplicando  $q$  e  $p$  na Equação (B.5), obtém-se

$$\begin{aligned}
 q \otimes p &= (\alpha + \mathbf{0})(p_0 + \mathbf{p}) \\
 &= \alpha p_0 + \alpha p_1\mathbf{i} + \alpha p_2\mathbf{j} + \alpha p_3\mathbf{k}
 \end{aligned}$$

Portanto, para a multiplicação de um escalar por um quatérnio, basta multiplicar o valor escalar por cada componente quaterniônica.

Algumas propriedades envolvendo a operação de multiplicação de quatérnios são úteis. Em resumo,

$$\begin{array}{llll}
 qp \neq pq; & \alpha q = q\alpha; & p(\alpha q + \beta r) = \alpha pq + \beta pr; & p(q + r) = pq + pr; \\
 1 \otimes q = q; & p(qr) = (pq)r; & (\alpha p + \beta q)r = \alpha pr + \beta qr; & (p + q)r = pr + qr.
 \end{array}$$

### Conjugado Complexo

Um conceito algébrico importante para o uso do quatérnio como operador rotacional, o conjugado complexo é simbolizado por  $q^*$ . Assim como estabelecido para o conjugado de um número complexo, a parte imaginária de  $q$  é modificada ao seu valor oposto, portanto

$$q^* = q_0 - \mathbf{q} = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}.$$

Como consequência, a parte real de  $q$  pode ser definida no cálculo

$$\operatorname{Re} e(q) = \frac{q + q^*}{2}.$$

Algumas propriedades são válidas na definição do conjugado complexo dos quatérnios. Sejam  $q$  e  $p \in \mathbb{H}$  e  $\alpha \in \mathbb{R}$ , então:

$$(q^*)^* = q; \quad (qp)^* = p^*q^*; \quad qq^* = q^*q; \quad (q + p)^* = q^* + p^*; \quad (\alpha q)^* = \alpha q^*.$$

### Norma

Em algumas literaturas, a norma de um quatérnio é chamada de valor absoluto, módulo ou comprimento de  $q$ . Sua definição tem o mesmo significado de uma norma Euclidiana considerando o espaço vetorial em  $\mathbb{R}^4$ .

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}.$$

Algumas relações da norma de um quatérnio são válidas:

$$\|q\| = \sqrt{qq^*}; \quad \|qp\| = \|q\|\|p\|; \quad \|q\| = \|q^*\|.$$

Como enunciado na Seção B.1 deste apêndice, o quatérnio que descreve a orientação espacial necessariamente precisa ter norma unitária. Para estes casos, será designado  $\mathbb{H}_1$  para representar o conjunto dos quatérnio unitários.

### Inversa

A fundamentação algébrica desenvolvida anteriormente, sustentada pelas definições e propriedades de conjugado complexo e norma quaterniônica, será usada na construção do inverso multiplicativo de um quatérnio. Por definição, o inverso  $q^{-1}$  é tal que

$$q^{-1}q = qq^{-1} = 1.$$

Pela propriedade da multiplicação do conjugado complexo  $q^*$  à direita de  $q$ ,

$$q^{-1}qq^* = q^*qq^{-1} = q^*.$$

E, sendo a norma de  $q$  definida como  $\|q\| = \sqrt{qq^*}$ , então

$$q^{-1} = \frac{q^*}{\|q\|^2}, \tag{B.6}$$



para todo quatérnio  $q$  não nulo.

A Equação (B.6) fornece um princípio que será usado como base do quatérnio como operador rotacional. Se  $q$  for um quatérnio unitário, ou seja  $\|q\| = 1$ , então seu inverso multiplicativo será simplesmente seu conjugado complexo  $q^*$ . A interpretação geométrica de rotação usando quatérnios, contida na Seção B.3 deste apêndice, reforça a importância da equivalência entre  $q^{-1}$  e  $q^*$  para composição de rotações em  $\mathbb{R}^3$ .

### Divisão

No desenvolvimento algébrico de algum algoritmo que utilize quatérnios, como no Capítulo 3 deste documento, muitas vezes será necessário realizar operações de divisão entre quatérnios. Devido à sua natureza não comutativa para a multiplicação, o cálculo de divisão entre os quatérnios  $q$  e  $p$  nesta ordem é definida como

$$\begin{aligned} q \setminus p &= p^{-1}q, & \text{divisão à esquerda,} \\ q/p &= qp^{-1}, & \text{divisão à direita.} \end{aligned}$$

## B.2.2 Representação Matricial

### Matriz em 2x2

$$Z = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \quad (\text{B.7})$$

$$q = \begin{bmatrix} Z & W \\ -W^* & Z^* \end{bmatrix} \equiv \left[ \begin{array}{cc|cc} q_0 + q_1\mathbf{i} & q_2 + q_3\mathbf{i} \\ -q_2 + q_3\mathbf{i} & q_0 - q_1\mathbf{i} \end{array} \right] \quad (\text{B.8})$$

$$q = Z + W\mathbf{j}$$

### Matriz em 4x4

$$q = \left[ \begin{array}{cc|cc} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ \hline -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{array} \right] \quad (\text{B.9})$$

$$q \otimes p = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & -q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (\text{B.10})$$

Outra forma

$$q \otimes p = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & -q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (\text{B.11})$$

$$q = \left[ \begin{array}{c|c} q_0 & -\mathbf{q}^T \\ \hline \mathbf{q} & q_0 I_3 + [\mathbf{q} \times] \end{array} \right] \quad (\text{B.12})$$

tal que  $[\mathbf{q} \times]$  é a matriz antissimétrica de produto vetorial

$$\mathbf{q} \times \mathbf{p} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ q_1 & q_2 & q_3 \\ p_1 & p_2 & p_3 \end{vmatrix} = \begin{bmatrix} q_2 p_3 - q_3 p_2 \\ q_3 p_1 - q_1 p_3 \\ q_1 p_2 - q_2 p_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}}_{[\mathbf{q} \times]} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (\text{B.13})$$

$$\omega(q) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \quad \text{e} \quad \Xi(q) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}. \quad (\text{B.14})$$

$$\mathcal{L}_{\mathcal{G}} q(t) = \mathcal{G} q^*(t)$$

### B.2.3 Diferenciação no Tempo

A taxa de movimento rotacional do quadro local  $\{\mathcal{L}\}$  em relação ao quadro global  $\{\mathcal{G}\}$  é calculada pela derivada do quatérnio correspondente.

$$\mathcal{L}_t \dot{q} = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left( \mathcal{L}_{t+\Delta t} q - \mathcal{L}_t q \right). \quad (\text{B.15})$$

O uso do quatérnio como operador rotacional é apresentado detalhadamente na Seção B.3 deste apêndice. Por enquanto, para o alinhamento do quadro  $\{\mathcal{L}_t\}$  em  $\{\mathcal{L}_{t+\Delta t}\}$ , mantendo o referencial global  $\{\mathcal{G}\}$ , é usada o produto entre quatérnios como

$$\mathcal{L}_{t+\Delta t} q = \mathcal{L}_{t+\Delta t} q \otimes \mathcal{L}_t q. \quad (\text{B.16})$$

A linearização da relação trigonométrica, estabelecida na Equação (B.4), para um ângulo infinitesimal fornece um operador rotacional linear na forma

$$\mathcal{L}_{t+\Delta t} q = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{u} \text{sen}\left(\frac{\theta}{2}\right) \end{bmatrix} \approx \begin{bmatrix} 1 \\ \frac{1}{2} \delta \boldsymbol{\theta} \end{bmatrix}, \quad (\text{B.17})$$

tal que  $\delta \boldsymbol{\theta} = \mathbf{u} \theta$  seja um vetor em  $\mathbb{R}^3$  que contém a direção do eixo de rotação e a magnitude do ângulo de rotação. Assim, a velocidade angular do quadro local  $\{\mathcal{L}\}$  é calculada por

$$\boldsymbol{\omega} = \lim_{\Delta t \rightarrow 0} \frac{\delta \boldsymbol{\theta}}{\Delta t}. \quad (\text{B.18})$$

Então, o cálculo da variação instantânea de  $\frac{\mathcal{L}}{\mathcal{G}}q$  no tempo é realizado a partir do conjunto de Equações (B.15-B.18):

$$\begin{aligned}\frac{\mathcal{L}^t}{\mathcal{G}}\dot{q} &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left( \frac{\mathcal{L}^{t+\Delta t}}{\mathcal{L}^t} q \otimes \frac{\mathcal{L}^t}{\mathcal{G}} q - 1 \otimes \frac{\mathcal{L}^t}{\mathcal{G}} q \right) \\ &\approx \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left( \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \otimes \frac{\mathcal{L}^t}{\mathcal{G}} q \\ &\approx \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \otimes \frac{\mathcal{L}^t}{\mathcal{G}} q.\end{aligned}\quad (\text{B.19})$$

Baseado em (B.12), o cálculo da derivada do quatérnio no tempo pode ser expressa por

$$\frac{\mathcal{L}^t}{\mathcal{G}}\dot{q} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}) \frac{\mathcal{L}^t}{\mathcal{G}} q, \quad (\text{B.20})$$

sendo  $\boldsymbol{\Omega}(\boldsymbol{\omega})$  a matriz antissimétrica

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 0 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (\text{B.21})$$

Caso seja conveniente, a derivada do quatérnio ainda pode ser implementada segundo (B.11) como

$$\frac{\mathcal{L}^t}{\mathcal{G}}\dot{q} = \left[ \begin{array}{c|c} \frac{\mathcal{L}}{\mathcal{G}}q_0 & -\frac{\mathcal{L}}{\mathcal{G}}\mathbf{q}^T \\ \hline \frac{\mathcal{L}}{\mathcal{G}}\mathbf{q} & \frac{\mathcal{L}}{\mathcal{G}}q_0 I_3 + [\frac{\mathcal{L}}{\mathcal{G}}\mathbf{q} \times] \end{array} \right] \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}. \quad (\text{B.22})$$

$$\mathcal{L}(q) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \quad (\text{B.23})$$

$$\mathcal{R}(q) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}. \quad (\text{B.24})$$

$$\frac{\mathcal{G}}{\mathcal{B}}\mathbf{C}(q(t)) = I_{3 \times 3} + 2[\mathbf{q} \times] - 2\mathbf{q}\mathbf{q}^T \quad (\text{B.25})$$

### B.3 Operador Rotacional

$$\mathbf{R}(q) = \begin{bmatrix} 1 - 2q_3^2 - 2q_2^2 & 2(q_2q_1 + q_3q_0) & 2(q_3q_1 - q_2q_0) \\ 2(q_2q_1 - q_3q_0) & 1 - 2q_3^2 - 2q_1^2 & 2(q_3q_2 + q_1q_0) \\ 2(q_3q_1 + q_2q_0) & 2(q_3q_2 - q_1q_0) & 1 - 2q_2^2 - 2q_1^2 \end{bmatrix} \quad (\text{B.26})$$



# Anexos



---

# ANEXO A

## Repositório GitHub

---

### A.1 Decomposições Matriciais

#### Arquivos

- ❑ `givens_rotation_QR.m`
- ❑ `givens_rotation_QL.m`
- ❑ `givens_rotation_RQ.m`
- ❑ `givens_rotation_LQ.m`

#### Descrição

Esta pasta contém quatro arquivos de funções desenvolvidas em MATLAB que realizam a decomposição matricial de uma matriz quadrada usando Rotação de Givens (Apêndice A), originando um produto entre uma matriz ortogonal  $Q$  e uma matriz triangular superior  $R$  ou inferior  $L$ . A ordem dos fatores matriciais resultantes dependerá da função executada, assim como o tipo de matriz triangular.

A sequência para anulação das células da matriz é fundamental no procedimento numérico. Por exemplo, para a decomposição  $QR$ , inicialmente é necessário anular todos os elementos da primeira coluna que estão abaixo de  $A(1, 1)$ . Depois, todos elementos da segunda coluna abaixo de  $A(2, 2)$ . Esses passos são seguidos até a coluna  $n - 1$ , em que um único elemento,  $A(n - 1, n)$ , é anulado. Para as funções numéricas desenvolvidas, a sucessão dos elementos que serão anulados seguem as sequências mostradas na Figura 35. É importante ressaltar que a sequência primária, representada pelas setas internas, não

precisa necessariamente ser a mesma das funções, desde que todos os elementos de interesse sejam anulados antes de saltar para a próxima coluna, no caso das decomposições  $QR$  e  $RQ$ , ou para a próxima linha, no caso das decomposições  $QL$  e  $LQ$ .

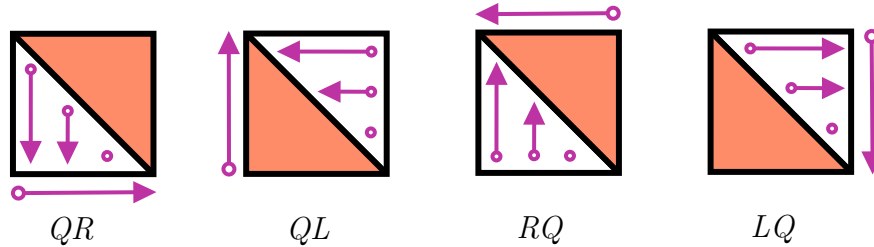


Figura 35 – Sequência usada nas funções MATLAB para anulação dos elementos nas decomposições  $QR$ ,  $QL$ ,  $RQ$  e  $LQ$ .

## A.2 Fast Givens Rotation

### Arquivos

- ❑ `FastGivens_QR.m`
- ❑ `FastGivens.m`

### Descrição

Esta pasta contém dois arquivos de funções desenvolvidas em MATLAB. A primeira é responsável por realizar um pré-escalamento na matriz  $A$  e organizar a sequência de rotações para gerar os fatores  $Q$  e  $R$ . A sequência de rotações segue a estratégia para decomposição  $QR$ , apresentada na Figura 35. O segundo arquivo contém a implementação do Algoritmo 2, tendo como entradas as linhas  $b_i$  e  $b_j$  de  $B$ , os elementos  $D_i$  e  $D_j$  de  $D$  e a coluna  $k$  referente ao elemento  $b_{jk}$  que será anulado. Em cada rotação,  $b_i$ ,  $b_j$ ,  $D_i$  e  $D_j$  são atualizados. De forma hierárquica, a função `FastGivens.m` é solicitada por `FastGivens_QR.m` em cada processo de anulação do elemento  $b_{jk}$ . Ao final, as matrizes  $Q$  e  $R$  são calculadas como no *Exemplo A.3.1*,  $R = \sqrt{D}B$  e  $Q = AR^{-1}$ .