



Universidade Estadual de Londrina
Centro de Tecnologia e Urbanismo
Departamento de Engenharia Elétrica

SIMULAÇÃO E IMPLEMENTAÇÃO DE UM
DETERMINADOR DE ATITUDE EM TEMPO REAL
UTILIZANDO SENSORES MICROFABRICADOS

Francisco Granziera Júnior

Marcelo Carvalho Tosin
Orientador

Roberto V. F. Lopes
Co-orientador

Banca Examinadora

Marcelo Carvalho Tosin - Presidente

Roberto V. F. Lopes - INPE/São José dos Campos

Taufik Abrão - UEL/Londrina

*Dissertação submetida ao Departamento de Engenharia Elétrica
da Universidade Estadual de Londrina, para preenchimento
dos pré-requisitos parciais para obtenção do título de
Mestre em Engenharia Elétrica*

Londrina, 25 de Janeiro de 2006



Universidade Estadual de Londrina
Centro de Tecnologia e Urbanismo
Departamento de Engenharia Elétrica

SIMULAÇÃO E IMPLEMENTAÇÃO DE UM
DETERMINADOR DE ATITUDE EM TEMPO REAL
UTILIZANDO SENSORES MICROFABRICADOS

Francisco Granziera Júnior

Londrina, 25 de Janeiro de 2006

**Catálogo na Publicação Elaborada pela Divisão de Processos Técnicos
da Biblioteca Central da Universidade Estadual de Londrina**

Dados Internacionais de Catalogação-na-Publicação (CIP)

G765s Granziera Júnior, Francisco.

Simulação e implementação de um determinador de atitude em tempo real utilizando sensores microfabricados / Francisco Granziera Júnior. – Londrina, 2006.
ix, 248f.

Orientador : Marcelo Carvalho Tosin.

Co-orientador: Roberto V. F. Lopes.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Estadual de Londrina, 2006.

Inclui bibliografia.

1. Engenharia elétrica – Teses. 2. Sensores microfabricados – Engenharia elétrica – Teses. 3. Sistemas eletrônicos – Teses. I. Tosin, Marcelo Carvalho. II. Lopes, Roberto V. F. III. Universidade Estadual de Londrina. IV. Título.

CDU 621.391

*à minha família: Prof^a Maria Luiza, seu Chico, Fran, Giovani e Emer
por existirem e resistirem comigo quando precisei.*

GRANZIERA Jr., Francisco. *Simulação e Implementação de um Determinador de Atitude em Tempo Real Utilizando Sensores Microfabricados*. 2006. Dissertação de Mestrado (Mestrado em Sistemas Eletrônicos) – Universidade Estadual de Londrina.

Resumo

Este trabalho é sobre o estudo, simulação, implementação e testes de um sistema de determinação de atitude baseado em sensores microfabricados. São apresentados brevemente conceitos de referências e sistemas de coordenadas para atitude e exploradas as formas matemáticas mais comuns para expressar a atitude, como a matriz de atitude, os ângulos de Euler e o quatérnion, bem como as relações entre as representações. Dois algoritmos de determinação da atitude são explorados com profundidade: o TRIAD e o QUEST. O método de filtragem de Kalman é introduzido como uma forma de unir informações de sensores de referências e giros para contornar limitações físicas e elétricas dos sensores. São feitas simulações para testar a convergência do filtro de Kalman e uma simulação gráfica é criada para ilustrar efeito do filtro sobre os quatérnions de atitude. Também são analisados os sensores utilizados e o hardware de aquisição e processamento da atitude construído. Alguns pontos importantes do hardware são discutidos e as funções do software em linguagem C são apresentadas e comentadas. Também são apresentados resultados de ensaios realizados em uma mesa de rotação tridimensional e em uma mesa unidimensional de mancal a ar. Finalmente, são discutidas as conclusões sobre todo o trabalho realizado, enfatizando os acertos e os erros sugerindo caminhos para a continuação do mesmo.

Abstract

This work is about the study, simulation, implementation and tests of a attitude determination system based on MEMS sensors. It briefly introduces concepts about attitude references and coordinate frames. The most common mathematical forms for attitude representation are shown, such as attitude matrix, Euler angles and the Quaternion as well as the relationships between these representations. Two algorithms for attitude determination are deeply explored: the TRIAD and QUEST algorithms. It is introduced the Kalman Filter method as a way to join the reference sensors information with gyros information and to contour both physical and electrical limitations of the MEMS sensors. Basic simulations were done to verify the Kalman Filter convergence. A graphical demonstration was created to visualize the Filter effect in the simulation. The sensors, the acquisition scheme and the hardware build for the attitude determination system are analyzed. Some important points about the hardware are discussed. A description of a test using a three dimensional table and its results are also shown. Tests and results at three dimensional rotation tables are also shown. Finally, conclusions are exposed and future works are proposed.

Agradecimentos

- A Deus pela sabedoria, conhecimento e boas companhias que tem me dado por toda minha vida.
- À minha família que me apoiou nos momentos que mais precisei.
- Aos meus colegas e amigos de trabalho, especialmente ao Marcelo, à Marília e ao Fernando, e a todos os outros que de alguma forma deram apoio moral, mental e braçal e contribuíram para esta realização.
- Aos meus orientadores, Marcelo e Roberto, que não mediram esforços para viabilizar idéias e recursos. E também pela liberdade dada à mim para o desenvolvimento do meu próprio conhecimento.
- A todos os institutos de fomento que ajudaram (e ainda ajudam) com recursos para o projeto iniciado nesta dissertação. Especialmente as instituições como a CAPES pela bolsa de mestrado, a AEB – Agência Espacial Brasileira – pela iniciativa de projetos como o UNIESPAÇO.
- À UEL e funcionários, pelos recursos de laboratório e recursos para viagens ao INPE em São José dos Campos.
- Ao INPE, especialmente ao Dr. Paulo G. Milani e ao Dr. Roberto V. F. Lopes, pela disponibilização de avançados recursos de ensaios do LabSim e auxílio imprescindível para realização dos ensaios.
- Às empresas de Londrina e região, como a Rota Indústria Ltda., que 5 anos antes, acreditou e creditou a essa idéia; e também a INTUEL que acolhe indiscriminadamente nossas **atitudes** empreendedoras.

Sumário

Resumo	vii
Abstract	ix
Agradecimentos	xi
Sumário	xiii
Lista de Acrônimos	xix
Lista de Figuras	xxi
Lista de Tabelas	xxvii
Lista de Símbolos	xxix
Capítulo 1 – Introdução	1
1.1 – O Tema	1
1.2 – A Meta	2
1.3 – A Organização	3
Capítulo 2 – Sistemas de Coordenadas e Referências para a Atitude	5
2.1 – Introdução	5
2.2 – Sistemas de Coordenadas	5
2.2.1 – Sistema de coordenadas horizontal local	6
2.2.2 – Sistema de coordenadas celestial ou equatorial	7
2.3 – Vetores de Referência	8
2.3.1 – As estrelas	9
2.3.2 – O Sol	9
2.3.3 – Albedo	10

2.3.4 – Campo gravitacional	10
2.3.5 – Campo magnético terrestre	13
2.4 – Conclusões e considerações sobre aplicações	21
 Capítulo 3 – Representações da Atitude	 23
3.1 – Introdução	23
3.2 – Atitude por Ângulos de Euler	23
3.2.1 – Teorema e definições	23
3.2.2 – A sequência de Euler	24
3.2.3 – Problemas com ângulos de Euler	26
3.3 – Atitude por Quatérnion	26
3.3.1 – Introdução ao quatérnion	27
3.3.2 – Definição das operações com quatérnions	28
3.3.3 – Representando um vetor no domínio dos quatérnions	32
3.3.4 – O quatérnion representando uma rotação	32
3.3.5 – Derivada de um quatérnion	34
3.4 – Relações entre as representações	35
3.4.1 – Ângulos de Euler para quatérnion	35
3.4.2 – Quatérnion para ângulos de Euler	38
3.5 – Comparação entre as Representações	39
 Capítulo 4 – Algoritmos para a Determinação da Atitude	 41
4.1 – Introdução	41
4.2 – O Algoritmo TRIAD (TRI-axis Attitude Determination)	42
4.2.1 – A Matriz de atitude do TRIAD	42
4.2.2 – A Matriz de covariância do TRIAD	46
4.2.3 – Codificação do TRIAD	65
4.2.4 – Sobre o TRIAD	76
4.3 – Algoritmo QUEST (QUaternion ESTimator)	77
4.3.1 – q-Método	77
4.3.2 – Quatérnion ótimo via QUEST	81
4.3.3 – A Matriz de covariância do QUEST	84
4.3.4 – Codificação do QUEST	87

4.4 – Outros algoritmos	95
4.5 – Considerações e Conclusões sobre o TRIAD e o QUEST	95
Capítulo 5 – Atitude por Intermédio de um Filtro de Kalman: Teoria e Simulação	99
5.1 – Introdução	99
5.2 – Filtro de Kalman	100
5.2.1 – Breve histórico	100
5.2.2 – Equações gerais	100
5.3 – Equações de Rotação	105
5.4 – Modelos para os Sensores	107
5.4.1 – Modelos para girômetros	107
5.4.2 – Sensores de atitude	109
5.5 – A Equação de Estado	109
5.5.1 – Predição	110
5.6 – A Matriz de Covariância dos Erros do Estado	111
5.6.1 – Predição	112
5.6.2 – Filtragem	113
5.6.3 – Matriz de covariância	113
5.7 – Propriedades da Matriz $\Xi(\tilde{q})$	114
5.8 – Representação da Matriz de Covariância na forma Reduzida	119
5.8.1 – Predição	119
5.8.2 – Filtragem	122
5.9 – Formatando as Equações para Simulação do Filtro de Kalman	122
5.9.1 – Introdução	122
5.9.2 – Algoritmo para simulação – Mecanização do filtro de Kalman	123
5.9.3 – Considerações sobre os intervalos	128
5.9.4 – Fluxograma da simulação	129
5.10 – Simulação do Filtro	130
5.10.1 – Código da simulação	131
5.10.2 – Funções de propagação e filtragem	134
5.10.3 – Funções de simulação dos sensores	138
5.10.4 – Simulação com atitude em repouso	140

5.10.5 – Simulação sob giro entorno de vetor fixo	143
5.11 – Recurso Gráfico de Visualização	147
5.11.1 – Introdução	147
5.11.2 – Fluxograma do programa	147
5.11.3 – Visualização	148
5.12 – Considerações e Conclusões sobre a Filtragem e Simulação	149
 Capítulo 6 – Implementação do Determinador de Atitude: projeto, programação e testes	 151
6.1 – Introdução	151
6.2 – Considerações sobre os Elementos de Hardware	152
6.2.1 – Sensores	152
6.2.2 – Microcontroladores	152
6.2.3 – Conversores AD	153
6.2.4 – Amplificadores	154
6.2.5 – Memória de dados não-voláteis	154
6.2.6 – Interfaces de comunicação	155
6.2.7 – Componentes discretos	155
6.2.8 – Conectores	156
6.2.9 – Considerações finais sobre a escolha dos componentes	156
6.3 – Diagrama de Blocos Ampliado – Visão geral do Sistema	156
6.4 – Os Sensores	157
6.4.1 – Sensor de aceleração ADXL202	158
6.4.2 – Sensor GYROSTAR	165
6.4.3 – Sensor magnético HMC2003	177
6.5 – Set/Reset	180
6.6 – Conversores Analógico-Digital	186
6.6.1 – Conversor analógico-digital ADS8344	186
6.6.2 – Conversor analógico-digital ADS8364	187
6.7 – O Microcontrolador	188
6.8 – Software Básico	189
6.8.1 – Inicializações	189
6.8.2 – Comunicação	190

6.8.3 – Aquisição	191
6.8.4 – Interrupção do timer	194
6.8.5 – Rotina ler e enviar pela serial – rotina de teste	196
6.9 – Implementação de Algoritmos de Atitude	197
6.9.1 – Implementação do algoritmo TRIAD	197
6.9.2 – Código C para o algoritmo TRIAD	200
6.10 – Ensaaios e Testes	202
6.10.1 – Testes preliminares – seguidor de movimentos	203
6.10.2 – Ensaio em mesa de rotação	203
6.12 – Considerações e Conclusões	227
 Capítulo 7 – Sumário e Conclusões	 229
 Referências Bibliográficas	 231
 Anexo I – Código do Programa para Representação da Atitude Graficamente	 235
 Anexo II – Esquemático do Hardware	 243

Lista de Acrônimos

AD	<i>Analog to Digital</i> – Analógico para Digital
ADC	<i>Analog Digital Converter</i> – Conversor Analógico-Digital (AD)
ASIC	<i>Application Specific Integrated Circuit</i> – Circuito Integrado para Aplicação Específica
BDM	<i>Background Debug Mode</i> – Modo de Depuração de Fundo
CCD	<i>Charge Couple Device</i> – Dispositivo de Carga Acoplada
CHAMP	<i>CHallenging Mini-Satellite Payload</i> – Desafiando Mini-satélites de Carga Útil
CISC	<i>Complex Instruction Set Computer</i> – Conjunto de Instruções Computacionais Complexas
ESOQ	<i>Estimator of the Optimum Quaternion</i> – Estimador do Quatérnion Ótimo
FK	Filtro de Kalman
FKE	Filtro de Kalman Estendido
FOAM	<i>Fast Optimal Attitude Matrix</i> – Matriz Rápida de Atitude Ótima
FOG	<i>Fiber Optic Gyro</i> – Giro de Fibra Óptica
FPGA	<i>Field Programmable Gate Array</i> – Matriz de Portas Programáveis por Campo
GND	Abreviação de <i>ground</i> (referência, 0 V, terra, massa)
GPIO	<i>Generic Pin Input/Output</i> – Pino Genérico de Entrada ou Saída
GPS	<i>Global Positional System</i> – Sistema de Posicionamento Global
GRACE	<i>Gravity Recovery and Climate Experiment</i> – Recuperação da Gravidade e Experimentação de Clima
HSC	<i>High Speed CMOS</i> – CMOS de Alta Velocidade
I2C	Inter-Integrated Circuit Bus – Protocolo de comunicação serial a dois fios
IR	<i>Infrared</i> – Infravermelho
IrDA	<i>Infrared Data Access</i> – Interface Infravermelho de Acesso a Dados
IRGF	<i>International Geomagnetic Reference Field</i> – Campo de Referência Geomagnético Internacional
LED	<i>Light Emission Diode</i> – Diodo Emissor de Luz
MARG	Magnetômetros, Acelerômetros e Giros (conjunto)

MASCO	Mascara Codificada
MAVS	<i>Modular Acoustic Velocity System</i> – Sensor de Velocidade Acústica Modular
MEMS	<i>Micro Eletro-Mechanic System</i> – Sistema Microeletromecânico
MOSFET	<i>Metal Oxide on Silicon Field Effect Transistor</i> – Transistor de Efeito de Campo
OpenGL	<i>Open Graphic Library</i> – Biblioteca Gráfica Aberta (em código)
PWM	<i>Pulse Width Modulation</i> – Modulação por Largura de Pulso
QUEST	<i>Quaternion Estimator</i> – Estimador do Quatérnion
RAM	<i>Random Access Memory</i> – Memória de Acesso Randômico
RLG	<i>Ring Laser Gyro</i> – Giro Laser em Anel
RMS	<i>Root Mean Square</i> – Raiz Quadrada da Média Quadrática
SCC	<i>Solar Compass Chip</i> – Compasso Solar Integrado
SMD	<i>Surface Mounting Devices</i> – Dispositivo de Montagem de Superfície
SPI	<i>Serial Paralell Interface</i> – Interface Paralelo Serial
SVD	<i>Singular Value Decomposition</i> – Decomposição por Valor Singular
TRIAD	<i>Tri-axis Attitude Determination</i> – Determinação de Atitude em Três Eixos
USB	<i>Universal Serial Bus</i> – Barramento Serial Universal
VHDL	<i>VHSIC Hardware Description Language</i> – Linguagem de Descrição de Hardware VHSIC
VHSIC	<i>Very High Speed Integrated Circuits</i> – Circuito Integrados de Velocidade Muito Alta
VI	<i>Virtual Instrument</i> – Instrumento Virtual
WMM	<i>World Magnetic Model</i> – Modelo Magnético Mundial

Lista de Figuras

Figura 2.1 – Sistema de coordenadas horizontal local.	6
Figura 2.2 – Sistema de coordenadas celestial.	8
Figura 2.3 – O formato de massa da terra visto pela missão CHAMP. Se a terra fosse uma distribuição esférica uniforme de massa, então a gravidade seria a mesma em todos os pontos da superfície terrestre.	12
Figura 2.4 – Variação da intensidade do campo gravitacional sobre a superfície do planeta Terra em relação à intensidade média de $9,8 \text{ m/s}^2$. As variações são dadas em mGal.	12
Figura 2.5 – Campo magnético entorno da na região de Londrina/PR/Brasil – Modelo WMM2005 – Intensidade do Campo em função da latitude e longitude.	20
Figura 2.6 – Campo magnético entorno da na região de Londrina/PR/Brasil – Modelo WMM2005 – Declinação do Campo em função da latitude e longitude.	20
Figura 2.7 – Campo magnético entorno da na região de Londrina/PR/Brasil – Modelo WMM2005 – Inclinação do Campo em função da latitude e longitude.	20
Figura 3.1 – Representação dos Ângulos de Euler.	24
Figura 4.1. Representação gráfica da rotação dada por matriz de atitude. A matriz A gira o sistema de coordenadas de referência para o sistema de coordenadas de observação.	43
Figura 4.2 – Versores e erros.	51
Figura 4.3 – Ilustração mostrando que \hat{W}_2 está no mesmo plano formado por \hat{s}_1 e \hat{s}_3 .	57
Figura 4.4 – Sequência de quatérnions de atitude dado por uma rotação em torno de um eixo qualquer.	74
Figura 4.5 – Sequência de quatérnions de atitude recuperados através das medidas dos vetores observados corrompidos por ruído gaussiano com $N(0; 0,05)$	75
Figura 4.6 – Erro de cada componente do quatérnion na comparação da atitude verdadeira com a atitude recuperada pelo algoritmo TRIAD.	75
Figura 4.7 – Componente vetorial do quatérnion calculada pelo TRIAD e plotada sobre superfície de uma esfera unitária.	76
Figura 4.8 – Componentes do quatérnion de atitude lido sequencialmente a partir da rotação sobre um eixo qualquer.	92
Figura 4.9 – Componentes do quatérnion recuperado a partir das leituras dos vetores observados corrompidos por ruído gaussiano.	93
Figura 4.10 – Erro de cada componentes do quatérnion na subtração da atitude verdadeira da recuperada pelo algoritmo QUEST.	93
Figura 4.11 – Componente vetorial do quatérnion estimado pelo algoritmo QUEST, plotada sobre superfície de uma esfera unitária.	94
Figura 4.12 – Erro quadrático médio acumulado. Em (a) erro médio acumulado do algoritmo QUEST. Em (b) o erro quadrático médio acumulado pelo algoritmo TRIAD.	95

Figura 4.13 – Erro quadrático médio acumulado para diferentes níveis de ruído. Em (a) tem-se o mesmo gráfico de 4.12, quando os níveis de ruído de cada sensor são idênticos; esse é o melhor caso de desempenho do QUEST. Em (b) considerou-se um sensor sendo muito superior ao outro sensor (10 vezes menos ruidoso); neste caso o desempenho de ambos os algoritmos foi praticamente o mesmo.	97
Figura 5.1 – Ordem dos eventos da filtragem e propagação.	129
Figura 5.2 – Fluxograma da simulação do filtro de Kalman.	130
Figura 5.3 – Componentes do quatérnion estimadas pelo filtro – simulação estática.	141
Figura 5.4 – Erros e desvio padrão das componentes do quatérnion.	142
Figura 5.5 – Componentes do <i>bias</i> do giro estimadas pelo filtro.	143
Figura 5.6 – Erros e desvio padrão das componentes do <i>bias</i> do giro.	143
Figura 5.7 – Componentes do quatérnion estimadas pelo filtro – simulação dinâmica.	144
Figura 5.8 – Convergência inicial das componentes do quatérnion estimadas pelo filtro.	145
Figura 5.9 – Erros e desvio padrão das componentes do quatérnion durante uma rotação.	145
Figura 5.10 – Componentes do <i>bias</i> do giro estimadas pelo filtro durante simulação de rotação.	146
Figura 5.11 – Erros e desvio padrão das componentes do <i>bias</i> do giro.	146
Figura 5.12 – Fluxograma da simulação gráfica.	148
Figura 5.13 – Visualização gráfica da simulação da atitude.	149
Figura 5.14 – Visualização gráfica da atitude – instantes 10 s e 11 s de uma simulação com erro na implementação.	149
Figura 6.1 – Diagrama básico de um sistema de determinação de atitude.	151
Figura 6.2 – Diagrama de blocos do hardware de determinação da atitude.	157
Figura 6.3 – Sensor de aceleração ADXL202 – Tamanho.	158
Figura 6.4 – Sensor de Aceleração ADXL202 – Diagrama interno.	158
Figura 6.5 – Acelerômetros de capacitância variável por deslocamento.	159
Figura 6.6 – Configuração mínima para o sensor de aceleração. Experiência para verificar o ruído de saída.	161
Figura 6.7 – Aquisição da aceleração via LabView. $f_s = 1$ kHz .	163
Figura 6.8 – Aquisição da aceleração via LabView. $f_s = 1$ kHz com posterior filtragem digital em $f_c = 1$ kHz .	165
Figura 6.9 – Sensores GYROSTAR 03JA	166
Figura 6.10 – Experiência com girômetro para verificar o ruído de saída e deriva.	167
Figura 6.11 – Dados adquiridos do sensor Murata 3JA. (a) Tensão de referência (b) Tensão de saída como sensor em repouso.	168
Figura 6.12 – Velocidade angular medida pelo sensor 3JA. Fator de escala médio 0,67 mV/°/s.	169
Figura 6.13 – Velocidade angular medida pelo sensor 3JA após filtragem para limitação da banda em 50 Hz. Fator de escala médio 0,67 mV/°/s.	170
Figura 6.14 – Sinal do girômetro e integral do sinal.	171

Figura 6.15 – Ilustração da necessidade de amplificação do sinal do girômetro 3JA da Murata.	172
Figura 6.16 – Circuito filtro ativo passa-faixa do sinal do giro 3JA sugerido pelo fabricante.	173
Figura 6.17 – Simulação do circuito da Figura 6.16.	173
Figura 6.18 – Saída do amplificador quando a entrada excursionsa entre seus possíveis extremos.	174
Figura 6.19 – Resposta em frequência do circuito amplificador passa-banda. Os pontos de corte estão em torno de 0,35 Hz e 1000 Hz.	174
Figura 6.20 – Solução de amplificação com zero em 2,5 V.	175
Figura 6.21 – Solução de amplificação com zero em 2,5 V.	176
Figura 6.22 – Medida real do giro para o circuito da Figura 6.21. Taxa de 1 kHz.	176
Figura 6.23 – Medida do Giro com e sem amplificação para estimação da deriva. Tempo de aquisição de 60 segundos. Taxa 1kHz.	177
Figura 6.24 – Sensor Híbrido HMC2003 – Contém uma pastilha com eixos X e Y e um partilha vertical para leitura no eixo Z. No DIP também estão incluídos amplificadores que ajustam a escala para 1V/Gauss.	178
Figura 6.25 – Circuito amplificador de ganho 8, não-inversor com frequência de corte em 50 Hz usado para amplificação do sinal do canal X dos magnetômetro.	170
Figura 6.26 – Processo do set/reset na membrana <i>permalloy</i> dos sensores que compõe o híbrido HMC2003.	181
Figura 6.27 – Circuito set/reset sugerido pelo fabricante.	182
Figura 6.28 – Formato das ondas de Set/Reset adquiridas por um osciloscópio nos pontos de entrada e saída do circuito.	183
Figura 6.29 – Esquemático de ensaio do sensor HMC2003	183
Figura 6.30 – Sinal adquirido do sensor HMC2003 sem amplificação e sem Set/Reset.	184
Figura 6.31 – Sinal adquirido do sensor HMC2003 com amplificação e com Set/Reset comparado ao da Figura 6.30.	185
Figura 6.32 – Sinal adquirido do sensor HMC2003 com amplificação e com Set/Reset plotado em função do tempo no eixo Z.	186
Figura 6.33 – Diagrama de blocos do conversor ADS8344.	187
Figura 6.34 – Diagrama de blocos do conversor ADS8364.	188
Figura 6.35 – Fluxograma implementado para algoritmo TRIAD.	199
Figura 6.36 – Cubo real e virtual atuando de forma mestre-escravo respectivamente.	203
Figura 6.37 – Ensaio em mesas de rotação. Cubo no interior da mesa de simulação de três eixos.	204
Figura 6.38 – A segunda foto trás todo o sistema, inclusive o <i>notebook</i> de aquisição sobre o prato da mesa de mancal a ar de um eixo.	204
Figura 6.39 – Componentes do quatérnion adquiridas durante rotação em passos de 30° entorno da gravidade (eixo z).	209
Figura 6.40 – Componente angular do quatérnion plotada em graus.	209

Figura 6.41 – Componentes do quatérnion durante giro contínuo a 20 °/s entorno do eixo z.	210
Figura 6.42 – Componente escalar do quatérnion convertida em graus para o ensaio de 20°/s contínuo entorno do eixo z.	211
Figura 6.43 – Componente vetorial do quatérnion normalizada. A direção do vetor se movimenta.	211
Figura 6.44 – Componentes do quatérnion adquiridas durante giro em velocidade constante entorno de z. Mecanismo movido por mancal externo.	212
Figura 6.45 – Componente escalar do quatérnion convertida em graus. Movimento contínuo, a 20°/s, do mancal externo.	212
Figura 6.46 – Componente vetorial do quatérnion normalizada. Rotação mancal externo. A direção da componente vetorial do quatérnion se movimenta.	213
Figura 6.47 – Componentes do quatérnion adquiridas em giro entorno do eixo z sobre a mesa de mancal ar. Velocidade de rotação constante.	
Figura 6.48 – Componente angular do quatérnion adquirida durante rotação entorno do eixo z sobre a mesa de mancal a ar.	214
Figura 6.49 – Componente vetorial normalizada plotada sobre a superfície de uma esfera unitária.	215
Figura 6.50 – Componentes do campo magnético medidas durante rotação entorno do eixo z pela movimentação do mancal interno. A componente z não deveria variar na realização desse movimento.	216
Figura 6.51 – Componentes de aceleração (gravidade). Somente o eixo z está medindo a gravidade. Os demais deveriam ser zero caso não houvesse <i>off-set</i> .	216
Figura 6.52 – Componentes do campo magnético medidas durante rotação do mancal externo entorno do eixo z.	217
Figura 6.53 – Comparação entre as componentes do campo magnético medidas no ensaio N e Q. Dois pontos de vista.	218
Figura 6.54 – Componentes do quatérnion obtidas a partir dos dados dos sensores magnéticos e de aceleração do ensaio N.	217
Figura 6.55 – Componentes do quatérnion obtidas a partir dos dados dos sensores magnéticos e de aceleração do ensaio Q.	219
Figura 6.56 – Componentes do campo magnético sem <i>off-set</i> . Dados do ensaio Q.	220
Figura 6.57 – Componentes de aceleração sem <i>off-set</i> . Dados do ensaio Q.	220
Figura 6.58 – Componentes do quatérnion recuperado com base nos dados do ensaio Q sem <i>off-set</i> nos sensores.	221
Figura 6.59 – Componente escalar do quatérnion recuperado dos dados do ensaio Q sem <i>off-set</i> nos sensores.	222
Figura 6.60 – Componente vetorial do quatérnion recuperado dos dados do ensaio Q sem <i>off-set</i> nos sensores.	222
Figura 6.61 - Componentes do campo magnético sem <i>off-set</i> . Ganho do sensor em x igualado ao ganho do sensor em y. Dados do ensaio Q.	223

Figura 6.62 – Comparação entre quatérnion calculado a partir de dados de sensores com ganhos iguais e com ganhos diferentes. Dados sem <i>off-set</i> em ambos.	223
Figura 6.63 – Saída do giro do eixo x, quando girando entorno de x em passos incrementais de 5°/s e em seguida passos de – 5°/s .	225
Figura 6.64 – Saída do giro do eixo y, quando girando entorno de y em passos incrementais de 5°/s e em seguida passos de – 5°/s .	226
Figura 6.65 – Saída do giro do eixo z, quando girando entorno de z em passos incrementais de 5°/s e em seguida passos de – 5°/s .	226

Lista de Tabelas

Tabela 2.1 – Faixa de variação observada para os elementos do campo geomagnético para a superfície terrestre.	15
Tabela 2.2 – Valores para campo geomagnético em Londrina – PR, calculados a partir do ELEMAG.	19
Tabela 2.3 – Valores para campo geomagnético em Londrina – PR calculados a partir do WMM2005. A taxa de variação não foi estimada.	21
Tabela 6.1 – Características principais do sensor ADXL202E	159
Tabela 6.2 – Características do sensor GYROSTAR 03JA	166
Tabela 6.3 – Principais especificações do sensor HCM2003	178
Tabela 6.4 – Configuração dos ensaios	206
Tabela 6.5 – Conversão para unidades de engenharia	207

Lista de Símbolos

$\mathbf{B}(\mathbf{r}, t)$	Campo geomagnético total
$\mathbf{B}_m(\mathbf{r}, t)$	Campo geomagnético principal
$\mathbf{B}_c(\mathbf{r}, t)$	Campo geomagnético crustal
$\mathbf{B}_d(\mathbf{r}, t)$	Campo geomagnético de perturbação
F	Intensidade total do campo geomagnético
H	Intensidade horizontal total do campo geomagnético
I	Ângulo entre o plano horizontal e o vetor do campo
D	Ângulo entre o norte verdadeiro e o vetor do campo
X	Intensidade do campo geomagnético na direção norte
Y	Intensidade do campo geomagnético na direção leste
Z	Intensidade do campo geomagnético na vertical, positiva na direção da gravidade
(φ', λ, r)	Latitude, longitude e raio.
$g_n^m(t)$	Coeficientes de Gauss variantes no tempo de grau m e ordem n
$h_n^m(t)$	
\tilde{P}_n^m	Polinômio de Legendre
ψ, θ e φ	Ângulos de Euler. Representam respectivamente <i>Yaw</i> , <i>Pitch</i> e <i>Roll</i>
\mathbf{I}	Matriz Identidade. Sua dimensão pode ser especificada no contexto.
\mathbf{A}	Matriz de Atitude
\mathbf{R}	Matriz de rotação construída a partir dos ângulos de Euler
\mathbf{R}_φ^x	Matriz de rotação de φ graus em torno de do eixo x
\mathbf{R}_θ^y	Matriz de rotação de θ graus em torno de do eixo y
\mathbf{R}_ψ^z	Matriz de rotação de ψ graus em torno de do eixo z
\tilde{q}, \tilde{p} e \tilde{r}	Quatérnions
q_0 ou q_4	Componente escalar do quatérnion
$q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$	Componentes vetoriais do quatérnion
\mathbf{q}	Parte vetorial do quatérnion

$\mathbf{i}, \mathbf{j} \text{ e } \mathbf{k}$	Números complexos que formando uma base canônica
\otimes	Operador do produto entre quatérnions
\bullet	Operador para produto interno
\times	Operador para produto vetorial
$*$	Operador para conjugado
$ \cdot $	Operador de norma
\hat{q}	Quatérnion unitário
$\hat{\mathbf{u}}$	Vetor unitário
$L_q(.)$	Operador linear de rotação
\mathbf{v}, \mathbf{w}	Vetores quaisquer
\mathbf{Q}	Matriz de rotação construída a partir de quatérnions
$\dot{\tilde{q}}(t)$	Derivada temporal do quatérnion \tilde{q}
$\hat{\mathbf{v}}_i$	Vetores unitários (versores) de referência com $\mathbf{A} = \mathbf{I}$
$\hat{\mathbf{w}}_i$	Vetores unitários (versores) de observação. Leitura dos vetores de referência com $\mathbf{A} \neq \mathbf{I}$
\mathbf{A}_{opt}	Atitude ótima que minimiza função de Wahba
a_i	Pesos da função de Wahba
$L(\mathbf{A})$	Função perda de Wahba
$\hat{\mathbf{r}}_i$	Tríade de referência levantada a partir de $\hat{\mathbf{v}}_i$
$\hat{\mathbf{s}}_i$	Tríade de observação levantada a partir de $\hat{\mathbf{w}}_i$
\mathbf{M}_{ref}	Matriz formada pelos $\hat{\mathbf{r}}_i$. Matriz de referência.
\mathbf{M}_{obs}	Matriz formada pelos $\hat{\mathbf{s}}_i$. Matriz de observação.
$\delta\boldsymbol{\theta}$	Vetor erro dos ângulos
$\langle \cdot \rangle$ ou $E[.]$	Operadores de média
$\boldsymbol{\Omega}(\cdot)$	Operador linear de distribuição
\mathbf{P}_{00}	Matriz de covariância cartesiana da atitude
\mathbf{P}	Matriz de covariância total
$\delta\mathbf{A}$	Matriz diferença de atitude (erro de atitude)
$\text{tr}(\cdot)$	Operador traço de uma matriz

\mathbf{P}_{obs}	Matriz de covariância da matriz de observação
\mathbf{P}_{ref}	Matriz de covariância da matriz de referência
$\delta \hat{\mathbf{w}}_i, \delta \hat{\mathbf{v}}_i$	Erros dos versores $\hat{\mathbf{w}}_i$, $\hat{\mathbf{v}}_i$.
$\sigma_{\mathbf{x}}^2$	Variância de um vetor qualquer
$g(\mathbf{A})$	Função ganho de Wahba
$[\mathbf{q}]$	Matriz distribuição de \mathbf{q}
\mathbf{K}	Matriz de informação para extração do quatérnion ótimo
λ_{max}	Maior autovalor da matriz \mathbf{K}
$\tilde{\mathbf{q}}_{\text{opt}}$	Autovetor correspondente ao maior autovalor da matriz \mathbf{K}
\mathbf{y}	Vetor de Gibbs
\mathbf{P}_{qq}	Matriz de covariância do quatérnion
σ_{tot}^2	Variância total.
$\mathbf{x}(t)$	Vetor de estados
$\mathbf{w}(t)$	Vetor ruído do processo
$\mathbf{Q}(t)$	Matriz covariância do ruído do sistema
$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$	Valor estimado do vetor de estados no instante t_0
Φ	Matriz ou função de transição
$\mathbf{P}(t)$	Matriz de covariância do estado
$\Delta \mathbf{x}(t)$	Vetor erro do estado
$\hat{\mathbf{x}}_k(-)$	Vetor estado propagado antes da estimação
$\mathbf{P}_k(-)$	Matriz de covariância propagada antes da estimação
$\hat{\mathbf{x}}_k(+)$	Vetor estado propagado após a estimação
$\mathbf{P}_k(+)$	Matriz de covariância propagada após a estimação
$\hat{\mathbf{x}}_{k+1}(-)$	Estado propagado
$\mathbf{P}_{k+1}(-)$	Matriz de covariância propagada
\mathbf{N}_k	Matriz correlação do erro da transição de estado
\mathbf{z}_k	Vetor de medidas
\mathbf{v}_k	Vetor de ruído aditivo sobre o vetor de medidas

\mathbf{R}_k	Covariância de \mathbf{v}_k
\mathbf{H}_k	Matriz de medida
\mathbf{K}_k	Matriz ganho de Kalman
$\mathbf{\Omega}_4(\cdot)$	Operador linear de distribuição
$\boldsymbol{\omega}$	Vetor velocidade angular
$\Delta\boldsymbol{\theta}$	Vetor ângulo incremental
$\mathbf{M}(\Delta\boldsymbol{\theta})$	Matriz de transição para o quatérnion
\mathbf{u}	Vetor de medidas dos girômetros
\mathbf{b}	Vetor <i>bias</i> adicionado às medidas dos girômetros
$\boldsymbol{\eta}_1$	Vetor de ruído aditivo presente nas medidas dos girômetros
$\mathbf{Q}_1(t)$	Matriz de covariância do ruído $\boldsymbol{\eta}_1$
$\boldsymbol{\eta}_2$	Vetor da taxa de deriva do bias dos girômetros
$\mathbf{Q}_2(t)$	Matriz de covariância do ruído $\boldsymbol{\eta}_2$
$\Xi(\cdot)$	Operador distribuição
$\hat{\boldsymbol{\omega}}$	Vetor velocidade angular estimada
\hat{q}	Quatérnion unitário estimado
$\hat{\mathbf{b}}$	<i>Bias</i> estimado
$\boldsymbol{\Theta}(t, t_k)$	Função de transição do instante t_k para t
$\boldsymbol{\Lambda}(t, t_0)$	Matriz de rotação da atitude
$\mathbf{S}(\hat{q}(t))$	Operador redução
$\tilde{\boldsymbol{\Phi}}$	Matriz de transição reduzida
$\tilde{\mathbf{H}}_k$	Matriz de medida reduzida
$\tilde{\mathbf{K}}_k$	Matriz ganho de Kalman reduzida
$\tilde{\mathbf{P}}_k$	Matriz de covariância reduzida
$\tilde{\mathbf{Q}}_k$	Matriz covariância do ruído reduzida
N_{rms}	Nível de ruído
BW	Largura de banda
d_{rms}	Densidade de ruído

Δv_s	Excursão da tensão de saída
s	Sensibilidade Volts/grandeza
$f_c^{-3\text{dB}}$	Frequência de corte
V_{out}	Tensão de saída do sensor
V_{set}	Tensão de saída para condição de Set
V_{rst}	Tensão de saída para condição de Reset
ξ	<i>off-set</i> e ruído aditivo
O_i	<i>Off-set</i> no canal i
$V_{i\text{max}}$	Tensão máxima no canal i
$V_{i\text{min}}$	Tensão mínima no canal i
G_i	Ganho no canal i

Capítulo 1 – Introdução

1.1 – O Tema

A busca da orientação de corpos rígidos tem sido cada vez mais cogitada em aplicações científicas, industriais e comerciais como em robôs, satélites artificiais, submarinos, na indústria automobilística, aplicações em realidade virtual e em dispositivos de auxílio a deficientes físicos.

A orientação no espaço pode ser expressa matematicamente por uma matriz de rotação que permita realizar a transformação entre sistemas de coordenadas associados a diferentes sistemas de referência. A orientação de um corpo não está relacionada a sua posição e pode ser estudada e determinada de forma independente.

Para determinar a atitude são utilizados os mais diversos tipos de sensores que baseiam-se em princípios físicos dos mais triviais aos mais complexos. Alguns sensores baseiam-se em referências como a gravidade, o campo geomagnético, uma fonte de radiação luminosa, acústica, IR ou eletromagnética. Outros sensores baseiam-se em princípios físicos sentidos pelo deslocamento da inércia, por isso são considerados sensores inerciais, como os acelerômetros, girômetros mecânicos, giros piezelétricos, giros de fibra óptica (FOG), giros lasers (RLG) dentre outros. Não apenas um sensor, mas as combinações de vários destes sensores, possibilitam determinar a atitude com boa e até ótima precisão. Pelos menos dois vetores de referência, em direções distintas, são necessários para resolver matematicamente a atitude em três dimensões.

Uma técnica comum é utilizar sensores que meçam a velocidade angular de forma que esta informação é integrada enquanto algum outro sensor, baseado em referência, atualiza a deriva do giro. Em [CHI00] são utilizados sensores de estrela, receptor GPS e giros juntamente com um algoritmo de fusão para determinar a atitude. Em [FAL01] são empregados giros e sensores solares na confecção de um sistema de determinação de orientação para nano-satélites. Em [WIL04] utilizou-se sensores magnéticos (bússolas eletrônicas) e um acelerômetro como sensor de inclinação para prover atitude a um sensor de velocidade acústica modular (MAVS). Em [GEB04] são

apresentados projetos de sistemas de determinação de atitude baseado em múltiplos sensores de baixo custo, inclusive a fusão da informação redundante provida por vários sensores MEMS – *Micro-Electro-Mechanical Systems*. Em [MAR01] propõe-se o uso de tríades de sensores magnéticos, acelerométricos e giros (ou composição MARG) para estimação da atitude. O método propunha o uso do algoritmo iterativo de Gauss-Newton para encontrar a atitude a partir das referências como o campo geomagnético e a gravidade enquanto os dados dos giros eram integrados também para estimar a atitude.

Como exposto acima, são diversos os trabalhos e as técnicas empregadas na determinação da atitude. Um ponto comum a quase todas é que a maioria envolve a estimação da orientação utilizando algum tipo de filtragem para unir e aproveitar a informação estatística dos dados de forma a otimizar a estimação. O filtro de Kalman e o filtro de Kalman Estendido são os mais utilizados. Outro fator na determinação da orientação é como obter a atitude ótima baseada em várias (mais de duas) referências vetoriais corrompidas pelo ruído. Esse problema é conhecido como problema de Wahba [WAH66] e uma solução elegante para ele é o algoritmo QUEST proposto por Shuster [SHU80]. Um outro algoritmo mais simples, mas também eficiente, é o TRIAD. Este algoritmo fornece equações simples para encontrar a atitude e a matriz de covariância quando existem apenas dois vetores de referência.

1.2 – A Meta

O objetivo deste trabalho é construir um protótipo de um determinador de atitude, que a partir de medidas tomadas em sensores MEMS e por meio de um hardware de aquisição e processamento, encontre em tempo real a atitude.

Os sensores a serem utilizados também são uma composição MARG, ou seja, contém magnetômetros, acelerômetros e giros. O sistema pode unir as informações dos sensores de várias formas para encontrar a atitude. A forma mais simples é lendo os vetores do campo geomagnético e da gravidade. A partir deles, fazendo uso do algoritmo TRIAD ou QUEST, é possível encontrar a orientação em relação ao sistema de referência. Essa solução está vulnerável ao fato da gravidade não poder ser lida diretamente, portanto, quando o sensor de aceleração se encontra em movimento acelerado o sistema é falho. A segunda solução é, partindo de uma atitude conhecida, integrar as velocidades angulares medidas pelos giros, atualizando a orientação a cada instante. Esta solução também é falha, pois erros de medida das velocidades angulares

dos giros causarão, quando integrados, erros intoleráveis em poucos minutos, até mesmo em segundos, dependendo da qualidade dos giros empregados e do sistema de aquisição. Agora, combinando ambas as soluções – em certos momentos processando as leituras dos giros, e em outros instantes os dados dos sensores magnéticos e acelerômetros – ter-se-á uma solução plausível e com menos limitações de aplicabilidade que formas isoladas. Resta apenas a questão de como unir tais informações e como saber o momento certo de usar cada uma delas. O uso do Filtro de Kalman (e Estendido) é uma solução generalizada muito utilizada e bastante custosa computacionalmente para os sistemas de processamento embarcados existentes e, portanto, ainda existe espaço para inovação.

1.3 – A Organização

O trabalho como um todo, foi dividido em quatro etapas: estudo, simulação, implementação e testes. O capítulo 2 apresenta brevemente conceitos de referências e sistemas de coordenadas para a atitude. No capítulo 3 são mostradas as formas matemáticas mais comuns de expressar a atitude, como a matriz de atitude, os ângulos de Euler e o quatérnion, bem como as relações entre as representações. No capítulo 4 dois algoritmos de determinação da atitude são explorados com profundidade: o TRIAD e o QUEST. No capítulo 5 o método de filtragem de Kalman é introduzido como uma forma de unir informações de sensores de referências e giros para contornar limitações físicas e elétricas de ambos. São feitas simulações para testar a convergência do filtro de Kalman. Uma simulação gráfica criada ilustra bem o efeito do filtro. No capítulo 6, são analisados os sensores utilizados e o hardware de aquisição e processamento da atitude construído. Alguns pontos importantes do hardware são discutidos e as funções do software em linguagem C são apresentadas e comentadas. Também são apresentados resultados de ensaios realizados no Instituto de Pesquisas Espaciais – INPE, em uma mesa de rotação 3D e em uma mesa de mancal a ar. Por fim, o capítulo final traz conclusões sobre todo o trabalho realizado, enfatizando acertos, erros e sugerindo caminhos para continuação deste trabalho.

Capítulo 2 – Sistemas de Coordenadas e Referências para a Atitude

2.1 – Introdução

Determinar a atitude de um corpo é determinar a rotação que devemos aplicar sobre o sistema de coordenadas de referência para que o mesmo coincida com o sistema de coordenadas definido no corpo, sem levar em conta a translação necessária para que eles coincidam sua posição no espaço. A comparação entre os sistemas de coordenadas do corpo e o de referência é feita superpondo as origens de ambos.

Quando se deseja determinar a atitude deve-se primeiro saber em relação a qual sistema de coordenadas se deseja fazer isso. Em segundo lugar, é preciso saber quais vetores de referência podem ser utilizados e também conhecer os seus modelos no sistema de coordenadas de referência. Um vetor de referência é um vetor que possa ser lido tanto no sistema de coordenadas referencial quando no sistema de coordenadas do corpo.

Na próxima seção serão apresentados brevemente dois sistemas de coordenadas referenciais, cada um servindo a uma aplicação. E na seção seguinte serão enunciadas as principais referências para determinação da atitude utilizadas tanto em aplicações terrestres quanto aeroespaciais.

2.2 – Sistemas de Coordenadas

Um sistema de coordenadas pode ser construído com vetores que sejam ortogonais entre si e a formação obedeça a regra da mão direita. Cada aplicação exige um sistema de coordenadas que seja adequado à questão. Por exemplo, em uma fábrica de automóveis não é preciso conhecer a orientação da garra ou do braço de um robô em relação ao Norte ou ao Sol, mas sim saber a sua orientação em relação ao sistema de coordenadas do robô, definido na sua própria base. Aplicações de navegação (terrestre, marinha ou aérea) exigem o conhecimento das direções Norte-Sul, Leste-Oeste e da

Inclinação. Outras aplicações, como em satélites artificiais, é necessário conhecer a orientação do mesmo em relação à Terra, ao Sol ou à alguma estrela.

A seguir será exposto dois exemplos de sistemas de coordenadas. O primeiro deles, o sistema de coordenadas horizontal local, é aplicável sobre a superfície terrestre enquanto o segundo, o sistema de coordenadas equatorial, é mais adequado para aplicações aeroespaciais. Vale reafirmar que o interesse nos sistemas de coordenadas é pela direção dos vetores ortogonais que compõe o sistema.

2.2.1 – Sistema de coordenadas horizontal local

Para aplicações sobre a superfície terrestre, geralmente é utilizado o sistema de coordenadas horizontal local, ou seja, o sistema constituído pelo Norte, pelo Leste e pela direção da Gravidade. A Figura 2.1 apresenta como o sistema está em relação à Terra. O *Zênite* é definido como o eixo, perpendicular à superfície do geóide de referência. Também pode ser associado à direção do centro gravitacional. Já o Norte, está no mesmo plano do eixo de rotação da Terra, e o Leste, pode ser compreendido como sendo o eixo resultado do produto vetorial dos outros dois acima citados. Ambos vetores, Norte e Leste, são tangentes à superfície enquanto o vetor Zênite é normal à superfície.

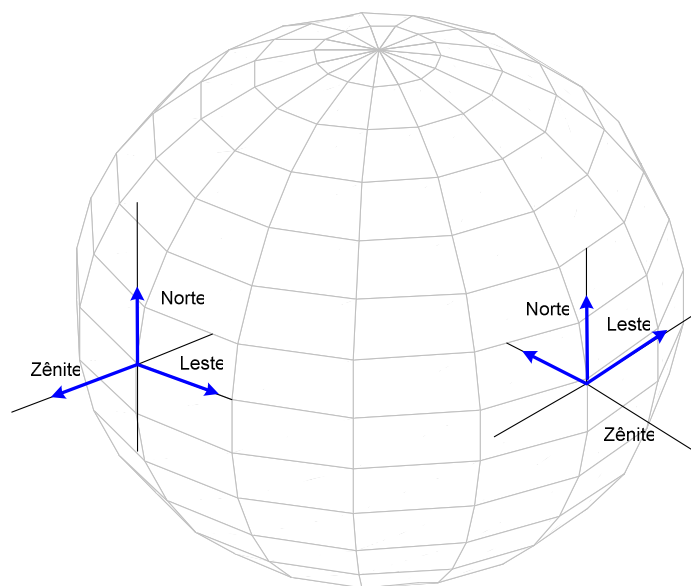


Figura 2.1 – Sistema de coordenadas horizontal local

É evidente que a direção dos vetores é dependente da Latitude e da Longitude adotada. Portanto, quando se utiliza esse sistema de coordenadas é importante saber a atual posição sobre o globo.

Assim como mostra a Figura 2.1, o Norte pode ser adotado como eixo Y, o Leste como X e pela regra da mão direita, para cima o Z. Outra forma, pode ser adotar o Norte como X e o Leste com Y, fazendo com Z aponte para baixo. Estas duas formas são utilizadas. A primeira porque está de acordo com as direções do o sistema de coordenadas polar. O segundo por ser a ordem adotada na representação dos ângulos de Euler de um objeto sobre a superfície.

2.2.2 – Sistema de coordenadas celestial ou equatorial

Para a determinação da atitude em uma aplicação aeroespacial, o sistema mais comum é o sistema de coordenadas inercial celestial. Neste sistema de coordenadas, assim como na maioria deles, é considerado que o sistema de coordenadas é uma esfera de raio unitário centrada no objeto (no corpo) e não na Terra.

Os eixos de referência são: O Norte Celestial, vetor normal ao plano paralelo ao equador da Terra, e o *Equinócio Vernal*.

O Norte Celestial fica a aproximadamente 1° da direção da estrela Polaris, a estrela do Pólo Norte. O Equinócio Vernal é o ponto onde o plano eclíptico, ou seja, o plano de translação da Terra, cruza o plano do equador passando do Sul para o Norte. Um vetor saindo normal ao plano tangente a este ponto apontará para a constelação de Áries. Esses dois eixos definem um terceiro eixo ortogonal e contido no plano do equador. A Figura 2.2 ilustra o que foi dito acima. É interessante notar que essa referência não é absolutamente inercial já que existe um movimento de *precessão dos Equinócios* com um período de 26000 anos, ou seja, uma variação de 50 segundos de arco por ano. Essa variação obriga, para aplicações de grande precisão, que seja vinculado o tempo (data atual) à posição.

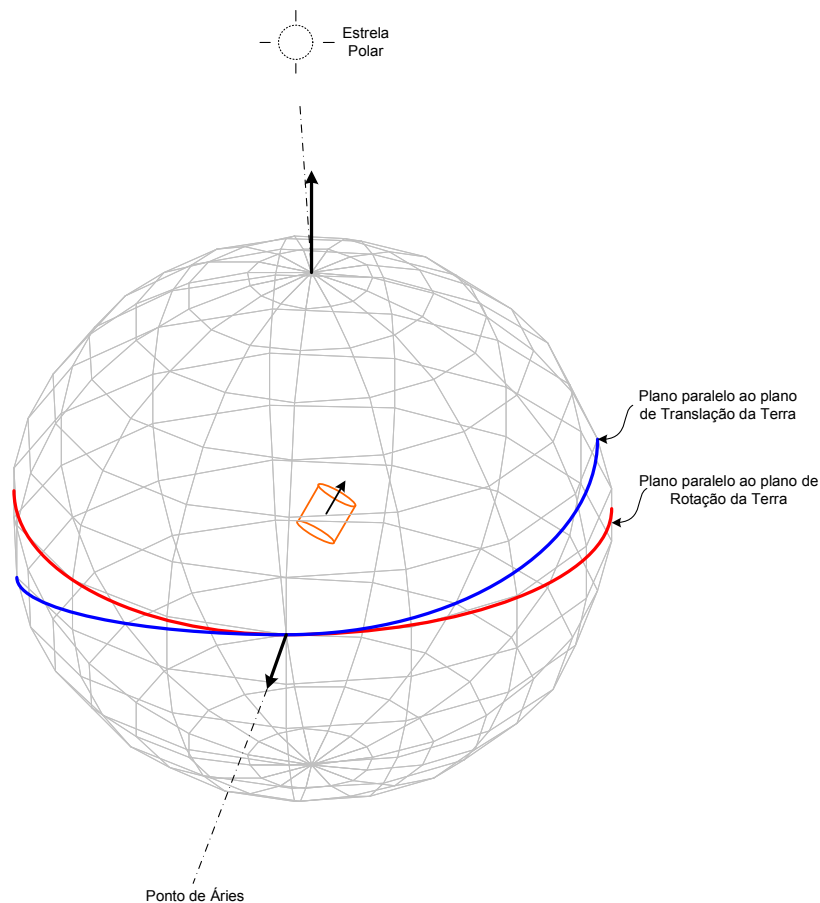


Figura 2.2 – Sistema de coordenadas celestial

2.3 – Vetores de Referência

Tendo definido o sistema de coordenadas agora é necessário observar vetores nesses sistemas e modelá-los para que sirvam como referências para determinação da atitude.

Existem diversos vetores que podem ser tomados como referência para a determinação da atitude. Esses, quando têm suas projeções observadas no sistema de coordenadas de referência e do corpo, podem ser comparadas resultando uma informação sobre a orientação do objeto. Os vetores de referência mais comuns que servem a aplicações terrestres e aplicações aeroespaciais são: as estrelas, o Sol, o albedo da Terra, o campo magnético terrestre, e o campo gravitacional. A seguir uma breve descrição de cada uma dessas referências.

2.3.1 – As estrelas

Sensores conhecidos como *Star Trackers*, os rastreadores de estrelas, fazem a identificação das coordenadas locais de estrelas cujas coordenadas num dado sistema de referência encontram-se armazenadas em um banco de dados que pode possuir até dezenas de milhares de estrelas catalogadas; esse banco é chamado catálogo ou Mapa Estelar. Quando se identifica uma estrela ou um grupo delas, sabe-se exatamente para que estrelas do catálogo o imageador está apontando e, portanto, a direção dessas estrelas são vetores de referência conhecidos.

Os imageadores são instrumentos ótico-mecânicos de precisão. Geralmente são câmeras CCD com resolução moderada.

As estrelas são vistas como pontos acesos na matriz CCD – *pixels*. Elas são reconhecidas ou pela intensidade luminosa ou pela posição relativa que ela se encontra em relação às outras estrelas.

Um satélite cujo posicionamento deve ser feito com grande exatidão deve conter um *Star Tracker* à bordo, pois esse tipo de sistema, é capaz de atingir exatidão próxima a 1 segundo de arco.

O balão telescópico de raios-X denominado MASCO [MEI00], é um exemplo da aplicação desse recurso como ajuda na determinação e controle da atitude. Em seu projeto, foi utilizado um sensor CCD de baixo custo capaz de determinar as coordenadas de apontamento com um erro menor que $0,5^\circ$. O instrumento consistia em uma câmera CCD acoplada em lentes ópticas que digitalizava as imagens com uma resolução de 512×400 pontos e com um campo de visão de apenas $6^\circ \times 5^\circ$. Infelizmente a implementação desse experimento não pôde ser testado em vôo devido a problemas de interface com o computador de bordo.

2.3.2 – O Sol

O Sol também é uma estrela, mas como está muito próxima da Terra, comparada com outras estrelas, essa referência pode ser verificada de outras formas. Para observar a atitude de um satélite em relação ao Sol é possível utilizar os próprios painéis solares de um satélite como sensores de observação [SAN00], já que os painéis apresentam seu rendimento de acordo com o ângulo que estão recebendo a luz do Sol. Conhecido assim a posição do Sol em relação ao sistema de coordenadas do satélite, então, é possível

utilizar o ângulo entre o vetor normal ao painel solar e o Sol como uma medida para determinação da atitude do satélite. Essa técnica tem sido aplicada em várias missões por ser de custo relativamente baixo, mas sua precisão não chega a 1° em muitos casos.

O Sol também pode ser monitorado por sensores mais simples que painéis solares ou câmeras CCD. Os SCC – *Solar Compass Chip* – compasso solar em circuito integrado [LIE04], por exemplo, são chips fotos-sensíveis fabricados com tecnologia óptica baseado em MEMS (Sistema Microeletromecânico) que integram o processamento geralmente na própria pastilha, o que torna o SCC um recurso barato para identificar a direção do Sol. A precisão atingida utilizando um SCC pode chegar a 0,1°.

2.3.3 – Albedo

O albedo de um objeto é a fração da energia incidente que é refletida de volta. Por exemplo, o albedo da Terra é a energia solar, o calor, a radiação, que são recebidos do céu e que retornam para o espaço. O albedo da Terra é de 0,3 na média para vegetação ou rochas, mas há que se considerar que o gelo e água refletem mais luz que a Terra por isso existe distorções. No caso de um satélite, saber para que lado está a Terra é uma ótima referência para a atitude, por esse motivo o albedo é estudado e modelado para que seja uma referência cada vez mais precisa.

A intensidade do albedo depende, além da refletividade da superfície terrestre, da posição que a Terra está sendo iluminada pelo Sol e da posição de onde é feita a leitura.

Como o cálculo analítico do albedo é impraticável, é possível levantar um modelo numérico e tabelá-lo [APP04]. A precisão que se pode atingir não chega a ser melhor que 1°. Mesmo assim, o albedo é utilizado como uma direção de referência em vários satélites que orbitam neste momento a Terra.

2.3.4 – Campo gravitacional

O campo gravitacional é produzido pela massa da Terra. Se a Terra fosse uma esfera perfeita, a gravidade seria a mesma em todos os pontos da superfície, e seria um vetor constante \vec{g} , perpendicular à superfície. Como a Terra tem formato geodésico, a

gravidade não será exatamente constante sobre toda a superfície porque a distribuição de massa não é totalmente homogênea.

A gravidade fornece uma referência interessante porque pode ser utilizada para indicar a inclinação de um objeto que se encontra sobre a superfície.

Não é possível medir a gravidade diretamente. Só é possível medir a força de reação devido à gravidade. Acelerômetros são dispositivos capazes de medir a aceleração gravitacional atuando sobre corpos não acelerados.

Não é possível utilizar acelerômetros para medir aceleração gravitacional em um corpo que não se encontra sob a ação de forças de reação. Um satélite em órbita da comporta-se com um corpo em queda livre, portanto não há forças de reação e a única aceleração que se lê é a aceleração devido ao arrasto aerodinâmico.

Mesmo com esse inconveniente, a gravidade é um vetor de referência importante na superfície terrestre ou sobre a superfície de qualquer outro planeta.

Modelagem do Campo Gravitacional

A não uniformidade do campo gravitacional é um dos fatores de perturbação da órbita dos satélites. Portanto, o campo é constantemente modelado para que se detecte as mínimas variações do módulo e da direção da gravidade sobre a superfície terrestre. Dados coletados pela missões CHAMP e GRACE [TAP05] resultaram em um modelo de altíssima resolução.

Com isso, a modelagem da geóide terrestre ficou milimetricamente precisa. Como pode ser observado na Figura 2.3, a modelagem da gravidade sobre o planeta permite estimar um formato para distribuição da massa do planeta. A Figura 2.4 apresenta uma representação da variação da gravidade em mGal¹ nas diferentes latitudes e longitudes do planeta.

¹ Um mGal é uma unidade conveniente para descrever as variações da gravidade sobre a superfície da Terra. 1 mGal = 0,00001 m/s² que pode ser comparado a gravidade total na superfície da Terra que é de aproximadamente 9,8 m/s². Assim, um miligal pode ser comparado como sendo a milionésima parte da aceleração padrão sobre a superfície da Terra.

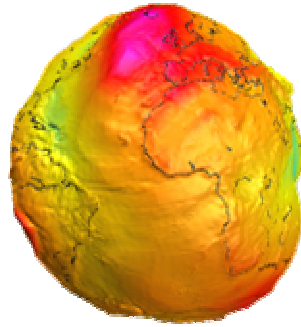


Figura 2.3 – O formato de massa da terra visto pela missão CHAMP. Se a terra fosse uma distribuição esférica uniforme de massa, então a gravidade seria a mesma em todos os pontos da superfície terrestre.

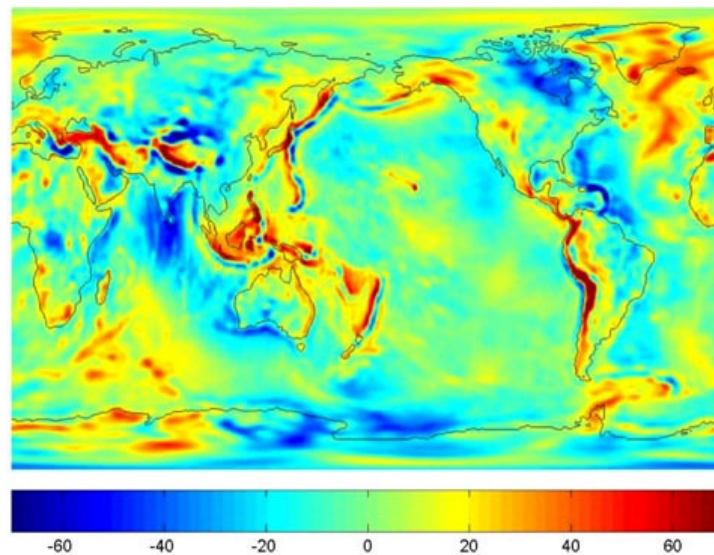


Figura 2.4 – Variação da intensidade do campo gravitacional sobre a superfície do planeta Terra em relação à intensidade média de $9,8 \text{ m/s}^2$. As variações são dadas em mGal^5

Para aplicações terrestres locais, o campo gravitacional pode ser aproximado para um valor constante e com direção coincidente à direção do eixo Z do sistema de coordenadas horizontal local. Para determinação da atitude, em aplicações sobre a superfície terrestre, o modelo gravitacional não é de extrema importância. Tanto a intensidade da gravidade, quanto sua direção local, mudam de acordo com a Latitude, Longitude e altitude.

Existem modelos matemáticos para o campo gravitacional, mas para aplicações terrestres, a direção da gravidade se confunde com a direção do eixo Z quando utilizado

o sistema de coordenadas horizontal local, portanto esse modelo matemático é um tanto desnecessário para aplicações de pouca precisão e por isso não será exposto aqui.

2.3.5 – Campo magnético terrestre

O campo geomagnético e as estrelas são sem dúvida as referências que propiciaram a navegação marítima desde a Idade Média e propiciam ainda nos dias de hoje. Embora as suas características gerais sejam conhecidas há séculos, foi o físico e matemático Karl Gauss quem primeiro estudou sistematicamente o campo magnético terrestre no início do século XIX [WER78]. Ainda nos dias de hoje, alguns efeitos observados são considerados fenômenos por não terem uma explicação física plausível. O campo da Terra é dividido em pelo menos três partes formadoras distintas que se somam para formar o campo que os sensores monitoram:

$$\mathbf{B}(\mathbf{r}, t) = \mathbf{B}_m(\mathbf{r}, t) + \mathbf{B}_c(\mathbf{r}, t) + \mathbf{B}_d(\mathbf{r}, t) \quad (2.1)$$

Onde:

- O campo principal, \mathbf{B}_m , que é campo gerado pela circulação do fluido no núcleo da Terra.
- O campo crustal, \mathbf{B}_c , que é o campo presente nas aglomerações rochosas da crosta.
- O campo de perturbação, \mathbf{B}_d , que é um combinado das correntes elétricas da alta atmosfera e da magnetosfera que também induzem correntes no mar e na terra.

O campo \mathbf{B}_m é a parte dominante com cerca de 95% do módulo do campo na superfície terrestre. O campo \mathbf{B}_m apresenta variações seculares, ou seja, sua variação é lenta com o passar dos anos.

O campo \mathbf{B}_c é aquele presente nas rochas magnetizadas. Embora ele varia espacialmente, ou seja, de acordo com a latitude e longitude sobre a Terra, ele é praticamente constante com o passar dos anos. O campo \mathbf{B}_c é muito menor em magnitude que \mathbf{B}_m .

As correntes que percorrem a ionosfera e a magnetosfera e suas correntes induzidas no manto e na crosta terrestre produzem o campo \mathbf{B}_d . Essas correntes variam com a posição e com o tempo.

O campo geomagnético é uma referência importante, mas para que ele seja utilizado como vetor de referência é necessário que haja um modelo que indique qual a direção e intensidade do campo em cada ponto do planeta. O modelo WMM [MCL05] (*World Magnetic Model* – Modelo Magnético Mundial), assim como o IGRF² (*International Geomagnetic Reference Field* – Campo Geomagnético Internacional de Referência) fornecem o modelo para \mathbf{B} excluindo as perturbações.

Modelagem do Campo Magnético

O campo geomagnético, devida a sua extrema importância à navegação e à orientação de forma geral, tem sido estudado e modelado. Não só seu estado atual, mas sua variação com o tempo. O campo geomagnético, diferentemente da gravidade, varia significativamente sua direção, inclinação e intensidade de ponto a ponto no globo terrestre e, ainda que em menor escala, varia no tempo.

Como dito anteriormente, qualquer referência precisa ser modelada para que sirva à navegação. O campo geomagnético não é uma exceção.

O campo geomagnético, \mathbf{B} , pode ser descrito por sete elementos indicadores, são eles: as componentes ortogonais X (intensidade na direção norte), Y (intensidade na direção leste) e Z (intensidade vertical, positiva na direção da gravidade), intensidade total F , intensidade horizontal H , inclinação I (ângulo entre o plano horizontal e o vetor do campo) e a declinação D (ângulo entre o norte verdadeiro e o vetor do campo).

As componentes F , H , I e D podem ser calculadas a partir de X , Y e Z :

$$H = \sqrt{X^2 + Y^2} \quad (2.2)$$

$$F = \sqrt{H^2 + Z^2} \quad (2.3)$$

$$D = \arctan(Y, X) \quad (2.4)$$

$$I = \arctan(Z, H) \quad (2.5)$$

² IGRF – www.igrf.org

Onde a declinação está na faixa de -180° a $+180^\circ$ e a inclinação está entre -90° e $+90^\circ$.

Para $H = 0$, a inclinação é indefinida. As componentes vetoriais são observadas no sistema de coordenadas elipsoidal ou geodésico, bem como, em coordenadas geocêntrica esférica.

A Tabela 2.1 apresenta as faixas observadas para os parâmetros do campo magnético na superfície da Terra.

Tabela 2.1 – Faixa de variação observada para os elementos do campo geomagnético para a superfície terrestre.

Elemento	Nome	Nome Alternativo	Faixa dos Elementos do Campo		
			MIN	MAX	Unidade
X	Componente Norte	Intensidade Norte	-17000	42000	nT
Y	Componente Leste	Intensidade Leste	-18000	18000	nT
Z	Componente para Baixo	Intensidade Vertical	-67000	61000	nT
H	Intensidade Horizontal		0	42000	nT
F	Intensidade Total	Campo Total	22000	67000	nT
I	Inclinação	Profundidade	- 90	+90	Graus
D	Declinação	Variação Magnética	-180	+180	Graus

Como é possível observar na Tabela 2.1, o campo magnético terrestre tem o valor de seu módulo confinado à 67000 nT. Esse dado é importante, pois especifica os tipos de sensores adequados para serem utilizados nesta faixa de medida. Os magnetômetros com tecnologia *fluxgate* e os magnetos-resistivos são exemplos de tecnologias adequadas para efetuar medidas do campo geomagnético.

Parametrização do Modelo

O campo magnético da Terra, medido na superfície da Terra ou na altitude de um satélite em órbita, é a soma das componentes geradas por fontes internas e externas.

O campo magnético \mathbf{B} é um campo potencial, e portanto pode ser escrito como o gradiente negativo de um potencial escalar, assim:

$$\mathbf{B}(\varphi', \lambda, r, t) = -\nabla V(\varphi', \lambda, r, t) \quad (2.6)$$

Este potencial pode ser expandido em termos dos harmônicos esféricos

$$\mathbf{B}(\varphi', \lambda, r, t) = a \left\{ \sum_{n=1}^N \sum_{m=0}^n \left(g_n^m(t) \cos(m\lambda) + h_n^m(t) \sin(m\lambda) \right) \left(\frac{a}{r} \right)^{n+1} \tilde{P}_n^m(\sin \varphi') \right\} \quad (2.7)$$

onde, $a = 6371,2$ km é o raio magnético de referência padrão, (φ', λ, r) são a latitude, a longitude e o raio no sistema de coordenadas esférico, no sistema de referências geocêntrico, e $(g_n^m(t), h_n^m(t))$ são os coeficientes de Gauss variantes no tempo, de grau n e ordem m , que descrevem fontes internas de geração do campo.

$\tilde{P}_n^m(\sin \varphi')$ são as Funções Associadas de Legendre semi-normalizadas de Schmidt, definidas como,

$$\tilde{P}_n^m(\sin \varphi') = \begin{cases} \sqrt{2 \frac{(n-m)!}{(n+m)!}} P_n^m(\sin \varphi') & \text{se } m > 0 \\ P_n^m(\sin \varphi') & \text{se } m = 0 \end{cases} \quad (2.8)$$

O Modelo WMM2005 considera $N = 36$, como nível de truncamento da expansão. Os coeficientes de Gauss são considerados com sendo de oitavo grau com dependência quadrática do tempo,

$$\begin{aligned} g_n^m(t) &= g_n^m + \dot{g}_n^m(t-t_0) + \frac{1}{2} \ddot{g}_n^m(t-t_0) \\ h_n^m(t) &= h_n^m + \dot{h}_n^m(t-t_0) + \frac{1}{2} \ddot{h}_n^m(t-t_0) \end{aligned} \quad (2.9)$$

Onde, $(g_n^m(t), h_n^m(t))$ são funções variantes no tempo, enquanto $g_n^m, h_n^m, \dot{g}_n^m, \dot{h}_n^m, \ddot{g}_n^m$ e \ddot{h}_n^m são constantes obtidas através dos dados coletados. O tempo é dado em ano decimal e o instante t_0 é a data de referência do modelo.

O modelo da equação de $\mathbf{B}(\varphi', \lambda, r)$ é válido somente para fontes internas à Terra, como o campo principal e o campo crustal.

De acordo com dados de anos anteriores, observou-se que a variação do modelo não chega a 1° em ângulo e alguns nT em módulo na maior parte do globo terrestre durante um período de 5 anos.

Cálculos do Modelo

O modelo do campo geomagnético é um tanto complicado e custoso para implementação embarcada (*onboard*), ou seja, uma implementação de recursos computacionais escassos. Soluções alternativas e simplificadas podem ser adotadas para fazer uso do modelo sem cometer grandes erros.

Idealmente, toda aplicação embarcada de determinação de atitude deveria carregar em memória os coeficientes e fórmulas necessárias para determinar a direção dos vetores de referência no local onde se encontra e assim determinar a atitude de forma precisa.

Os cálculos são custosos computacionalmente, e ainda mais para uma plataforma de processamento moderado, ou seja, que executa uma multiplicação em ponto flutuante em milisegundos, fica impraticável a realização destes cálculos na mesma taxa que se deseja o cálculo da orientação, já que o cálculo das referências envolve acesso a tabela de valores, dezenas de multiplicações, somas e cálculo de funções trigonométricas.

O melhor recurso é mapear o domínio da função do campo em intervalos discretos de latitude, longitude e elevação, tal que possam ser geradas previamente uma tabela do campo em cada um dos intervalos definidos. O custo computacional agora é em quantidade de memória apenas. Por exemplo, dividindo o globo terrestre em 128 pontos de latitude, 128 pontos de longitude e 128 pontos de altitude ter-se-á uma tabela de 2^{21} elementos, ou seja, armazenado em cada linha desta tabela três palavras de dados de 16 bits cada necessita-se de 12 Megabytes de memória com velocidade de acesso de menos

de 0,1 microsegundo, já que as memórias não-voláteis atuais possuem tempo de acesso³ de menores que 50 ns. Para posições intermediárias, o uso da interpolação dos valores pode aumentar a precisão.

Para realização dos cálculos do campo magnético em função da posição sobre o globo utilizando um modelo complexo como WMM2005, uma solução computacional interessante seria a montagem de um hardware dedicado, um chip programado em VHDL (*VHSIC Hardware Description Language* – Linguagem de descrição de hardware é uma linguagem usada para facilitar o design de circuitos digitais em FPGAs e ASICs) que realizasse cálculos paralelamente. A única memória necessária seria uma memória externa que armazenaria os coeficientes do modelo. Os *buffers* de interface e tudo mais pode ser programado dentro do próprio chip.

Uma outra solução é a atualização da direção do campo geomagnético sendo feita remotamente. Por exemplo, um veículo terrestre rastreado via satélite e monitorado pela estação base. O veículo pode ter seus referenciais atualizados enviando periodicamente sua posição lida por um GPS. A base então, calcula os vetor (ou vetores) de referência e envia uma atualização, se necessário. Neste caso o banco de dados é suportado por uma base de controle.

Vejamos a seguir duas implementações que foram feitas para modelo do campo. A primeira delas, uma implementação de 4ª ordem feita em 1995, o ELEMAG. A segunda, uma implementação do WMM2005, como já data, feita em 2005.

Implementação do Modelo – Software ELEMAG [ELE95]

O ELEMAG é uma versão da implementação do campo feita pelo Observatório Nacional. Sua implementação é conhecida por um código aberto escrito em Fortran 77. Entrando com dados de latitude, longitude e data são fornecidos os valores das componentes Declinação (D), Inclinação (I), a Intensidade Horizontal (H) e Intensidade Total (F) e as componentes cartesianas locais, Norte (X), Leste (Y) e Vertical (Z) do campo geomagnético e sua taxa de variação anual. Trata-se de um modelo polinomial do 4º grau em latitude, longitude e 2º grau em tempo.

Por exemplo, fornecendo as coordenadas de Londrina/PR e a época atual, a saída calculada é dada pela Tabela 2.2.

³ <http://www.hitachi.com/New/cnews/E/1996/960208B.html>

Tabela 2.2 – Valores para campo geomagnético em Londrina – PR, calculados a partir do ELEMAG.

Elemento	Valor Atual (nT)	Taxa de Variação (nT/ano)
<i>X</i>	18832	−91,8
<i>Y</i>	−5922	−13,9
<i>Z</i>	−11589	−100,0
<i>H</i>	19741	−83,4
<i>F</i>	22891	−21,3
<i>I</i>	−30°	−7,4′
<i>D</i>	−17°	−20,2′

Implementação do WMM2005

Outra implementação é a do WMM2005 em Matlab®. Trata-se de uma implementação aberta, mas é muito mais complexa que a implementação do ELEMAG, pois está adaptada ao poder de processamento dos computadores de hoje. Essa implementação permite o fornecimento dos dados de co-latidade ($90^\circ - \text{latitude}$), longitude e altitude de qualquer ponto da Terra, inclusive até centenas de quilômetros de altitude.

A saída é o campo em coordenadas polares, onde *Z* está apontando para o zênite, *X* para o Oeste e *Y* para o Sul.

Com o auxílio destes softwares é possível levantar o perfil do campo magnético de qualquer região com facilidade. Por exemplo, o perfil do campo em uma região que tem como centro a cidade de Londrina, situada na latitude -23° e longitude -51° tem para as componentes *F*, *D* e *I* os valores dados pelas Figuras 2.5, 2.6 e 2.7 respectivamente.

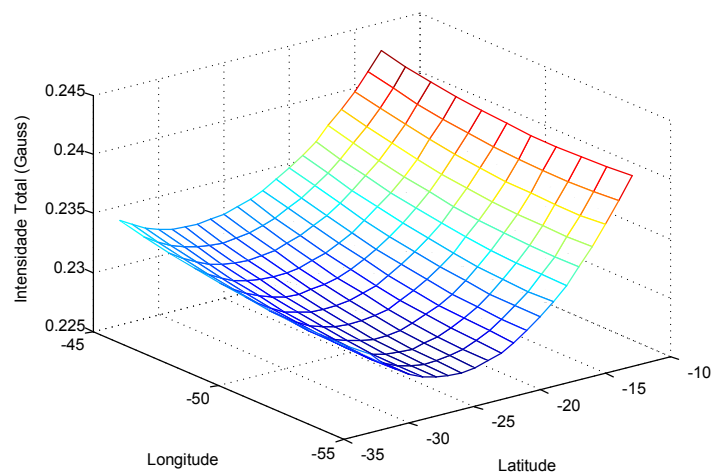


Figura 2.5 – Campo magnético entorno da na região de Londrina/PR/Brasil – Modelo WMM2005
– Intensidade do Campo em função da latitude e longitude.

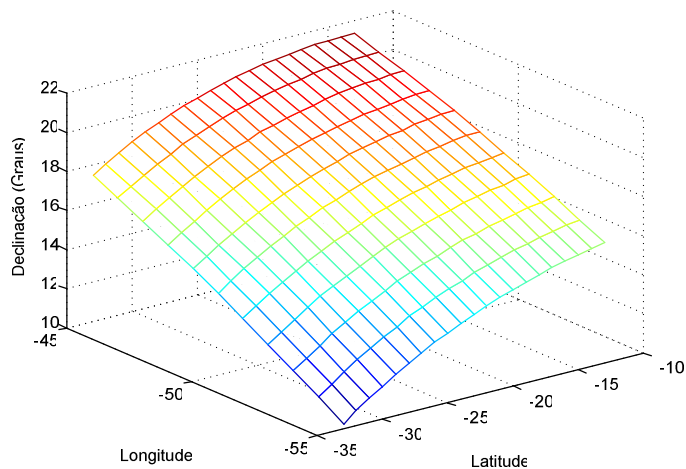


Figura 2.6 – Campo magnético entorno da na região de Londrina/PR/Brasil – Modelo WMM2005
– Declinação do Campo em função da latitude e longitude.

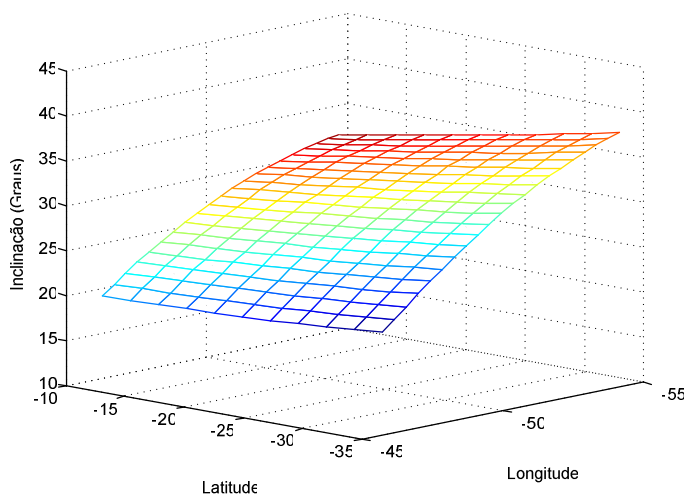


Figura 2.7 – Campo magnético entorno da na região de Londrina/PR/Brasil – Modelo WMM2005
– Inclinação do Campo em função da latitude e longitude.

A Tabela 2.3 apresenta as grandezas do campo geomagnético para a cidade de Londrina/PR calculadas a partir do modelo WMM2005.

Tabela 2.3 – Valores para campo geomagnético em Londrina – PR calculados a partir do WMM2005. A taxa de variação não foi estimada.

Elemento	Valor Atual (nT)
<i>X</i>	19213
<i>Y</i>	-5889
<i>Z</i>	-11135
<i>H</i>	20096
<i>F</i>	22974
<i>I</i>	-17,04°
<i>D</i>	-28,99°

Comparando os dados da Tabela 2.3 como os valores da Tabela 2.2 é possível observar que os dados ficaram muito próximos mesmo com os dados do ELEMAG sendo de baixa ordem e com uma projeção de 10 anos. As declinações de ambos são praticamente as mesmas e as inclinações diferem em 1° apenas. As amplitudes não são dados relevantes para orientação se forem utilizadas tríades de sensores que medem as projeções do campo em cada eixo, pois, tendo a medida de todas as projeções é possível calcular o campo total.

2.4 – Conclusões e considerações sobre aplicações

Para aplicações de determinação da orientação sobre a superfície terrestre, a gravidade e o campo magnético são boas referências. A gravidade, embora não possa ser medida diretamente, tem a vantagem de manter a mesma direção (sempre apontando para baixo) evitando a necessidade do uso de um modelo complexo.

O campo geomagnético tem a desvantagem de não possuir a mesma inclinação e declinação por todo globo. Por esse motivo é necessário o uso de um modelo que diga qual a sua inclinação e declinação em um determinado ponto sobre o globo terrestre.

A gravidade precisa ser medida indiretamente, pela força de reação de uma massa. Essa massa também estará sujeita a acelerações que se somam à gravidade, causando até mesmo seu cancelamento, como por exemplo, em um corpo em queda livre. Já o campo geomagnético possui o inconveniente de não poder ser medido próximo a objetos metálicos, pois o campo muda de direção buscando passagem pelo material de menor relutância.

Essas e outras limitações podem ser minimizadas se forem utilizados outros sensores ou algoritmos de filtrações adequados.

Capítulo 3 – Representações da Atitude

3.1 – Introdução

Existem diversas formas matemáticas de se representar a orientação. As mais comuns são na forma de ângulos de Euler, e pelo quatérnion¹. De qualquer forma, todas essas representações convergem para uma matriz de atitude que realiza a rotação de um sistema de coordenadas para outro, mais especificamente, realiza a rotação do sistema de coordenadas de referência para o sistema de coordenadas de observação. Para cada aplicação em particular, uma dessas formas é a mais adequada. Nas próximas seções serão vistos como representar a atitude pelos ângulos de Euler (como os ângulos representam uma rotação) e também como fazê-la utilizando quatérnions. Por fim, como ir de uma representação para outra e quais as vantagens e desvantagens do uso de cada uma delas.

3.2 – Atitude por Ângulos de Euler [KUI02]

3.2.1 – Teorema e definições

Leonard Euler, enunciou o seguinte Teorema sobre seqüências de rotações:

Dados dois sistemas de coordenadas ortogonais e independentes eles pode ser associados por uma seqüência de rotações (não mais que três) sobre os eixos de coordenadas desde que não haja duas rotações consecutivas sobre um mesmo eixo.

Analisando o teorema acima é possível concluir que uma matriz de atitude pode ser decomposta em três rotações, ou seja, a atitude por ser uma matriz de rotação, pode ser representada por uma seqüência dos ângulos de Euler.

¹ Recentemente foi adotado em alguns dicionários da Língua Portuguesa o termo *quatérnion* (repare o acento por ser proparoxítona). Do inglês, a muito tempo utilizado em textos no mundo todo como *quaternion* (sem acento).

Como são três rotações consecutivas sobre os eixos das coordenadas, então são possíveis até doze combinações

xyz	yzx	zxy
xzy	yxz	zyx
xyx	yyz	zxx
xzx	yxz	zyz

A seqüência mais popular dos ângulos de Euler é a Seqüência Aeroespacial que é constituída por uma rotação ψ em torno de z , ou outra de θ em torno de y' (ou y atual), e outra de ϕ em torno de x'' (ou x atual) com na Figura 3.1. Entretanto, é possível utilizar qualquer uma das doze seqüências.

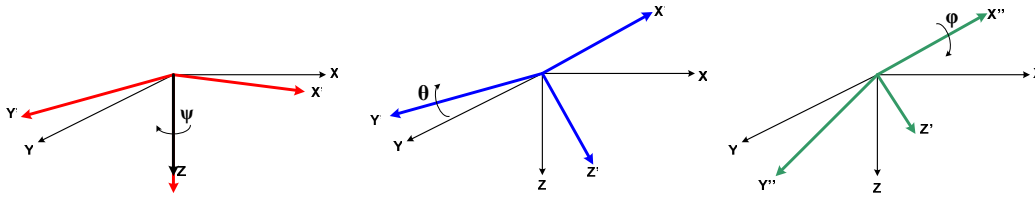


Figura 3.1 – Representação dos Ângulos de Euler

3.2.2 – A seqüência de Euler

Como dito acima, a seqüência zyx é a seqüência mais popular das doze, pois é muito utilizada em aplicações envolvendo atitude e navegação. Os ângulos de Euler estão presentes nos instrumentos dos *cockpits* dos aviões, nos algoritmos de atitude dos satélites, no jogos de computador, enfim, presentes na maioria dos casos onde é necessário determinar atitude ou realizar rotações.

É importante lembrar que o sistema de coordenadas de referência utilizado no caso da orientação do avião é o sistema de coordenadas horizontal local, onde o X é o norte, o Y o leste e o Z aponta para o centro da Terra. Caso o X seja o leste e o Y o norte, o Z apontará para cima [Cap. 2].

Por exemplo, para um avião, o eixo da cauda para o bico é o eixo X , o eixo da asa direita para esquerda é o Y e o eixo apontando de cima para baixo é o eixo Z . O ângulo de giro em Z é dito azimuth, arfagem ou apontamento, em torno de Y é a elevação, e em torno de X é a rolagem.

Os termos do inglês, *Yaw*, *Pitch* e *Roll* são utilizados para representar desvios ou perturbações que ocorrem nos respectivos eixos de rotação ZYX. Não devem ser utilizados para nomear os ângulos ψ , θ e ϕ .

O sistema de coordenadas de referência e o sistema de coordenadas de um corpo estão relacionados através dos ângulos de Euler pela seguinte matriz de rotação (ou matriz de atitude)

$$\mathbf{A} = \mathbf{R} = \mathbf{R}_\phi^x \mathbf{R}_\theta^y \mathbf{R}_\psi^z \quad (3.1)$$

onde,

$$\mathbf{R}_\phi^x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (3.2)$$

$$\mathbf{R}_\theta^y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3.3)$$

$$\mathbf{R}_\psi^z = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

portanto,

$$\mathbf{R} = \mathbf{R}_\phi^x \mathbf{R}_\theta^y \mathbf{R}_\psi^z = \mathbf{R}_\phi^x \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\sin \psi & \cos \psi & 0 \\ \sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \begin{pmatrix} -\cos \varphi \sin \psi \\ +\sin \varphi \sin \theta \cos \psi \end{pmatrix} & \begin{pmatrix} \cos \varphi \cos \psi \\ +\sin \varphi \sin \theta \sin \psi \end{pmatrix} & \sin \varphi \cos \theta \\ \begin{pmatrix} \sin \varphi \sin \psi \\ +\cos \varphi \sin \theta \cos \psi \end{pmatrix} & \begin{pmatrix} -\sin \varphi \cos \psi \\ +\cos \varphi \sin \theta \sin \psi \end{pmatrix} & \cos \varphi \cos \theta \end{bmatrix} \quad (3.5)$$

O sinal do ângulo é determinado via regra da mão direita. Cada um dos ângulos da seqüência possui uma faixa. A faixa para cada um deles são

$$\psi = [-\pi, +\pi] \quad \theta = [-\frac{\pi}{2}, +\frac{\pi}{2}] \quad \varphi = [-\pi, +\pi] \quad (3.6)$$

A matriz \mathbf{R} é uma matriz de atitude representada pelos ângulos de Euler da seqüência aeroespacial. É possível olhar para a matriz \mathbf{R} como se ela representasse uma única rotação em torno de um único eixo e que leva o corpo do sistemas de coordenadas de referência para o sistema de coordenadas atual.

3.2.3 – Problemas com ângulos de Euler

O primeiro deles é a divisão por zero, em equações envolvendo funções trigonométricas. Por exemplo, se $\cos \theta = 0$ [$\theta = \pm \pi / 2$], as funções

$$\tan \theta = \frac{\sin \theta}{\cos \theta} \quad \sec \theta = \frac{1}{\cos \theta} \quad (3.7)$$

resultam em valores indefinidos.

Neste ponto particularmente, ocorre um problema conhecido como *Gimbal Lock*, ou seja, os eixos de rolagem e arfagem se coincidem.

3.3 – Atitude por Quatérnion [KUI02]

O quatérnion é uma entidade matemática de 4 elementos. O quatérnion pode operar uma rotação da mesma forma que uma matriz de rotação, por esse fato, o

quatérnion também é capaz de representar a atitude. Veremos a seguir as definições e algumas propriedades dos quatérnions.

3.3.1 – Introdução ao quatérnion

Em 1843 Sir William Rowan Hamilton inventou, o então chamado número hipercomplexo, nomeado quatérnion. O quatérnion, como sugere o nome, é uma quantia híbrida feita da soma de um escalar a um vetor. Assim, o quatérnion \tilde{q} é dado por

$$\tilde{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} = q_0 + \mathbf{q} \quad (3.8)$$

Onde q_0 é a parte escalar e \mathbf{q} a parte vetorial. Os vetores \mathbf{i} , \mathbf{j} e \mathbf{k} são unitários e ortogonais entre si, formando uma base canônica.

O quatérnion também pode ser escrito na forma a seguir²

$$\tilde{q} = (q_0, q_1, q_2, q_3) = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} \text{ ou } \tilde{q} = (q_1, q_2, q_3, q_0) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_0 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_0 \end{bmatrix} \quad (3.9)$$

A parte fundamental da teoria dos quatérnions é que

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \quad (3.10)$$

Essa relação entre os vetores canônicos é o que dá suporte à álgebra do conjunto dos quatérnions.

Os quatérnions definidos como em (3.8) são objetos ‘estranhos’ na matemática usual e precisam ter suas operações básicas definidas, ou seja, é preciso mostrar que quatérnions podem ser igualados, somados, subtraídos e multiplicados.

² Existe uma notação dúbia que considera o quatérnion como vetor ora com sua parte escalar sendo o primeiro elemento, ora como último elemento. Quanto o quatérnion tem sua componente escalar como sendo o primeiro elemento, então este elemento é dito q_0 , quanto o elemento escalar é tido como o último elemento do vetor, geralmente é chamado por q_4 .

3.3.2 – Definição das operações com quatérnions

Soma e subtração

Dois quatérnions \tilde{p} e \tilde{q} , definidos como em (3.8) podem ser somados como a seguir

$$\tilde{p} + \tilde{q} = (p_0 + q_0) + (p_1 + q_1)\mathbf{i} + (p_2 + q_2)\mathbf{j} + (p_3 + q_3)\mathbf{k} \quad (3.11)$$

E consequentemente subtraídos como

$$\tilde{p} + (-\tilde{q}) = \tilde{p} - \tilde{q} = (p_0 - q_0) + (p_1 - q_1)\mathbf{i} + (p_2 - q_2)\mathbf{j} + (p_3 - q_3)\mathbf{k} \quad (3.12)$$

Note que na soma ou subtração de dois quatérnions, o resultado é um quatérnion também. Isso significa que o quatérnion é um conjunto fechado nestas operações.

Intuitivamente, é possível notar que o quatérnion zero é o elemento neutro da soma, assim como na soma dos números reais. Outras propriedades de número reais para soma também servem aos quatérnions, como a associatividade e comutatividade:

$$\tilde{p} + (\tilde{q} + \tilde{r}) = \tilde{p} + (\tilde{r} + \tilde{q}) \quad \tilde{p} + \tilde{q} = \tilde{q} + \tilde{p} \quad (3.13)$$

Definição da multiplicação entre quatérnion e escalar

Exatamente com no caso dos vetores para \mathbb{R}^3 , o produto entre um escalar e um quatérnion é definido de forma simples. Se c é um escalar e \tilde{q} é o quatérnion

$$\tilde{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

Então o produto entre eles será

$$c\tilde{q} = cq_0 + cq_1\mathbf{i} + cq_2\mathbf{j} + cq_3\mathbf{k} \quad (3.14)$$

Note que multiplicar um quatérnion por uma constante resulta em um quatérnion escalado, ou seja, o procedimento é o mesmo que multiplicar uma constante por um vetor ou uma matriz.

Definição do produto entre quatérnions

Dados \tilde{p} e \tilde{q}

$$\tilde{p} = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}$$

$$\tilde{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

O produto $\tilde{p} \otimes \tilde{q}$ será

$$\begin{aligned}\tilde{p} \otimes \tilde{q} &= (p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k})(q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}) \\ &= p_0q_0 + p_0q_1\mathbf{i} + p_0q_2\mathbf{j} + p_0q_3\mathbf{k} \\ &\quad + p_1q_0\mathbf{i} + p_1q_1\mathbf{i}^2 + p_1q_2\mathbf{ij} + p_1q_3\mathbf{ik} \\ &\quad + p_2q_0\mathbf{j} + p_2q_1\mathbf{ji} + p_2q_2\mathbf{j}^2 + p_2q_3\mathbf{jk} \\ &\quad + p_3q_0\mathbf{k} + p_3q_1\mathbf{ki} + p_3q_2\mathbf{kj} + p_3q_3\mathbf{k}^2\end{aligned}$$

Aplicando as relações entre os vetores \mathbf{i}, \mathbf{j} e \mathbf{k} vistos na Equação (3.10) o produto se torna

$$\begin{aligned}\tilde{p} \otimes \tilde{q} &= p_0q_0 + p_0q_1\mathbf{i} + p_0q_2\mathbf{j} + p_0q_3\mathbf{k} \\ &\quad + p_1q_0\mathbf{i} - p_1q_1 + p_1q_2\mathbf{k} - p_1q_3\mathbf{j} \\ &\quad + p_2q_0\mathbf{j} - p_2q_1\mathbf{k} - p_2q_2 + p_2q_3\mathbf{i} \\ &\quad + p_3q_0\mathbf{k} + p_3q_1\mathbf{j} - p_3q_2\mathbf{i} - p_3q_3\end{aligned}$$

Reagrupando os termos tem-se

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

Assim, é possível notar que o produto entre dois quatérnions pode ser representado em função das suas partes escalar e vetorial como

$$\tilde{r} = \tilde{p} \otimes \tilde{q} = p_0 q_0 - \mathbf{p} \bullet \mathbf{q} + p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q} \quad (3.15)$$

Observando a Equação (3.15) não é difícil concluir que o resultado do produto é também um quatérnion. Isso pelo fato do produto das constantes e o produto interno resultarem em constantes que se subtraem e, as operações escalar-vetor e vetor-vetor resultarem em vetores, portanto, sua soma também é um vetor.

De outra forma, pode-se considerar que o quatérnion \tilde{r} resultante pode ser dado pelo produto matricial

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3.16)$$

Agora, observando as formas do produto entre quatérnions das Equações (3.15) e (3.16) pode-se concluir que o elemento neutro em uma multiplicação entre quatérnions é o quatérnion unitário dado por

$$\tilde{p} = 1 + \mathbf{0} \quad (3.17)$$

O Complexo Conjugado de um Quatérnion

Um importante conceito algébrico dos quatérnions é seu complexo conjugado. O complexo conjugado do seguinte quatérnion

$$\tilde{q} = q_0 + \mathbf{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}$$

é definido como

$$\tilde{q}^* = q_0 - \mathbf{q} = q_0 - q_1 \mathbf{i} - q_2 \mathbf{j} - q_3 \mathbf{k} \quad (3.18)$$

Assim como nos números complexos, a parte escalar e vetorial de um quatérnion pode ser extraída fazendo

$$\begin{aligned}\frac{\tilde{q} + \tilde{q}^*}{2} &= q_0 \\ \frac{\tilde{q} - \tilde{q}^*}{2} &= \mathbf{q}\end{aligned}\tag{3.19}$$

A Norma do Quatérnion

Outro conceito importante relacionado ao quatérnion é sua norma. A norma de um quatérnion é denotada por $|\tilde{q}|$ e definida como

$$|\tilde{q}| = \sqrt{\tilde{q}^* \otimes \tilde{q}}\tag{3.20}$$

A Inversa do Quatérnion

A inversa de um número, por definição, deve ser um número cujo produto com seu inverso resulte no elemento neutro da multiplicação. Assim pela definição de inverso temos

$$\tilde{q}^{-1} \otimes \tilde{q} = \tilde{q} \otimes \tilde{q}^{-1} = 1$$

Multiplicando ambos os lados pelo complexo conjugado

$$\begin{aligned}\tilde{q}^{-1} \otimes \tilde{q} \otimes \tilde{q}^* &= \tilde{q}^* \otimes \tilde{q} \otimes \tilde{q}^{-1} = \tilde{q}^* \\ \tilde{q}^{-1} |\tilde{q}|^2 &= \tilde{q}^*\end{aligned}$$

Portanto, o inverso do quatérnion é dado por

$$\tilde{q}^{-1} = \frac{\tilde{q}^*}{|\tilde{q}|^2}\tag{3.21}$$

Se a norma do quatérnion for igual a 1, então o inverso do quatérnion é seu próprio conjugado.

Normalizando um Quatérnion

Um quatérnion pode ser normalizado da mesma forma que um número complexo, ou seja, dividindo o mesmo pela sua magnitude, assim

$$\hat{q} = \frac{\tilde{q}}{|\tilde{q}|} = \frac{\tilde{q}}{\sqrt{\tilde{q}^* \otimes \tilde{q}}} \quad (3.22)$$

Onde \hat{q} simboliza o quatérnion unitário.

3.3.3 – Representando um vetor no domínio dos quatérnions

Um vetor pode ser representado no domínio dos quatérnions adicionando-se zero a componente escalar, ou seja, um vetor \mathbf{v} qualquer será representado como

$$\tilde{\mathbf{v}} = 0 + \mathbf{v} = (0, \mathbf{v}) = \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (3.23)$$

Um quatérnion cuja parte escalar é zero, é denominado quatérnion *puro*.

3.3.4 – O quatérnion representando uma rotação

Restringindo o módulo do vetor a unidade, então o quatérnion unitário \hat{q} pode ser escrito como

$$\hat{q} = \cos \alpha + \hat{\mathbf{u}} \sin \alpha = (\cos \alpha, \hat{\mathbf{u}} \sin \alpha) \quad (3.24)$$

Onde $\hat{\mathbf{u}}$ é um vetor unitário e α um ângulo pertencente ao intervalo $[-\pi, \pi]$.

Um quatérnion pode atuar como um operador de rotação que gira um vetor \mathbf{v} de um ângulo θ sobre o vetor $\hat{\mathbf{u}}$ através do operador $L_q(.)$ [KUI98] da forma

$$L_q(\mathbf{v}) = \hat{q}^* \otimes \mathbf{v} \otimes \hat{q} = \hat{q}^* \otimes \tilde{\mathbf{v}} \otimes \hat{q} \quad (3.25)$$

Onde \hat{q} é um quatérnion na forma

$$\hat{q} = \cos \frac{\theta}{2} + \hat{\mathbf{u}} \sin \frac{\theta}{2} \quad (3.26)$$

E \mathbf{v} um vetor representando um quatérnion puro.

Aplicando o operador de rotação definido pela equação (3.25) no vetor \mathbf{v} e considerando que o vetor resultando seja \mathbf{w} , então

$$\begin{aligned} \mathbf{w} &= L_q(\mathbf{v}) = \tilde{q}^* \otimes \tilde{\mathbf{v}} \otimes \tilde{q} \\ &= (q_0 - \mathbf{q}) \otimes (0 + \mathbf{v}) \otimes (\mathbf{q} + q_0) \\ &= (2q_0^2 - 1)\mathbf{v} + 2(\mathbf{v} \cdot \mathbf{q})\mathbf{q} + 2q_0(\mathbf{v} \times \mathbf{q}) \end{aligned}$$

Expandindo cada termo em sua forma matricial, tem-se

$$(2q_0^2 - 1)\mathbf{v} = \begin{bmatrix} (2q_0^2 - 1) & 0 & 0 \\ 0 & (2q_0^2 - 1) & 0 \\ 0 & 0 & (2q_0^2 - 1) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$2(\mathbf{v} \cdot \mathbf{q})\mathbf{q} = \begin{bmatrix} 2q_1^2 & 2q_1q_2 & 2q_1q_3 \\ 2q_1q_2 & 2q_2^2 & 2q_2q_3 \\ 2q_1q_3 & 2q_2q_3 & 2q_3^2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$2q_0(\mathbf{v} \times \mathbf{q}) = \begin{bmatrix} 0 & 2q_0q_3 & -2q_0q_2 \\ -2q_0q_3 & 0 & 2q_0q_1 \\ 2q_0q_2 & -2q_0q_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Somando as matrizes acima, já que todas possuem \mathbf{v} em comum, então \mathbf{w} será dado por

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (3.27)$$

ou na forma,

$$\mathbf{w} = \mathbf{Q}\mathbf{v} \quad (3.28)$$

onde, a matriz \mathbf{Q} pode ser compreendida como uma matriz de rotação para \mathbf{v} , levando-o a \mathbf{w} .

3.3.5 – Derivada de um quatérnion

Dois quatérnions unitários podem ser associados por um quatérnion de transição, ou seja, dado os quatérnions $\tilde{q}(t)$, $\tilde{q}(t + \Delta t)$ e o quatérnion $\Delta\tilde{r}(t)$ é possível escrever

$$\tilde{q}(t + \Delta t) = \tilde{q}(t) \otimes \Delta\tilde{r}(t) \quad (3.29)$$

Onde $\Delta\tilde{r}(t) = \cos(\Delta\alpha/2) + \hat{\mathbf{u}}(t)\sin(\Delta\alpha/2)$ é o quatérnion de transição. Considerando o ângulo α incremental e substituindo em (3.29) tem-se

$$\tilde{q}(t + \Delta t) = \tilde{q}(t) \otimes (1 + \hat{\mathbf{u}}(t)\Delta\alpha/2)$$

$$\tilde{q}(t + \Delta t) - \tilde{q}(t) = \frac{1}{2} \tilde{q}(t) \otimes \hat{\mathbf{u}}(t)\Delta\alpha$$

Dividindo ambos os lados por Δt e aplicando limite com $\Delta t \rightarrow 0$ temos

$$\lim_{\Delta t \rightarrow 0} \frac{\tilde{q}(t + \Delta t) - \tilde{q}(t)}{\Delta t} = \frac{1}{2} \lim_{\Delta t \rightarrow 0} \frac{\tilde{q}(t) \otimes \hat{\mathbf{u}}(t)\Delta\alpha}{\Delta t}$$

$$\frac{d}{dt} \tilde{q}(t) = \frac{1}{2} \tilde{q}(t) \otimes \hat{\mathbf{u}}(t) \lim_{\Delta t \rightarrow 0} \frac{\Delta \alpha}{\Delta t}$$

$$\dot{\tilde{q}}(t) = \frac{1}{2} \tilde{q}(t) \otimes \hat{\mathbf{u}}(t) \omega(t)$$

Como $\omega(t)$ é uma função escalar e $\hat{\mathbf{u}}(t)$ um vetor unitário, então é possível escrever um quatérnion puro $\tilde{\omega}(t)$ como

$$\tilde{\omega}(t) = 0 + \hat{\mathbf{u}}(t) \omega(t) = 0 + \boldsymbol{\omega}(t)$$

e assim,

$$\dot{\tilde{q}}(t) = \frac{1}{2} \tilde{q}(t) \otimes \tilde{\omega}(t) \quad (3.30)$$

O quatérnion $\tilde{\omega}(t)$ pode ser compreendido com sendo um quatérnion “velocidade de rotação”. Ainda, como a derivada de $\tilde{q}(t)$ é feita por meio de um produto entre quatérnions, então é possível expressá-lo na forma matricial (vide Equação 3.16) como sendo

$$\dot{\tilde{q}} = \frac{1}{2} \tilde{q}(t) \otimes \tilde{\omega}(t) = \frac{1}{2} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3.31)$$

3.4 – Relações entre as representações

Como visto nas seções acima, tanto os ângulos de Euler em uma matriz, quanto o quatérnion, podem representar uma orientação, uma rotação, enfim, uma atitude. Vejamos como relacioná-los.

3.4.1 – Ângulos de Euler para quatérnion

Encontrando o Quatérnion a partir dos ângulos de Euler

Dados os ângulos de Euler como obter o quatérnion correspondente?

Se o quatérnion pode representar uma rotação pelo operador de rotação $L_q(.)$, então é possível reescrever as rotações da sequência aeroespacial de Ψ em torno de Z, Θ em torno de Y e Φ em torno de X, por

$$\tilde{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} = \tilde{q}_z \otimes \tilde{q}_y \otimes \tilde{q}_x \quad (3.32)$$

onde \tilde{q}_z , \tilde{q}_y e \tilde{q}_x são respectivamente

$$\tilde{q}_z = \cos \frac{\Psi}{2} + \mathbf{k} \sin \frac{\Psi}{2} \quad (3.33)$$

$$\tilde{q}_y = \cos \frac{\Theta}{2} + \mathbf{j} \sin \frac{\Theta}{2} \quad (3.34)$$

$$\tilde{q}_x = \cos \frac{\Phi}{2} + \mathbf{i} \sin \frac{\Phi}{2} \quad (3.35)$$

Calculando o produto entre os quatérnions como na Equação (3.32), ter-se-á então

$$\tilde{q} = q_z \otimes q_y \otimes q_x = \left(\cos \frac{\Psi}{2} + \mathbf{k} \sin \frac{\Psi}{2} \right) \otimes \left(\cos \frac{\Theta}{2} + \mathbf{j} \sin \frac{\Theta}{2} \right) \otimes \left(\cos \frac{\Phi}{2} + \mathbf{i} \sin \frac{\Phi}{2} \right)$$

Fazendo a distribuição dos produtos acima, encontramos as seguinte componentes de \tilde{q}

$$q_0 = \cos \frac{\Psi}{2} \cos \frac{\Theta}{2} \cos \frac{\Phi}{2} + \sin \frac{\Psi}{2} \sin \frac{\Theta}{2} \sin \frac{\Phi}{2} \quad (3.36)$$

$$q_1 = \cos \frac{\Psi}{2} \cos \frac{\Theta}{2} \sin \frac{\Phi}{2} + \sin \frac{\Psi}{2} \sin \frac{\Theta}{2} \cos \frac{\Phi}{2} \quad (3.37)$$

$$q_2 = \cos \frac{\Psi}{2} \sin \frac{\Theta}{2} \cos \frac{\Phi}{2} + \sin \frac{\Psi}{2} \cos \frac{\Theta}{2} \sin \frac{\Phi}{2} \quad (3.38)$$

$$q_3 = \sin \frac{\Psi}{2} \cos \frac{\Theta}{2} \cos \frac{\Phi}{2} + \cos \frac{\Psi}{2} \sin \frac{\Theta}{2} \sin \frac{\Phi}{2} \quad (3.39)$$

Encontrando o quatérnion a partir da matriz $\mathbf{A} = \mathbf{Q}$

Quando se tem somente a matriz de atitude (ou rotação) \mathbf{A} , e deseja-se saber o quatérnion que a representa, basta utilizar algumas observações sobre a matriz de interesse \mathbf{Q} . Primeiramente, duas matrizes são iguais, se somente se, são iguais termo a termo, então

$$\mathbf{A} = \mathbf{Q}$$

$$\begin{bmatrix} A_{11} & A_{12} & A_{31} \\ A_{21} & A_{22} & A_{32} \\ A_{31} & A_{23} & A_{33} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{31} \\ Q_{21} & Q_{22} & Q_{32} \\ Q_{31} & Q_{23} & Q_{33} \end{bmatrix}$$

A matriz \mathbf{Q} é dada pela Equação (3.28). Substituindo o lado direito da relação acima, tem-se

$$\begin{bmatrix} A_{11} & A_{12} & A_{31} \\ A_{21} & A_{22} & A_{32} \\ A_{31} & A_{23} & A_{33} \end{bmatrix} = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix}$$

Se as matrizes acima são iguais, então seu traços também serão, logo

$$\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{Q})$$

$$\sum_{i=1}^3 A_{ii} \equiv \text{tr}(\mathbf{Q}) = 4q_0^2 - 1$$

Assim é possível encontrar a componente escalar do quatérnion fazendo

$$q_0 = \frac{1}{2} \sqrt{\text{tr}(\mathbf{A}) + 1} = \frac{1}{2} \sqrt{\sum_{i=1}^3 A_{ii} + 1} \quad (3.40)$$

As demais componentes podem ser obtidas fazendo diferença

$$\mathbf{A} - \mathbf{A}^T = \mathbf{Q} - \mathbf{Q}^T$$

$$\begin{bmatrix} 0 & A_{12} - A_{21} & A_{31} - A_{31} \\ A_{21} - A_{12} & 0 & A_{32} - A_{23} \\ A_{31} - A_{12} & A_{23} - A_{32} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 4q_0q_3 & -4q_0q_2 \\ -4q_0q_3 & 0 & 4q_0q_1 \\ 4q_0q_2 & -4q_0q_1 & 0 \end{bmatrix}$$

Igualando os termos da diagonal superior ou da diagonal inferior, ter-se-á:

$$q_1 = \frac{A_{23} - A_{32}}{4q_0} \quad (3.41)$$

$$q_2 = \frac{A_{31} - A_{13}}{4q_0} \quad (3.42)$$

$$q_3 = \frac{A_{12} - A_{21}}{4q_0} \quad (3.43)$$

3.4.2 – Quatérnion para ângulos de Euler

Encontrando os ângulos a partir dos elementos do quatérnion

Dados as quatro componentes do quatérnion, como encontrar os ângulos de Euler para a seqüência aeroespacial?

Ambas as matrizes \mathbf{R} da Equação (3.5) e \mathbf{Q} da Equação (3.27) são matrizes de rotação. Se elas representam a mesma rotação elas devem ser iguais termo a termo. Assim, igualando-as é possível extrair a informação desejada. Então, considerando, a igualdade da Equação (3.44)

$$R_{ij} = Q_{ij} \quad (3.44)$$

onde o termos Q_{ij} são conhecidos numericamente.

É possível encontrar as relações para os ângulos como a seguir

$$\frac{R_{12}}{R_{11}} = \tan \psi = \frac{Q_{12}}{Q_{11}} = \frac{2q_1q_2 + 2q_0q_3}{2q_0^2 + 2q_1^2 - 1} \quad (3.45)$$

$$-R_{13} = \sin \theta = -Q_{13} = -(2q_1q_3 - 2q_0q_2) \quad (3.46)$$

$$\frac{R_{23}}{R_{33}} = \tan \phi = \frac{Q_{23}}{Q_{33}} = \frac{2q_2q_3 + 2q_0q_1}{2q_0^2 + 2q_3^2 - 1} \quad (3.47)$$

Ambigüidades trigonométricas (como saber o quadrante quando há duas soluções) podem ser resolvidas considerando que $\cos \theta$ deve ser sempre positivo.

3.5 – Comparação entre as Representações [DUM99]

Quatérnions e os ângulos de Euler são utilizados com frequência para gerar a matriz de atitude **A**, ou representá-la. Cada notação tem suas vantagens e desvantagens.

A mais significativa vantagem dos quatérnions é que não existe singularidade na representação quanto o ângulo de elevação atinge $\pm\pi/2$. Qualquer orientação pode ser descrita por quatérnions.

O uso dos quatérnions permite contornar as funções transcendentais de cálculos em equações dinâmicas. Isto reduz o custo computacional.

Os quatérnions são compactos e simples de entender. Nas funções de rotação da famosa biblioteca gráfica *OpenGL (Open Graphic Library)*, a função *glRotate* tem como argumento de entrada um vetor de rotação e um ângulo, ou seja, um quatérnion [SHR04].

Existem algumas desvantagens de se utilizar o quatérnion. Uma delas é que existem sempre dois quatérnions para indicar uma atitude. Por exemplo, o quatérnion

$$\left(-\cos \frac{\beta}{2}, \hat{\mathbf{u}} \sin \frac{\beta}{2}\right) \quad (3.48)$$

e o quatérnion

$$\left(\cos\frac{\beta}{2}, -\hat{\mathbf{u}}\sin\frac{\beta}{2}\right) \quad (3.49)$$

significam a mesma rotação, pois houve a inversão do sentido do eixo de rotação, mas também do sentido do ângulo, o que causa a mesma rotação.

Os quatérnions *unitários* estão sujeitos a seguinte restrição

$$|\tilde{q}|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = \left(\cos\frac{\beta}{2}\right)^2 + |\hat{\mathbf{u}}|^2 \left(\sin\frac{\beta}{2}\right)^2 = 1 \quad (3.49)$$

O que implica que a partir de três elementos conhecidos é possível calcular o que falta.

Capítulo 4 – Algoritmos para a Determinação da Atitude

4.1 – Introdução

Encontrar a atitude é calcular a matriz de rotação que realize a transformação de coordenadas de vetores arbitrários entre dois sistemas de referência, um inercial e outro fixo no corpo, com base nas coordenadas de um conjunto de vetores, conhecidas em ambos os sistemas. Em princípio, é possível definir uma matriz de atitude \mathbf{A} que satisfaça

$$\mathbf{A}\hat{\mathbf{v}}_i = \hat{\mathbf{w}}_i \quad (i = 1, \dots, n) \quad (4.1)$$

onde $\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_n$ é um conjunto de vetores unitários medidos no sistema de coordenadas inercial, denominados vetores de referência. As coordenadas dos vetores de referência são a projeção de grandezas vetoriais nos eixos do sistema de referência inercial. Por exemplo, a direção do campo magnético terrestre, a direção da gravidade e a direção do sol podem ser utilizados como grandezas vetoriais ou vetores fonte.

Os vetores $\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_n$ são também vetores unitários, cujas coordenadas são dadas pela projeção de cada vetor fonte nos eixos do sistema de coordenadas do corpo. Esses vetores, por sua vez, são ditos vetores de observação.

Na prática, tanto as medidas dos vetores no sistema de coordenadas do corpo quanto as suas medidas no sistema de coordenadas inercial são corrompidas por ruído e, portanto, a Equação (4.1) pode não ter solução exata nem mesmo quando há apenas duas referências vetoriais ($n = 2$). Para o caso particular em que $n = 2$, o algoritmo TRIAD oferece uma opção simples para encontrar um valor aproximado para a matriz \mathbf{A} de maneira determinística, embora não ótima, mas com um custo computacional relativamente baixo. Quando $n > 2$, o TRIAD torna-se ineficiente porque existem $C_{n,2}$

soluções e nenhuma delas é ótima. Além disso, a matriz de covariância ficaria demasiadamente complicada.

Shuster e Oh propuseram o algoritmo QUEST como uma solução ótima para determinar a atitude quando existem dois ou mais vetores fonte [SHU80]. Este algoritmo implementa de modo eficiente a solução formulada por Davenport, denominada “método q” [WER78] para o problema de minimização da função perda proposta por Wahba [WAH66].

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i |\hat{\mathbf{w}}_i - \mathbf{A} \hat{\mathbf{v}}_i|^2 \quad (4.2)$$

onde é desejável encontrar uma matriz \mathbf{A}_{opt} que minimiza $L(\mathbf{A})$ com $a_i, i = 1, 2, \dots, n$ sendo um conjunto de pesos escalares, reais e não negativos.

Além da otimização da determinação da atitude para n referências, o algoritmo QUEST mantém vantagens computacionais similares aos algoritmos determinísticos. Adicionalmente, e possivelmente mais importante ainda, este algoritmo fornece uma simples expressão analítica para o cálculo da matriz de covariância, baseada num modelo de erros simplificado para os vetores de referência e de observação.

Outros algoritmos foram desenvolvidos no decorrer dos últimos 20 anos, como o ESOQ, ESOQ2, FOAM e outros. Nas secções a seguir será dada ênfase principalmente aos algoritmos TRIAD e QUEST, desenvolvendo com profundidade a álgebra destes algoritmos. Não apenas a atitude, mas também a matriz de covariância de cada algoritmo será explorada. Isto porque estes algoritmos podem ser implementados em hardware embarcado, como será visto no capítulo 6.

4.2 – O Algoritmo TRIAD (*TRI-axis Attitude Determination*)

4.2.1 – A Matriz de atitude do TRIAD

Dado dois vetores unitários de referência $\hat{\mathbf{v}}_1$ e $\hat{\mathbf{v}}_2$ que são projeções dos vetores fonte no sistema de coordenadas inercial e os seus respectivos vetores unitários de observação $\hat{\mathbf{w}}_1$, $\hat{\mathbf{w}}_2$ que são as projeções dos vetores fonte no sistema de coordenadas

do corpo, é desejável encontrar uma matriz ortogonal¹ \mathbf{A} que satisfaça, ainda que de modo aproximado, às seguintes equações:

$$\mathbf{A}\hat{\mathbf{v}}_1 = \hat{\mathbf{w}}_1 \quad \mathbf{A}\hat{\mathbf{v}}_2 = \hat{\mathbf{w}}_2 \quad (4.3)$$

Para encontrar \mathbf{A} inicialmente constroem-se dois conjuntos de três vetores ortogonais unitários ou tríades, um baseado nos vetores de referência e outro baseado nos vetores de observação, a saber:

$$\hat{\mathbf{r}}_1 = \hat{\mathbf{v}}_1 \quad \hat{\mathbf{r}}_2 = \frac{(\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2)}{|\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2|} \quad \hat{\mathbf{r}}_3 = \frac{(\hat{\mathbf{v}}_1 \times (\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2))}{|\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2|} \quad (4.4)$$

$$\hat{\mathbf{s}}_1 = \hat{\mathbf{w}}_1 \quad \hat{\mathbf{s}}_2 = \frac{(\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \quad \hat{\mathbf{s}}_3 = \frac{(\hat{\mathbf{w}}_1 \times (\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2))}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \quad (4.5)$$

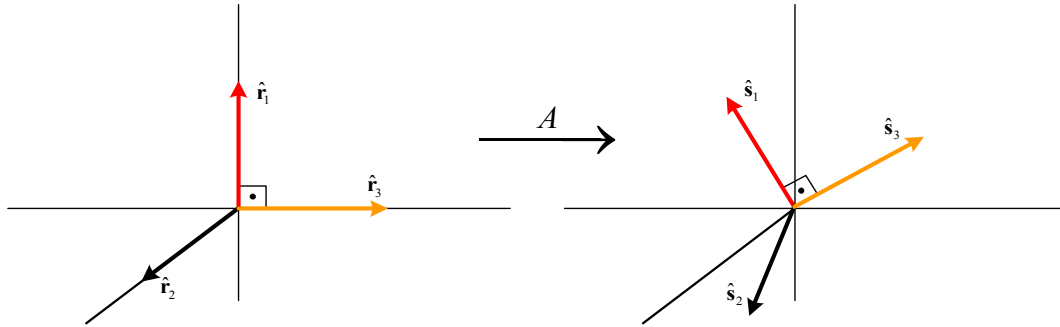


Figura 4.1. Representação gráfica da rotação dada por matriz de atitude. A matriz \mathbf{A} gira o sistema de coordenadas de referência para o sistema de coordenadas de observação.

Há somente uma matriz \mathbf{A} que roda $\hat{\mathbf{r}}_i$ para $\hat{\mathbf{s}}_i$, ou seja, que satisfaz a:

$$\mathbf{A}\hat{\mathbf{r}}_i = \hat{\mathbf{s}}_i \quad i = 1, 2, 3 \quad (4.6)$$

A Figura 4.1 mostra a rotação representada pela Equação (4.6). Os vetores $\hat{\mathbf{r}}_i$ da referência são rodados para a posição dos vetores $\hat{\mathbf{s}}_i$ no sistema de coordenadas de observação.

¹ Uma matriz *ortogonal*, que possui determinante unitário, tem sua transposta igual a sua inversa.

Dados os vetores $\hat{\mathbf{r}}_i$, é possível concatená-los para construir uma matriz de referência:

$$\mathbf{M}_{\text{ref}} = [\hat{\mathbf{r}}_1 : \hat{\mathbf{r}}_2 : \hat{\mathbf{r}}_3] \quad (4.7)$$

Da mesma forma a matriz de observação pode ser construída por $\hat{\mathbf{s}}_i$:

$$\mathbf{M}_{\text{obs}} = [\hat{\mathbf{s}}_1 : \hat{\mathbf{s}}_2 : \hat{\mathbf{s}}_3] \quad (4.8)$$

Note-se que $\hat{\mathbf{r}}_i$ e $\hat{\mathbf{s}}_i$ são vetores de dimensão 3×1 .

Agora, a Equação (4.6) pode ser reescrita de modo a associar a matriz de atitude às matrizes \mathbf{M}_{obs} e \mathbf{M}_{ref} como

$$\mathbf{A}\mathbf{M}_{\text{ref}} = \mathbf{M}_{\text{obs}} \quad (4.9)$$

Ora, as matrizes \mathbf{M}_{obs} e \mathbf{M}_{ref} satisfazem as seguintes condições:

$$\mathbf{M}_{\text{obs}}^T \mathbf{M}_{\text{obs}} = \begin{bmatrix} \hat{\mathbf{s}}_1^T \hat{\mathbf{s}}_1 & \hat{\mathbf{s}}_2^T \hat{\mathbf{s}}_1 & \hat{\mathbf{s}}_3^T \hat{\mathbf{s}}_1 \\ \hat{\mathbf{s}}_1^T \hat{\mathbf{s}}_2 & \hat{\mathbf{s}}_2^T \hat{\mathbf{s}}_2 & \hat{\mathbf{s}}_3^T \hat{\mathbf{s}}_2 \\ \hat{\mathbf{s}}_1^T \hat{\mathbf{s}}_3 & \hat{\mathbf{s}}_2^T \hat{\mathbf{s}}_3 & \hat{\mathbf{s}}_3^T \hat{\mathbf{s}}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_{3 \times 3}$$

$$\mathbf{M}_{\text{ref}}^T \mathbf{M}_{\text{ref}} = \begin{bmatrix} \hat{\mathbf{r}}_1^T \hat{\mathbf{r}}_1 & \hat{\mathbf{r}}_2^T \hat{\mathbf{r}}_1 & \hat{\mathbf{r}}_3^T \hat{\mathbf{r}}_1 \\ \hat{\mathbf{r}}_1^T \hat{\mathbf{r}}_2 & \hat{\mathbf{r}}_2^T \hat{\mathbf{r}}_2 & \hat{\mathbf{r}}_3^T \hat{\mathbf{r}}_2 \\ \hat{\mathbf{r}}_1^T \hat{\mathbf{r}}_3 & \hat{\mathbf{r}}_2^T \hat{\mathbf{r}}_3 & \hat{\mathbf{r}}_3^T \hat{\mathbf{r}}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_{3 \times 3}$$

Então, as matrizes \mathbf{M}_{obs} e \mathbf{M}_{ref} , pela definição, são matrizes ortogonais, portanto,

$$\mathbf{M}_{\text{obs}}^T = \mathbf{M}_{\text{obs}}^{-1} \quad \mathbf{M}_{\text{ref}}^T = \mathbf{M}_{\text{ref}}^{-1} \quad (4.10)$$

Então, é possível isolar a matriz de atitude multiplicando ambos os lados da Equação (4.9) por $\mathbf{M}_{\text{ref}}^T$, então $\mathbf{A}\mathbf{M}_{\text{ref}}\mathbf{M}_{\text{ref}}^T = \mathbf{M}_{\text{obs}}\mathbf{M}_{\text{ref}}^T$. Desta forma, a matriz de atitude será dada por

$$\mathbf{A} = \mathbf{M}_{\text{obs}}\mathbf{M}_{\text{ref}}^T \quad (4.11)$$

A Equação (4.11) representa a solução do algoritmo TRIAD para a atitude. Outra forma para expressá-la é

$$\mathbf{A} = \sum_{i=1}^3 \hat{\mathbf{s}}_i \hat{\mathbf{r}}_i^T \quad (4.12)$$

Uma condição necessária e suficiente para a matriz de atitude assim obtida satisfazer de modo exato à Equação (4.3) é a igualdade do produto interno a seguir

$$\hat{\mathbf{v}}_1 \bullet \hat{\mathbf{v}}_2 = \hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2 \quad (4.13)$$

Neste caso, como consequência, também será satisfeito para o caso do módulo do produto vetorial

$$|\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2| = |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2| \quad (4.14)$$

É interessante notar nas Equações (4.4) e (4.5) que os vetores $\hat{\mathbf{r}}_3$ e $\hat{\mathbf{s}}_3$ não são obtidos por cálculos simétricos. Isto acontece porque os vetores $\hat{\mathbf{v}}_1$ e $\hat{\mathbf{w}}_1$ são usados duas vezes enquanto os vetores $\hat{\mathbf{v}}_2$ e $\hat{\mathbf{w}}_2$ são utilizados apenas uma vez durante os cálculos de $\hat{\mathbf{r}}_3$ e $\hat{\mathbf{s}}_3$. Devido a esta assimetria deve-se escolher a referência mais precisa para as variáveis $\hat{\mathbf{v}}_1$ e $\hat{\mathbf{w}}_1$.

Como exemplo para o uso deste algoritmo, considere um sistema simples para a determinação da atitude, que utiliza o algoritmo TRIAD e possui duas tríades de sensores que adquirem informações de dois vetores fonte diferentes. Os vetores $\hat{\mathbf{v}}_i$ podem ser calculados inicialmente através da leitura dos sensores em uma posição pré-

determinada. Esta posição determina o sistema de coordenadas de referência. Os valores médios e normalizados destas leituras constituem os vetores $\hat{\mathbf{v}}_i$, tal como em um processo de calibração. Estes valores de calibração são empregados para calcular a atitude durante toda a utilização do sistema.

Assim, o sistema está pronto para rodar e a partir dos valores medidos dos sensores no sistema de coordenadas do corpo, podem ser construídos os vetores de observação $\hat{\mathbf{w}}_i$. Fazendo uso das Equações (4.4) e (4.5), os vetores $\hat{\mathbf{r}}_i$ e $\hat{\mathbf{s}}_i$ podem ser montados e em seguida as matrizes \mathbf{M}_{ref} e \mathbf{M}_{ref} podem ser construídas como mostrado em (4.7) e (4.8), respectivamente. Fazendo a transposta da matriz de referência e fazendo a multiplicação da Equação (4.11) tem-se a matriz de atitude. Os valores de $\hat{\mathbf{w}}_i$ são adquiridos freqüentemente dos sensores e o sistema deve ser capaz de calcular as matrizes de atitude em uma freqüência alta o suficiente para descrever corretamente eventuais movimentos de rotação ao longo do tempo.

4.2.2 – A Matriz de covariância do TRIAD

A Matriz de covariância do TRIAD é muito importante e pode ser utilizada para descrever estatisticamente o erro de atitude. O objetivo é descrever a matriz de covariância Cartesiana da atitude de forma simples e analítica, a qual depende somente das medidas dos valores $\hat{\mathbf{v}}_i$ e $\hat{\mathbf{w}}_i$. Esta matriz é muito útil em um sistema que use um Filtro de Kalman [LEF82] ou para fundir dados dos sensores de referência e dados de sensores giroscópicos, por exemplo [GEB04]. A simplicidade do algoritmo TRIAD tanto para determinação quanto na análise de erros, torna o sistema implementável em sistemas embarcados atuais, considerados de baixo poder de processamento.

O vetor erro dos ângulos é dado por

$$\delta\boldsymbol{\theta} = [\delta\theta_1 \quad \delta\theta_2 \quad \delta\theta_3]^T \quad (4.15)$$

Ele é definido como o conjunto de ângulos que roda a matriz exata da atitude para matriz de atitude medida. Então para pequenos ângulos

$$\mathbf{A} = (\mathbf{I} - \mathbf{\Omega}(\delta\boldsymbol{\theta})) \langle \mathbf{A} \rangle = \begin{bmatrix} 1 & \delta\theta_3 & -\delta\theta_2 \\ -\delta\theta_3 & 1 & \delta\theta_1 \\ \delta\theta_2 & -\delta\theta_1 & 1 \end{bmatrix} \langle \mathbf{A} \rangle \quad (4.16)$$

onde $\langle \cdot \rangle$ é definido como o operador média e $\mathbf{\Omega}(\cdot)$ é um operador linear (operador distribuição) definido como

$$\mathbf{\Omega}(\delta\boldsymbol{\theta}) = \begin{bmatrix} 0 & -\delta\theta_3 & \delta\theta_2 \\ \delta\theta_3 & 0 & -\delta\theta_1 \\ -\delta\theta_2 & \delta\theta_1 & 0 \end{bmatrix} \quad (4.17)$$

A matriz de covariância Cartesiana da atitude é definida como

$$\mathbf{P}_{\theta\theta} = \langle \delta\boldsymbol{\theta} \delta\boldsymbol{\theta}^T \rangle \quad (4.18)$$

A substituição do vetor de erros definido pela Equação (4.15) na (4.18), resulta em

$$\mathbf{P}_{\theta\theta} = \left\langle \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \\ \delta\theta_3 \end{bmatrix} \begin{bmatrix} \delta\theta_1 & \delta\theta_2 & \delta\theta_3 \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} \delta\theta_1^2 & \delta\theta_1\delta\theta_2 & \delta\theta_1\delta\theta_3 \\ \delta\theta_2\delta\theta_1 & \delta\theta_2^2 & \delta\theta_2\delta\theta_3 \\ \delta\theta_3\delta\theta_1 & \delta\theta_3\delta\theta_2 & \delta\theta_3^2 \end{bmatrix} \right\rangle \quad (4.19)$$

Para implementação computacional eficiente, é conveniente examinar a matriz de covariância Total definida como,

$$\mathbf{P} = \langle \delta\mathbf{A} \delta\mathbf{A}^T \rangle \quad (4.20)$$

onde,

$$\delta\mathbf{A} = \mathbf{A} - \langle \mathbf{A} \rangle \quad (4.21)$$

Considerando a Equação (4.16) que define o erro incluso dentro de \mathbf{A} , e a matriz de covariância Cartesiana da atitude definida pela Equação (4.18), tem-se

$$\begin{aligned}\mathbf{P} &= \langle \delta \mathbf{A} \delta \mathbf{A}^T \rangle \\ \mathbf{P} &= \langle ((\mathbf{I} - \boldsymbol{\Omega}(\delta \boldsymbol{\theta})) \langle \mathbf{A} \rangle - \langle \mathbf{A} \rangle) ((\mathbf{I} - \boldsymbol{\Omega}(\delta \boldsymbol{\theta})) \langle \mathbf{A} \rangle - \langle \mathbf{A} \rangle)^T \rangle \\ \mathbf{P} &= \langle (\langle \mathbf{A} \rangle - \boldsymbol{\Omega}(\delta \boldsymbol{\theta}) \langle \mathbf{A} \rangle - \langle \mathbf{A} \rangle) (\langle \mathbf{A} \rangle - \boldsymbol{\Omega}(\delta \boldsymbol{\theta}) \langle \mathbf{A} \rangle - \langle \mathbf{A} \rangle)^T \rangle \\ \mathbf{P} &= \langle \boldsymbol{\Omega}(\delta \boldsymbol{\theta}) \langle \mathbf{A} \rangle \langle \mathbf{A} \rangle^T \boldsymbol{\Omega}(\delta \boldsymbol{\theta})^T \rangle\end{aligned}$$

Como $\langle \mathbf{A} \rangle \langle \mathbf{A} \rangle^T = \mathbf{I}_{3 \times 3}$, \mathbf{P} é dada por

$$\mathbf{P} = \langle \boldsymbol{\Omega}(\delta \boldsymbol{\theta}) \boldsymbol{\Omega}(\delta \boldsymbol{\theta})^T \rangle \quad (4.22)$$

Abrindo a multiplicação da Equação (4.22), obtém-se

$$\mathbf{P} = \left\langle \begin{bmatrix} 0 & -\delta \theta_3 & \delta \theta_2 \\ \delta \theta_3 & 0 & -\delta \theta_1 \\ -\delta \theta_2 & \delta \theta_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -\delta \theta_3 & \delta \theta_2 \\ \delta \theta_3 & 0 & -\delta \theta_1 \\ -\delta \theta_2 & \delta \theta_1 & 0 \end{bmatrix}^T \right\rangle$$

resultando em

$$\mathbf{P} = \left\langle \begin{bmatrix} \delta \theta_3^2 + \delta \theta_2^2 & -\delta \theta_1 \delta \theta_2 & -\delta \theta_1 \delta \theta_3 \\ -\delta \theta_1 \delta \theta_2 & \delta \theta_3^2 + \delta \theta_1^2 & -\delta \theta_2 \delta \theta_3 \\ -\delta \theta_1 \delta \theta_3 & -\delta \theta_2 \delta \theta_3 & \delta \theta_1^2 + \delta \theta_2^2 \end{bmatrix} \right\rangle \quad (4.23)$$

Observando as Equações (4.19) e (4.23) há uma relação entre os traços de suas matrizes, que é

$$\text{tr}(\mathbf{P}) = 2 \langle \delta \boldsymbol{\theta}^T \delta \boldsymbol{\theta} \rangle = 2 \text{tr}(\mathbf{P}_{\theta\theta}) \quad (4.24)$$

Agora, expandindo \mathbf{P} em sua forma matricial, resulta em

$$\mathbf{P} = \langle \mathbf{\Omega}(\delta\mathbf{\theta}) \mathbf{\Omega}(\delta\mathbf{\theta})^T \rangle$$

$$\mathbf{P} = \langle \delta\mathbf{\theta}^T \delta\mathbf{\theta} \mathbf{I} - \delta\mathbf{\theta} \delta\mathbf{\theta}^T \rangle$$

$$\mathbf{P} = \langle \delta\mathbf{\theta}^T \delta\mathbf{\theta} \rangle \mathbf{I} - \langle \delta\mathbf{\theta} \delta\mathbf{\theta}^T \rangle$$

$$\mathbf{P} = \langle \delta\mathbf{\theta}^T \delta\mathbf{\theta} \rangle \mathbf{I} - \mathbf{P}_{\theta\theta}$$

Considerando a relação dada pela Equação (4.24) que fornece o valor de $\langle \delta\mathbf{\theta} \delta\mathbf{\theta}^T \rangle$ em função de \mathbf{P} e de $\mathbf{P}_{\theta\theta}$, é possível estabelecer as seguintes relações entre \mathbf{P} e $\mathbf{P}_{\theta\theta}$:

$$\mathbf{P} = \text{tr}(\mathbf{P}_{\theta\theta}) \mathbf{I} - \mathbf{P}_{\theta\theta} \quad (4.25)$$

$$\mathbf{P}_{\theta\theta} = \frac{1}{2} \text{tr}(\mathbf{P}) \mathbf{I} - \mathbf{P} \quad (4.26)$$

A relação descrita pela Equação (4.26) torna o cálculo de $\mathbf{P}_{\theta\theta}$ mais eficiente, pois o cálculo de \mathbf{P} está relacionado como a atitude e seu erro, como descrito pelas equações (4.20) e (4.21).

Por outro lado, da Equação (4.9) segue que

$$\delta \mathbf{A} \mathbf{M}_{\text{ref}} + \mathbf{A} \delta \mathbf{M}_{\text{ref}} = \delta \mathbf{M}_{\text{obs}}$$

Multiplicando ambos os lados desta equação por $\mathbf{M}_{\text{ref}}^T$ e isolando $\delta \mathbf{A}$ obtém-se:

$$\delta \mathbf{A} = (\delta \mathbf{M}_{\text{obs}} - \mathbf{A} \delta \mathbf{M}_{\text{ref}}) \mathbf{M}_{\text{ref}}^T$$

Com isso, é possível descrever \mathbf{P} como uma função das matrizes de observação, referência e atitude:

$$\mathbf{P} = \langle \delta \mathbf{A} \delta \mathbf{A}^T \rangle$$

$$\mathbf{P} = \langle (\delta \mathbf{M}_{\text{obs}} - \mathbf{A} \delta \mathbf{M}_{\text{ref}}) \mathbf{M}_{\text{ref}}^T \mathbf{M}_{\text{ref}} (\delta \mathbf{M}_{\text{obs}}^T - \delta \mathbf{M}_{\text{ref}}^T \mathbf{A}^T) \rangle$$

$$\mathbf{P} = \left\langle \delta \mathbf{M}_{\text{obs}} \delta \mathbf{M}_{\text{obs}}^T + \mathbf{A} \delta \mathbf{M}_{\text{ref}} \delta \mathbf{M}_{\text{ref}}^T \mathbf{A}^T - \delta \mathbf{M}_{\text{obs}} \delta \mathbf{M}_{\text{ref}}^T \mathbf{A}^T - \mathbf{A} \delta \mathbf{M}_{\text{ref}} \delta \mathbf{M}_{\text{obs}}^T \right\rangle$$

Sabendo que os erros lidos no frame de referência e de observação não estão correlacionados, os termos $\langle \delta \mathbf{M}_{\text{obs}} \delta \mathbf{M}_{\text{ref}}^T \rangle$ e $\langle \delta \mathbf{M}_{\text{ref}} \delta \mathbf{M}_{\text{obs}}^T \rangle$ desaparecem, resultando em

$$\mathbf{P} = \langle \delta \mathbf{M}_{\text{obs}} \delta \mathbf{M}_{\text{obs}}^T \rangle + \mathbf{A} \langle \delta \mathbf{M}_{\text{ref}} \delta \mathbf{M}_{\text{ref}}^T \rangle \mathbf{A}^T \quad (4.27)$$

Denominando o primeiro termo de \mathbf{P} como \mathbf{P}_{obs} e o segundo como $\mathbf{A} \mathbf{P}_{\text{ref}} \mathbf{A}^T$, então a matriz de covariância total fica

$$\mathbf{P} = \mathbf{P}_{\text{obs}} + \mathbf{A} \mathbf{P}_{\text{ref}} \mathbf{A}^T \quad (4.28)$$

A Equação (4.28) mostra a relação da matriz de covariância Total com as matrizes de covariância de observação e referência. Agora, as covariâncias têm que ser representadas em função dos vetores de observação e referência e de seus erros. Para fazer isto, o primeiro passo é analisar e entender a covariância destes vetores unitários, também denominados versores.

A Covariância de um Versor

Tanto $\hat{\mathbf{w}}_i$ quanto $\hat{\mathbf{v}}_i$ são vetores unitários. Eles apresentam variação ou erro somente em duas dimensões, já que os versores sempre mantêm seus módulos unitários. Assim, os erros que ocorrem nos vetores de observação e referência, respectivamente, são ortogonais aos próprios versores, pois eles devem estar contidos no plano perpendicular ao versor. Portanto, os erros dos versores $\delta \hat{\mathbf{w}}_i$ e $\delta \hat{\mathbf{v}}_i$, possuem somente dois graus de liberdade.

$$\begin{aligned} \delta \hat{\mathbf{w}}_i \bullet \hat{\mathbf{w}}_i &= 0 \\ \delta \hat{\mathbf{v}}_i \bullet \hat{\mathbf{v}}_i &= 0 \end{aligned} \quad (4.29)$$

Um vetor unitário pode ser definido como

$$\hat{\mathbf{x}} = \frac{\mathbf{x}}{|\mathbf{x}|} = \frac{\mathbf{x}}{\sqrt{\mathbf{x}^T \mathbf{x}}} \quad (4.30)$$

A variância do erro de um vetor unitário será

$$\text{var}(\hat{\mathbf{x}}) = \langle \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \rangle \quad (4.31)$$

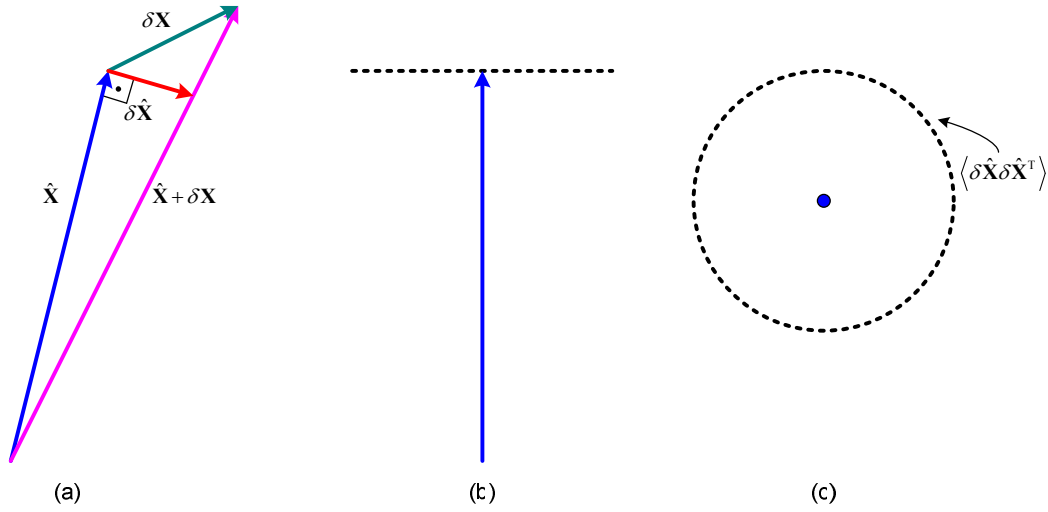


Figura 4.2 – Versores e erros.

A ilustração 4.2 (a) apresenta os vetores e seus erros. Note que o erro de um versor não muda seu módulo, pois, um versor é sempre unitário. A projeção do erro no plano perpendicular ao versor é o erro que importa realmente. Sendo o erro pequeno, então ele ocorre sempre no plano perpendicular ao versor, como no pontilhado na Figura 4.2 (b). Em 4.2 (c) é mostrada uma visão superior. O erro quadrático médio do erro da direção do versor (variância) é representado pela circunferência pontilhada.

A relação entre o erro em um vetor sem qualquer vínculo e o erro em seu valor normalizado é dada por,

$$\delta \hat{\mathbf{x}} = \frac{d\hat{\mathbf{x}}}{d\mathbf{x}} \delta \mathbf{x}$$

$$\delta \hat{\mathbf{x}} = \frac{\delta \mathbf{x}}{\sqrt{\mathbf{x}^T \mathbf{x}}} - \frac{\mathbf{x}}{2} (\mathbf{x}^T \mathbf{x})^{-\frac{3}{2}} (2\mathbf{x}^T) \delta \mathbf{x}$$

$$\delta \hat{\mathbf{x}} = \frac{1}{|\mathbf{x}|} \left[\mathbf{I} - \frac{\mathbf{x}\mathbf{x}^T}{|\mathbf{x}|^2} \right] \delta \mathbf{x},$$

onde \mathbf{I} representa a matriz identidade.

Colocando a expressão acima como função dos versores, então

$$\delta \hat{\mathbf{x}} = \frac{1}{|\mathbf{x}|} [\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T] \delta \mathbf{x} \quad (4.32)$$

A variância pode ser calculada substituindo a Equação (4.32) na (4.31), resultando em

$$\begin{aligned} \langle \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \rangle &= \left\langle (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T) \frac{\delta \mathbf{x}}{|\mathbf{x}|} \left((\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T) \frac{\delta \mathbf{x}}{|\mathbf{x}|} \right)^T \right\rangle \\ \langle \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \rangle &= \left\langle (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T) \frac{\delta \mathbf{x}}{|\mathbf{x}|} \frac{\delta \mathbf{x}^T}{|\mathbf{x}|} (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T)^T \right\rangle \\ \langle \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \rangle &= (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T) \frac{\langle \delta \mathbf{x} \delta \mathbf{x}^T \rangle}{|\mathbf{x}|^2} (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T)^T \end{aligned} \quad (4.33)$$

Visto que o módulo de \mathbf{x} não está restrito a um valor fixo, o erro $\delta \mathbf{x}$ pode ocorrer em qualquer direção. Considerando que a distribuição de $\delta \mathbf{x}$ seja a mesma em todas as direções, a sua covariância é dada por

$$\langle \delta \mathbf{x} \delta \mathbf{x}^T \rangle = \mathbf{I} |\mathbf{x}|^2 \sigma_{\mathbf{x}}^2,$$

onde $|\mathbf{x}| \sigma_{\mathbf{x}}$ representa o desvio padrão da distribuição de $\delta \mathbf{x}$.

Observando as seguintes propriedades

$$(\hat{\mathbf{x}}\hat{\mathbf{x}}^T)^T = \hat{\mathbf{x}}\hat{\mathbf{x}}^T \quad (4.34a)$$

$$(\hat{\mathbf{x}}\hat{\mathbf{x}}^T)^n = \hat{\mathbf{x}}\hat{\mathbf{x}}^T \quad (4.34b)$$

$$(\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T)^T = \mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T \quad (4.34c)$$

$$(\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T)^n = \mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T \quad (4.34d)$$

A Equação (4.33) pode ser reduzida para

$$\langle \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \rangle = (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T) \mathbf{I} \sigma_x^2 (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T)$$

$$\langle \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \rangle = \sigma_x^2 (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T) (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T)$$

$$\langle \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \rangle = \sigma_x^2 (\mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T) \quad (4.35)$$

Aplicando a Equação (4.35) para o cálculo das covariâncias de $\delta \hat{\mathbf{v}}_i$ e $\delta \hat{\mathbf{w}}_i$, as seguintes covariâncias são obtidas

$$\langle \delta \hat{\mathbf{w}}_1 \delta \hat{\mathbf{w}}_1^T \rangle = \sigma_{w_1}^2 (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T) \quad (4.36a)$$

$$\langle \delta \hat{\mathbf{w}}_2 \delta \hat{\mathbf{w}}_2^T \rangle = \sigma_{w_2}^2 (\mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T) \quad (4.36b)$$

$$\langle \delta \hat{\mathbf{v}}_1 \delta \hat{\mathbf{v}}_1^T \rangle = \sigma_{v_1}^2 (\mathbf{I} - \hat{\mathbf{v}}_1 \hat{\mathbf{v}}_1^T) \quad (4.36c)$$

$$\langle \delta \hat{\mathbf{v}}_2 \delta \hat{\mathbf{v}}_2^T \rangle = \sigma_{v_2}^2 (\mathbf{I} - \hat{\mathbf{v}}_2 \hat{\mathbf{v}}_2^T) \quad (4.36d)$$

$$\langle \delta \hat{\mathbf{v}}_1 \delta \hat{\mathbf{w}}_2^T \rangle = \langle \delta \hat{\mathbf{v}}_2 \delta \hat{\mathbf{w}}_1^T \rangle = 0 \quad (4.36e)$$

A Equação (4.36e) resulta da hipótese de que os erros nas observações não são correlacionados com os erros das referências.

Calculando \mathbf{P}_{ref} e \mathbf{P}_{obs}

A matriz de covariância \mathbf{P}_{obs} pode ser escrita em função dos vetores $\hat{\mathbf{s}}_i$.

$$\mathbf{P}_{\text{obs}} = \langle \delta \mathbf{M}_{\text{obs}} \delta \mathbf{M}_{\text{obs}}^T \rangle = \sum_{i=1}^3 \langle \delta \hat{\mathbf{s}}_i \delta \hat{\mathbf{s}}_i^T \rangle$$

$$\mathbf{P}_{\text{obs}} = \langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle + \langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle + \langle \delta \hat{\mathbf{s}}_3 \delta \hat{\mathbf{s}}_3^T \rangle \quad (4.37)$$

Da mesma forma \mathbf{P}_{obs} pode ser definido em função dos vetores $\hat{\mathbf{r}}_i$.

$$\mathbf{P}_{\text{ref}} = \langle \delta \mathbf{M}_{\text{ref}} \delta \mathbf{M}_{\text{ref}}^T \rangle = \sum_{i=1}^3 \langle \delta \hat{\mathbf{r}}_i \delta \hat{\mathbf{r}}_i^T \rangle$$

$$\mathbf{P}_{\text{ref}} = \langle \delta \hat{\mathbf{r}}_1 \delta \hat{\mathbf{r}}_1^T \rangle + \langle \delta \hat{\mathbf{r}}_2 \delta \hat{\mathbf{r}}_2^T \rangle + \langle \delta \hat{\mathbf{r}}_3 \delta \hat{\mathbf{r}}_3^T \rangle \quad (4.38)$$

Agora, as covariâncias descritas nas Equações (4.37) e (4.38) podem ser escritas em termos dos vetores $\delta \hat{\mathbf{v}}_i$ e $\delta \hat{\mathbf{w}}_i$. Fazendo uso da Equação (4.33) como ponto de partida, a covariância de $\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle$ será

$$\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle = \left\langle \frac{1}{|\mathbf{s}_1|} (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) \delta \mathbf{s}_1 \delta \mathbf{s}_1^T (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T)^T \frac{1}{|\mathbf{s}_1|} \right\rangle$$

$$\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle = \frac{1}{|\mathbf{s}_1|} (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) \langle \delta \mathbf{s}_1 \delta \mathbf{s}_1^T \rangle (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T)^T \frac{1}{|\mathbf{s}_1|}$$

$$\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle = (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) \mathbf{I} \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T)$$

$$\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle = \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T)$$

A expressão final torna-se

$$\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle = \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) \quad (4.39)$$

Da mesma maneira, $\langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle$ pode ser calculada a partir da Equação (4.33), então

$$\langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle = \left\langle \frac{1}{|\mathbf{s}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \delta \mathbf{s}_2 \delta \mathbf{s}_2^T (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T)^T \frac{1}{|\mathbf{s}_2|} \right\rangle$$

$$\left\langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \right\rangle = \frac{1}{|\mathbf{s}_2|^2} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \left\langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \right\rangle (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T)^T \quad (4.40)$$

Sabendo que $\mathbf{s}_2 = \hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2$ e $|\mathbf{s}_2| = |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|$, a covariância $\left\langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \right\rangle$ pode ser calculada, mas primeiro, o vetor erro $\delta \mathbf{s}_2$ tem de ser definido como função dos vetores $\delta \hat{\mathbf{w}}_i$, então,

$$\delta \mathbf{s}_2 = \delta(\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2) = \delta \hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2 + \hat{\mathbf{w}}_1 \times \delta \hat{\mathbf{w}}_2 = -\hat{\mathbf{w}}_2 \times \delta \hat{\mathbf{w}}_1 + \hat{\mathbf{w}}_1 \times \delta \hat{\mathbf{w}}_2$$

O produto vetorial pode ser escrito por meio do operador $\boldsymbol{\Omega}(\cdot)$, da seguinte maneira:

$$\mathbf{a} \times \mathbf{b} = \boldsymbol{\Omega}(\mathbf{a})\mathbf{b} = -\boldsymbol{\Omega}(\mathbf{b})\mathbf{a}$$

Assim sendo,

$$\delta \mathbf{s}_2 = -\boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \delta \hat{\mathbf{w}}_1 + \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \delta \hat{\mathbf{w}}_2$$

Então, a covariância de $\delta \mathbf{s}_2$ será

$$\left\langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \right\rangle = \left\langle \left(\boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \delta \hat{\mathbf{w}}_1 + \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \delta \hat{\mathbf{w}}_2 \right) \left(\boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \delta \hat{\mathbf{w}}_1 + \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \delta \hat{\mathbf{w}}_2 \right)^T \right\rangle$$

$$\left\langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \right\rangle = \left\langle \boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \delta \hat{\mathbf{w}}_1 \delta \hat{\mathbf{w}}_1^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T \right\rangle + \left\langle \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \delta \hat{\mathbf{w}}_2 \delta \hat{\mathbf{w}}_2^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_1)^T \right\rangle$$

$$\left\langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \right\rangle = \boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \left\langle \delta \hat{\mathbf{w}}_1 \delta \hat{\mathbf{w}}_1^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T + \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \left\langle \delta \hat{\mathbf{w}}_2 \delta \hat{\mathbf{w}}_2^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{w}}_1)^T$$

$$\left\langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \right\rangle = \boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \sigma_{\hat{\mathbf{w}}_1}^2 (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T) \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T + \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \sigma_{\hat{\mathbf{w}}_2}^2 (\mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{w}}_1)^T$$

$$\langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \rangle = \sigma_{\mathbf{w}_1}^2 \begin{pmatrix} \mathbf{\Omega}(\hat{\mathbf{w}}_2) \mathbf{\Omega}(\hat{\mathbf{w}}_2)^T \\ -\mathbf{\Omega}(\hat{\mathbf{w}}_2) \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T \mathbf{\Omega}(\hat{\mathbf{w}}_2)^T \end{pmatrix} + \sigma_{\mathbf{w}_2}^2 \begin{pmatrix} \mathbf{\Omega}(\hat{\mathbf{w}}_1) \mathbf{\Omega}(\hat{\mathbf{w}}_1)^T \\ -\mathbf{\Omega}(\hat{\mathbf{w}}_1) \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T \mathbf{\Omega}(\hat{\mathbf{w}}_1)^T \end{pmatrix} \quad (4.41)$$

A matriz $\mathbf{\Omega}$ tem a seguinte propriedade

$$\mathbf{\Omega}(\mathbf{a}) \mathbf{\Omega}(\mathbf{b}) = (\mathbf{a}^T \mathbf{b}) \mathbf{I} - \mathbf{a} \mathbf{b}^T \quad (4.42)$$

onde \mathbf{a} e \mathbf{b} são dois vetores quaisquer; assim, aplicando esta propriedade na Equação (4.41), segue que

$$\begin{aligned} \langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \rangle &= \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T |\mathbf{s}_2|^2) + \sigma_{\mathbf{w}_2}^2 (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T |\mathbf{s}_2|^2) \\ \langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \rangle &= \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T) + \sigma_{\mathbf{w}_2}^2 (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T) - (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T |\mathbf{s}_2|^2 \end{aligned} \quad (4.43)$$

agora, é possível continuar a descrever Equação (4.40), então

$$\begin{aligned} \langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle &= \frac{1}{|\mathbf{s}_2|^2} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \langle \delta \mathbf{s}_2 \delta \mathbf{s}_2^T \rangle (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T)^T \\ \langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle &= \frac{1}{|\mathbf{s}_2|^2} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \begin{bmatrix} \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T) + \sigma_{\mathbf{w}_2}^2 (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T) \\ -(\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T |\mathbf{s}_2|^2 \end{bmatrix} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \\ \langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle &= \frac{1}{|\mathbf{s}_2|^2} \begin{bmatrix} \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) + \sigma_{\mathbf{w}_2}^2 (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \\ -(\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T |\mathbf{s}_2|^2 + (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T |\mathbf{s}_2|^2 \end{bmatrix} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \end{aligned}$$

Assim,

$$\langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle = \frac{1}{|\mathbf{s}_2|^2} [\sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) + \sigma_{\mathbf{w}_2}^2 (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T)] \quad (4.44)$$

O resultado apresentado pela Equação (4.44) tem um termo dependente de $\hat{\mathbf{w}}_2$. O vetor $\hat{\mathbf{w}}_2$ não pertence à triade ortogonal formada por $\hat{\mathbf{s}}_1$, $\hat{\mathbf{s}}_2$ e $\hat{\mathbf{s}}_3$, mas pode ser escrita como uma função dos vetores $\hat{\mathbf{s}}_1$ e $\hat{\mathbf{s}}_3$, como

$$\hat{\mathbf{w}}_2 = \hat{\mathbf{s}}_1 (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) - \hat{\mathbf{s}}_3 |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2| \quad (4.45)$$

A Figura 4.3 mostra que $\hat{\mathbf{w}}_2$ está no plano formado por $\hat{\mathbf{s}}_1$ e $\hat{\mathbf{s}}_3$. Isto também pode ser verificado pela Equação (4.5).

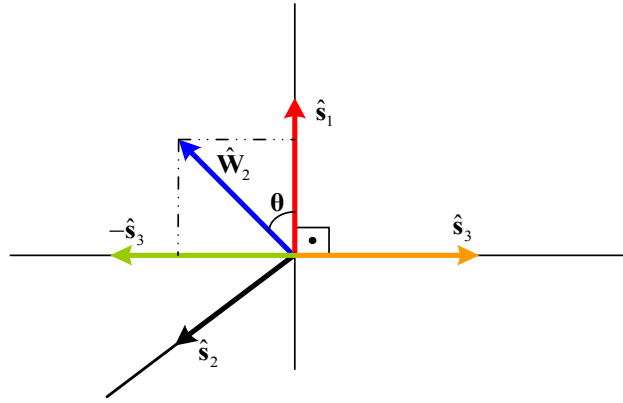


Figura 4.3 – Ilustração mostrando que $\hat{\mathbf{w}}_2$ está no mesmo plano formado por $\hat{\mathbf{s}}_1$ e $\hat{\mathbf{s}}_3$.

As constantes $(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)$ e $|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|$ são, respectivamente, o co-seno e o seno dos ângulos entre os vetores $\hat{\mathbf{w}}_1$ e $\hat{\mathbf{w}}_2$. Então, a seguinte identidade trigonométrica é válida.

$$(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)^2 + |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2 = 1 \quad (4.46)$$

Usando as relações descritas acima pelas Equações (4.45) e (4.46) e sabendo que $|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2| = |\mathbf{s}_2|$, o termo $\hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T$ será

$$\hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T = (\hat{\mathbf{s}}_1 (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) - \hat{\mathbf{s}}_3 |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|) (\hat{\mathbf{s}}_1 (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) - \hat{\mathbf{s}}_3 |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|)^T$$

$$\hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T = \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)^2 + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T |\mathbf{s}_2|^2 - (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T)(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) |\mathbf{s}_2|$$

$$\hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T = \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T (1 - |\mathbf{s}_2|^2) + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T |\mathbf{s}_2|^2 - (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T)(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) |\mathbf{s}_2|$$

$$\hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T = \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T |\mathbf{s}_2|^2 + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T |\mathbf{s}_2|^2 - (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T)(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) |\mathbf{s}_2|$$

assim, a Equação (4.44) pode ser escrita como

$$\langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle = \frac{1}{|\mathbf{s}_2|^2} \left[\sigma_{\mathbf{w}_1}^2 \left(\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T + \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T |\mathbf{s}_2|^2 - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T |\mathbf{s}_2|^2 \right) + \sigma_{\mathbf{w}_2}^2 \left(\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T \right) + (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T)(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) |\mathbf{s}_2| \right] \quad (4.47)$$

Uma simples, mas importante relação, é que a soma dos produtos de vetores unitários que formam uma tríade resulta a matriz Identidade, assim

$$\sum_{i=1}^3 \hat{\mathbf{s}}_i \hat{\mathbf{s}}_i^T = \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T + \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T = \mathbf{I} \quad (4.48)$$

Finalmente, é possível reagrupar os termos na Equação (4.47) e escrevê-la como

$$\langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle = \frac{1}{|\mathbf{s}_2|^2} \left(\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2 \right) \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \sigma_{\mathbf{w}_1}^2 \left(\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T \right) + \sigma_{\mathbf{w}_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\mathbf{s}_2|} \left(\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T \right) \quad (4.49)$$

Dado que $\mathbf{s}_3 = \hat{\mathbf{s}}_3$, já que é resultante do produto vetorial $\hat{\mathbf{s}}_1$ e $\hat{\mathbf{s}}_2$, vetores ortogonais e unitários, então a covariância de $\delta \mathbf{s}_3$ pode ser representada como

$$\langle \delta \hat{\mathbf{s}}_3 \delta \hat{\mathbf{s}}_3^T \rangle = \langle \delta \mathbf{s}_3 \delta \mathbf{s}_3^T \rangle \quad (4.50)$$

assim, o diferencial de \mathbf{s}_3 é dado por

$$\delta \mathbf{s}_3 = \delta(\hat{\mathbf{s}}_1 \times \hat{\mathbf{s}}_2) = \delta \hat{\mathbf{s}}_1 \times \hat{\mathbf{s}}_2 + \hat{\mathbf{s}}_1 \times \delta \hat{\mathbf{s}}_2$$

$$\delta \mathbf{s}_3 = -\hat{\mathbf{s}}_2 \times \delta \hat{\mathbf{s}}_1 + \hat{\mathbf{s}}_1 \times \delta \hat{\mathbf{s}}_2$$

$$\delta \mathbf{s}_3 = -\mathbf{\Omega}(\hat{\mathbf{s}}_2) \delta \hat{\mathbf{s}}_1 + \mathbf{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{s}}_2$$

então, a covariância de $\delta \mathbf{s}_3$ é dada por

$$\begin{aligned} \langle \delta \mathbf{s}_3 \delta \mathbf{s}_3^T \rangle &= \langle (-\mathbf{\Omega}(\hat{\mathbf{s}}_2) \delta \hat{\mathbf{s}}_1 + \mathbf{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{s}}_2) (-\mathbf{\Omega}(\hat{\mathbf{s}}_2) \delta \hat{\mathbf{s}}_1 + \mathbf{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{s}}_2)^T \rangle \\ \langle \delta \mathbf{s}_3 \delta \mathbf{s}_3^T \rangle &= \mathbf{P}_{11} - \mathbf{P}_{21} - \mathbf{P}_{12} + \mathbf{P}_{22} \end{aligned} \quad (4.51)$$

onde as covariâncias cruzadas definidas por \mathbf{P}_{ij} podem ser visualmente definidas das equações acima. As covariâncias \mathbf{P}_{ij} são dadas por

$$\mathbf{P}_{11} = \langle \mathbf{\Omega}(\hat{\mathbf{s}}_2) \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \mathbf{\Omega}(\hat{\mathbf{s}}_2)^T \rangle \quad (4.52a)$$

$$\mathbf{P}_{12} = \langle \mathbf{\Omega}(\hat{\mathbf{s}}_2) \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_2^T \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T \rangle \quad (4.52b)$$

$$\mathbf{P}_{21} = \langle \mathbf{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_1^T \mathbf{\Omega}(\hat{\mathbf{s}}_2)^T \rangle \quad (4.52c)$$

$$\mathbf{P}_{22} = \langle \mathbf{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T \rangle \quad (4.52d)$$

Os cálculos destas covariâncias em termos dos vetores $\hat{\mathbf{s}}_i$ são mostrados abaixo.

\mathbf{P}_{11} pode ser expresso como

$$\mathbf{P}_{11} = \langle \mathbf{\Omega}(\hat{\mathbf{s}}_2) \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \mathbf{\Omega}(\hat{\mathbf{s}}_2)^T \rangle$$

$$\mathbf{P}_{11} = \mathbf{\Omega}(\hat{\mathbf{s}}_2) \langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle \mathbf{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{11} = \mathbf{\Omega}(\hat{\mathbf{s}}_2) \sigma_{w_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) \mathbf{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{11} = \sigma_{\mathbf{w}_1}^2 \left(\mathbf{\Omega}(\hat{\mathbf{s}}_2) \mathbf{\Omega}(\hat{\mathbf{s}}_2)^T - \mathbf{\Omega}(\hat{\mathbf{s}}_2) \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T (\hat{\mathbf{s}}_2)^T \right)$$

$$\mathbf{P}_{11} = \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T)$$

Aplicando a Equação (4.48) \mathbf{P}_{11} se tornará

$$\mathbf{P}_{11} = \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T \quad (4.53)$$

da mesma forma, \mathbf{P}_{22} pode ser calculado como

$$\mathbf{P}_{22} = \left\langle \mathbf{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T \right\rangle$$

$$\mathbf{P}_{22} = \mathbf{\Omega}(\hat{\mathbf{s}}_1) \left\langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \right\rangle \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T$$

substituindo a Equação (4.49) na equação acima,

$$\mathbf{P}_{22} = \mathbf{\Omega}(\hat{\mathbf{s}}_1) \left[\frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \sigma_{\mathbf{w}_1}^2 (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T) + \sigma_{\mathbf{w}_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\mathbf{s}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \right] \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T$$

$$\begin{aligned} \mathbf{P}_{22} = & \frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \mathbf{\Omega}(\hat{\mathbf{s}}_1) \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T + \sigma_{\mathbf{w}_1}^2 \mathbf{\Omega}(\hat{\mathbf{s}}_1) (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T) \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T \\ & + \sigma_{\mathbf{w}_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\mathbf{s}_2|} \mathbf{\Omega}(\hat{\mathbf{s}}_1) (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \mathbf{\Omega}(\hat{\mathbf{s}}_1)^T \end{aligned}$$

e notando que os seguintes produtos cruzados são

$$\mathbf{\Omega}(\hat{\mathbf{s}}_1) \hat{\mathbf{s}}_1 = 0 \quad (4.54a)$$

$$\mathbf{\Omega}(\hat{\mathbf{s}}_1) \hat{\mathbf{s}}_3 = -\hat{\mathbf{s}}_2 \quad (4.54b)$$

a expressão descrevendo \mathbf{P}_{22} se reduz a

$$\mathbf{P}_{22} = \frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T - \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T \quad (4.55)$$

O termo \mathbf{P}_{12} pode ser descrito como,

$$\mathbf{P}_{12} = \left\langle \boldsymbol{\Omega}(\hat{\mathbf{s}}_2) \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_2^T \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T \right\rangle$$

$$\mathbf{P}_{12} = \boldsymbol{\Omega}(\hat{\mathbf{s}}_2) \left\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_2^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

mas $\delta \hat{\mathbf{s}}_2$, dado pela Equação (4.32), pode ser substituído na equação acima como segue

$$\mathbf{P}_{12} = \boldsymbol{\Omega}(\hat{\mathbf{s}}_2) \left\langle \delta \hat{\mathbf{s}}_1 \left[\frac{1}{|\mathbf{s}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \delta \mathbf{s}_2 \right]^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

Utilizando a relação para $\delta \mathbf{s}_2$ definida como

$$\delta \mathbf{s}_2 = \delta(\hat{\mathbf{s}}_1 \times \hat{\mathbf{w}}_2) = \delta \hat{\mathbf{s}}_1 \times \hat{\mathbf{w}}_2 + \hat{\mathbf{s}}_1 \times \delta \hat{\mathbf{w}}_2$$

a equação de \mathbf{P}_{12} se torna

$$\mathbf{P}_{12} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_2)}{|\mathbf{s}_2|} \left\langle \delta \hat{\mathbf{s}}_1 (\delta \hat{\mathbf{s}}_1 \times \hat{\mathbf{w}}_2 + \hat{\mathbf{s}}_1 \times \delta \hat{\mathbf{w}}_2)^T \right\rangle (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

$$\mathbf{P}_{12} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_2)}{|\mathbf{s}_2|} \left\langle \delta \hat{\mathbf{s}}_1 (-\boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \delta \hat{\mathbf{s}}_1 + \boldsymbol{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{w}}_2)^T \right\rangle (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

$$\mathbf{P}_{12} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_2)}{|\mathbf{s}_2|} \left\langle -\delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T + \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{w}}_2^T \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T \right\rangle (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

Pelo fato de, por hipótese, não haver correlação entre os erros em $\hat{\mathbf{w}}_1$ e em $\hat{\mathbf{w}}_2$, o termo $\delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{w}}_2^T$ se anula, então

$$\mathbf{P}_{12} = -\frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_2)}{|\mathbf{s}_2|} \langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

Substituindo a covariância de $\delta \hat{\mathbf{s}}_1$ da Equação (4.39), então

$$\mathbf{P}_{12} = -\frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_2)}{|\mathbf{s}_2|} \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

$$\mathbf{P}_{12} = -\frac{\sigma_{\mathbf{w}_1}^2}{|\mathbf{s}_2|} \left(\boldsymbol{\Omega}(\hat{\mathbf{s}}_2) \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T - \boldsymbol{\Omega}(\hat{\mathbf{s}}_2) \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_2)^T \right) (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

Agora, aplicando novamente a propriedade mostrada pela Equação (4.42) e $\mathbf{a} \times \mathbf{b} = \boldsymbol{\Omega}(\mathbf{a})\mathbf{b} = -\boldsymbol{\Omega}(\mathbf{b})\mathbf{a}$, então

$$\mathbf{P}_{12} = -\frac{\sigma_{\mathbf{w}_1}^2}{|\mathbf{s}_2|} (-\hat{\mathbf{s}}_2 \hat{\mathbf{w}}_2^T - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_2^T |\mathbf{s}_2|) (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

$$\mathbf{P}_{12} = -\frac{\sigma_{\mathbf{w}_1}^2}{|\mathbf{s}_2|} (-\hat{\mathbf{s}}_2 \hat{\mathbf{w}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

Substituindo $\hat{\mathbf{w}}_1$ da Equação (4.45) então

$$\mathbf{P}_{12} = -\frac{\sigma_{\mathbf{w}_1}^2}{|\mathbf{s}_2|} [-\hat{\mathbf{s}}_2 \hat{\mathbf{s}}_1^T (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) + \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_3^T |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|] \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

$$\mathbf{P}_{12} = -\frac{\sigma_{\mathbf{w}_1}^2}{|\mathbf{s}_2|} [-\hat{\mathbf{s}}_2 \hat{\mathbf{s}}_1^T (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) + \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_3^T |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|] \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

$$\mathbf{P}_{12} = -\frac{\sigma_{\mathbf{w}_1}^2}{|\mathbf{s}_2|} |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2| \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_3^T \boldsymbol{\Omega}(\hat{\mathbf{s}}_1)^T$$

e, finalmente, P_{12} pode ser escrito como

$$\mathbf{P}_{12} = \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T \quad (4.56)$$

O termo \mathbf{P}_{21} pode ser calculado da mesma forma descrita para \mathbf{P}_{12}

$$\mathbf{P}_{21} = \left\langle \boldsymbol{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_1^T \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T \right\rangle$$

$$\mathbf{P}_{21} = \boldsymbol{\Omega}(\hat{\mathbf{s}}_1) \left\langle \frac{1}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_1^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{21} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_1)}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \left\langle \delta(\hat{\mathbf{s}}_1 \times \hat{\mathbf{w}}_2) \delta \hat{\mathbf{s}}_1^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{21} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_1)}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \left\langle (\hat{\mathbf{w}}_2 \times \delta \hat{\mathbf{s}}_1 + \hat{\mathbf{s}}_1 \times \delta \hat{\mathbf{w}}_2) \delta \hat{\mathbf{s}}_1^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{21} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_1)}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \left\langle \boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T + \boldsymbol{\Omega}(\hat{\mathbf{s}}_1) \delta \hat{\mathbf{w}}_2 \delta \hat{\mathbf{s}}_1^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{21} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_1)}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \left\langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \right\rangle \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{21} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_1)}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T$$

$$\mathbf{P}_{21} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_1)}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \sigma_{\mathbf{w}_1}^2 [\boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T - \boldsymbol{\Omega}(\hat{\mathbf{w}}_2) \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T \boldsymbol{\Omega}(\hat{\mathbf{s}}_2)^T]$$

$$\mathbf{P}_{21} = \frac{\boldsymbol{\Omega}(\hat{\mathbf{s}}_1)}{|\hat{\mathbf{s}}_2|} (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \sigma_{\mathbf{w}_1}^2 (\hat{\mathbf{w}}_2^T \hat{\mathbf{s}}_2 \mathbf{I} - \hat{\mathbf{w}}_2 \hat{\mathbf{s}}_2^T - |\hat{\mathbf{s}}_2| \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T)$$

$$\mathbf{P}_{21} = \frac{\Omega(\hat{\mathbf{s}}_1)}{|\mathbf{s}_2|} \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) (-\hat{\mathbf{w}}_2 \hat{\mathbf{s}}_2^T - |\mathbf{s}_2| \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_3^T)$$

$$\mathbf{P}_{21} = \frac{\Omega(\hat{\mathbf{s}}_1)}{|\mathbf{s}_2|} \sigma_{\mathbf{w}_1}^2 (-\hat{\mathbf{w}}_2 \hat{\mathbf{s}}_2^T - |\mathbf{s}_2| \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_3^T + |\mathbf{s}_2| \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_3^T)$$

$$\mathbf{P}_{21} = -\frac{\Omega(\hat{\mathbf{s}}_1)}{|\mathbf{s}_2|} \sigma_{\mathbf{w}_1}^2 (\hat{\mathbf{s}}_1 (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) - \hat{\mathbf{s}}_3 |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|) \hat{\mathbf{s}}_2^T$$

$$\mathbf{P}_{21} = -\frac{\Omega(\hat{\mathbf{s}}_1)}{|\mathbf{s}_2|} \sigma_{\mathbf{w}_1}^2 (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_2^T (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_2^T |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|)$$

$$\mathbf{P}_{21} = \frac{\sigma_{\mathbf{w}_1}^2}{|\mathbf{s}_2|} \Omega(\hat{\mathbf{s}}_1) \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_2^T |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|$$

resultando em

$$\mathbf{P}_{21} = -\sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T \quad (4.57)$$

Agora, continuando o cálculo da covariância de $\delta \hat{\mathbf{s}}_3^T$ da Equação (4.51), então

$$\langle \delta \hat{\mathbf{s}}_3 \delta \hat{\mathbf{s}}_3^T \rangle = \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T + \frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T - \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T - \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T + \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T$$

Simplificando esta expressão, a covariância final para $\delta \hat{\mathbf{s}}_3^T$ se torna

$$\langle \delta \hat{\mathbf{s}}_3 \delta \hat{\mathbf{s}}_3^T \rangle = \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T + \frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T - \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T \quad (4.58)$$

Os resultados apresentados nas Equações (4.39), (4.49) e (4.58) são termos parciais que quando substituídos na Equação (4.37), resultam

$$\mathbf{P}_{\text{obs}} = \langle \delta \hat{\mathbf{s}}_1 \delta \hat{\mathbf{s}}_1^T \rangle + \langle \delta \hat{\mathbf{s}}_2 \delta \hat{\mathbf{s}}_2^T \rangle + \langle \delta \hat{\mathbf{s}}_3 \delta \hat{\mathbf{s}}_3^T \rangle$$

$$\begin{aligned} \mathbf{P}_{\text{obs}} = & \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) + \frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \sigma_{\mathbf{w}_1}^2 (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T) + \sigma_{\mathbf{w}_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\mathbf{s}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) + \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T \\ & + \frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T - \sigma_{\mathbf{w}_1}^2 \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T \end{aligned}$$

Agrupando os termos similares e substituindo $|\mathbf{s}_2| = |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|$, então

$$\begin{aligned} \mathbf{P}_{\text{obs}} = & \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \\ & + \frac{1}{|\mathbf{s}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) + \sigma_{\mathbf{w}_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\mathbf{s}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \\ \mathbf{P}_{\text{obs}} = & \sigma_{\mathbf{w}_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \\ & + \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\sigma_{\mathbf{w}_1}^2 + \sigma_{\mathbf{w}_2}^2) (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) + \sigma_{\mathbf{w}_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \end{aligned} \quad (4.59)$$

O cálculo de P_{ref} segue o mesmo procedimento para o cálculo de \mathbf{P}_{obs} . A solução será similar a Equação (4.59), então

$$\mathbf{P}_{\text{ref}} = \langle \delta \mathbf{M}_{\text{ref}} \delta \mathbf{M}_{\text{ref}}^T \rangle = \sum_{i=1}^3 \langle \delta \hat{\mathbf{r}}_i \delta \hat{\mathbf{r}}_i^T \rangle$$

$$\mathbf{P}_{\text{ref}} = \langle \delta \hat{\mathbf{r}}_1 \delta \hat{\mathbf{r}}_1^T \rangle + \langle \delta \hat{\mathbf{r}}_2 \delta \hat{\mathbf{r}}_2^T \rangle + \langle \delta \hat{\mathbf{r}}_3 \delta \hat{\mathbf{r}}_3^T \rangle$$

$$\begin{aligned} \mathbf{P}_{\text{ref}} = & \sigma_{\mathbf{v}_1}^2 (\mathbf{I} - \hat{\mathbf{r}}_3 \hat{\mathbf{r}}_3^T + \hat{\mathbf{r}}_1 \hat{\mathbf{r}}_1^T - \hat{\mathbf{r}}_2 \hat{\mathbf{r}}_2^T) \\ & + \frac{1}{|\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2|^2} (\sigma_{\mathbf{v}_1}^2 + \sigma_{\mathbf{v}_2}^2) (\mathbf{I} - \hat{\mathbf{r}}_1 \hat{\mathbf{r}}_1^T) + \sigma_{\mathbf{v}_1}^2 \frac{(\hat{\mathbf{v}}_1 \bullet \hat{\mathbf{v}}_2)}{|\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2|} (\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_3^T + \hat{\mathbf{r}}_3 \hat{\mathbf{r}}_1^T) \end{aligned} \quad (4.60)$$

Considerando o ângulo de referência o mesmo na observação, a Equação (4.60) torna-se

$$\begin{aligned} \mathbf{P}_{\text{ref}} = & \sigma_{v_1}^2 (\mathbf{I} - \hat{\mathbf{r}}_3 \hat{\mathbf{r}}_3^T + \hat{\mathbf{r}}_1 \hat{\mathbf{r}}_1^T - \hat{\mathbf{r}}_2 \hat{\mathbf{r}}_2^T) \\ & + \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\sigma_{v_1}^2 + \sigma_{v_2}^2) (\mathbf{I} - \hat{\mathbf{r}}_1 \hat{\mathbf{r}}_1^T) + \sigma_{v_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_3^T + \hat{\mathbf{r}}_3 \hat{\mathbf{r}}_1^T) \end{aligned} \quad (4.61)$$

Agora, fazendo os cálculos de $\mathbf{A}\mathbf{P}_{\text{obs}}\mathbf{A}^T$ das Equações (4.60) e considerando que $\hat{\mathbf{s}}_i = \mathbf{A}\hat{\mathbf{r}}_i$ e $\hat{\mathbf{s}}_i^T = \hat{\mathbf{r}}_i^T \mathbf{A}^T$, isso resultará em

$$\begin{aligned} \mathbf{A}\mathbf{P}_{\text{ref}}\mathbf{A}^T = & \sigma_{v_1}^2 (\mathbf{A}\mathbf{A}^T - \mathbf{A}\hat{\mathbf{r}}_3 \hat{\mathbf{r}}_3^T \mathbf{A}^T + \mathbf{A}\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_1^T \mathbf{A}^T - \mathbf{A}\hat{\mathbf{r}}_2 \hat{\mathbf{r}}_2^T \mathbf{A}^T) \\ & + \frac{(\sigma_{v_1}^2 + \sigma_{v_2}^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\mathbf{A}\mathbf{A}^T - \mathbf{A}\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_1^T \mathbf{A}^T) + \sigma_{v_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\mathbf{A}\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_3^T \mathbf{A}^T + \mathbf{A}\hat{\mathbf{r}}_3 \hat{\mathbf{r}}_1^T \mathbf{A}^T) \\ \\ \mathbf{A}\mathbf{P}_{\text{ref}}\mathbf{A}^T = & \sigma_{v_1}^2 (\mathbf{I} - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \\ & + \frac{(\sigma_{v_1}^2 + \sigma_{v_2}^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) + \sigma_{v_1}^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \end{aligned} \quad (4.62)$$

Considerando que $\sigma_1^2 = \sigma_{v_1}^2 + \sigma_{w_1}^2$ e $\sigma_2^2 = \sigma_{v_2}^2 + \sigma_{w_2}^2$ a covariância total é dada por $\mathbf{P} = \mathbf{P}_{\text{obs}} + \mathbf{A}\mathbf{P}_{\text{ref}}\mathbf{A}^T$, o qual resultará em

$$\mathbf{P} = \sigma_1^2 (\mathbf{I} - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) + \frac{(\sigma_1^2 + \sigma_2^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) + \sigma_1^2 \frac{\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \quad (4.63)$$

O traço da matriz \mathbf{P} da Equação (4.63) será

$$\text{tr}(\mathbf{P}) = 2\sigma_1^2 + \frac{2(\sigma_1^2 + \sigma_2^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \quad (4.64)$$

Portanto, partindo da Equação (4.26) a matriz de covariância Cartesiana da atitude será

$$\mathbf{P}_{\theta\theta} = \sigma_1^2 + \frac{(\sigma_1^2 + \sigma_2^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} - \left[\frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\sigma_1^2 + \sigma_2^2) (\mathbf{I} - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T) + \sigma_1^2 (\mathbf{I} - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) \right. \\ \left. + \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \right]$$

$$\mathbf{P}_{\theta\theta} = \sigma_1^2 (\hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T - \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T + \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) + \frac{(\sigma_1^2 + \sigma_2^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T)$$

$$\mathbf{P}_{\theta\theta} = \frac{(\sigma_1^2 + \sigma_2^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T - \sigma_1^2 \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T + \sigma_1^2 (\hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) - \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T)$$

Rearranjando alguns termos, a forma final da matriz de covariância Cartesiana da atitude é dada em termos dos vetores ortogonais de observação e a variância dos vetores de observação.

$$\mathbf{P}_{\theta\theta} = \left[\frac{(\sigma_1^2 + \sigma_2^2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} - \sigma_1^2 \right] \hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T + \sigma_1^2 (\hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T) - \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T) \quad (4.65)$$

A Equação (4.65) pode ser reescrita somente em termos dos vetores de observação $\hat{\mathbf{w}}_1$ e $\hat{\mathbf{w}}_2$. Considerando que

$$\hat{\mathbf{s}}_1 = \hat{\mathbf{w}}_1$$

$$\hat{\mathbf{s}}_2 = \frac{\hat{\mathbf{s}}_1 \times \hat{\mathbf{w}}_2}{|\hat{\mathbf{s}}_1 \times \hat{\mathbf{w}}_2|} = \frac{\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} = \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \hat{\mathbf{w}}_2$$

$$\hat{\mathbf{s}}_3 = \hat{\mathbf{s}}_1 \times \hat{\mathbf{s}}_2 = \hat{\mathbf{w}}_1 \times \frac{\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} = \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \hat{\mathbf{s}}_2$$

Os termos que dependem de $\hat{\mathbf{s}}_i$ na Equação (4.65) podem ser escritos como

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_1^T = \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T \quad (4.66)$$

$$\hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T + \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T = (\mathbf{I} - \hat{\mathbf{s}}_3 \hat{\mathbf{s}}_3^T) = \mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T \quad (4.67)$$

Os termos $\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T$ tornam-se

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T = \hat{\mathbf{w}}_1 (\hat{\mathbf{w}}_1 \times \hat{\mathbf{s}}_2)^T = \hat{\mathbf{w}}_1 (\boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \hat{\mathbf{s}}_2)^T$$

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T = \hat{\mathbf{w}}_1 (\hat{\mathbf{s}}_2 \boldsymbol{\Omega}(\hat{\mathbf{w}}_1))^T = \hat{\mathbf{w}}_1 \hat{\mathbf{s}}_2^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_1)^T$$

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T = \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 (\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2)^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_1)^T$$

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T = \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 (\boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \hat{\mathbf{w}}_2)^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_1)^T$$

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T = -\frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T \boldsymbol{\Omega}(\hat{\mathbf{w}}_1) \boldsymbol{\Omega}(\hat{\mathbf{w}}_1)^T$$

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T = -\frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T (\mathbf{I} - \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T)$$

$$\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3^T = \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T \quad (4.68)$$

e o termo $\hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T$, torna-se

$$\hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T = (\hat{\mathbf{s}}_1 \hat{\mathbf{s}}_3)^T$$

$$\hat{\mathbf{s}}_3 \hat{\mathbf{s}}_1^T = \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_1^T \quad (4.69)$$

substituindo os termos das Equações (4.66) e (4.69) na Equação (4.65), \mathbf{P}_{00} se torna

$$\mathbf{P}_{00} = \frac{\sigma_1^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T + \frac{\sigma_2^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T - \sigma_1^2 \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T + \sigma_1^2 I - \sigma_1^2 \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \left(2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T - \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T \right)$$

$$\mathbf{P}_{00} = \sigma_1^2 \mathbf{I} + \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T + \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_1^T) + \frac{\sigma_1^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T + \frac{\sigma_2^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T - 2\sigma_1^2 \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T \left(\frac{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} + \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \right)$$

$$\mathbf{P}_{00} = \sigma_1^2 \mathbf{I} + \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T + \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_1^T) + \frac{\sigma_1^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T + \frac{\sigma_2^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T$$

$$\mathbf{P}_{00} = \sigma_1^2 \mathbf{I} + \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T + \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_1^T) - \frac{\sigma_1^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T + \frac{\sigma_2^2}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T$$

$$\mathbf{P}_{00} = \sigma_1^2 \mathbf{I} + \sigma_1^2 \frac{(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2)}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} (\hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T + \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_1^T) + (\sigma_2^2 - \sigma_1^2) \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T$$

Rearranjando a Equação acima, finalmente, uma equação analítica para a matriz de covariância Cartesiana da atitude é dada em termos dos vetores de observação $\hat{\mathbf{w}}_1$ e $\hat{\mathbf{w}}_2$ em termos das variâncias $\sigma_1^2 = \sigma_{v_1}^2 + \sigma_{w_1}^2$ e $\sigma_2^2 = \sigma_{v_2}^2 + \sigma_{w_2}^2$. Assim, o resultado apresentado por Shuster e Oh é reescrito abaixo [SHU80].

$$\mathbf{P}_{00} = \sigma_1^2 \mathbf{I} + \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} [\sigma_1^2 (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) (\hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T + \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_1^T) + (\sigma_2^2 - \sigma_1^2) \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T] \quad (4.70)$$

4.2.3 – Codificação do TRIAD

Implementação no Matlab

Embora seja trabalhoso chegar ao resultado da Equação (4.70) é possível notar que como poucas multiplicações e somas e somente uma divisão, é possível chegar a um valor numérico para a covariância. A atitude via TRIAD é simples, bastando a construção de duas matrizes 3x3 e algumas multiplicações e somas. Ambas, matriz de atitude e covariância, podem ser implementadas no Matlab de forma simples como no código a seguir

```
% ===== Função de Determinação de Atitude =====
% =====          Algoritmo TRIAD          =====
% Os parâmetros de Entrada são:
% V1      : vetor de referência V1
% V2      : vetor de referência V2
% W1N     : vetor de observação W1N
% W2N     : vetor de observação W2N
% SigW1   : desvio padrão das observação WN1
% SigW2   : desvio padrão das observação WN2
% =====
% Os parâmetros de Retorno são:
% A        : matriz de atitude
% P        : covariância da atitude
% =====

function [A, P] = Triad(V1,V2,W1N,W2N,SigW1,SigW2)
% --- Obtendo a Triade de Referência
r1 = V1;
r2 = cross(V1,V2); % função do produto vetorial
r2 = r2 / norm(r2);
r3 = cross(r1,r2);

% --- Montando a Matriz de Referência
Mref = [r1 r2 r3];

% --- Obtendo a triade de observação
s1 = W1N;
s2 = cross(W1N,W2N);
s2 = s2 / norm(s2);
s3 = cross(s1,s2);

% --- Montando a Matriz de Observação
Mobs = [s1 s2 s3];
```



```

% --- Atitude pelo TRIAD
A = Mobs * Mref';

% --- Covariância do TRIAD
SigTot1 = SigW1;
SigTot2 = SigW2;
P = SigTot1^2 * eye(3) ...
    + (1/norm(cross(W1N,W2N)))^2 * ...
    (SigTot1^2 * (W1N'*W2N)*(W1N*W2N' + W2N*W1N') ...
    + (SigTot2^2 - SigTot1^2)*W1N*W1N');

```

Os parâmetros de entrada são os vetores de referência v_1 e v_2 (considerados verdadeiros, e portanto sem variância) e vetores de observação w_{1N} e w_{2N} corrompidos com ruído e suas respectivas covariâncias $SigW_1$ e $SigW_2$. A função retorna a atitude e a covariância, ou seja, A e P .

Exemplo numérico

Considere a matriz de atitude A como a seguir

$$A = \begin{bmatrix} 0,333333 & 0,933333 & -0,133333 \\ -0,666666 & 0,333333 & 0,666666 \\ 0,666666 & -0,133333 & 0,733333 \end{bmatrix} \quad (4.71)$$

O quatérnion correspondente a essa matriz é dado por (3.40) a (3.43), então

$$\tilde{q} = \begin{bmatrix} 0,258198 \\ 0,258198 \\ 0,516397 \\ 0,774596 \end{bmatrix} \quad (4.72)$$

e vetores de referência v_1 e v_2 como sendo

$$\mathbf{v}_1 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 0,6 \\ 0,8 \end{bmatrix} \quad (4.73)$$

Rodando os vetores \mathbf{v}_1 e \mathbf{v}_2 pela uso da Equação (4.1), são obtidos

$$\mathbf{w}_1 = \begin{bmatrix} 0,133333 \\ -0,666666 \\ -0,733333 \end{bmatrix} \quad \mathbf{w}_2 = \begin{bmatrix} 0,453333 \\ 0,733333 \\ 0,506666 \end{bmatrix} \quad (4.74)$$

Somando em ambos os vetores acima ruído AWGN de média $\mu = 0$ e variância $\sigma^2 = 0,05$, então os vetores observados ficam

$$\mathbf{w}_1 + N(\mu; \sigma^2) = \begin{bmatrix} 0,133333 \\ -0,666666 \\ -0,733333 \end{bmatrix} + \begin{bmatrix} 0,059458 \\ -0,001881 \\ 0,016364 \end{bmatrix} = \begin{bmatrix} 0,192791 \\ -0,668548 \\ -0,716968 \end{bmatrix} \quad (4.75)$$

$$\mathbf{w}_2 + N(\mu; \sigma^2) = \begin{bmatrix} 0,453333 \\ 0,733333 \\ 0,506666 \end{bmatrix} + \begin{bmatrix} 0,008731 \\ -0,009335 \\ 0,036289 \end{bmatrix} = \begin{bmatrix} 0,462065 \\ 0,723997 \\ 0,542956 \end{bmatrix} \quad (4.76)$$

onde $N(\mu; \sigma^2)$ é um vetor de ruído aleatório em função da média e variância do ruído. Utilizando os vetores de (4.73), (4.75) e (4.76) como parâmetros do algoritmo da função TRIAD, a matriz de atitude recuperada será

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0,242133 & 0,949989 & -0,192791 \\ -0,676278 & 0,307729 & 0,668548 \\ 0,695714 & -0,031496 & 0,716968 \end{bmatrix} \quad (4.77)$$

A matriz de covariância Cartesiana é dada por

$$\mathbf{P}_{00} = \begin{bmatrix} 0,001659 & 0,000798 & 0,001069 \\ 0,000798 & 0,007067 & 0,004161 \\ 0,001069 & 0,004161 & 0,006173 \end{bmatrix} \quad (4.78)$$

A forma de comparação mais comum seria subtrair a matriz de (4.77) da matriz verdadeira de (4.71). Essa é a forma utilizada no cálculo do erro que resulta na matriz de covariância Total, como em (4.20) e (4.21).

Mas, o erro que realmente importa ser analisado é o erro em ângulo. Por isso, multiplicando a matriz verdadeira de (4.71) pela matriz recuperada de (4.77) pode-se encontrar a matriz de rotação que causa o erro, assim

$$\mathbf{A}\tilde{\mathbf{A}}^T = \begin{bmatrix} 0,993073 & -0,027351 & 0,106912 \\ 0,026713 & 0,999127 & 0,003670 \\ -0,106623 & -0,001614 & 0,993786 \end{bmatrix} \quad (4.79)$$

Quanto mais perto da Identidade a matriz acima estiver, menor o erro.

O quatérnion correspondente a matriz $\tilde{\mathbf{A}}$ é dado por

$$\tilde{q} = \begin{bmatrix} 0,233149 \\ 0,295223 \\ 0,540163 \\ 0,752800 \end{bmatrix} \quad (4.80)$$

Também para fins de comparação, em termos dos Ângulos de Euler dados por (3.45–3.47), ter-se-á para (4.71) e (4.77), respectivamente

$$\begin{aligned} \psi &= 70,3^\circ \\ \theta &= 7,7^\circ \\ \varphi &= 42,3^\circ \end{aligned} \quad (4.81)$$

$$\begin{aligned} \tilde{\psi} &= 75,7^\circ \\ \tilde{\theta} &= 11,1^\circ \end{aligned} \quad (4.82)$$

$$\tilde{\Phi} = 43,1^\circ$$

Exemplo gráfico

É evidente, pela Equação (4.70) da covariância, que a amplitude da estatística do erro da matriz da atitude depende da atitude atual. Isso fica claro quando são tomadas medidas em diversos pontos e calculadas a atitude e seu erro, assim como no exemplo ilustrativo a seguir.

Considere uma rotação em que o quatérnion de atitude é dado pela seguinte função

$$\tilde{q} = \cos \frac{\theta(t)}{2} + \hat{\mathbf{u}} \sin \frac{\theta(t)}{2}$$

que resulta em uma seqüência de quatérnions de atitude cujos elementos são dados no gráfico da Figura 4.4. Os quatérnions são lidos com um período de $T_s = 1/50$. Os quatérnions representam uma rotação sobre um eixo fixo. Apenas o ângulo é modificado.

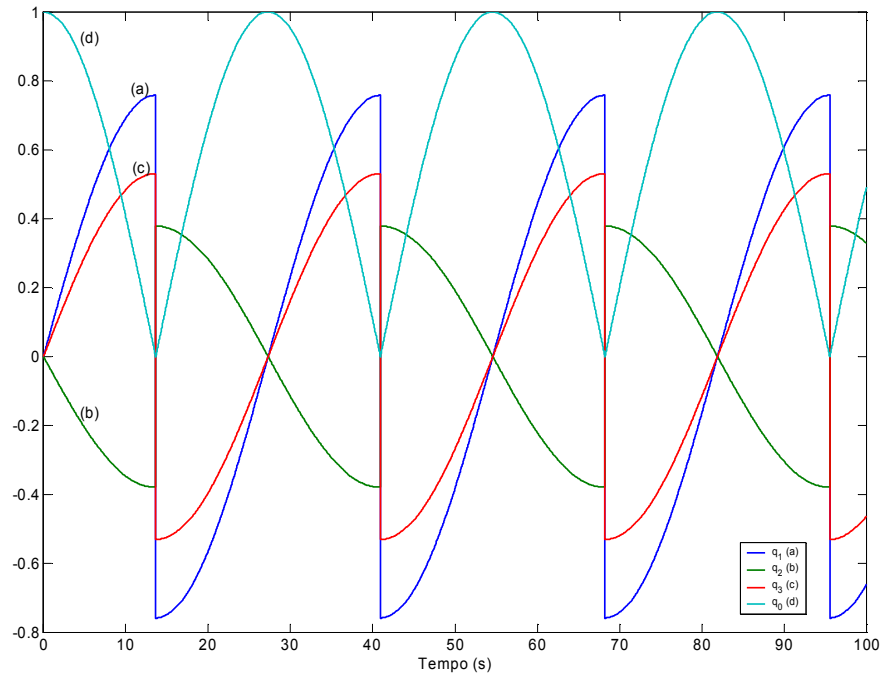


Figura 4.4 – Seqüência de quatérnions de atitude dado por uma rotação em torno de um eixo qualquer.

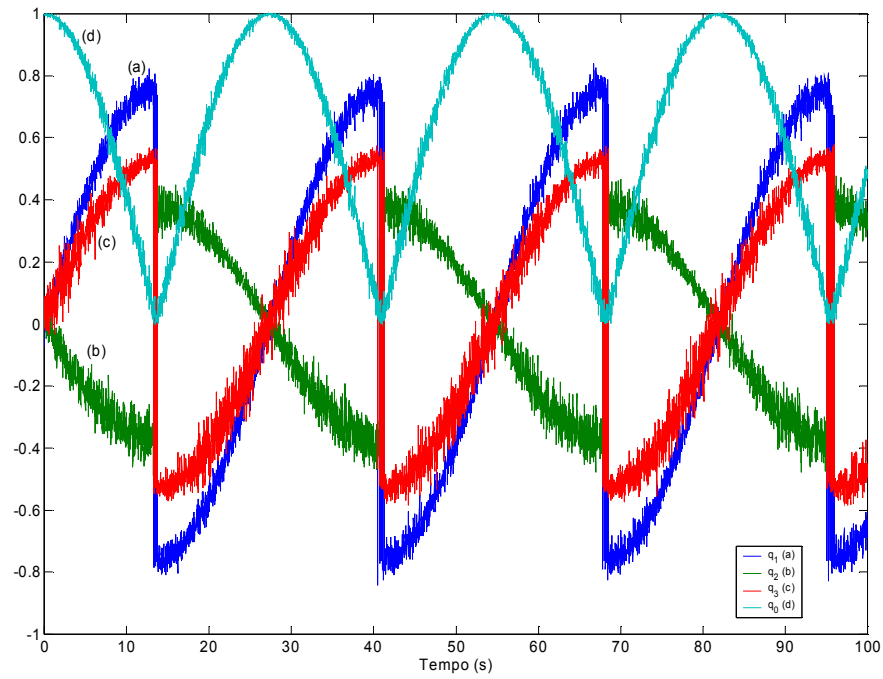


Figura 4.5 – Seqüência de quatérnions de atitude recuperados através das medidas dos vetores observados corrompidos por ruído gaussiano com $N(0; 0,05)$

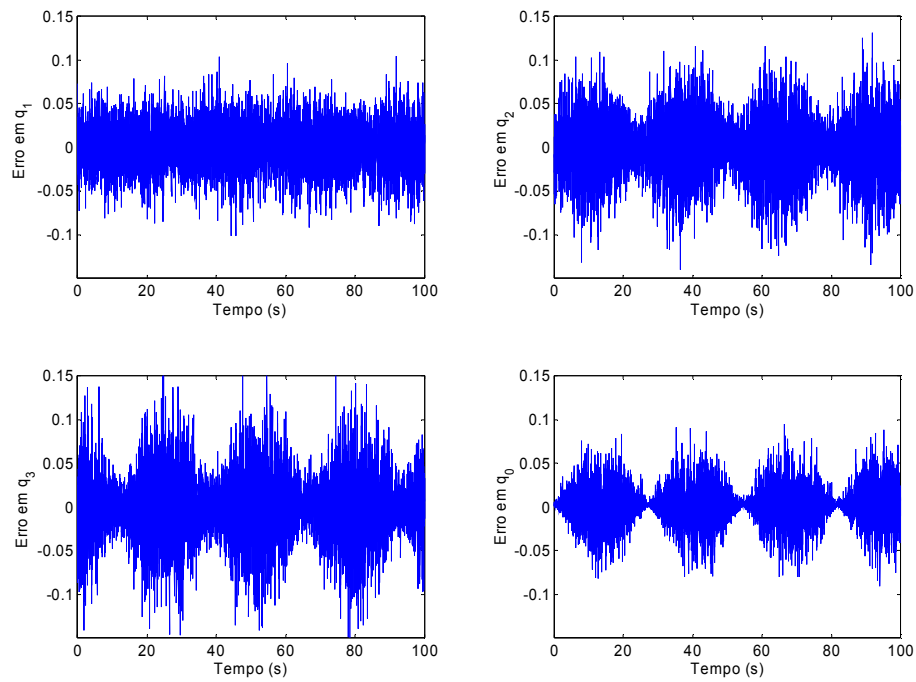


Figura 4.6 – Erro de cada componente do quatérnion na comparação da atitude verdadeira com a atitude recuperada pelo algoritmo TRIAD.

Utilizando a seqüência de quatérnions para gerar uma seqüência de vetores de observação, adicionando ruído aos vetores e em seguida usando o algoritmo TRIAD, a matriz de atitude, e conseqüentemente o quatérnion de atitude corrompidos por ruído, podem ser recuperados. A Figura 4.5 mostra as componentes do quatérnion recuperadas. A Figura 4.6 mostra o erro de cada componente do quatérnion. É interessante notar que o erro não é constante, mas apresenta momentos de máximos e mínimos.

A Figura 4.7 apresenta, pela plotagem sobre a esfera unitária, uma outra forma de visualizar os desvios na direção do quatérnion.

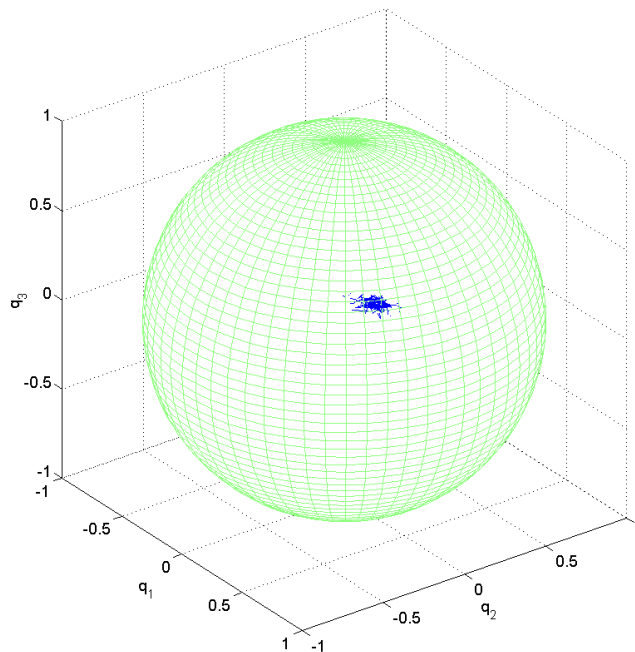


Figura 4.7 – Componente vetorial do quatérnion calculada pelo TRIAD e plotada sobre superfície de uma esfera unitária.

4.2.4 – Sobre o TRIAD

O Algoritmo TRIAD e sua matriz de covariância foram revisados neste texto com o objetivo de compreender a manipulação dos erros dos versores sem fazer uso de funções trigonométricas. A matriz de covariância Cartesiana da atitude é descrita em termos dos vetores de observação e dos vetores de referência. A álgebra de tensores e a estatística envolvida nos cálculos são apresentadas em seus detalhes, provendo uma introdução para o entendimento do problema da determinação da atitude.

O Algoritmo TRIAD e sua matriz de covariância dependem dos valores dos vetores observados e da informação estatística destes vetores. Nenhuma função trigonométrica é utilizada neste algoritmo. Isto torna o algoritmo viável para implementação em sistemas de baixo custo e baixo poder de processamento, permitindo calcular a atitude e a covariância em altas taxas.

Tudo que foi visto até aqui, pelo algoritmo TRIAD, está focado em apenas dois vetores fonte. Os cálculos são assimétricos e não-ótimos pelo fato dos vetores $\hat{\mathbf{v}}_1$ e $\hat{\mathbf{w}}_1$ serem mais requisitados nos cálculos que $\hat{\mathbf{v}}_2$ e $\hat{\mathbf{w}}_2$. Para resolver o problema de forma simétrica, inclusive para casos em que possam existir mais de dois vetores, será explorado o algoritmo conhecido como q-Método (ou Método-q) e sua versão para implementação em hardware, o QUEST.

4.3 – Algoritmo QUEST (QUaternion ESTimator)

4.3.1 – q-Método

Um método elegante de resolver a atitude que minimiza a função de Wahba [WAH66] é o q-Método.

A função perda é escrita como

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i |\hat{\mathbf{w}}_i - \mathbf{A} \hat{\mathbf{v}}_i|^2 \quad (4.83)$$

sendo a_i um conjunto de pesos não negativos, onde

$$\sum_{i=1}^n a_i = 1 \quad (4.84)$$

A Equação (4.83) pode ser reescrita como

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i (\hat{\mathbf{w}}_i - \mathbf{A} \hat{\mathbf{v}}_i)^T (\hat{\mathbf{w}}_i - \mathbf{A} \hat{\mathbf{v}}_i) \quad (4.85)$$

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i (\hat{\mathbf{w}}_i^T \hat{\mathbf{w}}_i - \hat{\mathbf{w}}_i^T \mathbf{A} \hat{\mathbf{v}}_i - \hat{\mathbf{v}}_i^T \mathbf{A}^T \hat{\mathbf{w}}_i + \hat{\mathbf{v}}_i^T \mathbf{A}^T \mathbf{A} \hat{\mathbf{v}}_i)$$

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i (1 - \hat{\mathbf{w}}_i^T \mathbf{A} \hat{\mathbf{v}}_i - \hat{\mathbf{v}}_i^T \mathbf{A}^T \hat{\mathbf{w}}_i + 1)$$

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i (2 - 2 \hat{\mathbf{w}}_i^T \mathbf{A} \hat{\mathbf{v}}_i)$$

$$L(\mathbf{A}) = \sum_{i=1}^n a_i (1 - \hat{\mathbf{w}}_i^T \mathbf{A} \hat{\mathbf{v}}_i) \quad (4.86)$$

já que,

$$\hat{\mathbf{w}}_i^T \mathbf{A} \hat{\mathbf{v}}_i = \hat{\mathbf{v}}_i^T \mathbf{A}^T \hat{\mathbf{w}}_i \quad (4.87)$$

A função de Equação (4.86) pode ser minimizada se o termo negativo for maximizado, ou seja,

$$g(\mathbf{A}) = 1 - L(\mathbf{A}) = \sum_{i=1}^n a_i \hat{\mathbf{w}}_i^T \mathbf{A} \hat{\mathbf{v}}_i \quad (4.88)$$

Ainda, é possível escrever a Equação (4.86) como

$$g(\mathbf{A}) = \text{tr}[\mathbf{A} \mathbf{B}^T] \quad (4.89)$$

onde a matriz \mathbf{B} é definida por

$$\mathbf{B} = \sum_{i=1}^n a_i \hat{\mathbf{w}}_i \hat{\mathbf{v}}_i^T \quad (4.90)$$

A maximização de \mathbf{A} é complicada pelo fato dela possuir 9 elementos. É possível reescrever a função ganho, em função do quatérnion correspondente, na forma

$$\mathbf{A}(\tilde{q}) = (|q_0|^2 - |\mathbf{q}|^2) \mathbf{I}_{3 \times 3} + 2\mathbf{q}\mathbf{q}^T + 2q_0[\tilde{\mathbf{q}}] \quad (4.91)$$

onde,

$$[\tilde{\mathbf{q}}] = \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \quad (4.92)$$

assim, a função ganho pode ser reescrita como

$$g(\tilde{q}) = \text{tr}[\mathbf{A}(\tilde{q})\mathbf{B}^T]$$

$$g(\tilde{q}) = \text{tr}\{(|q_0|^2 - |\mathbf{q}|^2)\mathbf{I} + 2\mathbf{q}\mathbf{q}^T + 2q_0[\tilde{\mathbf{q}}]\mathbf{B}^T\}$$

$$g(\tilde{q}) = \text{tr}\{(|q_0|^2 - |\mathbf{q}|^2)\mathbf{B}^T\} + \text{tr}\{2\mathbf{q}\mathbf{q}^T\mathbf{B}^T\} + 2q_0\text{tr}\{[\tilde{\mathbf{q}}]\mathbf{B}^T\} \quad (4.93)$$

Considerando as quantidades,

$$\sigma = \text{tr}(\mathbf{B}) = \sum_{i=1}^n a_i (\hat{\mathbf{w}}_i \bullet \hat{\mathbf{v}}_i) \quad (4.94)$$

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^T = \sum_{i=1}^n a_i (\hat{\mathbf{w}}_i \hat{\mathbf{v}}_i^T + \hat{\mathbf{v}}_i \hat{\mathbf{w}}_i^T) \quad (4.95)$$

$$\mathbf{z} = \sum_{i=1}^n a_i (\hat{\mathbf{w}}_i \times \hat{\mathbf{v}}_i) = \boldsymbol{\Omega}^{-1}(\mathbf{B} - \mathbf{B}^T) \quad (4.96)$$

a Equação (4.93) pode ser reescrita como

$$g(\tilde{q}) = \tilde{q}^T \mathbf{K} \tilde{q} \quad (4.97)$$

onde a matriz \mathbf{K} é dada por

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \sigma \mathbf{I} & \mathbf{z} \\ \mathbf{z}^T & \sigma \end{bmatrix} \quad (4.98)$$

O quatérnion está sujeito à restrição de ser unitário, então o problema de determinar a atitude ótima está resumido a otimizar a função da Equação (4.97) sujeito à restrição

$$\tilde{\mathbf{q}}^T \tilde{\mathbf{q}} = 1 \quad (4.99)$$

assim é possível escrever a função a ser otimizada como

$$g'(\tilde{\mathbf{q}}) = \tilde{\mathbf{q}}^T \mathbf{K} \tilde{\mathbf{q}} - \lambda \tilde{\mathbf{q}}^T \tilde{\mathbf{q}} \quad (4.100)$$

derivando a equação acima

$$\frac{\partial}{\partial \tilde{\mathbf{q}}^T} g'(\tilde{\mathbf{q}}) = \mathbf{K} \tilde{\mathbf{q}} - \lambda \tilde{\mathbf{q}} = 0$$

tem-se que o quatérnion ótimo deve satisfazer a

$$\mathbf{K} \tilde{\mathbf{q}} = \lambda \tilde{\mathbf{q}} \quad (4.101)$$

Repare que na equação acima, λ é um autovalor e $\tilde{\mathbf{q}}$ é um autovetor de \mathbf{K} . Então, para todos os autovalores

$$g(\mathbf{q}) = \tilde{\mathbf{q}}^T \mathbf{K} \tilde{\mathbf{q}} = \lambda \tilde{\mathbf{q}}^T \tilde{\mathbf{q}} = \lambda \quad (4.102)$$

assim, a Equação (4.100) será maximizada somente se $\tilde{\mathbf{q}}_{\text{opt}}$ for escolhido como o autovetor de \mathbf{K} correspondente ao maior autovalor de \mathbf{K} , ou seja

$$\mathbf{K}\tilde{\mathbf{q}}_{\text{opt}} = \lambda_{\text{max}}\tilde{\mathbf{q}}_{\text{opt}} \quad (4.103)$$

Encontrar no MatLab os autovalores e autovetores de \mathbf{K} em um PC é simples e rápido, mais precisamente, custa 135 flops (operações em ponto flutuante), mas quando essa operação tem de ser escrita em linguagem C (parte em *assembly*) para rodar em um computador de baixo poder de processamento, chegar à solução da Equação (4.103) fica custoso.

A seguir vejamos uma forma de se ‘baratear’ os cálculos de autovalores e autovetores, para isso, utilizando o algoritmo QUEST.

4.3.2 – Quatérnion ótimo via QUEST

Rearranjando a Equação (4.103), é válido para qualquer autovalor que

$$\mathbf{y} = [(\lambda + \sigma)\mathbf{I} - \mathbf{S}]^{-1}\mathbf{z} \quad (4.104)$$

$$\lambda = \sigma + \mathbf{z} \bullet \mathbf{y} \quad (4.105)$$

onde o vetor \mathbf{y} é definido pela razão entre a parte vetorial e escalar do quatérnion, também conhecido como vetor de Gibbs [WER78]

$$\mathbf{y} = \frac{\mathbf{q}}{q_0} = \hat{\mathbf{u}} \tan(\theta/2) \quad (4.106)$$

Em termos do vetor de Gibbs o quatérnion é escrito com

$$\tilde{\mathbf{q}} = \frac{1}{\sqrt{1 + |\mathbf{y}|^2}} \begin{bmatrix} \mathbf{y} \\ 1 \end{bmatrix} \quad (4.107)$$

Quando λ_{max} for usado para calcular \mathbf{y} , então $\tilde{\mathbf{q}}$ dado pela equação acima representará a atitude ótima. Substituindo (4.105) em (4.104) então,

$$\lambda = \sigma + \mathbf{z}^T [(\lambda + \sigma)\mathbf{I} - \mathbf{S}]^{-1} \mathbf{z}^T \quad (4.108)$$

A forma acima é equivalente à Equação característica para os autovalores de \mathbf{K} , a qual é desejável evitar. Entretanto é possível notar das Equações (4.88) e (4.102) que

$$\lambda_{\max} = 1 - L(\mathbf{A}) \approx 1 \quad (4.109)$$

ou seja, o autovalor que minimiza a função custo faz com que o ganho se torne praticamente unitário.

Quando o ângulo de rotação fica próximo a π , a Equação (4.106) tende a valores muito grandes, tornando os cálculos imprecisos e portanto a Equação (4.109) não pode ser utilizada. Para evitar as singularidades, o quatérnion deve ser encontrado sem fazer uso do vetor \mathbf{y} . Uma alternativa é reescrever a Equação (4.108) de maneira diferente. Notando que a equação característica de \mathbf{S} é dada por

$$\det(\mathbf{S} - \xi\mathbf{I}) = 0 \quad (4.110)$$

que abrindo como polinômio fica

$$-\xi^3 + 2\sigma\xi^2 - \kappa\xi + \Delta = 0 \quad (4.111)$$

onde

$$\sigma = \frac{1}{2}\text{tr}(\mathbf{S}) \quad \kappa = \text{tr}(\text{adj}(\mathbf{S})) \quad \Delta = \det(\mathbf{S}) \quad (4.112)$$

Pelo Teorema de Cayley-Hamilton \mathbf{S} satisfaz sua própria equação característica portanto,

$$\mathbf{S}^3 - 2\sigma\mathbf{S}^2 + \kappa\mathbf{S} - \Delta\mathbf{I} = 0 \quad (4.113)$$

a equação pode ser fatorada na forma

$$\gamma^{-1}(\alpha \mathbf{I} + \beta \mathbf{S} + \mathbf{S}^2) = [(\omega + \sigma) \mathbf{I} - \mathbf{S}]^{-1} \quad (4.114)$$

onde

$$\alpha = \omega^2 - \sigma^2 + \kappa \quad \beta = \omega - \sigma \quad \gamma = (\omega + \sigma)\alpha - \Delta \quad (4.115)$$

Quando ω assume o valor de λ_{\max} então

$$\mathbf{y}_{\text{opt}} = \frac{\mathbf{x}}{\gamma} \quad (4.116)$$

assim, a Equação (4.104) pode ser escrita utilizando a igualdade da Equação (4.114)

$$\mathbf{x} = (\alpha \mathbf{I} + \beta \mathbf{S} + \mathbf{S}^2) \mathbf{z} \quad (4.117)$$

das Equações (4.107) e (4.116) pode-se escrever o quatérnion como

$$\tilde{q} = \frac{1}{\sqrt{\gamma^2 + |\mathbf{x}|^2}} \begin{bmatrix} \mathbf{x} \\ \gamma \end{bmatrix} \quad (4.118)$$

Também a partir da forma da Equação (4.114) é possível substituí-la na Equação (4.108) resultando em uma equação característica na forma

$$\lambda^4 - (a + b)\lambda^2 - c\lambda + (ab + c\sigma - d) = 0 \quad (4.119)$$

onde,

$$a = \sigma^2 - \kappa \quad b = \sigma^2 + \mathbf{z}^T \mathbf{z} \quad c = \Delta + \mathbf{z}^T \mathbf{S} \mathbf{z} \quad d = \mathbf{z}^T \mathbf{S}^2 \mathbf{z} \quad (4.120)$$

Como λ_{\max} é sabido ser próximo da unidade, utilizando o método de Newton-Raphson na Equação (4.119), é possível, como uma ou duas iterações, atingir precisão disponível a uma palavra de 64 bits.

Quando existem apenas dois vetores de observação, o autovalor λ_{\max} é algebricamente dado por

$$\lambda_{\max} = \sqrt{a_1^2 + 2a_1a_2 \cos(\theta_v - \theta_w) + a_2^2} \quad (4.121)$$

sendo,

$$\cos(\theta_v - \theta_w) = (\hat{\mathbf{v}}_1 \bullet \hat{\mathbf{v}}_2)(\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) + |\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2| |\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2| \quad (4.122)$$

4.3.3 – A Matriz de covariância do QUEST

O quatérnion ótimo está sujeito a erros porque os vetores fonte, lidos tanto na referência quanto na observação, estão corrompidos por ruído. Assim como feito para o algoritmo TRIAD, o objetivo agora é encontrar uma matriz de covariância da atitude do quatérnion de atitude.

A matriz de covariância do quatérnion é definida pela covariância de um δq que gira o quatérnion verdadeiro para o quatérnion ótimo. Assim,

$$\tilde{q}_{\text{opt}} = \tilde{q}_{\text{true}} \otimes \delta \tilde{q} \quad (4.123)$$

E a média do quatérnion erro será

$$\langle \delta \tilde{q} \rangle = \left\langle \begin{bmatrix} \delta \mathbf{q} \\ \delta \tilde{q} \end{bmatrix} \right\rangle = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (4.124)$$

A covariância da parte vetorial do quatérnion erro é dada por

$$\mathbf{P}_{\mathbf{q}\mathbf{q}} = \langle \delta \mathbf{q} \delta \mathbf{q}^T \rangle \quad (4.125)$$

Da mesma forma que a matriz de covariância Cartesiana da Equação (4.70), então a Equação (4.125) pode ser dada por

$$\mathbf{P}_{\mathbf{q}\mathbf{q}} = \frac{1}{4} \mathbf{P}_{\theta\theta} \quad (4.126)$$

sendo que,

$$\delta \mathbf{q} = \frac{1}{2} [\delta \theta_1 \quad \delta \theta_2 \quad \delta \theta_3]^T \quad (4.127)$$

É possível notar que a covariância do quatérnion ótimo não depende da atitude atual, ou seja, não depende da posição, assim \tilde{q}_{true} pode ser escolhido como o quatérnion identidade. Assim pode se considerar os vetores de referência e os vetores de observação como sendo iguais para calcular a matriz a matriz de covariância, então

$$\bar{\mathbf{w}}_i = \bar{\mathbf{v}}_i \quad (4.128)$$

Se a rotação causada por $\delta \tilde{q}$ é pequena, então, temos que

$$\mathbf{y} = [(\lambda + \sigma) \mathbf{I} - \mathbf{S}]^{-1} \mathbf{z}$$

$$\delta \mathbf{q} = \delta \mathbf{y} = [(1 + 1) \mathbf{I} - \sum_{i=1}^n \bar{\mathbf{w}}_i \bar{\mathbf{w}}_i^T]^{-1} \delta \mathbf{z}$$

$$\delta \mathbf{q} = [2 \mathbf{I} - \sum_{i=1}^n \bar{\mathbf{w}}_i \bar{\mathbf{w}}_i^T]^{-1} \delta \mathbf{z}$$

$$\delta \mathbf{q} = \mathbf{M}^{-1} \delta \mathbf{z}$$

onde,

$$\mathbf{M} = 2 \mathbf{I} - 2 \sum_{i=1}^n \bar{\mathbf{w}}_i \bar{\mathbf{w}}_i^T \quad (4.129)$$

e também,

$$\delta \mathbf{z} = \sum_{i=1}^n a_i \delta (\bar{\mathbf{w}}_i \times \bar{\mathbf{v}}_i) = \sum_{i=1}^n a_i (\delta \bar{\mathbf{w}}_i \times \bar{\mathbf{v}}_i + \bar{\mathbf{w}}_i \times \delta \bar{\mathbf{v}}_i) \quad (4.130)$$

Assim, a Equação (4.125) torna-se

$$\mathbf{P}_{\text{qq}} = \langle \mathbf{M}^{-1} \delta \mathbf{z} (\mathbf{M}^{-1} \delta \mathbf{z})^T \rangle$$

$$\mathbf{P}_{\text{qq}} = \mathbf{M}^{-1} \langle \delta \mathbf{z} \delta \mathbf{z}^T (\mathbf{M}^T)^{-1} \rangle$$

e finalmente,

$$\mathbf{P}_{\text{qq}} = \mathbf{M}^{-1} \langle \delta \mathbf{z} \delta \mathbf{z}^T \rangle \mathbf{M}^{-1} \quad (4.131)$$

Utilizando as covariâncias dos vetores de referência e observação e seguindo procedimentos semelhante os da seção anterior (TRIAD), então

$$\langle \delta \mathbf{z} \delta \mathbf{z}^T \rangle = \left\langle \sum_{i=1}^n a_i^2 (\delta \bar{\mathbf{w}}_i \times \bar{\mathbf{v}}_i + \bar{\mathbf{w}}_i \times \delta \bar{\mathbf{v}}_i) (\delta \bar{\mathbf{w}}_i \times \bar{\mathbf{v}}_i + \bar{\mathbf{w}}_i \times \delta \bar{\mathbf{v}}_i)^T \right\rangle$$

$$\langle \delta \mathbf{z} \delta \mathbf{z}^T \rangle = \sum_{i=1}^n a_i^2 \sigma_i^2 [\mathbf{I} - \bar{\mathbf{w}}_i \bar{\mathbf{w}}_i^T] \quad (4.132)$$

Considerando,

$$\sigma_i^2 = \sigma_{\mathbf{v}_i}^2 + \sigma_{\mathbf{w}_i}^2 \quad (4.133)$$

Note que é desejável minimizar a covariância da Equação (4.131), portanto os pesos a_i^2 devem ser escolhidos para fazê-lo. Uma simples escolha é tomar os pesos que minimizem a função de Wahba (4.2), assim

$$a_i = \frac{\sigma_{\text{tot}}^2}{\sigma_i^2} \quad (4.134)$$

sendo σ_{tot}^2 dado por

$$\frac{1}{\sigma_{\text{tot}}^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \cdots + \frac{1}{\sigma_n^2} \quad (4.135)$$

A partir das Equações (4.129), (4.132) e com os pesos de (4.134), substituído na Equação (4.131), então

$$\mathbf{P}_{\text{qq}} = (2\mathbf{I} - 2\sum_{i=1}^n \bar{\mathbf{w}}_i \bar{\mathbf{w}}_i^T)^{-1} \left(\sum_{i=1}^n a_i^2 \sigma_i^2 [\mathbf{I} - \bar{\mathbf{w}}_i \bar{\mathbf{w}}_i^T] \right) (2\mathbf{I} - 2\sum_{i=1}^n \bar{\mathbf{w}}_i \bar{\mathbf{w}}_i^T)^{-1}$$

finalmente chegando a forma contida em [SHU80],

$$\mathbf{P}_{\text{qq}} = \frac{1}{4} \sigma_{\text{tot}}^2 \left(\mathbf{I} - \sum_{i=1}^n a_i \hat{\mathbf{w}}_i \hat{\mathbf{w}}_i^T \right)^{-1} \quad (4.136)$$

Para o caso de haver apenas dois vetores, então a Equação (4.136) fica

$$\begin{aligned} \mathbf{P}_{\text{qq}} = \frac{1}{4} \{ & \sigma_{\text{tot}}^2 \mathbf{I} + \frac{1}{|\hat{\mathbf{w}}_1 \times \hat{\mathbf{w}}_2|^2} [(\sigma_2^2 - \sigma_{\text{tot}}^2) \hat{\mathbf{w}}_1 \hat{\mathbf{w}}_1^T \\ & (\sigma_1^2 - \sigma_{\text{tot}}^2) \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_2^T + \sigma_{\text{tot}}^2 (\hat{\mathbf{w}}_1 \bullet \hat{\mathbf{w}}_2) (\hat{\mathbf{w}}_1 \hat{\mathbf{w}}_2^T + \hat{\mathbf{w}}_2 \hat{\mathbf{w}}_1^T)] \} \end{aligned} \quad (4.137)$$

4.3.4 – Codificação do QUEST

Codificação no Matlab

O algoritmo QUEST é bem mais custoso que o TRIAD. O calculo do quatérnion ótimo envolve mais contas e talvez não garanta melhoria de desempenho para todos os casos. Vale a pena ser aplicado quando houver mais que duas referências ou quando se busca aumento de performance, mesmo que pequena.

Abaixo é listado o código em Matlab que implementa o QUEST como opção de entrada para dois ou mais vetores de referência e observação mais o vetor de covariância.

```
% ===== Função de Determinação de Atitude =====
% =====                Algoritmo QUEST                =====
% Os parâmetros de Entrada são:
% V      : matriz de vetores de referência V1:V2:...Vn
% W      : matriz de vetores de observação W1:W2:...Wn
% SigVet : vetor de desvio padrão das observações
% =====
% Os parâmetros de Retorno são:
% q      : quatérnion de atitude
% P      : covariância da atitude
% =====

function [q,P] = Quest(V,W,SigVet)

[L,NVet] = size(V);

% === Calculando os Pesos
SigTot = sqrt(1/(sum((1./(SigVet.^2)))));
ai = SigTot^2 ./ (SigVet.^2);

% === B para dois vetores
B = zeros(3,3);
for i = 1:NVet
    B = B + ai(i) * W(:,i) * V(:,i)';
end

Sg = trace(B);
S  = B + B';
Z  = Omega3Inv(B - B');

% === Pela matriz K
% K  = [(S - Sg * eye(3))    Z;
%       Z'                  Sg];
% [D,V] = eig(K);
%
% Obs: Fazendo eig(K), basta escolher a coluna
```

```

% da matriz de autovetores que possua o maior autovalor
% correspondente, como isso é custoso, então vamos para
% um método mais longo, menos custoso.

% === Calculando o Máximo autovalor
cof(1,1) = S(2,2)*S(3,3) - S(3,2)*S(2,3);
cof(2,2) = S(1,1)*S(3,3) - S(3,1)*S(1,3);
cof(3,3) = S(1,1)*S(2,2) - S(2,1)*S(1,2);

Sg = 0.5 * trace(S);
Kapa = trace(cof);
Delta = det(S);

if (NVet == 2)
    csdTh = (V(:,1)'*V(:,2))*(W(:,1)'*W(:,2)) + ...
            norm(cross(V(:,1),V(:,2))) * norm(cross(W(:,1),W(:,2)));
    Lb = sqrt(ai(1)^2 + 2*ai(1)*ai(2)*csdTh + ai(2)^2);
else
    a = Sg^2 - Kapa;
    b = Sg^2 + Z'*Z;
    c = Delta + Z'*S*Z;
    d = Z'*(S*S)*Z;

    % === Encontrando a raiz do Polinômio

    % --- Teste:
    %  $L^4 - (a+b)L^2 - cL + (a*b + c*Sg-d) = 0$ 
    % --- Pra conferir usou-se:
    % p = [1 0 -(a+b) -c (a*b + c*Sg-d)];
    % Rz = roots(p);

    % === Newton-Raphson
    Lb = 1;
    for i = 1:3;
        Funp = Lb^4 - (a+b)*Lb^2 - c*Lb + (a*b + c*Sg-d);
        Fundp = 4*Lb^3 - 2*(a+b)*Lb - c;
        Lb = Lb - Funp/Fundp;
    end
end

% === Calculando o Quatérnion Ótimo

```

```

Omg = Lb;
Alpha = Omg^2 - Sg^2 + Kapa;
Beta = Omg - Sg;
Gamma =(Omg + Sg)*Alpha - Delta;

X      = (Alpha*eye(3) + Beta*S + S*S)*Z;
q      = 1/sqrt(Gamma^2 + norm(X)^2) * [X;Gamma];

% === Covariância do QUEST p/ dois vetores
if (NVet == 2)
    P = 1/4 * (SigTot^2 * eye(3) ...
        + 1/norm(cross(W(:,1),W(:,2)))^2 * ((SigVet(2)^2 -
        SigTot^2)*W(:,1)*W(:,1)' ...
        + (SigVet(1)^2 - SigTot^2)*W(:,2)*W(:,2)' ...
        + SigTot^2*(W(:,1)'*W(:,2))*(W(:,1)*W(:,2)' + W(:,2)*W(:,1)')));

% === Covariância do QUEST p/ mais de dois vetores
else
    SomaCov = zeros(3,3);
    for i = 1:NVet
        SomaCov = SomaCov + ai(i) * W(:,i) * W(:,i)';
    end
    P = 1/4 * (SigTot^2) * inv(eye(3,3) - SomaCov);
end

```

A função QUEST acima leva em conta que se forem apenas dois vetores, então algumas contas podem ser simplificadas. O maior autovalor passa a ser calculado diretamente pela Equação (4.109) e a matriz de covariância, embora use mais linhas de código, fica mais simples porque economiza o cálculo da inversa de uma matriz 3x3 no cálculo da covariância.

Exemplo Numérico

Para efeito de comparação, utilizando os mesmo vetores de (4.73), (4.75) e (4.76) como parâmetros do algoritmo da função QUEST, são obtidos como quaternion estimado e matriz de covariância do quaternion erro, para $n = 2$, respectivamente

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0,214079 & 0,961436 & -0,172658 \\ -0,677364 & 0,273469 & 0,682928 \\ 0,703808 & -0,029247 & 0,709787 \end{bmatrix} \quad (4.138)$$

$$\mathbf{P}_{\theta\theta} = \begin{bmatrix} 0,000395 & 0,000254 & 0,000218 \\ 0,000254 & 0,001613 & 0,001176 \\ 0,000218 & 0,001176 & 0,001379 \end{bmatrix} \quad (4.139)$$

Um parâmetro “visível” de comparação é a matriz erro angular entre (4.71) e (4.138) que é dada pelo produto

$$\mathbf{A}\tilde{\mathbf{A}}^T = \begin{bmatrix} 0,991720 & -0,061607 & 0,112666 \\ 0,062653 & 0,998017 & 0,005763 \\ -0,112088 & -0,012775 & 0,993615 \end{bmatrix} \quad (4.140)$$

e o quatérnion correspondente a matriz $\tilde{\mathbf{A}}$ é dado por

$$\tilde{q} = \begin{bmatrix} 0,240220 \\ 0,295636 \\ 0,552774 \\ 0,741170 \end{bmatrix} \quad (4.141)$$

Também para fins de comparação, em termos dos Ângulos de Euler dados por (3.46–3.48), ter-se-á para (4.71) e (4.138), respectivamente

$$\begin{aligned} \psi &= 70,3^\circ \\ \theta &= 7,7^\circ \\ \phi &= 42,3^\circ \end{aligned} \quad (4.142)$$

$$\begin{aligned} \tilde{\psi} &= 77,4^\circ \\ \tilde{\theta} &= 9,9^\circ \\ \tilde{\phi} &= 43,9^\circ \end{aligned} \quad (4.143)$$

Exemplo gráfico

Dada uma rotação em torno de um eixo, essa atitude gera determinada projeção sobre N vetores de referência e a partir destes vetores é possível através do algoritmo QUEST encontrar a atitude ótima levando em conta o nível de ruído presente na leitura de cada vetor. A Figura 4.8 mostra as componentes do quatérnion de atitude. A Figura 4.9 mostra o quatérnion recuperado. A Figura 4.10 apresenta o erro de cada componente do quatérnion como sendo a diferença entre a componente verdadeira e a estimada. A Figura 4.11 traz a parte direcional do quatérnion plotada sobre a superfície de uma esfera unitária.

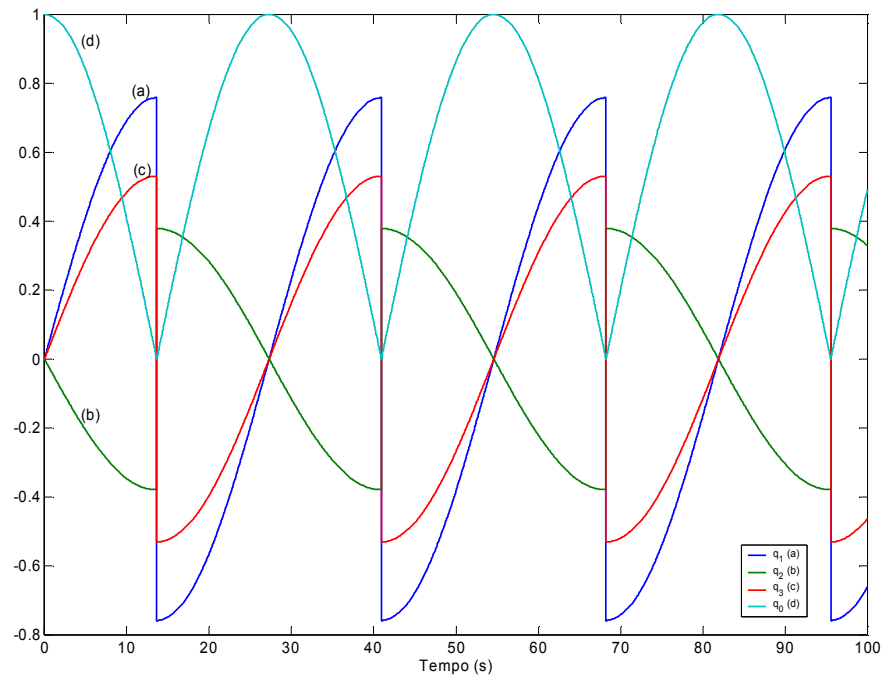


Figura 4.8 – Componentes do quatérnion de atitude lido sequencialmente a partir da rotação sobre um eixo qualquer.

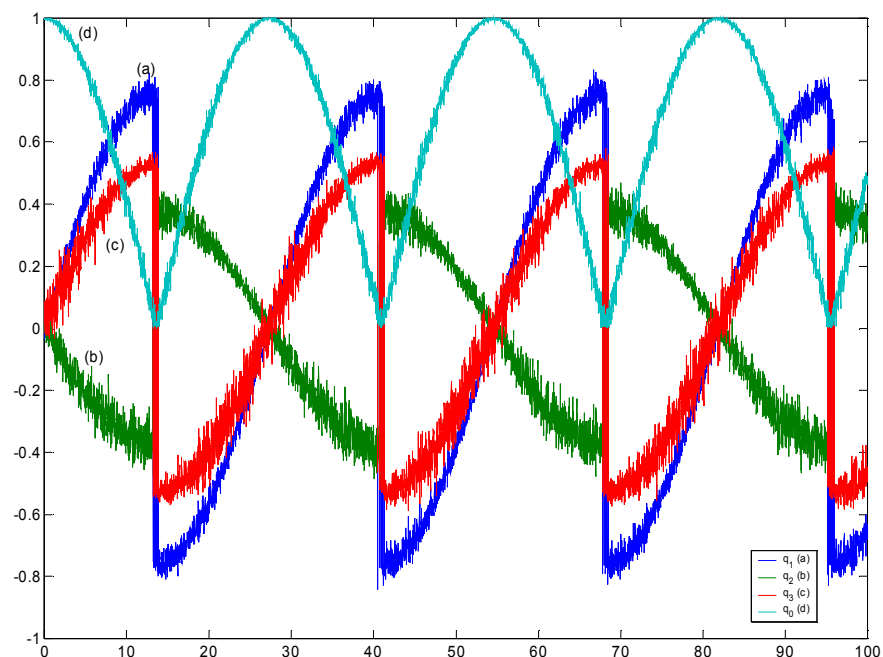


Figura 4.9 – Componentes do quatérnion recuperado a partir das leituras dos vetores observados corrompidos por ruído gaussiano.

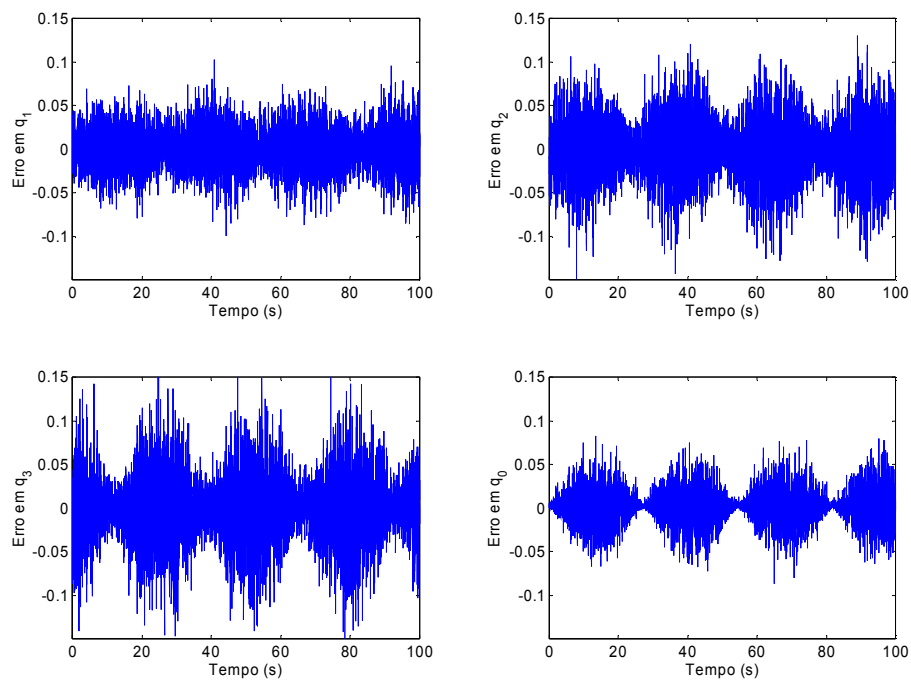


Figura 4.10 – Erro de cada componentes do quatérnion na subtração da atitude verdadeira da recuperada pelo algoritmo QUEST.

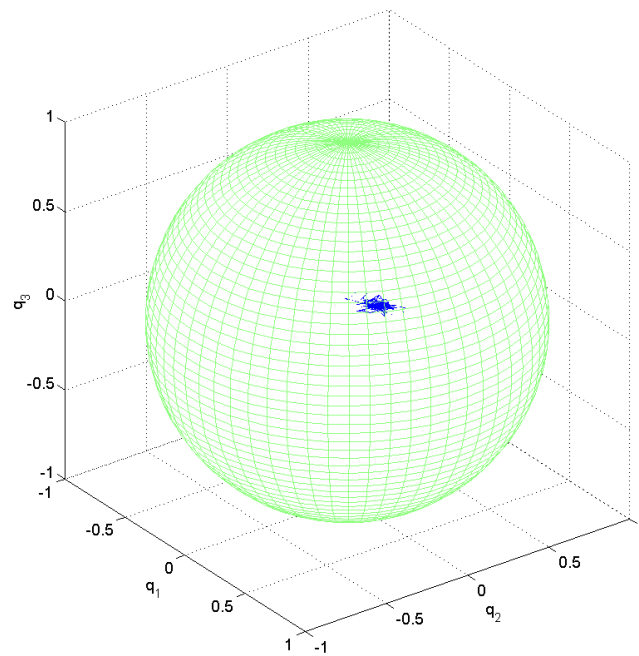


Figura 4.11 – Componente vetorial do quatérnion estimado pelo algoritmo QUEST, plotada sobre superfície de uma esfera unitária.

Embora as Figuras de 4.10 e 4.6 sejam semelhantes, pode-se observar pelo erro quadrático médio acumulado, visto na Figura 4.12, que o algoritmo QUEST tem um nível de ruído menor.

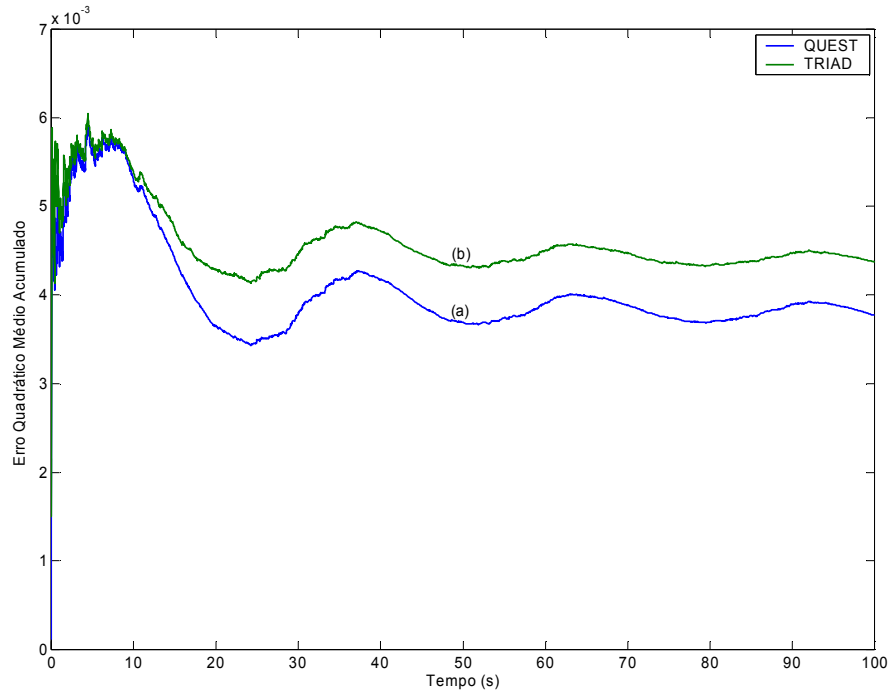


Figura 4.12 – Erro quadrático médio acumulado. Em (a) erro médio acumulado do algoritmo QUEST. Em (b) o erro quadrático médio acumulado pelo algoritmo TRIAD.

4.4 – Outros algoritmos

Existem diversos algoritmos de determinação de atitude além do TRIAD, do q-método e do QUEST. Dentre os mais famosos estão o método SVD (*Singular Value Decomposition* – Decomposição por valor singular), o FOAM (*Fast Optimal Attitude Matrix* – Rápida matriz de atitude ótima) e também ESOQ, ESOQ1 e ESOQ2 (*Estimator of the Optimal Quatérnion* – Estimador do quatérnion ótimo). Uma comparação destes algoritmos pode ser encontrada em [MAR00].

4.5 – Considerações e Conclusões sobre o TRIAD e o QUEST

O algoritmo TRIAD é o mais simples algoritmo para determinar atitude. Na forma que foi tratado aqui exige o uso de funções trigonométricas, o que o torna rápido. Pode ser utilizado somente quando houver dois vetores de medidas. O TRIAD não é um algoritmo simétrico, já que um vetor é utilizado duas vezes enquanto o outro é utilizado

apenas uma vez, na formação das matrizes de observação e de referência. Depois de encontrada a matriz de atitude, essa matriz tem que ser convertida para o quatérnion correspondente pois o algoritmo não calcula o quatérnion diretamente.

O algoritmo QUEST é uma implementação menos custosa do método-q para encontrar o maior autovalor e o autovetor correspondente que minimiza a função modificada de Wahba. O algoritmo QUEST retorna diretamente o quatérnion ótimo, livre de singularidades. O algoritmo leva em conta o ruído de cada sensor na determinação da atitude, o que não é feito pelo TRIAD. Por outro lado, quando houver apenas dois vetores de referência, o QUEST deve ser usado somente em casos onde um aumento de precisão for essencial, embora acarrete aumento da carga computacional. Quando houver mais de dois vetores de medida, então o uso do QUEST (ou outro algoritmo) passa a ser uma boa opção.

A matriz de covariância de ambos algoritmos são muito próximas conceitualmente quando existem apenas dois vetores, por esse motivo possível considerar a existência da relação entre as covariâncias [SHU80],

$$\mathbf{P}_{qq}^{\text{TRIAD}} = \mathbf{P}_{qq}^{\text{QUEST}} + \Delta \mathbf{P}_{qq} \quad (4.144)$$

onde,

$$\mathbf{P}_{qq}^{\text{TRIAD}} = \frac{1}{4} \mathbf{P}_{\theta\theta} \quad (4.145)$$

Assim, a partir da Equação 4.145 e 4.137, chega-se à diferença

$$\Delta \mathbf{P}_{qq} = \frac{1}{4} (\sigma_1^2 - \sigma_{\text{tot}}^2) \hat{\mathbf{s}}_2 \hat{\mathbf{s}}_2^T \quad (4.146)$$

A variância total σ_{tot}^2 pode ser compreendida com sendo o ‘paralelo’ das variâncias dos sensores. Por isso, quando os dois sensores possuem qualidade de medida muito diferentes ($\sigma_1^2 \ll \sigma_2^2$) então a variância total fica praticamente igual a variância do melhor sensor (σ_1^2 é a variância do melhor sensor)

$$\frac{1}{\sigma_{\text{tot}}^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \approx \frac{1}{\sigma_1^2} \longrightarrow \sigma_{\text{tot}}^2 \approx \sigma_1^2 \quad (4.147)$$

logo a diferença em (4.146) tende a zero e portanto a covariância da atitude pelo QUEST fica igual a do TRIAD indicando que não vale a pena utilizar o QUEST, pois o resultado será praticamente o mesmo. Mas, quando as covariâncias forem parecidas ou iguais (este é caso do exemplo numérico) atinge-se o pior caso para a covariância do TRIAD em relação ao QUEST, pois a variância total tende à metade da variância do melhor sensor em (4.146). Neste caso pode-se obter melhor exatidão utilizando o QUEST, isso pode ser observado na Figura 4.13.

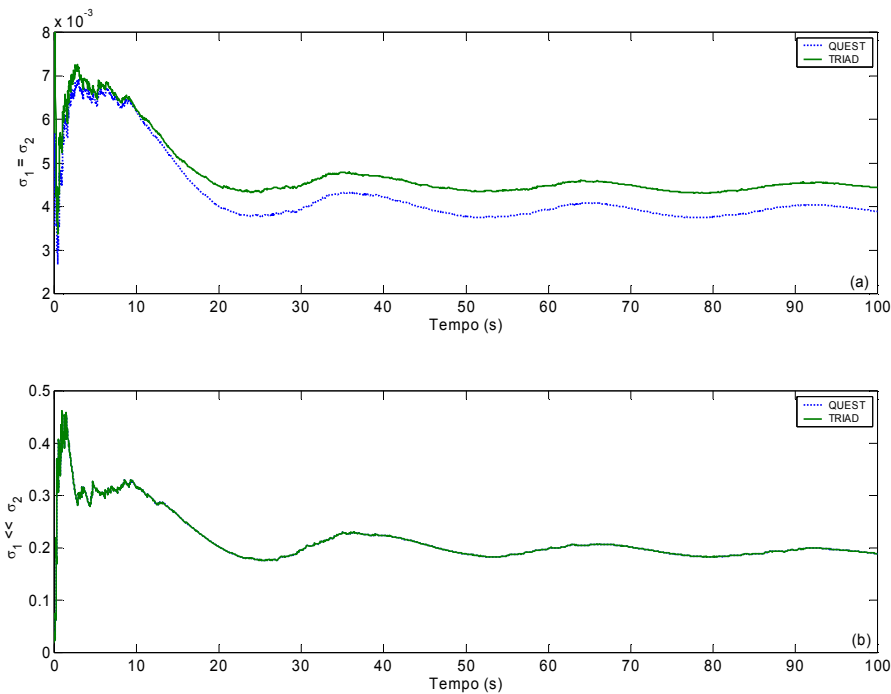


Figura 4.13 – Erro quadrático médio acumulado para diferentes níveis de ruído. Em (a) tem-se o mesmo gráfico de 4.12, quando os níveis de ruído de cada sensor são idênticos; esse é o melhor caso de desempenho do QUEST. Em (b) considerou-se um sensor sendo muito superior ao outro sensor (10 vezes menos ruidoso); neste caso, o desempenho de ambos os algoritmos foi praticamente o mesmo.

Capítulo 5 – Atitude por Intermédio de um Filtro de Kalman: Teoria e Simulação

5.1 – Introdução

Utilizando somente o campo geomagnético e o campo gravitacional como fontes de vetores de referências na confecção de hardware determinador de atitude, a atitude está sujeita a alguns tipos de erros que não são apenas por causa do ruído ou deriva dos sensores.

O sensor de campo magnético terá sua medida equivocada se algum material metálico chegar a uma certa distância capaz provocar um desvio na medida da direção do campo. Da mesma forma, o acelerômetro realiza uma medida de reação à gravidade e por esse motivo quando o mesmo se encontrar em movimento acelerado, haverá um erro significativo na medida da direção da gravidade. Por exemplo, quando um sensor de aceleração está em “queda livre” o sinal de saída do sensor indica 0 g.

Por outro lado, um girômetro, ou seja, um sensor que mede velocidade angular, não tem sua leitura gravemente afetada pelo ambiente a sua volta, ou por acelerações, mas em contrapartida, a informação lida pelo giro precisa ser integrada constantemente para gerar a informação de atitude desejada. Portanto, como o *bias* do giro é um sinal que apresenta deriva, a integração diverge rapidamente e a informação da posição angular se perde.

Perante tudo isso que foi dito, é possível concluir que nem os sensores de atitude citados, nem o giro, sozinhos, representam uma ótima solução para determinação da atitude. Contudo, uma combinação de ambos pode criar um sistema de determinação de atitude viável.

Uma idéia seria utilizar os sensores de referência para identificar o *bias* do girômetro e quando for identificada alguma falha no sistema de referência, como por

exemplo, se o módulo da aceleração lida difere de 1 g, então o giro pode ser usado até que se restabeleçam as condições de funcionamento dos sensores de referência.

Uma outra forma, é escrever a equação de estados do sistema considerando os giros como sendo os fornecedores da dinâmica do sistema enquanto os sensores de atitude fornecem as leituras do estado atual, e então, utilizando a informação estatística dos ruídos do sistema, encontrar os estados do sistema utilizando um processo conhecido como Filtragem de Kalman.

Na seção a seguir será apresentada uma visão geral sobre o Filtro de Kalman, e na próxima, o equacionamento para o caso em questão. No tópico 5.10 são utilizadas informações dos modelos dos ruídos dos sensores a serem empregados na implementação, para realizar uma simulação do Filtro de Kalman no Matlab. A descrição do equacionamento é basicamente um ajuste ao texto de [LEF82]. Na simulação também serão utilizados os algoritmos TRIAD e QUEST descritos no Capítulo 4. Por fim, será apresentada uma visualização gráfica para a apresentação do quatérnion de atitude que facilita identificar visualmente problemas de convergência.

5.2 – Filtro de Kalman

5.2.1 – Breve Histórico

O Filtro de Kalman [KAL60] foi originalmente desenvolvido como uma ferramenta na teoria da estimação linear. Foi prontamente aplicada pela NASA no programa Apollo [SMI61] no problema de estimação orbital. Embora as aplicações de hoje para o Filtro de Kalman sejam as mais diversas imagináveis, ele tem sido extensivamente utilizado na área de navegação como em: determinação e controle de atitude, estimação de órbita, trajetória e posição dentre outras. Sua utilização pode ser em filtragem, suavização e predição.

O tratamento teórico padrão dado a Filtragem de Kalman pode ser encontrado em [SOR66], [JAZ70] e [GEL74]; sobre teoria de filtragem adaptativa em [HAY96], e sobre implementação do Filtro de Kalman e semelhantes em [ZAR02].

5.2.2 – Equações Gerais

Para aplicar a teoria da Filtragem de Kalman, o modelo do sistema deve estar representado na forma de um conjunto de equações diferenciais. Essas equações são descritas de forma matricial ou em forma de *equação de estado*:

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t), t) + g(\mathbf{x}(t), t)\mathbf{w}(t) \quad (5.1)$$

onde $\mathbf{x}(t)$ é o vetor de estados, um vetor coluna $n \times 1$, f é uma função vetorial diferenciável e não necessariamente linear; g é uma matriz $n \times r$ com elementos contínuos no tempo; $\mathbf{w}(t)$, o ruído do processo, é um vetor de dimensão $r \times 1$ com um ruído gaussiano branco com média e funções de covariância dadas por

$$E[\mathbf{w}(t)] = \langle \mathbf{w}(t) \rangle = 0 \quad (5.2)$$

$$E[\mathbf{w}(t)\mathbf{w}^T(t')] = \langle \mathbf{w}(t)\mathbf{w}^T(t') \rangle = \mathbf{Q}(t)\delta(t-t') \quad (5.3)$$

onde ambos $E[\cdot]$ e $\langle \cdot \rangle$ denotam o operador esperança. $\mathbf{Q}(t)$ é a matriz de covariância do ruído do sistema. Os estados iniciais são dados por

$$E[\mathbf{x}(t_0)] \equiv \hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0 \quad (5.4)$$

$$E\{[\mathbf{x}(t_0) - \mathbf{x}_0][\mathbf{x}(t_0) - \mathbf{x}_0]^T\} = \mathbf{P}(t_0) = \mathbf{P}_0 \quad (5.5)$$

A. Predição

Dadas as condições iniciais do vetor de estado e da matriz de covariância do estado, a mínima variância estimada do vetor de estado em um tempo futuro t é dado, na ausência de medidas, pela esperança condicional

$$\hat{\mathbf{x}}(t) = E[\mathbf{x}(t) | \hat{\mathbf{x}}(t_0) = \mathbf{x}_0] \quad (5.6)$$

A estimativa predita satisfaz a equação diferencial

$$\frac{d}{dt} \hat{\mathbf{x}}(t) = E\{f(\hat{\mathbf{x}}(t), t)\} \equiv \hat{f}(\mathbf{x}(t), t) \quad (5.7)$$

Aproximadamente pode ser escrito como

$$\frac{d}{dt} \hat{\mathbf{x}}(t) = f(\hat{\mathbf{x}}(t), t) \quad (5.8)$$

A equação anterior pode ser integrada formalmente resultando em

$$\hat{\mathbf{x}}(t) = \Phi(t, \hat{\mathbf{x}}(t_0), t_0) \quad (5.9)$$

onde a função Φ é a função de transição integral no intervalo $t_0 \rightarrow t$. O vetor erro de estado e a matriz de covariância deste erro são dados respectivamente por

$$\Delta \mathbf{x} = \mathbf{x}(t) - \hat{\mathbf{x}}(t) \quad (5.10)$$

$$\mathbf{P}(t) = E[\Delta \mathbf{x}(t) \Delta \mathbf{x}^T(t)] \quad (5.11)$$

Desconsiderando termos de ordem superior a primeira ordem no vetor de erro de estado e do vetor ruído do processo, o vetor de erro de estado satisfaz à seguinte equação

$$\frac{d}{dt} \Delta \mathbf{x}(t) = \mathbf{F}(t) \Delta \mathbf{x}(t) + \mathbf{G}(t) \mathbf{w}(t) \quad (5.12)$$

onde,

$$\mathbf{F}(t) = \left. \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, t) \right|_{\hat{\mathbf{x}}(t)} \quad (5.13)$$

$$\mathbf{G}(t) = g(\hat{\mathbf{x}}(t), t) \quad (5.14)$$

Integrando a Equação (5.12):

$$\Delta \mathbf{x}(t) = \Phi(t, t_0) \Delta \mathbf{x}(t_0) + \int_{t_0}^t \Phi(t, t') G(t') \mathbf{w}(t') dt' \quad (5.15)$$

onde $\Phi(t, t_0)$ é a matriz de transição, a qual satisfaz

$$\frac{\partial}{\partial t} \Phi(t, t_0) = \mathbf{F}(t) \Phi(t, t_0) \quad (5.16)$$

$$\Phi(t_0, t_0) = \mathbf{I} \quad (5.17)$$

Note que para sistemas não-lineares a função $\Phi(t, t_0)$ também dependerá de $\hat{\mathbf{x}}(t_0)$, que será suprimido nesta notação.

A matriz de covariância do erro dos estados predita satisfaz a equação de Riccati [ZAR02]

$$\frac{d}{dt} \mathbf{P}(t) = \mathbf{F}(t) \mathbf{P}(t) + \mathbf{P}(t) \mathbf{F}^T(t) + \mathbf{G}(t) \mathbf{Q}(t) \mathbf{G}^T(t) \quad (5.18)$$

a qual pode ser integrada resultando em

$$\mathbf{P}(t) = \Phi(t, t_0) \mathbf{P}(t_0) \Phi^T(t, t_0) + \int_{t_0}^t \Phi(t, t') \mathbf{G}(t') \mathbf{Q}(t') \mathbf{G}^T(t') \Phi^T(t, t') dt' \quad (5.19)$$

Utilizando uma notação mais compacta, $\hat{\mathbf{x}}_k(-)$ e $\mathbf{P}_k(-)$ denotam os valores preditos do vetor de estado e da matriz de covariância no tempo t_k , e $\hat{\mathbf{x}}_k(+)$ e $\mathbf{P}_k(+)$ denotam as mesmas quantidades no instante imediatamente seguinte a medida no tempo t_k . Assim, seguindo esta notação

$$\hat{\mathbf{x}}_{k+1}(-) = \Phi(t_{k+1}, \hat{\mathbf{x}}_k(+), t_k) \quad (5.20)$$

$$\mathbf{P}_{k+1}(-) = \Phi_k \mathbf{P}_k(+) \Phi_k^T + \mathbf{N}_k \quad (5.21)$$

A matriz \mathbf{N}_k é a matriz correlação do erro da transição de estado.

B. Filtragem (propagação)

O vetor de medidas no tempo t_k é relatado ao vetor de estado por

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (5.22)$$

onde \mathbf{v}_k , o ruído gaussiano branco aditivo, de média e variância, respectivamente

$$E[\mathbf{v}_k] = 0 \quad (5.23)$$

$$E[\mathbf{v}_k \mathbf{v}_{k'}^T] = \mathbf{R}_k \delta_{kk'} \quad (5.24)$$

A mínima variância estimada de \mathbf{x}_k imediatamente seguinte a medida é dada por

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k [\mathbf{z}_k - h(\hat{\mathbf{x}}_k(-))] \quad (5.25)$$

onde o ganho de Kalman é dado por

$$\mathbf{K}_k = \mathbf{P}_k(-) \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k(-) \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (5.26)$$

e a matriz de medida é dada por

$$\mathbf{H}_k = \left. \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k(-)} \quad (5.27)$$

A matriz de covariância dos erros dos estados imediatamente após a medida é dada por

$$\mathbf{P}_k(+) = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k(-) \quad (5.28)$$

$$\mathbf{P}_k(+) = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k(-) (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (5.29)$$

sendo a última, uma forma numericamente mais estável que a primeira.

A seguir serão apresentadas as equações do movimento de rotação para que possam ser ajustadas as equações do Filtro vistas acima.

5.3 – Equações de Rotação

Considere um quatérnion unitário na forma vetorial, como definido na Equação (3.9), repetido abaixo para maior clareza.

$$\tilde{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q_0 \end{bmatrix} \quad (5.30)$$

onde

$$\mathbf{q} = \hat{\mathbf{u}} \sin(\theta/2) \quad q_0 = \cos(\theta/2) \quad (5.31)$$

O vetor unitário $\hat{\mathbf{u}}$ é o eixo de rotação e θ o ângulo de rotação. O quatérnion unitário tem três graus de liberdade e está restrito a condição

$$\tilde{\mathbf{q}}^T \tilde{\mathbf{q}} = 1 \quad (5.32)$$

A matriz de atitude pode ser formada a partir dos quatro elementos que compõem o quatérnion. Na forma vetorial é possível escrever essa relação sendo

$$\mathbf{A}(\tilde{\mathbf{q}}) = (|q_0|^2 - |\mathbf{q}|^2) \mathbf{I} + 2\mathbf{q}\mathbf{q}^T + 2q_0[\mathbf{q}] \quad (5.33)$$

onde

$$-\mathbf{\Omega}(\mathbf{q}) = [\mathbf{\tilde{q}}] = \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \quad (5.34)$$

A composição de rotações pode ser feita pelo produto entre quatérnions ou pelo produto de matrizes de rotação. Assim,

$$\mathbf{A}(\tilde{q}')\mathbf{A}(\tilde{q}) = \mathbf{A}(\tilde{q}' \otimes \tilde{q}) \quad (5.35)$$

O vetor de velocidade angular $\mathbf{\omega}$ é definido pela derivada da matriz de atitude, assim,

$$\frac{d}{dt}\mathbf{A}(t) = [\mathbf{\tilde{\omega}}(t)]\mathbf{A}(t) = -\mathbf{\Omega}(\mathbf{\omega}(t))\mathbf{A}(t) \quad (5.36)$$

Já, a derivada do quatérnion é dada por

$$\frac{d}{dt}\tilde{q}(t) = \frac{1}{2}\mathbf{\Omega}_4(\mathbf{\omega}(t))\tilde{q}(t) \quad (5.37)$$

onde

$$\mathbf{\Omega}_4(\mathbf{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \quad (5.38)$$

Definindo um quatérnion puro para representar a velocidade angular como

$$\tilde{\omega} = \begin{bmatrix} \mathbf{\omega} \\ 0 \end{bmatrix} \quad (5.39)$$

então, é possível reescrever a Equação (5.37) em termos do produto entre quatérnions

$$\frac{d}{dt}\tilde{q}(t) = \frac{1}{2}\tilde{\omega}(t) \otimes \tilde{q}(t) \quad (5.40)$$

Para resolver a função da Equação (5.37), considere que o vetor velocidade angular ω é constante sobre o intervalo de interesse ou que se o vetor ângulo de rotação definido a seguir é pequeno, então

$$\Delta\theta = \int_t^{t+\Delta t} \omega(t) dt \quad (5.41)$$

assim, a solução fica

$$\tilde{q}(t + \Delta t) = \mathbf{M}(\Delta\theta)\tilde{q}(t) \quad (5.42)$$

onde

$$\mathbf{M}(\Delta\theta) = \cos(|\Delta\theta|/2)\mathbf{I}_{4 \times 4} + \sin(|\Delta\theta|/2) \frac{\mathbf{\Omega}_4(\Delta\theta)}{|\Delta\theta|} \quad (5.43)$$

Uma passagem semelhante a essa foi feita na Equação (3.29).

5.4 – Modelos para os Sensores

O determinador de atitude a ser implementado conta com uma tríade de giros para medir as velocidades angulares, uma tríade de acelerômetros para ler a aceleração gravitacional e uma tríade de sensores magnéticos para ler o campo geomagnético.

5.4.1 – Modelos para girômetros

O modelo adotado será o mesmo descrito em [LEF82]. O modelo é simples mas compatível com a realidade deste tipo de sensor. Considere a velocidade angular como sendo

$$\omega = \mathbf{u} - \mathbf{b} - \eta_l \quad (5.44)$$

onde ω é a velocidade angular, \mathbf{u} é a saída dada pelo giro, \mathbf{b} é taxa de deriva do *bias* do giro e $\boldsymbol{\eta}_1$ é ruído da taxa de deriva. O ruído branco $\boldsymbol{\eta}_1$ é considerado Gaussiano, então

$$E[\boldsymbol{\eta}_1(t)] = 0 \quad (5.45)$$

$$E[\boldsymbol{\eta}_1(t)\boldsymbol{\eta}_1^T(t')] = \mathbf{Q}_1(t)\delta(t-t') \quad (5.46)$$

A taxa de deriva do *bias* não é estática, e sim um ruído branco com

$$\frac{d}{dt}\mathbf{b} = \boldsymbol{\eta}_2 \quad (5.47)$$

onde

$$E[\boldsymbol{\eta}_2(t)] = 0 \quad (5.48)$$

$$E[\boldsymbol{\eta}_2(t)\boldsymbol{\eta}_2^T(t')] = \mathbf{Q}_2(t)\delta(t-t') \quad (5.49)$$

Os dois processos estocásticos contidos na saída do giro são descorrelacionados, tal que

$$E[\boldsymbol{\eta}_1(t)\boldsymbol{\eta}_2^T(t')] = 0 \quad (5.50)$$

Uma equação alternativa para deriva do *bias* do giro é dada por

$$\frac{d}{dt}\mathbf{b} = -\frac{\mathbf{b}}{\tau} + \boldsymbol{\eta}_2 \quad (5.51)$$

Ou seja, o ruído está exponencialmente correlacionado ao *bias*, como no modelo de [HAM49]. Considerar $\tau \rightarrow \infty$, como na equação (5.47), é adequado para a maioria das aplicações.

5.4.2 – Sensores de atitude

Os sensores de atitude geralmente tem como informação de saída valores que dependem apenas da atitude. Sensores de estrela, sensores solar ou magnetômetros exemplificam bem o que é um sensor de atitude. Assim, podemos considerar que

$$\mathbf{w} = \mathbf{A}(\tilde{\mathbf{q}})\mathbf{v} \quad (5.52)$$

onde \mathbf{v} é a leitura do sensor na referência, e \mathbf{w} a leitura do sensor quando rodado pela matriz de atitude.

Um sensor de inclinação, que faz uso de acelerômetros tem sua saída dependente da velocidade angular, que por sua vez tem sua medida dependente do giro e seus erros, além, é claro, de depender da aceleração linear. Não será considerado na equação de estados a seguir, que existam todas essas dependências. Simplificações serão feitas, como considerar que as velocidades angulares são pequenas demais para afetar a medida da aceleração e também que não existem acelerações lineares significativas.

5.5 – A Equação de Estado

O vetor de estados é composto pelas componentes do quatérnion e pelos *bias* dos girômetros. Assim, o estado é um vetor dado por

$$\mathbf{x}(t) = \begin{bmatrix} \tilde{\mathbf{q}}(t) \\ \mathbf{b}(t) \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (5.53)$$

As derivadas do vetor de estados, dadas em duas partes, como já visto anteriormente nas Equações (5.37) e (5.47), são

$$\frac{d}{dt}\tilde{q}(t) = \frac{1}{2}\mathbf{\Omega}_4(\mathbf{u}(t) - \mathbf{b}(t) - \mathbf{\eta}_1(t))\tilde{q}(t) \quad (5.54)$$

$$\frac{d}{dt}\mathbf{b}(t) = \mathbf{\eta}_2(t) \quad (5.55)$$

O operador de distribuição matricial $\mathbf{\Omega}_4$ é uma função linear e homogênea. Pode-se definir o novo operador $\mathbf{\Xi}$ pela relação

$$\mathbf{\Omega}_4(\mathbf{b})\tilde{q} = \mathbf{\Xi}(\tilde{q})\mathbf{b} \quad (5.56)$$

onde $\mathbf{\Xi}$ é o operador que realiza a seguinte distribuição

$$\mathbf{\Xi}(q) = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (5.57)$$

Pela definição de $\mathbf{\Xi}(\tilde{q})$, é possível aplicá-la à Equação (5.54) para reescrevê-la como

$$\begin{aligned} \frac{d}{dt}\tilde{q}(t) &= \frac{1}{2}\mathbf{\Omega}_4(\mathbf{u}(t) - \mathbf{b}(t) - \mathbf{\eta}_1(t))\tilde{q}(t) \\ \frac{d}{dt}\tilde{q}(t) &= \frac{1}{2}\mathbf{\Omega}_4(\mathbf{u}(t) - \mathbf{b}(t))\tilde{q}(t) - \frac{1}{2}\mathbf{\Omega}_4(\mathbf{\eta}_1(t))\tilde{q}(t) \\ \frac{d}{dt}\tilde{q}(t) &= \frac{1}{2}\mathbf{\Omega}_4(\mathbf{u}(t) - \mathbf{b}(t))\tilde{q}(t) - \frac{1}{2}\mathbf{\Xi}(\tilde{q}(t))\mathbf{\eta}_1(t) \end{aligned} \quad (5.58)$$

5.5.1 – Predição

A predição do vetor de estados como definido na Seção 5.2.2, é dada por

$$\frac{d}{dt} \hat{q}(t) = \frac{1}{2} \mathbf{\Omega}_4(\hat{\omega}) \hat{q}(t) = \frac{1}{2} \mathbf{\Omega}_4(\mathbf{u} - \hat{\mathbf{b}}) \hat{q}(t) \quad (5.59)$$

$$\frac{d}{dt} \hat{\mathbf{b}} = \mathbf{0}_{3 \times 1} \quad (5.60)$$

Durante todo o intervalo de predição, \mathbf{b} é constante. Assim, ω dependerá somente do vetor de medidas $\mathbf{u}(t)$ e do valor inicial do vetor de estado. Portanto a Equação (5.59) pode ser integrada diretamente resultando em

$$\hat{q}(t) = \mathbf{\Theta}(t, t_k) \hat{q}(t_k) \quad (5.61)$$

com

$$\frac{\partial}{\partial t} \mathbf{\Theta}(t, t_k) = \frac{1}{2} \mathbf{\Omega}_4(\hat{\omega}(t)) \mathbf{\Theta}(t, t_k) \quad (5.62)$$

$$\mathbf{\Theta}(t, t_k) = \mathbf{I}_{4 \times 4} \quad (5.63)$$

Quando a direção de $\omega(t)$ é constante durante o intervalo de integração, ou se o deslocamento dos eixos é pequeno, então a função $\mathbf{\Theta}(t, t_k)$ pode ser aproximada pela função da Equação (5.43)

$$\mathbf{\Theta}(t, t_k) = \mathbf{M}(\Delta\theta) = \cos(|\Delta\theta|/2) \mathbf{I}_{4 \times 4} + \sin(|\Delta\theta|/2) \frac{\mathbf{\Omega}_4(\Delta\theta)}{|\Delta\theta|} \quad (5.64)$$

5.6 – A Matriz de Covariância dos Erros do Estado

A matriz de covariância é fundamental nos cálculos da Filtragem de Kalman. Pela Equações (5.18) e (5.19) é possível observar que a matriz de covariância depende de diversas funções, como $\mathbf{F}(t)$, $\mathbf{G}(t)$, $\mathbf{\Phi}(t)$ e $\mathbf{Q}(t)$. A seguir a construção matricial de cada uma delas

5.6.1 – Predição

O vetor de estado do erro satisfaz a equação diferencial (5.12), repetida a seguir

$$\frac{d}{dt} \Delta \mathbf{x}(t) = \mathbf{F}(t) \Delta \mathbf{x}(t) + \mathbf{G}(t) \mathbf{w}(t) \quad (5.65)$$

sendo

$$\mathbf{F}(t) = \begin{bmatrix} \frac{1}{2} \mathbf{\Omega}_4(\hat{\mathbf{w}}) & -\frac{1}{2} \mathbf{\Xi}(\hat{\mathbf{q}}) \\ \mathbf{0}_{3 \times 4} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (5.66)$$

$$\mathbf{G}(t) = \begin{bmatrix} -\frac{1}{2} \mathbf{\Xi}(\hat{\mathbf{q}}) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.67)$$

$$\mathbf{w}(t) = \begin{bmatrix} \boldsymbol{\eta}_1(t) \\ \boldsymbol{\eta}_2(t) \end{bmatrix} \quad (5.68)$$

e a partir de $\mathbf{w}(t)$, pela Equação (5.3), tem-se

$$\mathbf{Q}(t) = \begin{bmatrix} \mathbf{Q}_1(t) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q}_2(t) \end{bmatrix} \quad (5.69)$$

A matriz de transição é dada por

$$\boldsymbol{\Phi}(t) = \begin{bmatrix} \boldsymbol{\Theta}(t, t_0) & \boldsymbol{\Psi}(t, t_0) \\ \mathbf{0}_{3 \times 4} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.70)$$

onde $\boldsymbol{\Theta}(t, t_0)$ é dado pela Equação (5.64) e aproximado pela Equação (5.43) e o outro termo

$$\frac{\partial}{\partial t} \boldsymbol{\Psi}(t, t_0) = \frac{1}{2} \mathbf{\Omega}_4(\hat{\mathbf{w}}(t, t_0)) \boldsymbol{\Psi}(t, t_0) - \frac{1}{2} \mathbf{\Xi}(\hat{\mathbf{q}}(t)) \quad (5.71)$$

com

$$\Psi(t, t_0) = \mathbf{0}_{4 \times 3}$$

A integração da Equação (5.71) resulta em

$$\Psi(t, t_0) = -\frac{1}{2} \int_{t_0}^t \Theta(t, t') \Xi(\hat{q}(t')) dt' \quad (5.72)$$

5.6.2 – Filtragem

Assumindo que as medidas da atitude são escalares e dependem somente da atitude em si, e não do vetor de *bias*, então, a matriz de medidas terá que satisfazer

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (5.73)$$

Se considerarmos o caso onde a atitude é medida através de um sistema que não depende de giros, resultando em uma medida que é o próprio quatérnion, a matriz \mathbf{H} será dada por

$$\mathbf{H} = [\mathbf{I}_{4 \times 4} \quad \mathbf{0}_{4 \times 3}] \quad (5.74)$$

5.7 – Propriedades da Matriz $\Xi(\tilde{q})$

Antes de continuar com o cálculo da matriz de covariância reduzida dos erros do estado, vejamos algumas propriedades essenciais da matriz $\Xi(\tilde{q})$.

A matriz $\Xi(\tilde{q})$ é formada pelo elemento quatérnion distribuído como mostrado na Equação (5.57), repetido abaixo

$$\Xi(\tilde{q}) = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}_{4 \times 3} \quad (5.76)$$

A matriz é ortogonal ao quatérnion que a forma, assim

$$\Xi^T(\tilde{q})\tilde{q} = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}^T \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0}_{3 \times 1} \quad (5.77)$$

O produto $\Xi(\tilde{q})\Xi^T(\tilde{q})$ resulta

$$\Xi(\tilde{q})\Xi^T(\tilde{q}) = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} q_0 & q_3 & -q_2 & -q_1 \\ -q_3 & q_0 & q_1 & -q_2 \\ q_2 & -q_1 & q_0 & -q_3 \end{bmatrix}$$

$$\Xi(\tilde{q})\Xi^T(\tilde{q}) = \begin{bmatrix} q_0^2 + q_3^2 + q_2^2 & -q_2q_1 & -q_3q_1 & -q_0q_1 \\ -q_1q_2 & q_3^2 + q_0^2 + q_1^2 & -q_3q_2 & -q_0q_2 \\ -q_3q_1 & -q_3q_2 & q_2^2 + q_1^2 + q_0^2 & -q_0q_3 \\ -q_0q_1 & -q_0q_2 & -q_0q_3 & q_1^2 + q_2^2 + q_3^2 \end{bmatrix}$$

Como o quatérnion está sujeito a restrição de ser unitário, como mostrado Equação (5.32), então

$$\Xi(\tilde{q})\Xi^T(\tilde{q}) = \begin{bmatrix} 1-q_1^2 & -q_2q_1 & -q_3q_1 & -q_0q_1 \\ -q_1q_2 & 1-q_2^2 & -q_3q_2 & -q_0q_2 \\ -q_3q_1 & -q_3q_2 & 1-q_3^2 & -q_0q_3 \\ -q_0q_1 & -q_0q_2 & -q_0q_3 & 1-q_0^2 \end{bmatrix}$$

$$\Xi(\tilde{q})\Xi^T(\tilde{q}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} q_1^2 & q_2q_1 & q_3q_1 & q_0q_1 \\ q_1q_2 & q_2^2 & q_3q_2 & q_0q_2 \\ q_3q_1 & q_3q_2 & q_3^2 & q_0q_3 \\ q_0q_1 & q_0q_2 & q_0q_3 & q_0^2 \end{bmatrix}$$

finalmente,

$$\Xi(\tilde{q})\Xi^T(\tilde{q}) = \mathbf{I}_{4 \times 4} - \tilde{q}\tilde{q}^T \quad (5.78)$$

agora, o produto $\Xi^T(\tilde{q})\Xi(\tilde{q})$ é dado por

$$\Xi^T(\tilde{q})\Xi(\tilde{q}) = \begin{bmatrix} q_0 & q_3 & -q_2 & -q_1 \\ -q_3 & q_0 & q_1 & -q_2 \\ q_2 & -q_1 & q_0 & -q_3 \end{bmatrix} \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}$$

resultando claramente em

$$\Xi^T(\tilde{q})\Xi(\tilde{q}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_{3 \times 3} \quad (5.79)$$

É possível escrever as equações da derivada do quatérnion e sua integral respectivamente como,

$$\frac{d}{dt}\tilde{q}(t) = \frac{1}{2}\mathbf{\Omega}_4(\boldsymbol{\omega})\tilde{q}(t) \quad (5.80)$$

$$\tilde{q}(t) = \mathbf{\Theta}(t, t_0)\tilde{q}(t_0) \quad (5.81)$$

onde $\mathbf{\Omega}_4$ e $\mathbf{\Theta}(t, t_0)$ são dadas pelas Equações (5.38) e (5.43) respectivamente.

Existe uma clara relação entre as matrizes $\mathbf{\Xi}(\tilde{q})$ e $\mathbf{\Omega}_4$, dada por

$$\mathbf{\Xi}(\tilde{q})\mathbf{c} = \mathbf{\Omega}_4(\mathbf{c})\tilde{q} \quad (5.82)$$

onde \mathbf{c} é um vetor qualquer. Assim diferenciando ambos os lados da Equação acima em relação ao tempo, tem-se

$$\dot{\mathbf{\Xi}}(\tilde{q})\mathbf{c} + \mathbf{\Xi}(\tilde{q})\dot{\mathbf{c}} = \mathbf{\Omega}_4(\dot{\mathbf{c}})\tilde{q} + \mathbf{\Omega}_4(\mathbf{c})\dot{\tilde{q}} \quad (5.83)$$

lembrando que a função $\mathbf{\Omega}_4$ é linear e homogênea.

A derivada de \mathbf{c} pode ser escrita em função da velocidade de variação $\boldsymbol{\omega}$, assim

$$\dot{\mathbf{c}} = [\boldsymbol{\omega}] \mathbf{c} = -\boldsymbol{\omega} \times \mathbf{c} \quad (5.84)$$

Substituindo a relação as Equações (5.37), (5.84) e (5.82), na Equação (5.83), tem-se

$$\dot{\mathbf{\Xi}}(\tilde{q})\mathbf{c} + \mathbf{\Xi}(\tilde{q})[\boldsymbol{\omega}] \mathbf{c} = -\mathbf{\Omega}_4(\boldsymbol{\omega} \times \mathbf{c})\tilde{q} + \frac{1}{2}\mathbf{\Omega}_4(\mathbf{c})\mathbf{\Omega}_4(\boldsymbol{\omega})\tilde{q} \quad (5.85)$$

Uma propriedade da função $\mathbf{\Omega}_4$ facilmente demonstrável que

$$\mathbf{\Omega}_4(\boldsymbol{\omega} \times \mathbf{c}) = -\frac{1}{2}[\mathbf{\Omega}_4(\boldsymbol{\omega})\mathbf{\Omega}_4(\mathbf{c}) - \mathbf{\Omega}_4(\mathbf{c})\mathbf{\Omega}_4(\boldsymbol{\omega})] \quad (5.86)$$

Substituindo a relação acima na Equação (5.85), então

$$\dot{\Xi}(\tilde{q})\mathbf{c} + \Xi(\tilde{q})[\tilde{\omega}]\mathbf{c} = \frac{1}{2}\mathbf{\Omega}_4(\omega)\mathbf{\Omega}_4(\mathbf{c})\tilde{q} - \frac{1}{2}\mathbf{\Omega}_4(\mathbf{c})\mathbf{\Omega}_4(\omega)\tilde{q} + \frac{1}{2}\mathbf{\Omega}_4(\mathbf{c})\mathbf{\Omega}_4(\omega)\tilde{q}$$

$$\dot{\Xi}(\tilde{q})\mathbf{c} + \Xi(\tilde{q})[\tilde{\omega}]\mathbf{c} = \frac{1}{2}\mathbf{\Omega}_4(\omega)\mathbf{\Omega}_4(\mathbf{c})\tilde{q}$$

$$\dot{\Xi}(\tilde{q})\mathbf{c} = -\Xi(\tilde{q})[\tilde{\omega}]\mathbf{c} + \frac{1}{2}\mathbf{\Omega}_4(\omega)\Xi(\tilde{q})\mathbf{c}$$

assim, cortando \mathbf{c} de ambos os lados

$$\frac{d}{dt}\Xi(\tilde{q}(t)) = \frac{1}{2}\mathbf{\Omega}_4(\omega)\Xi(\tilde{q}(t)) - \Xi(\tilde{q}(t))[\tilde{\omega}] \quad (5.87)$$

Uma relação também importante para $\Xi(\tilde{q})$ é que é possível escrever o produto,

$$\Theta(t, t_0)\Xi(\tilde{q}(t_0)) = \Xi(\tilde{q}(t))\Lambda(t, t_0) \quad (5.88)$$

Onde $\Lambda(t, t_0)$ é uma matriz de rotação que será vista mais adiante. Para provar esse relação consideremos que

$$\mathbf{C}(t, t_0) = \Theta(t, t_0)\Xi(\tilde{q}(t_0)) - \Xi(\tilde{q}(t))\Lambda(t, t_0) \quad (5.89)$$

Diferenciando a relação acima em relação ao tempo

$$\frac{\partial}{\partial t}\mathbf{C}(t, t_0) = \frac{1}{2}\mathbf{\Omega}_4(\omega)\mathbf{C}(t, t_0)$$

que integrada resulta em

$$\mathbf{C}(t, t_0) = \Theta(t, t_0)\mathbf{C}(t_0, t_0)$$

mas,

$$\mathbf{C}(t_0, t_0) = \mathbf{0}_{4 \times 3}$$

Verificando assim a Equação (5.89). Pré-multiplicando a Equação (5.88) de ambos os lados por $\Xi^T(\tilde{q}(t))$ ter-se-á

$$\Xi^T(\tilde{q}(t))\Theta(t, t_0)\Xi(\tilde{q}(t_0)) = \Xi^T(\tilde{q}(t))\Xi(\tilde{q}(t))\Lambda(t, t_0)$$

$$\Xi^T(\tilde{q}(t))\Theta(t, t_0)\Xi(\tilde{q}(t_0)) = \mathbf{I}_{4 \times 4}\Lambda(t, t_0)$$

resultando em

$$\Lambda(t, t_0) = \Xi^T(\tilde{q}(t))\Theta(t, t_0)\Xi(\tilde{q}(t_0)) \quad (5.90)$$

e por fim, pós-multiplicando por $\Xi^T(\tilde{q}(t))$ a Equação (5.88), tem-se

$$\Theta(t, t_0)\Xi(\tilde{q}(t_0))\Xi^T(\tilde{q}(t_0)) = \Xi(\tilde{q}(t))\Lambda(t, t_0)\Xi^T(\tilde{q}(t_0))$$

$$\Theta(t, t_0)(\mathbf{I}_{4 \times 4} - \tilde{q}(t_0)\tilde{q}^T(t_0)) = \Xi(\tilde{q}(t))\Lambda(t, t_0)\Xi^T(\tilde{q}(t_0))$$

$$\Theta(t, t_0) - \Theta(t, t_0)\tilde{q}(t_0)\tilde{q}^T(t_0) = \Xi(\tilde{q}(t))\Lambda(t, t_0)\Xi^T(\tilde{q}(t_0))$$

Notando que

$$\Theta(t, t_0)\tilde{q}(t_0) = \tilde{q}(t)$$

então,

$$\Theta(t, t_0) - \tilde{q}(t)\tilde{q}^T(t_0) = \Xi(\tilde{q}(t))\Lambda(t, t_0)\Xi^T(\tilde{q}(t_0))$$

Chegando finalmente em

$$\Theta(t, t_0) = \Xi^T(\tilde{q}(t))\Lambda(t, t_0)\Xi(\tilde{q}(t_0)) + \tilde{q}(t)\tilde{q}^T(t_0) \quad (5.91)$$

5.8 – Representação da Matriz de Covariância na forma Reduzida

A Matriz de Covariância para o vetor de estados é singular. Isto se deve ao fato da norma do quatérnion ser unitária, ou seja,

$$\Delta \tilde{q}^T \hat{\tilde{q}} \approx 0$$

Esta singularidade é difícil de se manter matematicamente devido à acumulação de erros de arredondamento. De fato, $\mathbf{P}(t)$ pode até mesmo desenvolver um autovalor negativo. Existem diversas maneiras de contornar este problema, três formas são descritas em [LEF82]. A forma mais simples é a redução da representação da Matriz de Covariância.

5.8.1 – Predição

Dadas as propriedades da matriz $\Xi(\hat{\tilde{q}})$ da Seção 5.7, tem-se

$$\Theta(t, t')\Xi(\hat{\tilde{q}}(t')) = \Xi(\hat{\tilde{q}}(t))\Lambda(t, t') \quad (5.92)$$

onde

$$\Lambda(t, t') = \mathbf{A}(\hat{\tilde{q}}(t))\mathbf{A}^T(\hat{\tilde{q}}(t')) \quad (5.93)$$

é a matriz de rotação que propaga a matriz de atitude do tempo t para t' . Substituindo esta relação na Equação (5.72) tem-se

$$\Psi(t, t_0) = -\frac{1}{2} \int_{t_0}^t \Theta(t, t')\Xi(\hat{\tilde{q}}(t'))dt'$$

$$\Psi(t, t_0) = -\frac{1}{2} \int_{t_0}^t \Xi(\hat{\tilde{q}}(t))\Lambda(t, t')dt'$$

$$\Psi(t, t_0) = -\frac{1}{2} \Xi(\hat{q}(t)) \int_{t_0}^t \Lambda(t, t') dt'$$

resultando então em

$$\Psi(t, t_0) = -\Xi(\hat{q}(t)) \mathbf{J}(t, t_0) \quad (5.94)$$

onde

$$\mathbf{J}(t, t_0) = -\frac{1}{2} \int_{t_0}^t \Lambda(t, t') dt' \quad (5.95)$$

Também, pelas propriedades da matriz $\Xi(\tilde{q})$ tem-se de (5.91) que

$$\Theta(t, t_0) = \Xi(\hat{q}(t)) \Lambda(t, t_0) \Xi^T(\hat{q}(t_0)) + \hat{q}(t) \hat{q}^T(t_0) \quad (5.96)$$

Substituindo essa expressão dentro da expressão para a matriz de transição, então a matriz de transição (5.70) será expressa por

$$\Phi(t, t_0) = \mathbf{S}(\hat{q}(t)) \tilde{\Phi}(t, t_0) \mathbf{S}^T(\hat{q}(t_0)) + \begin{bmatrix} \hat{q}(t) \hat{q}^T(t_0) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 4} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (5.97)$$

onde

$$\mathbf{S}(\hat{q}(t)) = \begin{bmatrix} \Xi(\hat{q}(t)) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}_{7 \times 6} \quad (5.98)$$

$$\tilde{\Phi}(t, t_0) = \begin{bmatrix} \Lambda(t, t_0) & \mathbf{J}(t, t_0) \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}_{6 \times 6} \quad (5.99)$$

A Equação (5.97) é uma fatoração da matriz de transição de (5.70). Note que, $\Phi(t, t_0)$ é 7×7 enquanto $\tilde{\Phi}(t, t_0)$ é de apenas 6×6 .

Pré-multiplicando a Equação de Riccati (5.19) por $\mathbf{S}^T(\hat{q}(t))$ e pós-multiplicando por $\mathbf{S}(\hat{q}(t))$ tem-se

$$\mathbf{S}^T(\hat{q}(t))\mathbf{P}(t)\mathbf{S}(\hat{q}(t)) = \mathbf{S}^T[\Phi\mathbf{P}(t_0)\Phi^T]\mathbf{S} + \int_{t_0}^t \mathbf{S}^T[\Phi\mathbf{G}\mathbf{Q}\mathbf{G}^T\Phi^T]\mathbf{S}dt' \quad (5.100)$$

Por questão de simplificação termos entre parênteses foram omitidos.

É possível reescrever \mathbf{P} na forma reduzida como

$$\tilde{\mathbf{P}}(t) = \mathbf{S}^T(\hat{q}(t))\mathbf{P}(t)\mathbf{S}(\hat{q}(t)) \quad (5.101)$$

E considerando que $\Phi(t, t_0)$ é fatorado pela equação (5.97), então a Equação (5.100) fica

$$\begin{aligned} \tilde{\mathbf{P}}(t) = & (\mathbf{S}^T\mathbf{S}\tilde{\Phi}\mathbf{S}^T + \mathbf{S}^T\mathbf{\Pi})\mathbf{P}(t)(\mathbf{S}\tilde{\Phi}^T\mathbf{S}^T\mathbf{S} + \mathbf{\Pi}^T\mathbf{S}) \\ & + \int_{t_0}^t \mathbf{S}^T[(\mathbf{S}\tilde{\Phi}\mathbf{S}^T + \mathbf{\Pi})\mathbf{G}\mathbf{Q}\mathbf{G}^T(\mathbf{S}\tilde{\Phi}^T\mathbf{S}^T + \mathbf{\Pi}^T)]\mathbf{S} dt' \end{aligned} \quad (5.102)$$

onde

$$\mathbf{\Pi} = \begin{bmatrix} \hat{q}(t)\hat{q}^T(t_0) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 4} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (5.103)$$

Notando a propriedade de ortogonalidade entre $\Xi(\tilde{q})$ e \tilde{q} , então

$$\mathbf{S}^T\mathbf{\Pi} = \mathbf{\Pi}^T\mathbf{S} = \mathbf{0}_{6 \times 6} \quad (5.104)$$

Finalmente, a forma da Equação (5.102) fica reduzida a forma da Equação de Riccati,

$$\tilde{\mathbf{P}}(t) = \tilde{\Phi}(t, t_0)\tilde{\mathbf{P}}(t_0)\tilde{\Phi}^T(t, t_0) + \int_{t_0}^t \tilde{\Phi}(t, t')\tilde{\mathbf{G}}(t')\mathbf{Q}(t')\tilde{\mathbf{G}}^T(t')\tilde{\Phi}^T(t, t')dt' \quad (5.105)$$

onde,

$$\tilde{\mathbf{G}}(t) = \mathbf{S}^T(\hat{q}(t))\mathbf{G}(t) = \begin{bmatrix} -\frac{1}{2}\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.106)$$

A matriz $\mathbf{P}(t)$ pode ser reconstruída via

$$\mathbf{P}(t) = \mathbf{S}(\hat{q}(t))\tilde{\mathbf{P}}(t)\mathbf{S}^T(\hat{q}(t)) \quad (5.107)$$

5.8.2 – Filtragem

O Filtro de Kalman pode ser implementado fazendo o uso da matriz reduzida $\tilde{\mathbf{P}}(t)$. Pode-se definir a matriz reduzida $\tilde{\mathbf{H}}$ e a matriz $\tilde{\mathbf{K}}$ como

$$\tilde{\mathbf{H}}_k = \mathbf{H}_k \mathbf{S}(\hat{q}_k(-)) \quad (5.108)$$

$$\tilde{\mathbf{K}}_k = \tilde{\mathbf{P}}_k(-)\tilde{\mathbf{H}}_k^T [\tilde{\mathbf{H}}_k^T \tilde{\mathbf{P}}_k(-)\tilde{\mathbf{H}}_k^T + \mathbf{R}_k]^{-1} \quad (5.109)$$

Assim a atualização da covariância reduzida é dada por

$$\tilde{\mathbf{P}}_k(+) = \tilde{\mathbf{P}}_k(-) - \tilde{\mathbf{K}}_k \tilde{\mathbf{H}}_k \tilde{\mathbf{P}}_k(-) \quad (5.110)$$

A implementação da matriz reduzida torna desnecessário o uso da matriz de covariância 7×7 . A matriz reduzida pode ser propagada e atualizada pelas Equações (5.105) e (5.110) respectivamente.

A matriz ganho de Kalman 7×7 é dada por

$$\mathbf{K}_k = \mathbf{S}(\hat{q}_k(-))\tilde{\mathbf{K}}_k \quad (5.111)$$

5.9 – Formatando as Equações para Simulação do Filtro de Kalman

5.9.1 – Introdução

Toda a descrição feita nos tópicos acima teve o objetivo de apresentar os conceitos da filtragem de Kalman que estão relacionados com atitude. Agora, visando uma implementação do filtro, as equações serão compiladas para uma forma numérica discreta e uma simulação será feita no Matlab® para comprovar o funcionamento do modelo para um caso específico.

5.9.2 – Algoritmo para simulação – mecanização do filtro de Kalman

A. Predição (propagação) do estado

As etapas de cálculos para propagação do vetor de estados são:

1. Ler a velocidade angular \mathbf{u} de uma tríade de girômetros durante o intervalo de predição. Encontrar $\hat{\boldsymbol{\omega}}$ subtraindo $\hat{\mathbf{b}}$;
2. A partir da integral de $\hat{\boldsymbol{\omega}}$, encontra ângulo vetorial $\Delta\boldsymbol{\theta}$ e seu módulo;
3. Calcular a matriz de rotação $\mathbf{M}(\Delta\boldsymbol{\theta})$;
4. Propagar o estado \mathbf{x}_k .

Para predição da atitude é utilizado um girômetro. A integração da saída do girômetro durante um curto intervalo de tempo resulta em um pequeno delta ângulo. Esse delta ângulo pode ser aproximado por

$$\Delta\boldsymbol{\theta} = \sum_{i=1}^n [(\mathbf{u}_{k,i} - \hat{\mathbf{b}}_k) T_s]$$

$$\Delta\boldsymbol{\theta} = T_s \sum_{i=1}^n (\mathbf{u}_{k,i} - \hat{\mathbf{b}}_k)$$

$$\Delta\boldsymbol{\theta} = \frac{\Delta T}{n} \sum_{i=1}^n (\mathbf{u}_{k,i} - \hat{\mathbf{b}}_k)$$

$$\Delta\boldsymbol{\theta} = \frac{\Delta T}{n} \sum_{i=1}^n \mathbf{u}_{k,i} - \hat{\mathbf{b}}_k \frac{\Delta T}{n} \quad (5.112)$$

onde, $\Delta\boldsymbol{\theta}$ é o incremento angular vetorial, n o número de amostras tomada no intervalo de predição. A relação entre o intervalo de predição ΔT e o período de amostragem da velocidade angular do sensor é dada por

$$\Delta T = nT_s$$

Utilizando a equação (5.64), repetida abaixo por clareza, com o ângulo obtido pela equação (5.112)

$$\boldsymbol{\Theta}(t, t_0) = \mathbf{M}(\Delta\boldsymbol{\theta}) = \cos(|\Delta\boldsymbol{\theta}|/2)\mathbf{I}_{4 \times 4} + \sin(|\Delta\boldsymbol{\theta}|/2)\frac{\boldsymbol{\Omega}_4(\Delta\boldsymbol{\theta})}{|\Delta\boldsymbol{\theta}|} \quad (5.113)$$

A atualização do estado pode ser feita pelo produto matricial

$$\mathbf{x}(t) = \begin{bmatrix} \tilde{\mathbf{q}}(t) \\ \mathbf{b}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\Delta\boldsymbol{\theta}) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{q}}(t_0) \\ \mathbf{b}(t_0) \end{bmatrix} \quad (5.114)$$

Considerando $\tilde{\mathbf{q}}(t_0) = \tilde{\mathbf{q}}_k$ e $\tilde{\mathbf{q}}(t) = \tilde{\mathbf{q}}_{k+1}$ é possível reescrever como

$$\mathbf{x}_{k+1} = \begin{bmatrix} \tilde{\mathbf{q}}_{k+1} \\ \mathbf{b}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\Delta\boldsymbol{\theta}) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{q}}_k \\ \mathbf{b}_k \end{bmatrix} \quad (5.115)$$

ou de forma isolada, simplesmente

$$\begin{aligned} \tilde{\mathbf{q}}_{k+1} &= \mathbf{M}(\Delta\boldsymbol{\theta})\tilde{\mathbf{q}}_k \\ \mathbf{b}_{k+1} &= \mathbf{b}_k \end{aligned} \quad (5.116)$$

B. Predição (propagação) da covariância

A propagação da covariância feita pela Equação (5.105), depende de vários termos, e portanto, a ordem dos cálculos é dada por:

1. Calcular as matrizes $\mathbf{A}(\hat{q}_k)$ e $\mathbf{A}(\hat{q}_{k+1})$ em função dos quatérnions.
2. Calcular $\mathbf{\Lambda}$
3. Integrar $\mathbf{\Lambda}$ para encontrar \mathbf{J}
4. Integrar segunda parte da Equação (5.105)
5. Calcular a covariância propagada $\tilde{\mathbf{P}}_{k+1}$

A matriz $\tilde{\mathbf{G}}(t')$ da Equação (5.106), possui apenas constantes, e a matriz $\mathbf{Q}(t')$, da Equação (5.69) é diagonal. A diagonal da matriz $\mathbf{Q}(t')$ é formada pelas variâncias dos girômetros. A matriz $\mathbf{Q}(t')$ pode ser escrita como

$$\tilde{\mathbf{Q}}_k = \tilde{\mathbf{G}}_k \mathbf{Q}_k \tilde{\mathbf{G}}_k^T \quad (5.117)$$

e substituindo os valores,

$$\tilde{\mathbf{Q}}_k = \begin{bmatrix} -\frac{1}{2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1(t) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q}_2(t) \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}$$

$$\tilde{\mathbf{Q}}_k = \begin{bmatrix} -\frac{1}{2} \mathbf{Q}_1(t) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q}_2(t) \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}$$

tem-se

$$\tilde{\mathbf{Q}}_k = \begin{bmatrix} \frac{1}{4} \mathbf{Q}_1(t) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q}_2(t) \end{bmatrix}_{6 \times 6} \quad (5.118)$$

A matriz diagonal $\mathbf{Q}_1(t)$ é formada pela variância da velocidade medida em cada eixo da tríade ortogonal de giros. A matriz $\mathbf{Q}_2(t)$ é formada pelas variâncias da deriva do giro.

A matriz de transição reduzida, Equação (5.99), é formada pelas funções matriciais $\mathbf{\Lambda}_k$ e \mathbf{J}_k . Então a matriz $\mathbf{\Lambda}_k$ fica

$$\Lambda_k = \mathbf{A}(\hat{q}_{k+1})\mathbf{A}^T(\hat{q}_k) \quad (5.119)$$

e a matriz \mathbf{J}_k pode ser aproximada pela regra do trapézio por

$$\mathbf{J}(t, t_0) = -\frac{1}{2} \int_{t_0}^t \Lambda(t, t') dt' = -\frac{1}{2} [\Lambda_k + \mathbf{I}_{3 \times 3}] \frac{\Delta T}{2} = -[\Lambda_k + \mathbf{I}_{3 \times 3}] \frac{\Delta T}{4} \quad (5.120)$$

Assim, a matriz de transição fica montada como

$$\tilde{\Phi}_k = \begin{bmatrix} \Lambda_k & \mathbf{J}_k \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\hat{q}_{k+1})\mathbf{A}^T(\hat{q}_k) & -[\Lambda_k + \mathbf{I}_{3 \times 3}] \frac{\Delta T}{4} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.121)$$

Utilizando as formas de (5.117) e (5.121), parte direita de equação (5.105) pode ser integrada pela regra do trapézio como

$$\Gamma_k = (\tilde{\Phi}_k \tilde{\mathbf{Q}}_k \tilde{\Phi}_k^T + \mathbf{I}_{6 \times 6} \tilde{\mathbf{Q}}_k \mathbf{I}_{6 \times 6}) \frac{\Delta T}{2} = (\tilde{\Phi}_k \tilde{\mathbf{Q}}_k \tilde{\Phi}_k^T + \tilde{\mathbf{Q}}_k) \frac{\Delta T}{2} \quad (5.122)$$

Agora a covariância pode finalmente ser predita por

$$\tilde{\mathbf{P}}_{k+1} = \tilde{\Phi}_k \tilde{\mathbf{P}}_k \tilde{\Phi}_k^T + \Gamma_k = \tilde{\Phi}_k \tilde{\mathbf{P}}_k \tilde{\Phi}_k^T + (\tilde{\Phi}_k \tilde{\mathbf{Q}}_k \tilde{\Phi}_k^T + \tilde{\mathbf{Q}}_k) \frac{\Delta T}{2} \quad (5.123)$$

C. Atualização (Filtragem) do Estado

Os passos para realizar as atualizações do vetor de estados e da matriz de covariância dos erros do estado são:

1. A partir de \mathbf{H} e \mathbf{S} , calcular $\tilde{\mathbf{H}}$
2. Calcular o ganho $\tilde{\mathbf{K}}$ (muitas operações)
3. Encontrar \mathbf{K}
4. Atualizar estado $\hat{\mathbf{x}}_k(+)$

5. Atualizar $\tilde{\mathbf{P}}_k(+)$

Como a matriz de medida \mathbf{H} é simples, alguns passos podem ser abreviados algebricamente. Primeiramente, considerando que a atitude atual é fornecida por um algoritmo de determinação de atitude (como TRIAD ou QUEST), a matriz de medida será

$$\mathbf{H} = [\mathbf{I}_{4 \times 4} \quad \mathbf{0}_{4 \times 3}]_{4 \times 7} \quad (5.124)$$

Dadas as Equação (5.98) e (5.108) então

$$\tilde{\mathbf{H}}_k = \mathbf{H}_k \mathbf{S}(\hat{q}_k) = [\mathbf{I}_{4 \times 4} \quad \mathbf{0}_{4 \times 3}] \begin{bmatrix} \Xi(\hat{q}_k) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} = [\Xi(\hat{q}_k) \quad \mathbf{0}_{4 \times 3}]_{4 \times 7} \quad (5.125)$$

A matriz ganho reduzida dada pela Equação (5.109) fica

$$\tilde{\mathbf{K}}_k = \tilde{\mathbf{P}}_k(-) [\Xi(\hat{q}_k) \quad \mathbf{0}_{4 \times 3}]^T \left([\Xi(\hat{q}_k) \quad \mathbf{0}_{4 \times 3}] \tilde{\mathbf{P}}_k(-) [\Xi(\hat{q}_k) \quad \mathbf{0}_{4 \times 3}]^T + \mathbf{R}_k \right)^{-1} \quad (5.126)$$

A covariância do quatérnion observado (TRIAD ou QUEST) pode ser representada por

$$\mathbf{R}_k = \Xi(\hat{q}_k) \tilde{\mathbf{R}}_k \Xi^T(\hat{q}_k) \quad (5.127)$$

E a matriz de covariância $\tilde{\mathbf{P}}_k$ pode subdividida em quatro matrizes de posto três na forma,

$$\tilde{\mathbf{P}}_k = \begin{bmatrix} \tilde{\mathbf{P}}_{\theta\theta} & \mathbf{P}_{\theta b} \\ \mathbf{P}_{b\theta} & \mathbf{P}_{bb} \end{bmatrix} \quad (5.128)$$

Assim, a Equação (5.126) pode ser simplificada

$$\tilde{\mathbf{K}}_k = \begin{bmatrix} \tilde{\mathbf{P}}_{00} & \tilde{\mathbf{P}}_{0b} \\ \tilde{\mathbf{P}}_{b0} & \tilde{\mathbf{P}}_{bb} \end{bmatrix}_k \left[\begin{bmatrix} \Xi(\hat{q}_k) & \mathbf{0}_{4 \times 3} \end{bmatrix}^T \left(\begin{bmatrix} \Xi(\hat{q}_k) & \mathbf{0}_{4 \times 3} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{P}}_{00} & \tilde{\mathbf{P}}_{0b} \\ \tilde{\mathbf{P}}_{b0} & \tilde{\mathbf{P}}_{bb} \end{bmatrix}_k \begin{bmatrix} \Xi(\hat{q}_k) & \mathbf{0}_{4 \times 3} \end{bmatrix}^T + \Xi(\hat{q}_k) \tilde{\mathbf{R}}_k \Xi^T(\hat{q}_k) \right)^{-1} \right]$$

$$\tilde{\mathbf{K}}_k = \begin{bmatrix} \tilde{\mathbf{P}}_{00} & \tilde{\mathbf{P}}_{0b} \\ \tilde{\mathbf{P}}_{b0} & \tilde{\mathbf{P}}_{bb} \end{bmatrix}_k \begin{bmatrix} \Xi^T(\hat{q}_k) \\ \mathbf{0}_{3 \times 4} \end{bmatrix} \left(\Xi(\hat{q}_k) \tilde{\mathbf{P}}_{00,k} \Xi^T(\hat{q}_k) + \Xi(\hat{q}_k) \tilde{\mathbf{R}}_k \Xi^T(\hat{q}_k) \right)^{-1}$$

$$\tilde{\mathbf{K}}_k = \begin{bmatrix} \tilde{\mathbf{P}}_{00} \\ \tilde{\mathbf{P}}_{b0} \end{bmatrix} \Xi^T(\hat{q}_k) \Xi(\hat{q}_k) (\tilde{\mathbf{P}}_{00,k} + \tilde{\mathbf{R}}_k)^{-1} \Xi^T(\hat{q}_k)$$

chegando finalmente a forma reduzida do ganho da Kalman

$$\tilde{\mathbf{K}}_k = \begin{bmatrix} \tilde{\mathbf{P}}_{00} \\ \tilde{\mathbf{P}}_{b0} \end{bmatrix} (\tilde{\mathbf{P}}_{00,k} + \tilde{\mathbf{R}}_k)^{-1} \Xi^T(\hat{q}_k) \quad (5.129)$$

A covariância é atualizada pela Equação (5.110) repetida a seguir

$$\tilde{\mathbf{P}}_k(+) = \tilde{\mathbf{P}}_k(-) - \tilde{\mathbf{K}}_k \tilde{\mathbf{H}}_k \tilde{\mathbf{P}}_k(-) \quad (5.130)$$

Vale notar que a notação de $(-)$ se refere ao valor propagado e imediatamente antes a filtragem, enquanto o $(+)$ e o valor imediatamente após a filtragem.

Finalmente, a atualização do estado, é dada pela equação a seguir

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k [\tilde{q}_{\text{obs}} - \mathbf{H} \hat{\mathbf{x}}_k(-)] \quad (5.131)$$

O ganho de Kalman ampliado é calculado pela Equação (5.111). O quatérnion de atitude observado \tilde{q}_{obs} é obtido por intermédio de algum algoritmo de determinação de atitude como por exemplo, o TRIAD ou QUEST. A matriz de covariância desta medida é $\tilde{\mathbf{R}}_k$ (\mathbf{P}_{00} ou $4\mathbf{P}_{\text{qq}}$). Lembrando que se o algoritmo retornar a matriz de atitude e não o quatérnion, o mesmo deve ser calculado pela relação em (5.33).

5.9.3 – Considerações sobre os intervalos

A ordem em que os eventos de predição e filtragem ocorrem são ilustrados pela Figura 5.1. Um intervalo de predição pode durar o tempo que for adequado à aplicação, mas devido à aproximação da Equação (5.64) o tempo entre cada propagação não pode ser demasiadamente grande para evitar que ocorra erros na integração. Vale lembrar que a atitude é lida somente nos instantes de filtragem (atualização dos estados). Durante o intervalo de propagação, somente são feitas leituras do girômetros. A flexibilização do tamanho do intervalo de predição é um valioso recurso que pode ser adotado para contornar as acelerações lineares. Neste trabalho não será estudada nem empregada essa flexibilização que pode ser vista em [GEB04].

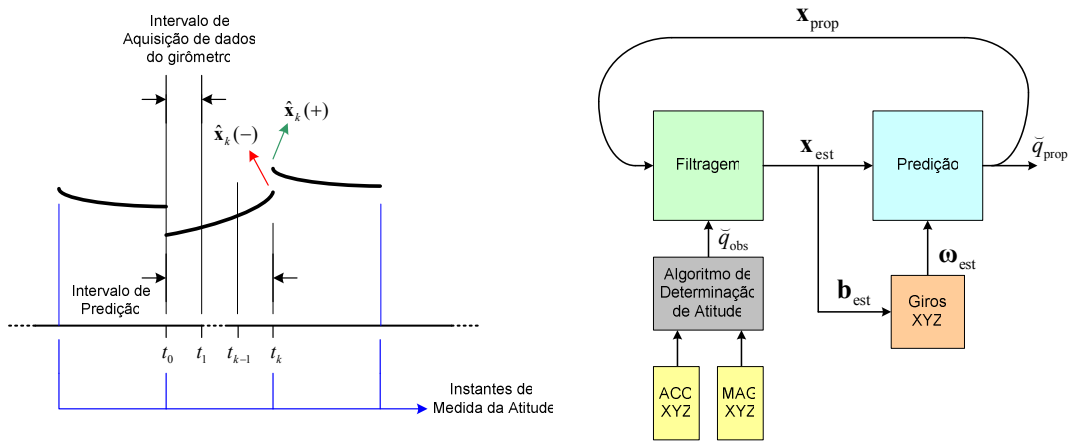


Figura 5.1 – Ordem dos eventos da filtragem e propagação. (a) Vetor de estado em dois momentos: filtragem e predição. (b) Diagrama ilustrativo da implementação do filtro.

5.9.4 – Fluxograma da simulação

A Figura 5.2 apresenta o fluxograma da simulação, na seqüência em que as operações são feitas. A simulação é controlada pelas velocidades angulares em cada eixo fornecidas na inicialização. O estado, composto pelo quatérnion e *bias* dos giros, é inicializado com os valores da primeira leitura. O níveis de tensão do ruído que é adicionado a tensão simulada de cada um dos sensores tem valor médio quadrático muito próximo do valor do ruído medido.

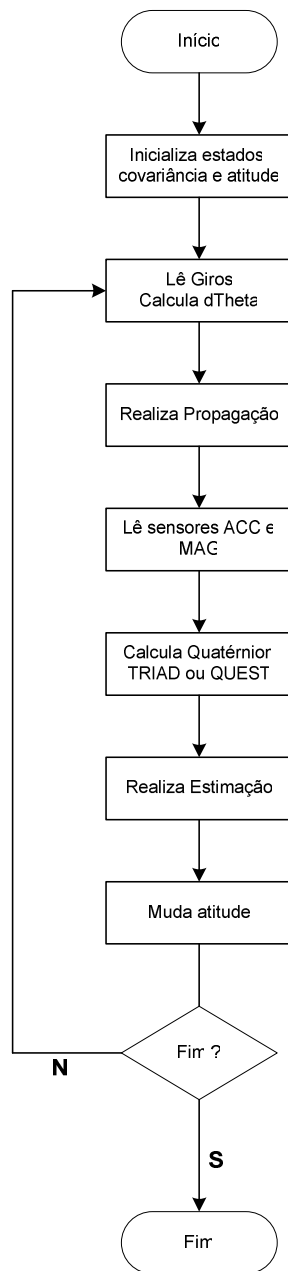


Figura 5.2 – Fluxograma da simulação do filtro de Kalman

As funções de atualização e propagação são apresentadas a seguir em Matlab.

5.10 – Simulação do Filtro

A simulação do filtro de Kalman envolve a leitura de dados simulados de diversos sensores. Neste caso são: duas tríades de sensores de direção, como inclinômetros e magnetômetros, e também uma tríade de sensores de velocidade de

rotação: girômetros. Em ambas as tríades será adicionado ruído gaussiano de variância fixa. Aos giros também será adicionado *bias*. Os parâmetros que regem a simulação são as velocidades angulares impostas, o tempo de amostragem (tempo de propagação) e o tempo total de simulação.

5.10.1 – Código da simulação

```
% Inicializa Diretório das funções
chdir('C:\Documents and Settings\Junior\Desktop\Tese
Mestrado\FiltroKalmanSim_A\NovosAlgs');

% 1. Inicialização do Filtro e das Constantes
% Tempo de Simulação em Seg
Tempo = 72;

% Intervalo de Amostragem
Ts = 0.02;

% Número de Loops
N = Tempo/Ts;

% Intervalo de Integração
IT = 20;

% Vetor Unitário de Referência 1
V1 = [0 0 1]';
V1 = V1 / norm(V1);

% Vetor Unitário de Referência 2
V2 = [0.5 0.5 0.5]';
V2 = V2 / norm(V2);

% Desvio padrão dos Sensores
SigMag = 0.071;
SigAcc = 0.071;
SigGyro = 0.071;
SigDGyro = 0.071; % desvio padrão da deriva do Gyro

% Definindo bias para cada Gyro
b(1,1) = 0.1;
```

```

b(2,1) = 0.2;
b(3,1) = 0.3;

% Dado de Entrada: velocidades angulares
% Entrada em graus por segundo:
w1 = 5.0 * pi/180;
w2 = 0.0 * pi/180;
w3 = 0.0 * pi/180;

% Formação do vetor velocidade angular:
w = [w1 w2 w3]';

% Dado de Entrada: posição angular inicial
% Entrada: quatérnion com atitude inicial
q1 = 0.0;
q2 = 0.0;
q3 = 0.0;
q0 = 1.0;

% Formação do quatérnion de atitude inicial
qtrue(:,1) = [q1 q2 q3 q0]';

% Inicialização da Matriz de ruído do Giro %
Q = [SigGyro^2 * eye(3) zeros(3,3);
     zeros(3,3) SigDGyro^2 * eye(3)];

% Inicialização da Matriz de Covariância %

Ptt = 1e6 * eye(3); % eye é uma função que cria uma matriz identidade
%
Ptb = zeros(3,3); % zeros é uma função que cria uma matriz de zeros %
Pbt = zeros(3,3);
Pbb = 1e6 * eye(3);
PT_est = [Ptt Ptb;
          Pbt Pbb];

% Matriz H - Matriz de Medida
H = [eye(4) zeros(4,3)];

% Matriz Constante
GT = [ -1/2*eye(3) zeros(3,3); zeros(3,3) eye(3) ];

```

```

QT = GT * Q * GT';

% Inicializa vetores com zero
x_est(7,N) = 0;
x_prop(7,N) = 0;
q_Tri(4,N) = 0;

% Inicializa primeiro valor
x_est(:,1) = [0 0 0 1 0 0 0]';
x_prop(:,1) = [0 0 0 1 0 0 0]';

% Marca Instante de Início
tic;

for i = 1:N
    % Item de Simulação, Gera Leitura do Sensores c/ ruído
    [W1,W2,W1N,W2N] = GetMagAccOuts(qtrue(:,i),V1,V2,SigMag,SigAcc);

    % 2. Item de Simulação, Obtem Leitura dos Gyros
    for j = 1:IT
        u(1:3,j) = GetGyroOut(SigGyro,w);
    end

    % 3. Calculo do deslocamento angular (vetorial)
    Th = sum(u,2)*Ts - x_est(5:7,i)*Ts*IT;

    % 4. Propaga o estado
    [x_prop(:,i+1), PT_prop] = KalmanProp(x_est(:,i),PT_est,QT,Th,Ts);

    % 5. Lê Atitude atual
    [A_Tri, P_Tri] = Triad(V1,V2,W1N,W2N,SigMag,SigAcc);
    q_Tri(:,i) = Mat2q(A_Tri)';

    % 6. Filtragem pelo Estimador
    [x_est(:,i+1),PT_est,P_est] = KalmanEst(x_prop(:,i+1),PT_prop,
    q_Tri(:,i), P_Tri);

    % Tira desvio padrão de cada variável
    for c = 1:7
        std_desv(c,i) = sqrt(P_est(c,c));
    end
end

```

```

% 7. Muda Atitude. Integração para o Próximo quatérnion verdadeiro
Th = w*Ts;
M = eye(4) * cos(norm(Th)/2) + Omega4(Th)*sin(norm(Th)/2)/norm(Th);
qtrue(:,i+1) = M * qtrue(:,i);

if mod(i,floor(N/100)) == 0
    disp(i);
end

end

% Marca o tempo da simulação
toc

% Função que Gera os sinais dos sensores Acc e Mag com base na Atitude
function [W1,W2,W1N,W2N] = GetMagAccOuts(qtrue,V1,V2,SigMag,SigAcc)

% Matriz de Atitude verdadeira
A = q2Mat(qtrue);

% Vetor Unitário de Observação: W1
W1 = A * V1;
W1N = W1 + SigMag * randn(3,1);
% Vetor Unitário de Observação c/ ruído: W1N
W1N = W1N / norm(W1N);

% Vetor Unitário de Observação: W2
W2 = A * V2;
W2N = W2 + SigAcc * randn(3,1);
% Vetor Unitário de Observação c/ ruído: W2N
W2N = W2N / norm(W2N);

% Função que gera o valor de u para cada giro
function u = GetGyroOut(SigGyro,w)

% Definindo bias para cada Gyro
b(1,1) = 0.1;
b(2,1) = 0.2;
b(3,1) = 0.3;

% A saída lida do giro é composta da velocidade angular
% mais uma bias, mais o ruído, como a seguir

u = w + SigGyro * randn(3,1) + b;

```

5.10.2 – Funções de propagação e filtragem

Três funções do código acima são o núcleo da simulação: a função de propagação, a função de determinação do quatérnio de atitude e a função de filtragem. Ainda existem as funções que simulam os sinais dos sensores com ruído adicionado e a função que gera uma rotação incremental a atitude a partir das velocidades angulares e do tempo de amostragem.

A. Função de propagação

Para realizar a propagação, deve-se obter os dados dos girômetros e então integrá-los de forma a obter o deslocamento angular ocorrido no intervalo de aquisição. O deslocamento angular é pequeno o suficiente para considerar que a direção do vetor velocidade angular se altere.

A seguir o código que implementa a leitura e integração dos dados dos giros e também a função de propagação do filtro. Note que a função de propagação é chamada somente após se ter a medida do “vetor ângulo” de deslocamento.

```
% ===== Função do Filtro de Kalman =====
% Os parâmetros de Entrada são:
% x_est : estado estimado
% PT_est : matriz de covariância estimada
% QT : matriz de ruído do giros
% Th : vetor deslocamento angular
% Ts : intervalo de propagação
% =====
% Os parâmetros de Retorno são:
% x_prop : estado propagado
% PT_prop : covariância reduzida propagada
% =====

function [x_prop, PT_prop] = KalmanProp(x_est, PT_est, QT, Th, Ts)

% ===== Propagação =====
% ----- Propagação do Estado
% São medidos os Girômetros e sua saída subtraída
% do bias é integrada no período de propagação
```

```

q_est = x_est(1:4,1);
b_est = x_est(5:7,1);

% Matriz de atualização do quatérnion
M = eye(4) * cos(norm(Th)/2) + Omega4(Th) * sin(norm(Th)/2) / norm(Th);
q_prop = M * q_est;
b_prop = b_est;
x_prop = [q_prop ; b_prop];

% ----- Propagação da Matriz de Covariância
L = q2Mat(q_prop) * q2Mat(q_est)';
KK = -1/2 * (L + eye(3)) * Ts/2; % Integral Trapézio
PhiT = [L KK; zeros(3,3) eye(3)];
SiG = (PhiT * QT * PhiT' + QT) * Ts/2; % Integral Trapézio
PT_prop = PhiT * PT_est * PhiT' + SiG;

```

Na função acima são utilizadas funções trigonométricas para gerar a matriz de rotação M que propaga o quatérnion. As integrais necessárias à propagação da matriz de covariância reduzida, como das Equações (5.120) e (5.123), são resolvidas pela regra do trapézio.

B. Função de filtragem

Para realização da filtragem é necessário realizar uma leitura da atitude. A atitude pode ser encontrada com o uso do algoritmo TRIAD ou QUEST (Cap. 4). O código do estimador segue abaixo.

```

% ===== Função do Filtro de Kalman =====
% Os parâmetros de Entrada são:
% x_prop    : estado propagado
% PT_prop   : matriz de covariância propagada
% q_Tri     : quatérnion de atitude (pelo TRIAD/QUEST)
% P_Tri     : matriz de covariância da medida de q_Tri
% =====
% Os parâmetros de Retorno são:
% x_esti    : estado estimado
% PT_esti   : covariância reduzida estimada
% P_esti    : covariância dos estimada
% =====

```

```

function [x_esti, PT_esti, P_esti] = KalmanEst(x_prop, PT_prop, q_Tri,
P_Tri)

% ---- Separando as componentes do estado
q_prop = x_prop(1:4,1);
b_prop = x_prop(5:7,1);

% ---- Montando a matriz E (Ksi)
E = [ q_prop(4) -q_prop(3)  q_prop(2);
      q_prop(3)  q_prop(4) -q_prop(1);
      -q_prop(2)  q_prop(1)  q_prop(4);
      -q_prop(1) -q_prop(2) -q_prop(3)];

% ---- Montando a matriz S
S = [E zeros(4,3); zeros(3,3) eye(3)];

% ---- Montando a matriz H e HT.
H = [eye(4) zeros(4,3)];
HT = [E zeros(4,3)];

% ---- Separando covariancias dos estados
Ptt_prop = PT_prop(1:3,1:3);
Pbt_prop = PT_prop(4:6,1:3);

% ---- Cálculo do novo ganho (reduzido)
KT = [Ptt_prop; Pbt_prop] * inv(Ptt_prop + P_Tri) * E';

% ---- Atualização da covariância reduzida
PT_esti = PT_prop - KT * HT * PT_prop;

% ---- Cálculo da covariância estendida
P_esti = S * PT_esti * S' ;

% ---- Cálculo do ganho estendido
K = S * KT;

% ---- Atualização do estado
x_esti = x_prop + K * (q_Tri - H * x_prop);

% ---- renormalização do quatérnion

```

```
n = norm(x_esti(1:4));           % norma do quatérnio
x_esti(1:4,1) = x_esti(1:4,1) / n; % renormalização do quatérnio
```

Nenhuma função especial é chamada no código acima. São feitas apenas operações matriciais. A operação mais custosa de codificar em linguagem C no código acima é uma inversa de uma matriz 3×3 , mas que no Matlab é facilmente implementada com o uso da função `inv`.

5.10.3 – Funções de simulação dos Sensores

Simular os sensores é dar uma saída de acordo com a atitude. Como simplificação, não serão considerados modelos dos sensores, mas apenas leituras escaladas corretamente para a grandeza em questão e com ruído gaussiano adicionado e *off-set* (se for o caso).

A. Simulação do giro

A simulação do giro é feita pela breve função a seguir

```
% ===== Função para Ler o Gyro =====
% Parâmetros de Entrada
% b      : vetor bias
% w      : vetor velocidade angular verdadeira
% SigGyro : desvio padrão do ruído da saída
% =====
% Parâmetro de Saída
% u      : medida de velocidade corrompida
% =====

function u = GetGyroOut(SigGyro,w,b)

% A saída lida do giro é composta da velocidade angular
% mais uma bias, mais o ruído, como a seguir

u = w + SigGyro*randn(3,1) + b;
```

O vetor velocidade angular é um parâmetro de entrada da simulação, assim como o vetor *bias*. A função `randn` gera um vetor de números randômicos de média zero e desvio padrão unitário. A saída `u` é uma soma destes efeitos.

B. Simulação dos vetores observados

A simulação do vetores observados tem como parâmetro principal a atitude atual. Pela definição, os vetores observados são vetores de referência girados pela matriz de atitude atual, como visto na Equação (5.1). Por simplificação será desconsiderado o *bias* destes sensores, bem como não-linearidades, erro do fator de escala, variação como temperatura, alinhamento, efeitos da aceleração, etc. O que os sensores lêem são simplesmente vetores de observação adicionado a um ruído gaussiano de média zero e de desvio padrão fixo. Depois de adicionado o ruído, os vetores são renormalizados.

A seguir o simples código da função em questão

```
% ===== Função GeraVetores =====
% Parâmetros de Entrada
% qtrue   : atitude atual verdadeira
% V1      : vetor de referência V1
% V2      : vetor de referência V2
% SigMag  : desvio padrão do ruído de W1
% SigAcc  : desvio padrão do ruído de W2
% =====
% Parâmetros de Saída
% W1      : vetor de observação W1
% W2      : vetor de observação W2
% W1N     : vetor de observação W1 + Noise
% W2N     : vetor de observação W2 + Noise
% =====
function [W1,W2,W1N,W2N] = GeraVetores(qtrue,V1,V2,SigMag,SigAcc)

% ---- Matriz de Atitude verdadeira
A = q2Mat(qtrue);

% ---- Vetor Unitário de Observação: W1
W1 = A * V1;
W1N = W1 + SigMag * randn(3,1);
```

```

% ---- Vetor Unitário de Observação c/ ruído: W1N
W1N = W1N / norm(W1N);

% ---- Vetor Unitário de Observação: W2
W2 = A * V2;
W2N = W2 + SigAcc * randn(3,1);

% ---- Vetor Unitário de Observação c/ ruído: W2N
W2N = W2N / norm(W2N);

```

5.10.4 – Simulação com atitude em repouso

O primeiro teste do filtro será com atitude em repouso, ou seja, as velocidades angulares são ajustadas para zero e o quatérnion inicial é um quatérnion unitário qualquer. O *bias* do giro será considerado fixo. Assim, rodando o código da seção 5.10.1 com esses parâmetros, a Figura 5.3¹ mostra o quatérnion estimado (primeiros quatro elementos do vetor de estados), em comparação com o quatérnion resultante do algoritmo QUEST e o quatérnion verdadeiro.

¹ Com dito no Cap. 3, às vezes o quatérnion é descrito como $[q_0 \ q_1 \ q_2 \ q_3]^T$ e outras vezes como $[q_1 \ q_2 \ q_3 \ q_0]^T$, sendo que nesta última o elemento q_0 pode ser compreendido como q_4 para facilitar computacionalmente.

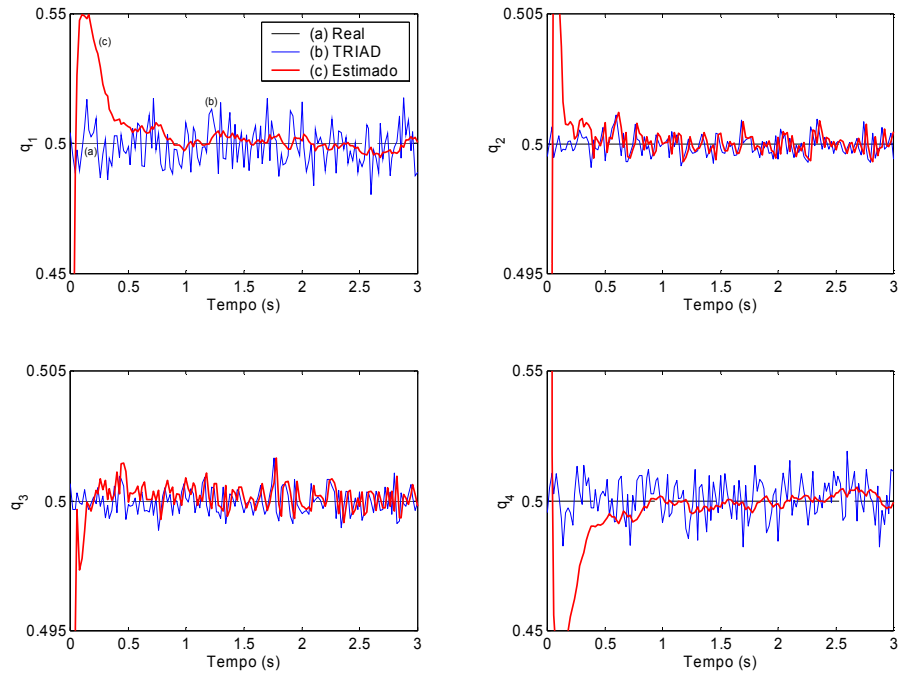


Figura 5.3 – Componentes do quatérnion estimadas pelo filtro – simulação estática

A Figura 5.4 apresenta o erro e o desvio padrão para essas componentes ao longo do tempo. O desvio de cada elemento do quatérnion é obtido da diagonal principal da matriz de covariância.

$$\sigma_i = \sqrt{\mathbf{P}_{ii}} \quad (5.132)$$

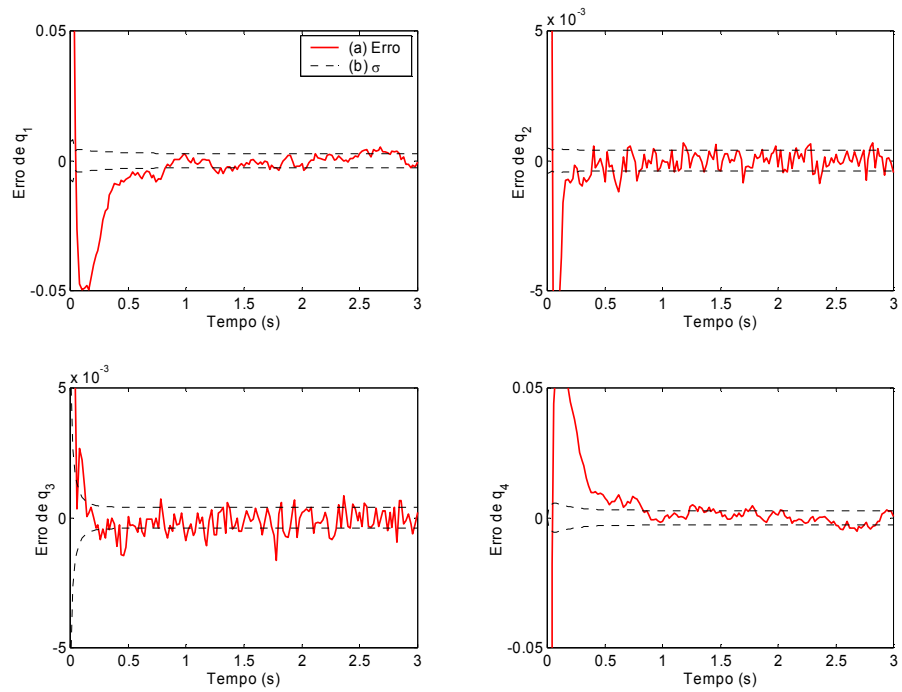


Figura 5.4 – Erros e desvio padrão das componentes do quatérnion

A Figura 5.5 mostra a convergência dos estados do *bias* do giros (três últimos do vetor de estados). Eles são comparados com o *bias* verdadeiro imposto na simulação. A Figura 5.6 mostra a plotagem do erro e seu respectivo desvio padrão retirado da diagonal da matriz de covariância estimada.

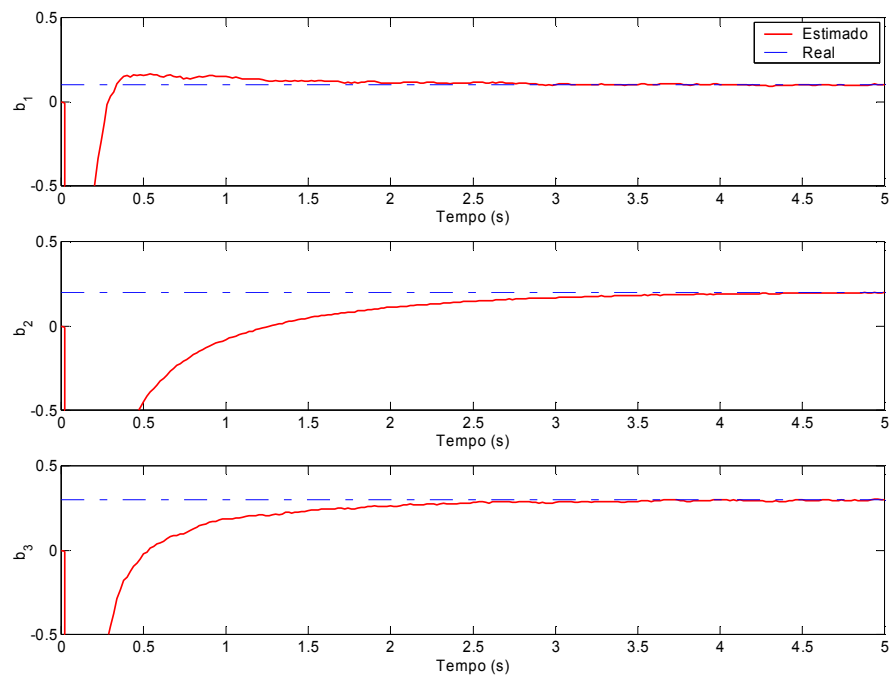


Figura 5.5 – Componentes do *bias* do giro estimadas pelo filtro

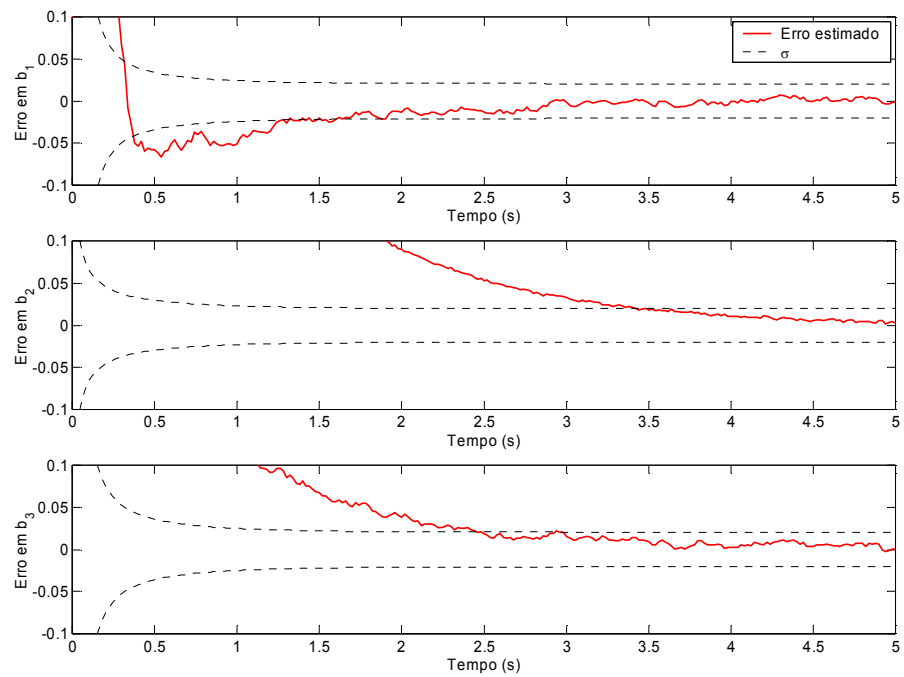


Figura 5.6 – Erros e desvio padrão das componentes do *bias* do giro.

É possível observar nas Figuras 5.4 e 5.6, que apresentam os erros do estados, que há convergência dos valores estimados para os valores reais. O tempo de convergência não chega a 5 segundos.

5.10.5 – Simulação sob giro entorno de vetor fixo

Neste teste são ajustadas velocidades angulares fixas e diferente de zero para os três eixos. Os *bias* simulado do giro é mantido fixo. A Figura 5.7 compara as componentes do quatérnion estimados com o quatérnion de atitude e o real. O objetivo de fazer uma simulação com velocidades angulares diferentes de zero, é verificar se em algum ponto da rotação existe singularidades. Como pode ser observado, a Figura 5.7 não apresenta nenhuma descontinuidade em qualquer uma das componentes do quatérnion.

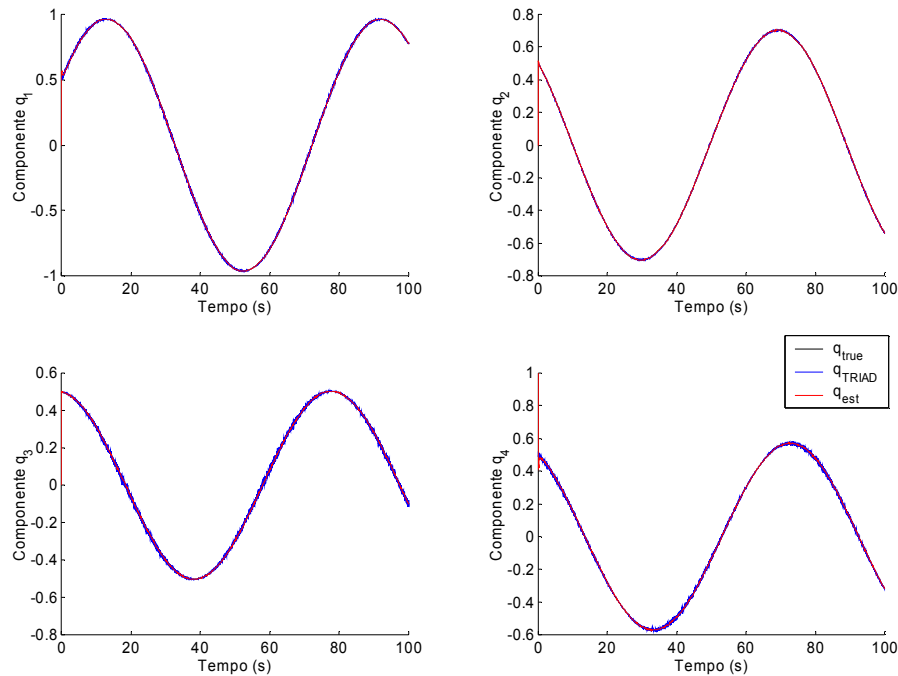


Figura 5.7 – componentes do quatérnion estimadas pelo filtro – simulação dinâmica

A Figura 5.8 mostra uma visão ampliada do ponto de convergência inicial. Nota-se que o tempo de convergência é de aproximadamente 0,5 segundo, no pior caso.

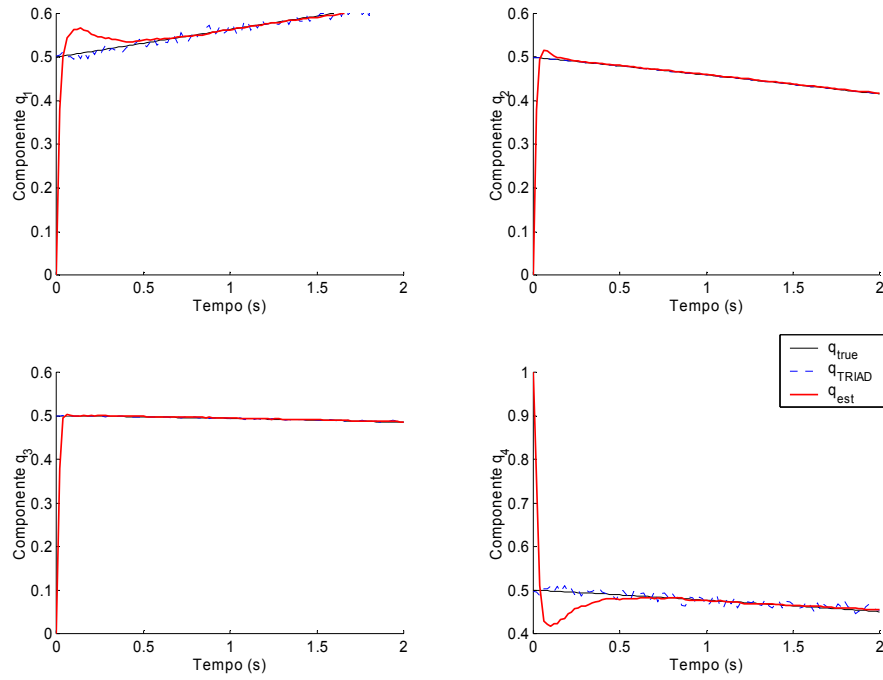


Figura 5.8 – Convergência inicial das componentes do quatérnion estimadas pelo filtro

A Figura 5.9 apresenta o erro das componentes estimadas do quatérnion dentro da faixa de desvio padrão esperado.

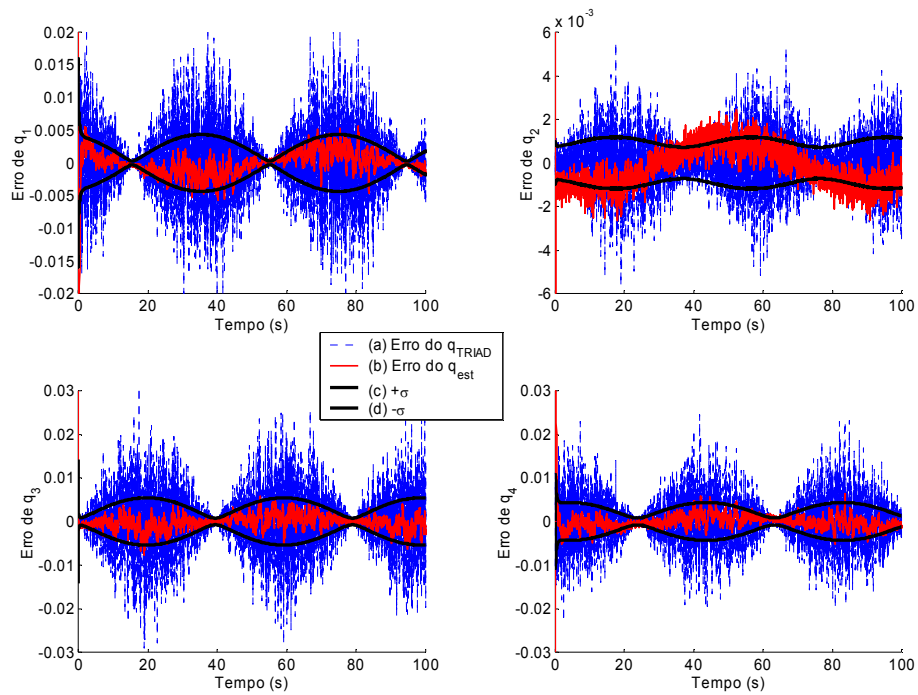


Figura 5.9 – Erros e desvio padrão das componentes do quatérnion durante uma rotação.

A Figura 10 mostra a estimação dos componentes do *bias* do giro.

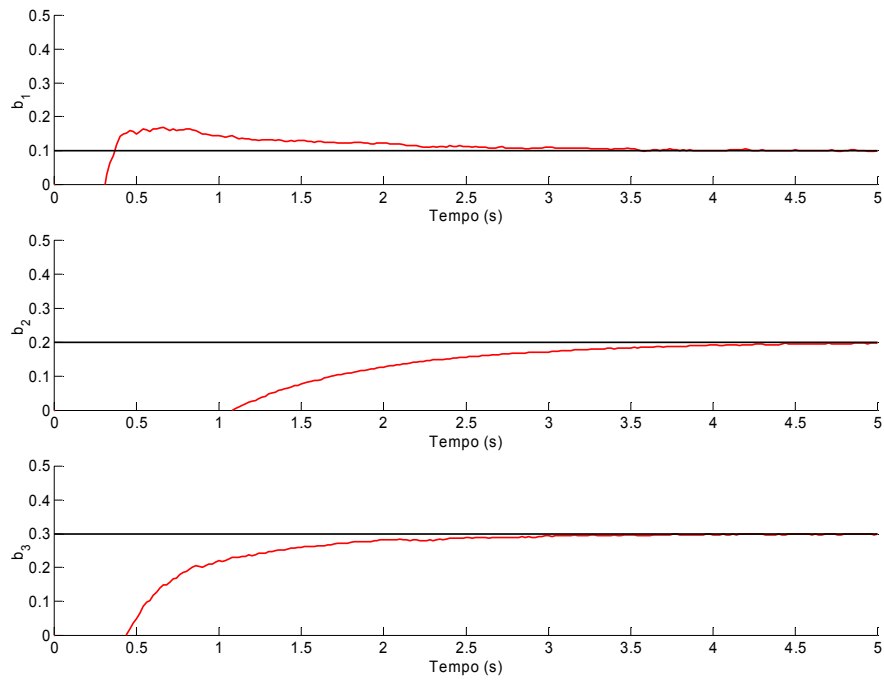


Figura 5.10 – Componentes do *bias* do giro estimadas pelo filtro durante simulação de rotação.

A Figura 5.11 trás o erro de cada componente do *bias* estimado na comparação com *bias* verdadeiro.

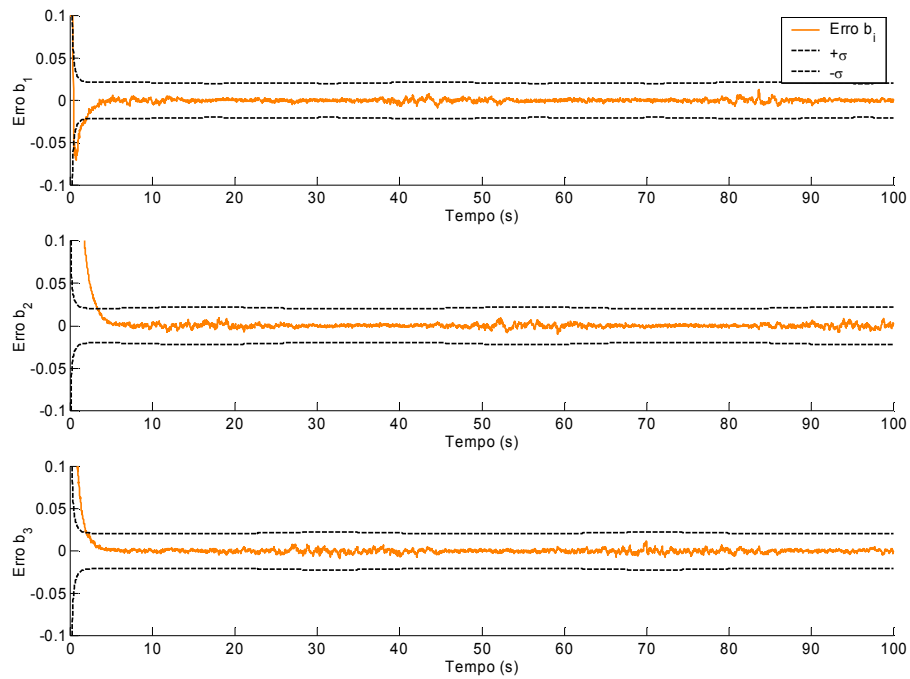


Figura 5.11 – Erros e desvio padrão das componentes do *bias* do giro.

Perante os erros dos estados estimados apresentados na Figura 5.9 e na Figura 5.11, pode-se concluir que o filtro está funcionando adequadamente para um *bias* fixo e velocidade de rotação constante.

5.11 – Recurso Gráfico de Visualização

5.11.1 – Introdução

Quando os componentes do quatérnion são analisados em um gráfico 2D pouco se pode compreender sobre o que representam aqueles quatro números ou a seqüência temporal deles. Um recurso desenvolvido para visualizar e compreender melhor o quatérnion foi uma simulação gráfica que apresenta a atitude de um objeto em tempo real de acordo com o quatérnion que lhe é dado.

Durante a simulação no Matlab os quatérnions verdadeiro, reconstruído (via TRIAD ou QUEST) e o estimado são salvos em arquivos. O software gráfico lê os arquivos com os quatérnions e apresenta objetos (cubos de faces coloridas) na tela. Para programar esse recurso são utilizados basicamente comandos de leitura de arquivo, um Timer, para ler arquivo com o mesmo período da simulação, e algumas funções da biblioteca gráfica OpenGL². Embora o software tenha sido escrito em C++ e compilado pelo software da Borland® para Windows®, esse recursos são disponíveis também a compiladores gratuitos.

5.11.2 – Fluxograma do programa

A Figura 5.12 trás um fluxograma da simulação gráfica. A simulação tem uma inicialização de parâmetros e uma interrupção periódica. No início são feitas algumas inicializações padrões para C++, inicializações do OpenGL e a inicialização do Timer, que neste caso deve ser configurado com o mesmo período da simulação para que ele rode em tempo-real. Também na inicialização é criado um objeto, neste caso um cubo de faces coloridas. Também são abertos os arquivos que contêm as seqüências de quatérnions que serão lidos um por vez a cada interrupção do Timer.

² www.opengl.org

Durante a interrupção, o quatérnion lido do primeiro arquivo (quatérnion verdadeiro) é utilizado para girar o objeto para a atitude correspondente. O objeto também é transladado para esquerda. Antes de mexer na matriz de informação do objeto uma cópia é feita na pilha para que se possa restaurar seus dados originais para o próximo posicionamento. Como a tela não é limpa entre cada posicionamento, são acumuladas três imagens até o último quatérnion. Ao fim da interrupção, se houver acabado um dos arquivos o programa desliga o Timer e pára.

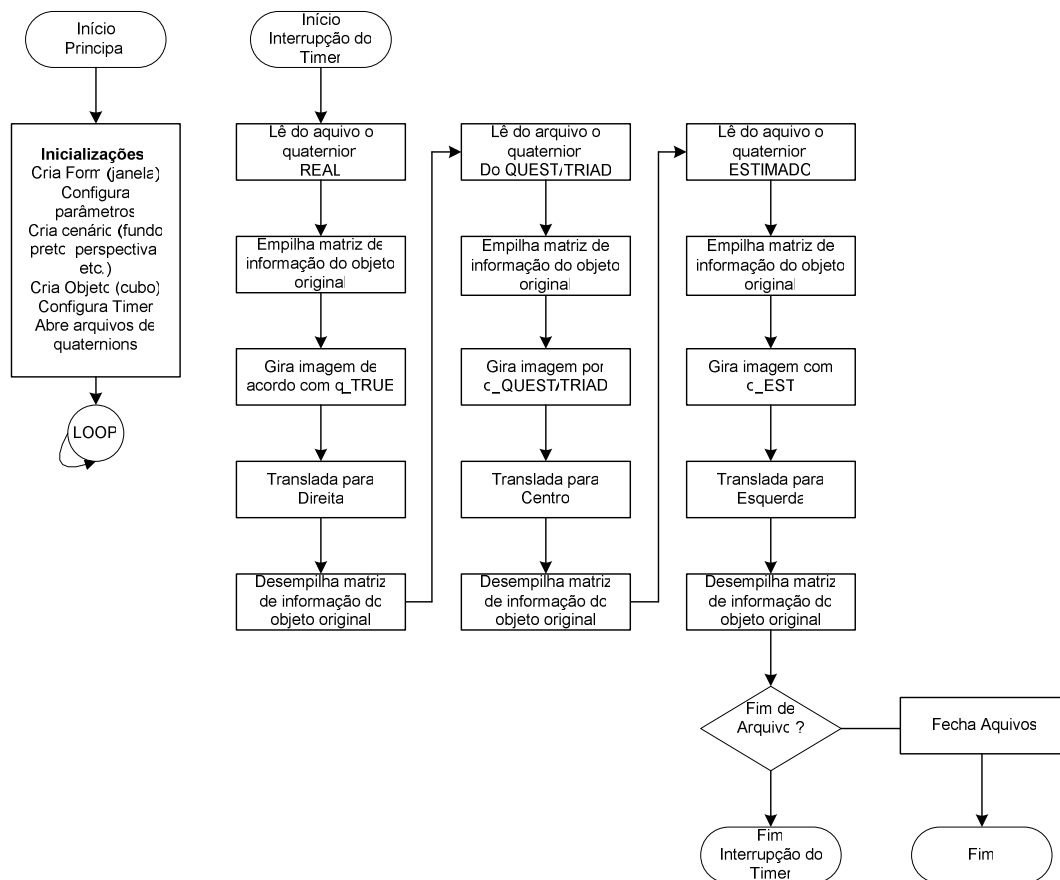


Figura 5.12 – Fluxograma da simulação gráfica

O Fluxograma da Figura 5.12 não é completo em todas as funções utilizadas, mas está coerente com o modo que a simulação ocorre. O código completo está lista do Anexo I os trechos mais importantes estão comentados.

5.11.3 – Visualização

Durante a simulação, a atitude verdadeira a reconstruída e a estimada são visualizadas respectivamente pelos três cubos coloridos que aparecem na tela. A Figura 5.13 ilustra a visualização em alguns instantes. Repare que a visão está em perspectiva.

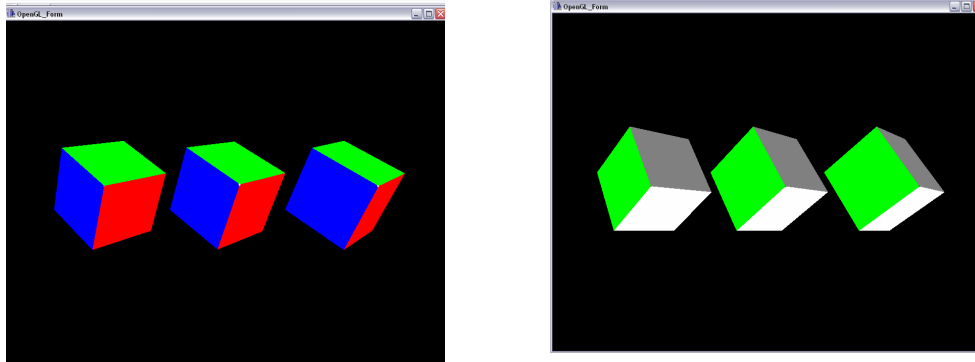


Figura 5.13 – Visualização gráfica da simulação da atitude

Problemas na convergência do filtro puderam ser facilmente identificados através desta útil ferramenta de visualização gráfica. Na Figura 5.14 pode-se observar um erro que fora cometido na codificação do filtro.

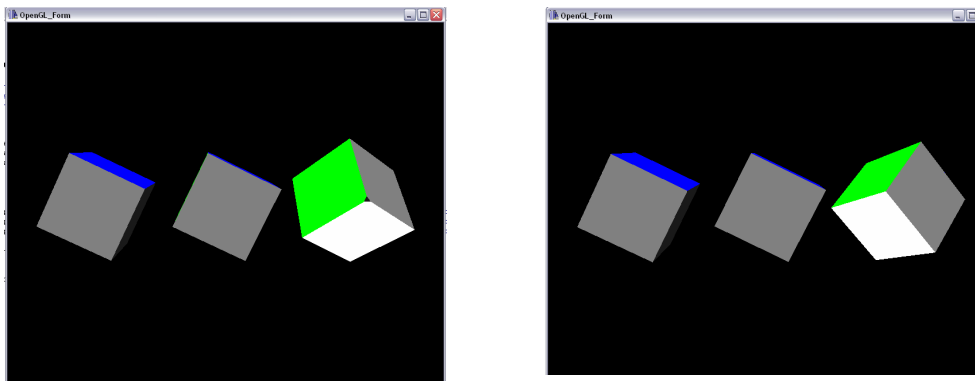


Figura 5.14 – Visualização gráfica da atitude – instantes 10 s e 11 s de uma simulação com erro na implementação.

5.12 – Considerações e Conclusões sobre a Filtragem e Simulação

A teoria sobre filtragem de Kalman e a estatística envolvida é bastante complexa. Alguns conceitos foram exposto no texto até aqui para aplicar o FK como uma ferramenta e não para exaurir todas as possibilidades.

Uma idéia original adicionada foi a visualização gráfica da atitude com a possibilidade de se fazer uma comparação visual entre o quatérnion de atitude real, o reconstruído pelo TRIAD ou QUEST, e o estimado.

Capítulo 6 – Implementação do Determinador de Atitude: projeto, programação e testes

6.1 – Introdução

Neste capítulo será descrito o projeto de um hardware de determinação de atitude que condiciona, adquire e processa sinais provenientes de acelerômetros, girômetros e magnetômetros.

Um hardware de determinação de atitude pode ser dividido em partes como mostra o diagrama da Figura 6.1. O sistema é composto por sensores, por um sistema de digitalização dos dados, pelo núcleo de processamento e pela interface de saída de dados, ou seja, a interface de saída da atitude.

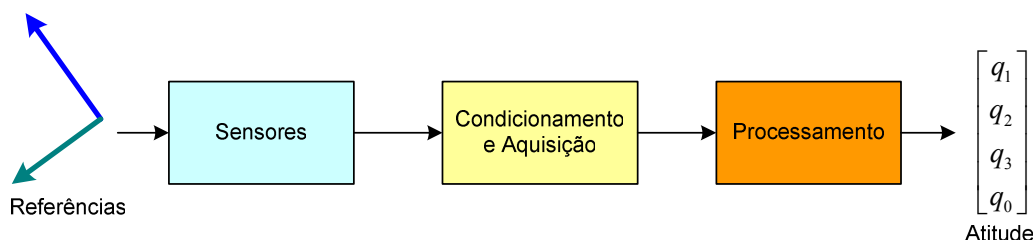


Figura 6.1 – Diagrama básico de um sistema de determinação de atitude.

O primeiro bloco, que é composto por sensores, faz a medida das grandezas físicas e as convertem para grandezas elétricas. Os sinais elétricos proveniente dos sensores passam por uma etapa de amplificação e filtragem antes do estágio de conversão AD (analógico-digital). O condicionamento do sinal juntamente com a conversão AD formam o bloco de digitalização. Os sinais, já digitalizados, são tratados (filtragem digital e redução da taxa de amostragem) e passados ao algoritmo de

determinação da atitude e ao filtro de Kalman. A saída do algoritmo fornece o quaternion ou matriz de atitude que finalmente é disponibilizado por uma interface de comunicação. Na próxima seção serão feitas algumas considerações sobre os elementos que compõe o hardware.

6.2 – Considerações sobre os Elementos de Hardware

Antes de projetar um hardware como este é necessário ficar atento às especificações do projeto e ainda levar em conta a disponibilidade de componentes, assim como o custo e tempo para adquiri-los. Os componentes que definem o projeto do determinador de atitude são: sensores, amplificadores, microcontrolador, conversores analógico-digital, interfaces, componentes discretos e conectores. A seguir uma breve discussão sobre parâmetros que levam escolha de cada um deles.

6.2.1 – Sensores

Os sensores são elementos geralmente caros. Os sensores mecânicos aplicados à atitude são caríssimos e geralmente de acesso restrito, por serem de especificação militar. Sensores como girômetros de poucos graus por hora de deriva e sensores de aceleração com ruído na ordem de micro-g's são alguns exemplos de sensores que custam milhares de dólares e são de difícil aquisição. Sensores de baixo custo são geralmente sensores MEMS. Chegam a custar dezenas dólares, alguns são fáceis de adquirir, mas apresentam resolução limitada e ruído característico elevado. Embora a tecnologia MEMS possua certas limitações, os sensores apresentaram uma grande melhoria de desempenho na última década (1995-2005). Um exemplo são os sensores de aceleração, que saltaram de 1 mg de resolução para menos de 100 micro-g em pouco mais de 5 anos.

Em um primeiro projeto, o uso de sensores MEMS é adequado e pode atingir resultados satisfatórios para diversas aplicações [GEB04].

6.2.2 – Microcontroladores

São em geral baratos e de fácil acesso. A especificação geralmente é feita comparando a arquitetura, velocidade, quantidade de memória de programa e memória

RAM disponível, ferramentas de depuração para desenvolvimento do software e periféricos disponíveis. Existem dezenas de fabricantes e centenas de famílias de microcontroladores que poderiam ser utilizados, mas quando feita uma análise cautelosa fica difícil encontrar algum que atenda a todos os requisitos simultaneamente. Para resolver ou emular operações em ponto flutuante com certa eficiência, é necessário que o núcleo do processador seja preferencialmente de 16 bits. Como várias matrizes ficarão armazenadas em memória durante os cálculos, pelo menos 2KB de memória RAM são necessários. Como os algoritmos são diversos, e sua codificação está constantemente sofrendo modificações e ainda não se conhece com exatidão os resultados, é essencial que o fabricante ou terceiros tenham à disposição ferramentas para gravação e depuração, preferencialmente gratuitas.

Uma facilidade que também deve ser levada em conta são os periféricos disponíveis no chip do microcontrolador. Por exemplo, se um microcontrolador possui periférico de interface serial SPI (*Serial Paralell Inteface* – Interface Serial Paralela), I2C (*Inter-Integrated Circuit*) ou SCI (*Serial Communication Interface*) os mesmos não precisarão ser implementados por software, resultando assim em uma redução no tamanho do código e melhorando a confiabilidade do programa. Os barramentos seriais, assim conhecidos porque apresentam poucas linhas de interface com os periféricos externos, ajudam a reduzir o número de conexões e consequentemente o tamanho da placa.

6.2.3 – Conversores analógico-digital

Uma vez que uma das tarefas do sistema é adquirir dados dos sensores, os mesmos terão de ser convertidos para o formato digital. O ADC (*analog to digital converter* – conversor analógico-digital) é especificado principalmente pela quantidade de bits efetivos em que é capaz de resolver o sinal analógico lido em função das referências máxima e mínima impostas a ele. Outras características são: arquitetura de conversão e o tempo de conversão, número de canais, impedância de entrada, linearidade, distorção, qualidade do circuito amostrador (*sample-and-hold*), o tipo da interface de saída dos dados (serial ou paralelo), dentre outras. Considerando o uso de sensores MEMS, os conversores devem possuir resolução entre 12 a 16 bits.

O ideal seria o uso de um microcontrolador que já possuísse um periférico de conversão AD com características razoáveis, simplificando os circuitos externos.

6.2.4 – Amplificadores

Uma técnica muitas vezes empregada na tentativa de evitar interferências de outros componentes no sinal elétrico proveniente do sensor é não utilizar nenhum componente, ou o mínimo possível de componentes, entre o sinal do sensor e o ADC. Os sensores MEMS geralmente possuem resposta em frequência limitada a poucos kHz, o que torna o próprio sensor um filtro *anti-aliasing*. Em casos onde o sinal disponível na saída do sensor é muito baixo, pode ser vantajoso amplificar o sinal, apenas para aumentar a faixa dinâmica da conversão AD, mas tendo a consciência de que é necessária a utilização de amplificadores e componentes discretos de boa qualidade para evitar a degradação do sinal.

Os amplificadores devem ser estáveis em temperatura e possuir deriva mínima. Os componentes discretos devem ser estáveis em temperatura, os resistores devem ser utilizados em menor número possível para evitar o ruído térmico e os capacitores devem ser preferivelmente de tântalo por possuírem baixa resistência série e baixa indutância. Muito cuidado deve ser tomado com a fonte de alimentação do circuito. A fonte analógica deve ser estável e separada da fonte digital. Os terras do circuito analógico e digital também devem ser separados.

Geralmente é possível atender a estas especificações com um baixo custo.

6.2.5 – Memória de dados não-voláteis

Geralmente microcontroladores de memória Flash permitem a gravação de dados na memória de programa, mas esse procedimento demanda tempo e causa uma pausa indesejável na execução do programa. Devido à elevada carga de processamento demandada, o uso de uma memória externa para gravação de dados não-voláteis faz parte da especificação do projeto.

As memórias de alta performance são paralelas, ou seja, apresentam pinos de endereço e dados, geralmente multiplexados, para leitura e gravação de dados. Muitas linhas de interface acarretam complexidade para realizar as ligações no design do circuito impresso e necessidade de *buffers* externos para demultiplexar as linhas de dados e endereços. Pelo fato da máxima taxa de gravação não ser superior a 1200 B/s,

uma memória de dados com barramento serial seria mais conveniente. O custo não é significativo.

6.2.6 – Interfaces de comunicação

Pelo fato deste hardware ser dedicado para desenvolvimento, é indispensável que ele apresente interfaces diversas para comunicação, inclusive interface com o usuário. As interfaces com o usuário geralmente são botões e LEDs. A interface com o computador é geralmente feita pela serial, ou SCI, conhecida no computador como COM.

Alguns microcontroladores possuem uma interface de gravação e depuração que permite ao desenvolvedor do software criar e testar simultaneamente. Nos microcontroladores da família HCS da Freescale®, por exemplo, essa interface é conhecida como BDM (*Background Debug Mode* – Modo de Depuração de Fundo). Essa interface dedicada usa apenas um pino do processador e não exige componentes adicionais na placa. A depuração em tempo real necessita de um hardware¹ para interfacear com o computador por uma interface de alta velocidade como USB ou rede *Ethernet*.

6.2.7 – Componentes discretos

Esses componentes são de maneira geral os resistores, capacitores, indutores, transistores, MOSFETS, LED e outros do gênero. Os principais componentes discretos são aqueles capazes de interferir no funcionamento do circuito e consequentemente na sua precisão. Estes componentes são os resistores e capacitores porque como são utilizados nos amplificadores e nos filtros, se eles apresentarem alto ruído térmico e deriva térmica, podem prejudicar a qualidade das medidas. Outro fator que deve ser levado em conta é o tamanho destes componentes. Na última década iniciou-se uma migração irreversível para o uso de componentes discretos SMD – (*Surface Mounting Devices* – componentes para montagem em superfície) que possibilitam uma maior densidade de componentes na montagem, pois, são muito menores. O tamanho desses componentes influi no desempenho porque permite que os circuitos integrados fiquem

¹ No caso das famílias HCS08 e HCS12 da Freescale, o hardware de interface mais popular para gravação e depuração é o Cyclone® da Pemicro: www.pemicro.com

mais próximos uns dos outros reduzindo a área dos laços de ligação das trilhas e consequentemente reduzindo a adição de ruído e interferência eletromagnética. Os SMD são de baixo custo e de fácil acesso, portanto são preferenciais ao projeto, a não ser em casos em que se necessite maior dissipação de potência.

6.2.8 – Conectores

Um erro comum quando se especifica um projeto é não escolher corretamente os conectores que o sistema deve possuir. Quando já existir um conector padrão para algum tipo de interface, ou seja, existir comercialmente cabos prontos para conexão com aquela interface, eles devem ser usados. A SCI é um bom exemplo para este caso. Os demais conectores necessários devem ser pinos paralelos, tipo IDE.

6.2.9 – Considerações finais sobre a escolha dos componentes

Em um hardware de determinação de atitude, os sensores afetam diretamente a saída desejada por não possuírem uma função de transferência totalmente linear e também por sofrerem variação com a temperatura e diversos outros fatores. Por isso, todos os outros componentes que interfaceam diretamente com os sensores têm de ser de qualidade superior. Os sensores mais caros possuem cartas de calibração e modelos de alta ordem para não linearidades e correção pela temperatura. Os demais componentes não. Portanto, os únicos componentes que podem apresentar variações são os sensores, pois essas variações podem ser minimizadas pelo processamento do sinal perante seu modelo.

6.3 – Diagrama de Blocos Ampliado – Visão geral do Sistema

Como pode ser visto na Figura 6.2, o sistema é composto por tríades de sensores, amplificadores e filtros, conversores analógico-digital, memória de dados, diversas interfaces como IrDA (*Infrared Data Access* – Interface de Dados por Infravermelho), SCI, SPI, PWM (*Pulse Width Modulation* – Modulação por Largura de Pulso), uma entrada de simulação denominada SIMUCAN², uma interface especial de

² O SIMUCAN é um módulo de simulação de canais analógicos; uma espécie de gerador de funções capaz de gerar um sinal analógico de 0 a 5 V na saída de até 12 canais de conversores digital-analógicos

depuração de software denominada BMD e uma entrada de alimentação com reguladores de tensão distintos para sinais analógicos e digitais.

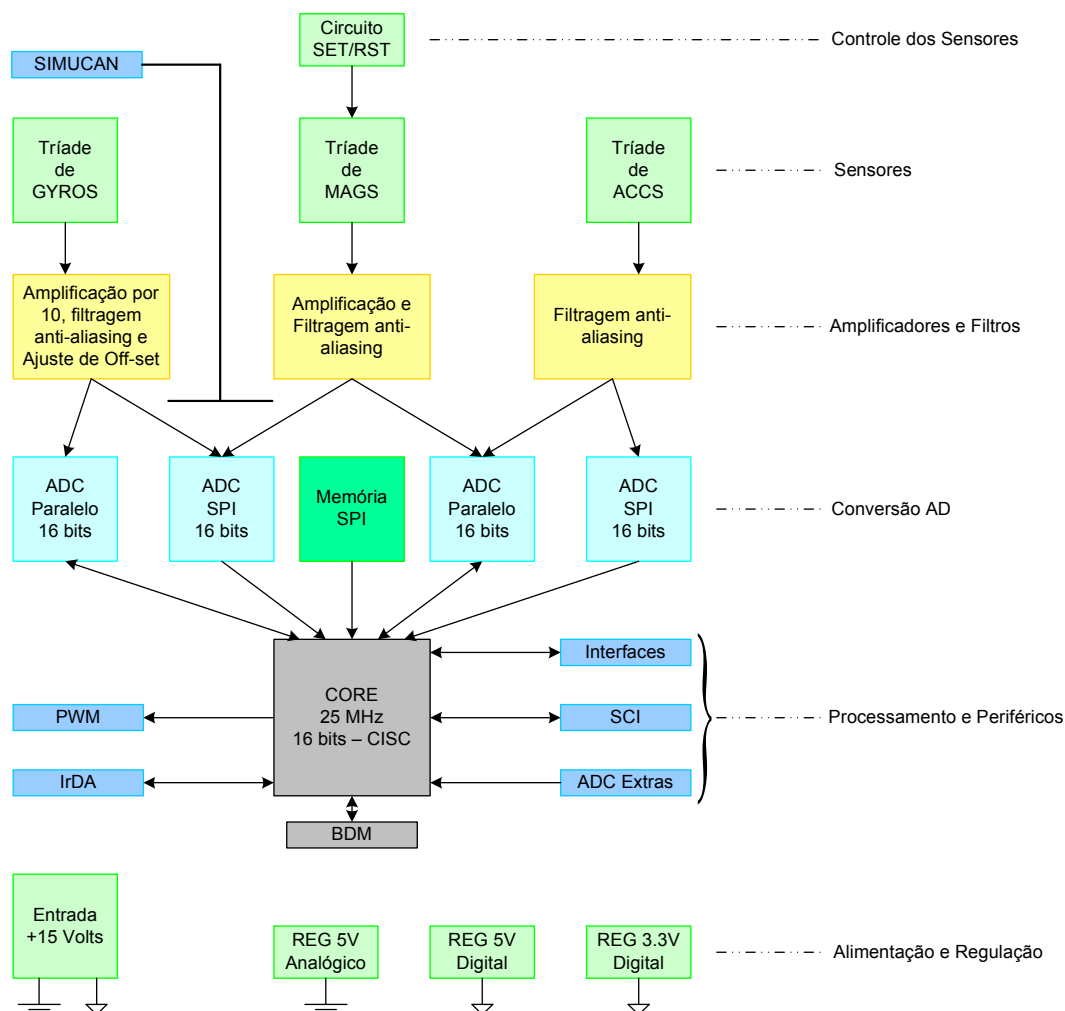


Figura 6.2 – Diagrama de blocos do hardware de determinação da atitude

Para preencher cada bloco do circuito acima foram levadas em consideração as especificações da secção anterior. A seguir serão descritos os componentes escolhidos para cada bloco, as características fornecidas pelo fabricante e sua funcionalidade no circuito. No anexo III estão as folhas de rosto das folhas de dados de todos os componentes empregados.

6.4 – Os Sensores

simultâneos a uma taxa de até 50 Hz. Ele é controlado por uma interface serial simplificada e pode ser comandado pela interface serial através do HyperTerminal® ou por qualquer software de controle que siga o protocolo. Obs.: Esta ferramenta foi desenvolvida para testes dos ADCs de vários canais.

Os sensores utilizados no hardware de desenvolvimento, como pode ser visto na Figura 6.2, são seis acelerômetros ADXL202, três girômetros GYROSTAR e três magnetômetros HMC2003. Todos são sensores MENS, ou seja, são sensores em estado sólido, não há partes mecânicas móveis, apenas micro-mecânica. Todos estão posicionados em forma de tríade e mantendo ortogonalidade³ entre os eixos. Os fabricantes fornecem especificações dos seus sensores, alguns fornecem folha de dados mais completa que outros, por isso é necessário estudar cada um deles para se levantar características reais de funcionamento, principalmente quanto ao ruído do sinal de saída.

6.4.1 – Sensor de aceleração ADXL202

A. Descrição geral do sensor ADXL202

Esse sensor fabricado pela ANALOG DEVICES [ADX05] é provavelmente o sensor MEMS mais popular utilizado em aplicações de baixo custo. Sua resolução não chega a 1 mg, mas isso não impede que seja utilizado em diversas aplicações, inclusive aplicações de navegação e determinação de atitude [GEB04]. Este sensor tem dimensões mínimas da ordem de $0,5 \text{ cm}^2$, como é possível observar na Figura 6.3.



Figura 6.3 – Sensor de aceleração ADXL202 – Tamanho

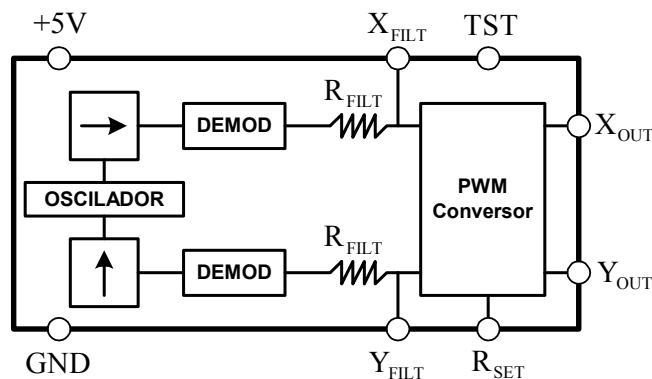


Figura 6.4 – Sensor de aceleração ADXL202 – diagrama interno

³ É possível que haja pequeno desalinhamento entre os eixos X, Y e Z.

O princípio de funcionamento deste sensor se baseia em medir o deslocamento de uma massa m presa a uma lâmina de elasticidade k , tal que uma aceleração externa seja compensada pelo deslocamento da massa em uma certa distância x . Se a massa é uma placa móvel de área A , posicionada entre duas placas de área também A então é possível medir a capacitância diferencial que varia de acordo com a posição da placa central. Aplicando um sinal alternado de amplitude v e de frequência f na placas imóveis, a tensão medida na placa central será proporcional ao deslocamento sofrido por ela, que por sua vez é proporcional à aceleração.

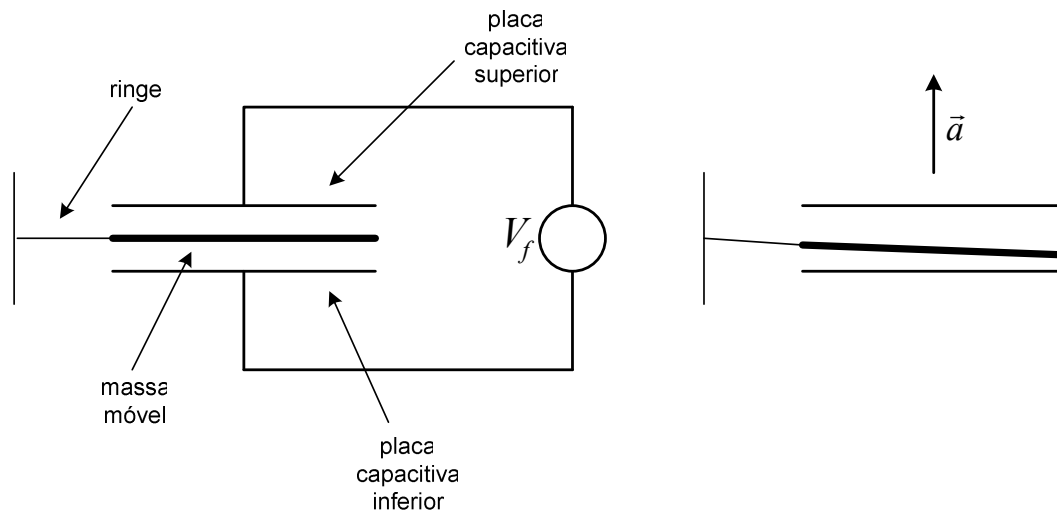


Figura 6.5 – Acelerômetros de capacitância variável por deslocamento

O Sensor ADXL possui encapsulado a micro-mecânica, o oscilador modulador e também os demoduladores para os 2 eixos sensitivos. O sensor ADXL202E possui também lógica para modular o sinal analógico para sinal PWM.

As principais características desse sensor são dadas na Tabela 6.1. Foram omitidos os parâmetros relacionados ao PWM.

Tabela 6.1 – Características principais do sensor ADXL202E

Parâmetro	Min	Max	Típico	Unidade
Faixa de Medida	-2	+2		g
Saída	0,5	4,5		V
Não-linearidade			0,2	%
Erro de alinhamento interno entre eixo			±1	Graus

verdadeiro e eixo de medida				
Erro de alinhamento entre X e Y			±0,01	Graus
Sensibilidade Cruzada			±2	%
Sensibilidade $V_{DD} = 5\text{ V}$	250	375	312	mV/g
Deriva com a temperatura (referência em 25°)			±0,5	%
Bias (0 g)	2,0	3,0	2,5	V
Densidade espectral de potência do ruído @ 25 °C		1000	200	$\mu\text{g}/\sqrt{\text{Hz}}$ rms
Resposta em Frequência			6	kHz
Ressonância do Sensor			10	kHz
Tolerância $R_{FILT} 32\text{k}\Omega$			±15	%

Analisando os dados da Tabela 6.1 é possível verificar que, se a função do acelerômetro é medir a gravidade, então a excursão do sinal será entre 2,1 e 2,9 V. Considerando que é possível existir acelerações que se somam com a gravidade, então essa variação pode ser ainda maior, portanto a faixa dinâmica do sinal de saída do sensor é apropriada à aplicação. Isso praticamente exige o uso de amplificadores para aumentar a faixa dinâmica. O sinal deste acelerômetro pode ser digitalizado diretamente, bastando apenas realizar a filtragem *anti-aliasing*, que de acordo com a resposta em frequência do sensor, deve ser em 6 kHz ou menos. Outro fator importante é a densidade de ruído, que para este sensor é tipicamente $200\text{ }\mu\text{g}/\sqrt{\text{Hz}}$, mas pode chegar até $1000\text{ }\mu\text{g}/\sqrt{\text{Hz}}$. A relação entre o ruído de saída e a densidade de ruído é dada por [ADX05]

$$N_{\text{rms}} = d_{\text{rms}} \sqrt{BW \times 1,6} \quad (6.1)$$

onde N_{rms} é o ruído da saída, BW é largura da banda, 1,6 é um fator devido a banda de guarda do filtro de primeira ordem e d_{rms} a densidade de ruído.

Por exemplo, se a banda do sinal for 500 Hz, tem-se

$$N_{\text{rms}} = 200 \times 10^{-6} \sqrt{500 \times 1,6} \approx 5,6\text{ mg} \quad (6.2)$$

Esse ruído, a princípio elevado, será reduzido por filtragem digital, já que a banda requerida para o determinador de atitude não é superior a 50 Hz. Assim o ruído de saída após nova filtragem será

$$N_{\text{rms}} = 200 \times 10^{-6} \sqrt{50 \times 1,6} \approx 1,8 \text{ mg} \quad (6.3)$$

B. Experimentação do sensor de aceleração

Uma simples montagem foi realizada para medir o ruído do sinal do sensor de aceleração. Esta montagem foi realizada no intuito de estimar o ruído intrínseco do acelerômetro ADXL202 e compará-lo com os dados do fabricante. A Figura 6.6 mostra a configuração mínima para ligar o sensor de aceleração.

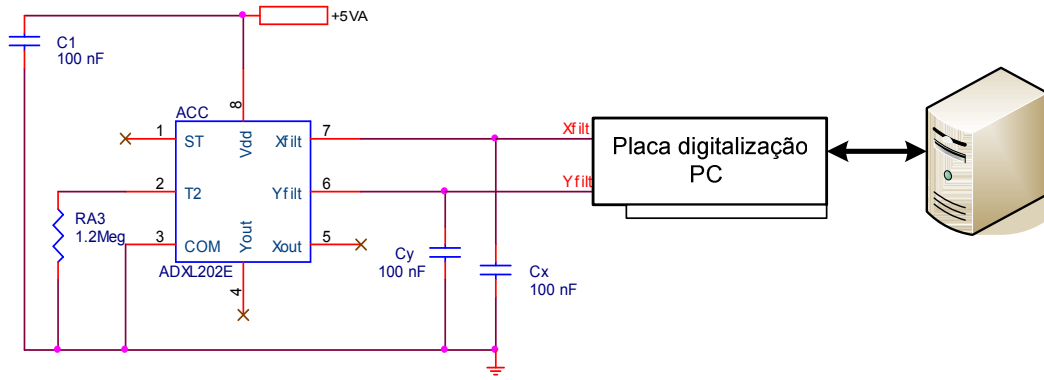


Figura 6.6 – Configuração mínima para o sensor de aceleração. Experiência para verificar o ruído de saída.

A resistência de saída R_{FILT} , tanto do canal X quanto do canal Y , é de cerca de 32 k Ω . Os capacitores C_Y e C_X são calculados de forma estabelecer uma frequência de corte f_c . Considerando, para fins de teste, $f_c = 50 \text{ Hz}$, tem-se

$$C = \frac{1}{2\pi f_c R_{\text{FILT}}} = \frac{1}{2\pi \cdot 50 \cdot 32 \cdot 10^3} = 99,4 \text{ nF} \quad (6.4)$$

O Capacitor C_1 é apenas um capacitor de desacoplamento, recomendado pelo fabricante. O Resistor R_A controla o período do PWM para as saídas moduladas X_{OUT} e Y_{OUT} , que neste teste não serão utilizadas.

Para realizar a aquisição de dados fora utilizada uma placa de digitalização PCI da NI (National Instruments). Um instrumento virtual (VI) de aquisição pode ser criado no LabView® para realizar o acesso ao hardware de aquisição e gravação dos dados. Uma outra maneira é usar o pacote de aquisição de dado do Matlab. Uma simples, mas útil função de aquisição foi escrita para ler os dados a uma dada taxa. O código da função é listado e comentado a seguir.

```
%=== rotina de aquisição de dados da NIDAQ via Matlab ===%
function [Y,FS] = LeCanais(tempo,taxa,vetcanais,nomearq)
daqreset;
ai = analoginput('nidaq', 0);
[Y,FS] = daqrecord('nidaq', ai, tempo, taxa, vetcanais)
save nomearq Y -ascii
%=== fim
```

A Figura 6.7 mostra o gráfico dos dados de uma das aquisições. Essa aquisição foi feita realizando movimentos para que o sensor medisse +g e -g. O objetivo principal é levantar a característica do ruído, ou seja, seu nível de ruído RMS. O código a seguir apresenta o procedimento para retirar o *off-set* das medidas feitas nos canais X e Y . Pelo fato de ter sido utilizado o fator médio de escala da folha de dados do componente, o módulo de g não ficou em 9,8.

```
%=== Rotina para retirar offset dos dados ===%
%=== Plot da figura &*
load dados_acc.txt % Arquivo Texto salvo na aquisição
acc = dados_acc; % acc passa a ser o vetor de manipulação do dados
fs = 1000; % constante fs: frequência de amostragem utilizada
s = 25; % constante s: fator de escala médio
t = (1:6e4)/1000; % constante vetor tempo t: em segundos
acc1_med = mean(acc(1:5000,1)); % média dos 5 primeiros segundos do
canal X
acc2_med = mean(acc(1:5000,2)); % média dos 5 primeiros segundos do
canal Y
acc1_L = acc(:,1) - acc1_med; % tirando a média (offset) dos dados
```

```

acc2_L = acc(:,2) - acc2_med;      % tirando a média (offset) dos dados
acc1_LS = acc1_L * s;              % aplicando o fator de escala para
m/s2 em X
acc2_LS = acc2_L * s;              % aplicando o fator de escala para
m/s2 em Y
plot(t, acc1_LS);                  % plot somente do canal X
%=== Fim

```

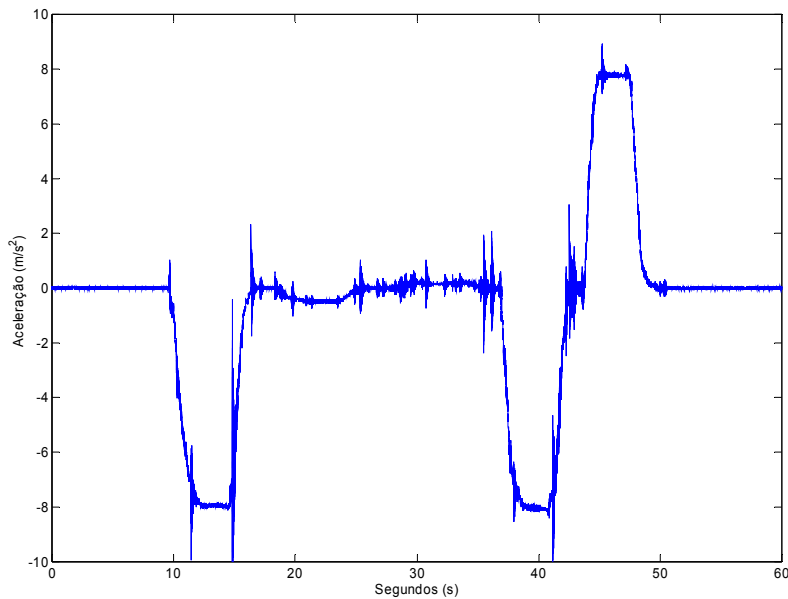


Figura 6.7 – Aquisição da aceleração via LabView. $f_s = 1$ kHz .

O ruído RMS do sinal do sensor pode ser estimado nos primeiros 5 segundos, considerando que o acelerômetro estava em repouso sobre a bancada de ensaios.

Sendo \mathbf{v} um vetor de dados contendo os N aquisições, o ruído RMS, ou o desvio padrão do sinal, será dado por

$$\sigma = \sqrt{\frac{\mathbf{v}^T \mathbf{v}}{N}} \quad (6.5)$$

Assim, para os primeiros 5 segundos ou 5000 aquisições, a simples linha de código realiza o cálculo desejado:

```

% === Procedimento para cálculo do ruído rms, ou desvio padrão
STD = sqrt((acc1_LS(1:5000)' * acc1_LS(1:5000))/5000);

```

```
% === Fim
```

$$\sigma = 21 \text{ mm/s}^2 = 2,1 \text{ mg} \quad (6.6)$$

O valor encontrado é menos que os 5,6 mg sugeridos pela equação dada pelo fabricante quando considerada uma banda de 500 Hz. Mas, se o sinal foi adquirido a uma taxa de 1000 Hz, ou seja, contém componentes de até 500 Hz, mesmo com o filtro de saída em 50 Hz, e ainda considerando o ruído de alta frequência que foi inserido pelo próprio conversor AD do sistema de digitalização, é possível melhorar as características do sinal aplicado uma filtragem digital de alta ordem, eliminando quase que completamente as componentes acima de 50 Hz.

Aplicando filtragem digital Butterworth de quarta ordem obtém-se agora um sinal como da Figura 6.8. O código no MatLab é simples, utilizando apenas a função `butter`:

```
% === Filtro Digital Butterworth de 4ª ordem no MatLab
BandaW = 50/(fs/2); % Definição da banda normalizada
[B,A] = butter(4,BandaW,'low'); % Uso da função 'butter' para
calcular o vetor de coeficientes A e B para uso posterior da função
'filter'.
acc1_LSF = filter(B,A,acc1_LS); % Filtrando o sinal acc1_LS e
armazenando o resultado vetor acc1_LSF
% === Fim do procedimento de filtragem
```

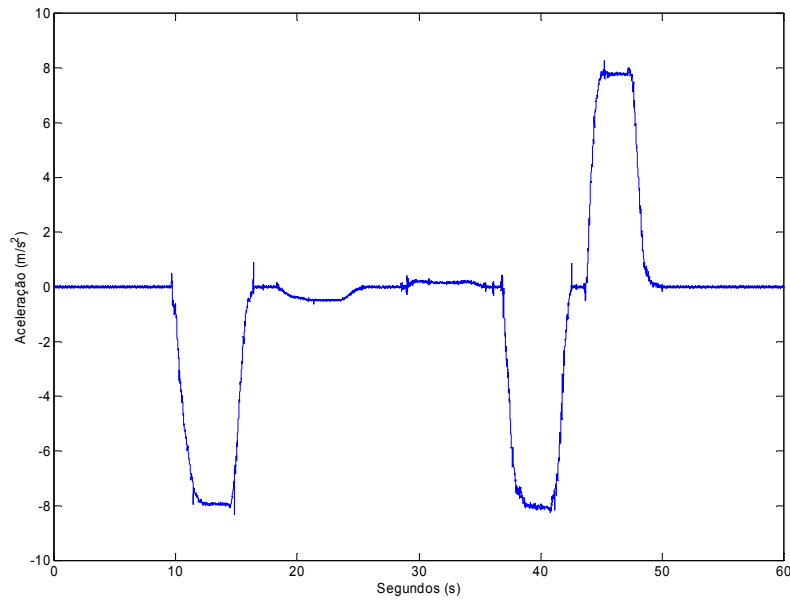


Figura 6.8 – Aquisição da aceleração via LabView. $f_s = 1$ kHz com posterior filtragem digital em $f_c = 1$ kHz .

Realizando novamente o cálculo do ruído RMS para as primeiras 5000 amostras, chega-se a

$$\sigma = 8,5 \text{ mm/s}^2 = 0,0008 \text{ g} = 0,8 \text{ mg} \quad (6.7)$$

Esse nível de ruído está abaixo dos 1,8 mg sugeridos pela equação do fabricante. Um ruído de 0,8 mg é equivalente em graus a

$$\delta\theta = \delta \cos^{-1} \frac{0,8\text{mg}_{\text{rms}}}{1\text{g}} \approx \delta \sin^{-1} \frac{0,8\text{mg}_{\text{rms}}}{1\text{g}} \approx 0,1^\circ \quad (6.8)$$

6.4.2 – Sensor GYROSTAR

A. Descrição e princípio de funcionamento

O sensor GYROSTAR 03JA [GYR04] da Murata® é um sensor de velocidade angular que utiliza o fenômeno da força de Coriolis que é gerada quando uma velocidade angular é aplicada a um corpo oscilante.

Uma configuração interna típica são três transdutores piezelétricos montados sobre três lados de um prisma triangular. Se um dos transdutores é excitado em sua frequência de ressonância as vibrações são sentidas pelos dois outros transdutores em igual intensidade. Quando o prisma é rodado em torno do seu eixo longitudinal, a força resultante de Coriolis causará uma suave diferença na intensidade de vibração medida nos dois transdutores. A diferença da tensão medida é uma saída que varia linearmente com a taxa de variação angular.



Figura 6.9 – Sensores GYROSTAR 03JA

A Tabela 6.2 apresenta as características do sensor GYROSTAR 03JA.

Pela tabela de dados do fabricante é possível observar que a excursão do sinal de saída será de

$$\begin{aligned}\Delta v_s &= s\Delta\omega \\ \Delta v_s &= 0,67 \times (300 - (-300)) = 400 \text{ mV} = \pm 200 \text{ mV}\end{aligned}\tag{6.9}$$

Onde s é a sensibilidade do sensor e $\Delta\omega$ faixa de velocidades angulares. Essa baixa excursão indica a necessidade de tratamento do sinal. Além disso, o nível de tensão da saída em $\omega = 0 \text{ rad/s}$ é 1,35 V, fora do padrão comum para aplicações em 5 V, onde o zero está em 2,5 V.

Tabela 6.2 – Características do sensor GYROSTAR 03JA

Componente	ENC – 03JA	Unidade
Tensão de Alimentação	+2,7 a +5,5	V
Consumo Máximo	5	mA
Máxima velocidade angular	± 300	°/s

Saída $\omega = 0$ °/s	1,35	V
Fator de Escala	0,67	mV/°/s
Coefficiente de Temperatura do fator de escala	± 20	%
Linearidade	± 5	%FS
Resposta em Frequência	50	Hz

Outro fator a ser levado em consideração é que o sensor tem uma baixa resposta em frequência, 50 Hz. Esse fator é limitante, mas isso não é um problema considerável para a aplicação pretendida.

O coeficiente de temperatura é um grande problema, pois afeta o ganho e nível de zero do sensor.

O fabricante não fornece parâmetros de ruído ou algo do gênero, portanto é necessário realizar algumas medidas para levantar parâmetros de deriva, pelo menos a deriva em repouso, e o ruído RMS.

B. Experimentação do Girômetro – sem Amplificação do Sinal

O girômetro 03JA é um componente de ligação extremamente simples. Ele apresenta apenas quatro pinos de saída. Dois de alimentação, um de saída do sinal e o outro com uma tensão de referência. O esquema da Figura 6.10 foi o primeiro montado para realizar as medidas.

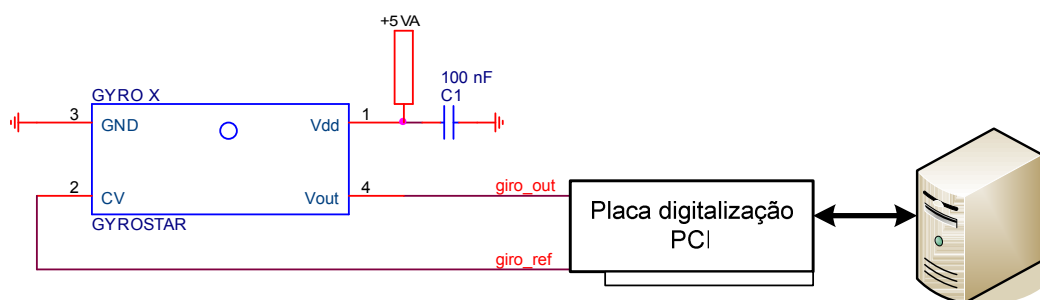


Figura 6.10 – Experiência com girômetro para verificar o ruído de saída e deriva.

O capacitor C_1 é apenas um desacoplamento para tensão de alimentação.

O mesmo VI da Figura 6.6 foi utilizado para realizar a aquisição dos sinais provenientes do sensor. Os sinais de interesse são a referência de tensão e a saída de

tensão proporcional à velocidade angular. Quando o giro estiver em repouso espera-se que ambas sejam iguais, a menos de um *off-set*. O objetivo de se fazer medidas com o sensor em repouso é verificar quão intenso é o ruído RMS e a deriva do *off-set*.

A Figura 6.11 traz os dados adquiridos.

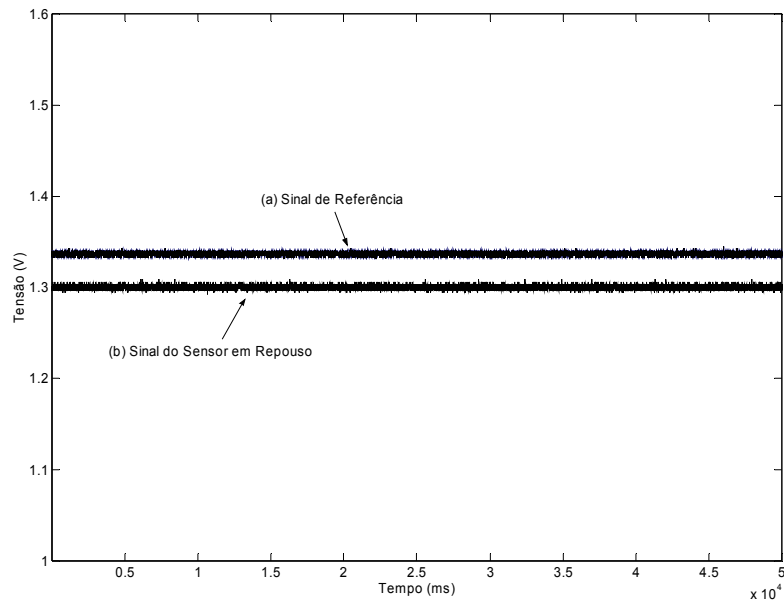


Figura 6.11 – Dados adquiridos do sensor Murata 3JA. (a) Tensão de referência (b) Tensão de saída como sensor em repouso

Convertendo os sinais de tensão de saída do sensor para velocidade angular, considerando o fator de escala do fabricante, a tensão do sinal do sensor da Figura 6.11 se torna os dados na Figura 6.12.

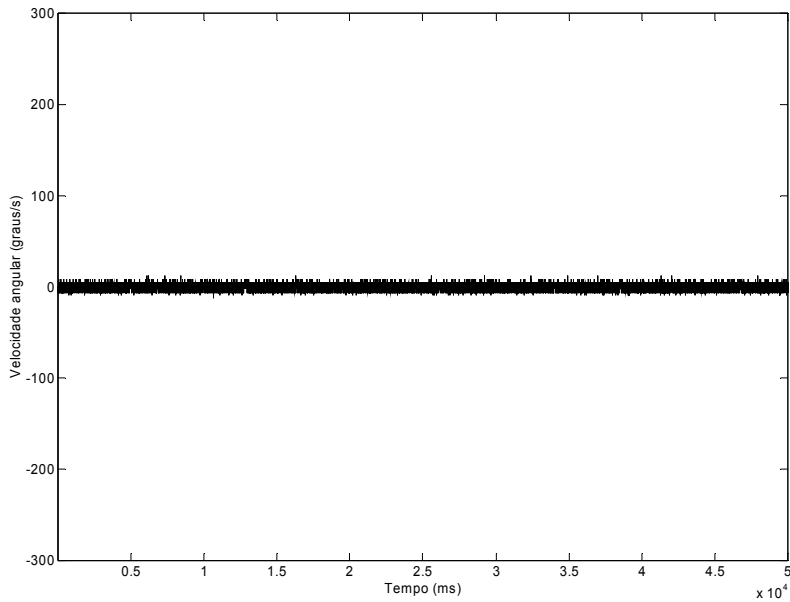


Figura 6.12 – Velocidade angular medida pelo sensor 3JA.
Fator de escala médio 0,67 mV/°/s

É possível notar pela Figura 6.12 que mesmo a velocidade angular medida sendo zero, o ruído está deveras intenso quando comparado com o fundo de escala que é de 300 °/s. Através do cálculo RMS, como utilizado anteriormente

$$\sigma = \sqrt{\frac{\mathbf{v}^T \mathbf{v}}{N}}$$

```
% === Procedimento para cálculo do ruído rms, ou desvio padrão
STD = sqrt((giro(1:5e4)' * giro(1:5e4))/5e4);
% === Fim
```

$$\sigma = 2,3633 \text{ °/s} \quad (6.10)$$

O sinal está ruidoso porque a banda do mesmo é de 500 Hz, já que foi adquirido a uma taxa de 1 kHz. Realizando uma filtragem digital idêntica a utilizada com os dados do sensor ADXL202, são obtidos dados de melhor qualidade, e ruído significativamente menor, como mostrado na Figura 6.13.

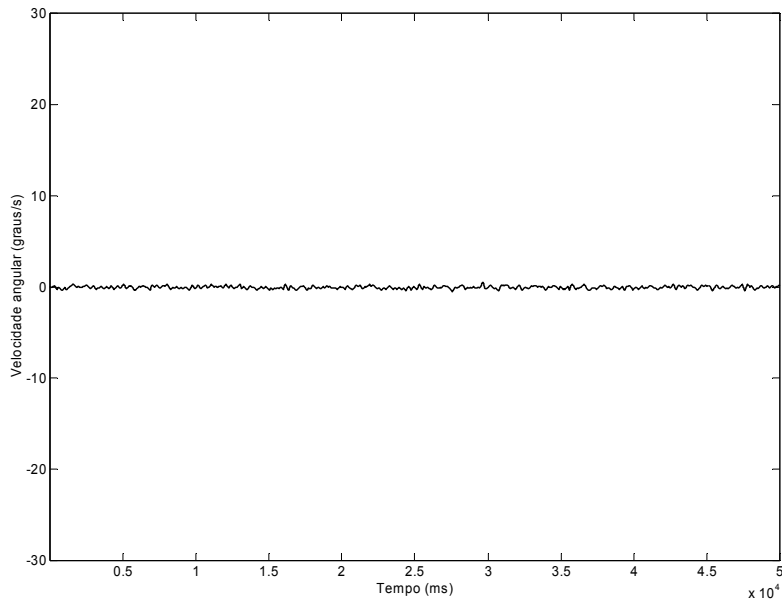


Figura 6.13 – Velocidade angular medida pelo sensor 3JA após filtragem para limitação da banda em 50 Hz. Fator de escala médio 0,67 mV/°/s.

O RMS da medida agora é de

$$\sigma = 0,156 \text{ } ^\circ/\text{s} \quad (6.11)$$

Agora, um outro fator importante é determinar a intensidade da deriva existente no sinal de saída do sensor. Durante os primeiros segundos com o sensor em repouso é tirada a média dos valores adquiridos. Esta média, considerada fixa durante o restante do tempo de aquisição, é subtraída de todo vetor, portanto, se o restante da aquisição tem a mesma média que os segundos iniciais, o novo vetor possuirá média zero. Quando um vetor de média zero é integrado no tempo, o mesmo irá produzir um outro vetor também com média zero. O girômetro mede velocidade angular, portanto se ele está em repouso medindo zero, mais o *off-set*, mais o ruído, é razoável esperar que, após a retirada de seu *off-set*, a integral de sua saída resulte em deslocamento angular zero.

A Figura 6.14 mostra que a integral do sinal do giro não é zero durante o passar do tempo. Isto já era esperado. Provavelmente porque a média do sinal variou entre os instantes inicial e final da aquisição dos dados. Como o sinal do giro está reduzido a uma faixa de excursão pequena, um circuito amplificador adicionado à saída do giro pode melhorar a qualidade do sinal.

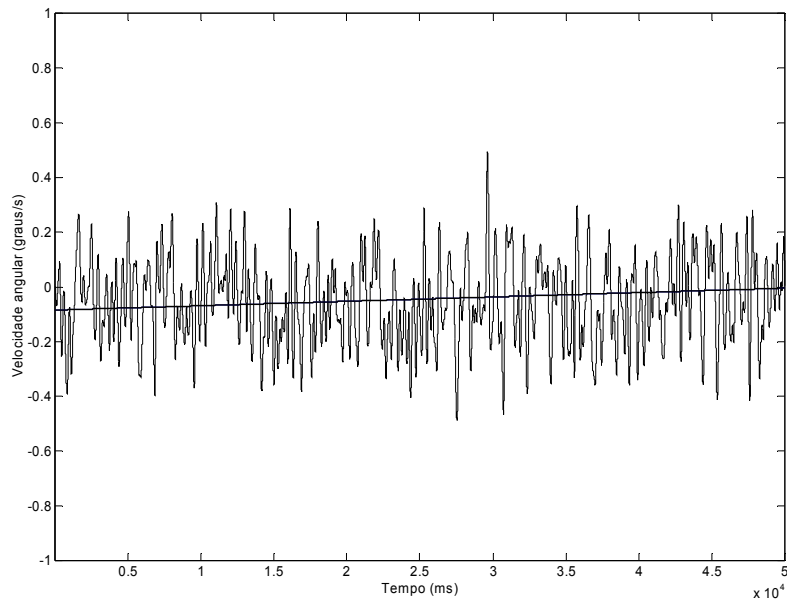


Figura 6.14 – Sinal do girômetro e integral do sinal.

C. Amplificação do sinal do girômetro

A maioria dos conversores AD existentes convertem um sinal de 0 a 5V para uma palavra binária com referência em 2,5V para o zero, são os conversores ditos bipolares. Logo o sinal ideal para este tipo de conversor dever ter faixa de excursão, por segurança, de 0,5 a 4,5V e o zero em 2,5V.

O sinal proveniente do giro possui pequena faixa de excursão que vai de 1,15 a 1,55V com zero em 1,35V. E ainda, como o sinal possui pequena faixa dinâmica, o SNR (relação sinal/ruído) fica muito baixa, prejudicando uma leitura eficiente do sinal do sensor pelo conversor AD.

Para atingir tal meta, entre o giro 3JA e o ADC é necessário, além da filtragem *anti-aliasing*, amplificar o sinal em pelo menos 5 vezes e ajustar o zero para que fique em 2,5V.

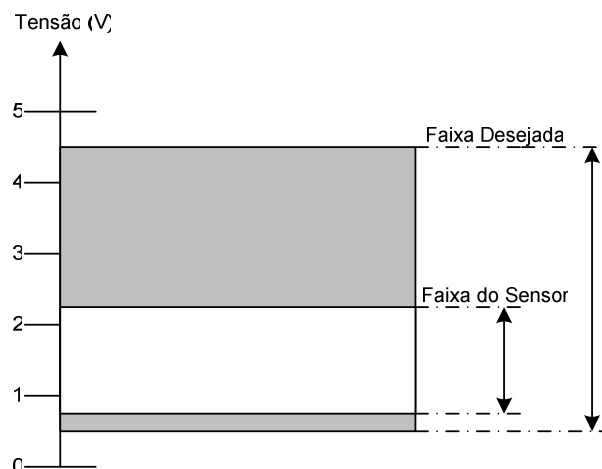


Figura 6.15 – Ilustração da necessidade de amplificação do sinal do girômetro 3JA da Murata.

Amplificação do *Application Note* do Fabricante

O fabricante do sensor recomenda o uso de um circuito amplificador da Figura 6.16. Este estágio de amplificação também age como um filtro ativo passa-faixa que seleciona sinais entre 0,5 Hz e 1k Hz. Os capacitores C_x e C_y assim como os resistores R_x e R_y são parte dos filtros passa-alta da entrada e passa-baixa da realimentação, respectivamente. A partir destes dados é possível conferir que o fabricante sugere frequência de corte passa-alta em

$$f_c^{-3dB} = \frac{1}{2\pi R_x C_x} = \frac{1}{2\pi \cdot 100 \cdot 10^3 \cdot 4700 \cdot 10^{-9}} = 0,34 \text{ Hz}$$

e frequência de corte passa-baixa em

$$f_c^{-3dB} = \frac{1}{2\pi R_y C_y} = \frac{1}{2\pi \cdot 90 \cdot 10^3 \cdot 1800 \cdot 10^{-9}} = 982 \text{ Hz}$$

Os resistores R_y e R_1 definem o ganho de amplificação. O capacitor C_1 apenas filtra o sinal DC da referência.

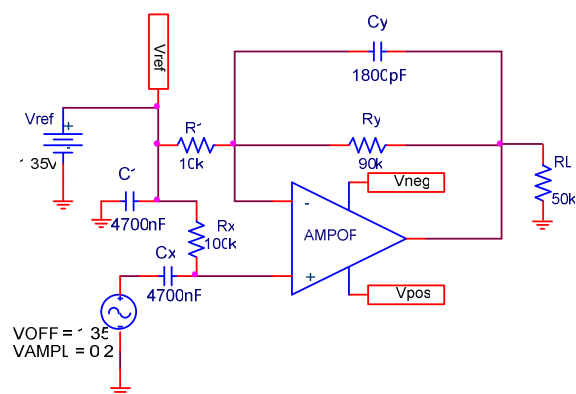


Figura 6.16 – Circuito filtro ativo passa-faixa do sinal do giro 3JA sugerido pelo fabricante.

A Figura 6.17 ilustra a simulação deste circuito no PSPICE. Foi injetado um sinal senoidal de 3 Hz com características semelhantes a do sensor, ou seja, sinal como *off-set* em 1,35V e pico de 100mV.

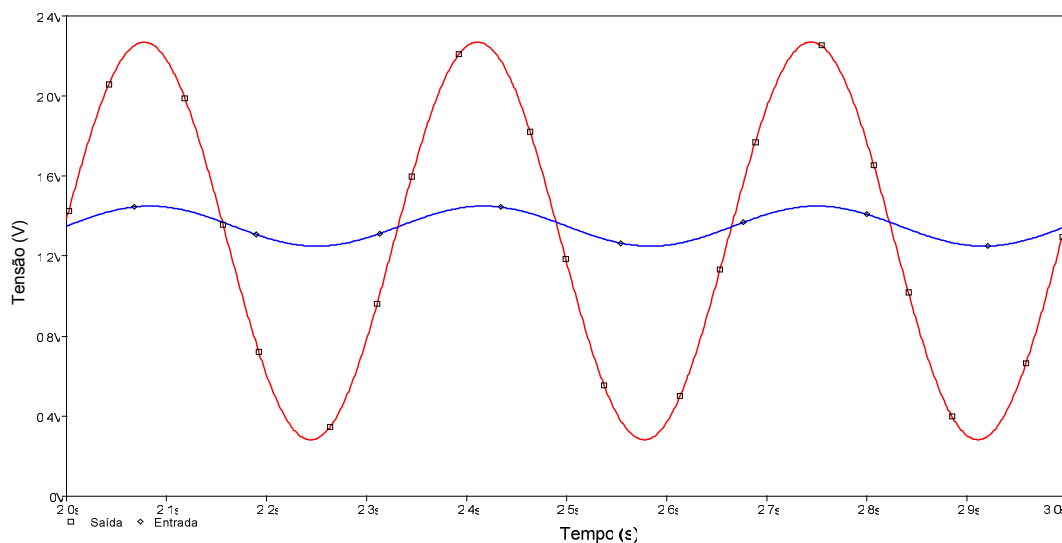


Figura 6.17 – Simulação do circuito da Figura 6.16.

A amplificação do sinal é de cerca de 10 vezes, mas é importante notar que com esse ganho, quando a entrada for máxima, ou seja, 300 °/s, o sinal amplificado nesta proporção resultará em saturação na saída, já que não será permitida alimentação simétrica nos amplificadores e conversores AD por motivos de portabilidade. A Figura 6.18 apresenta uma ilustração do que acontece neste caso.

Considerações sobre este circuito

Ocorre que quando o sinal não está centrado em 2,5V não é possível aplicar-lhe o máximo ganho, por esse motivo essa configuração não é plenamente adequada. Ainda, a resposta em frequência do circuito implementa uma filtragem passa-baixa pretendendo a eliminação da deriva do sensor. Em parte adequada, é um tanto desnecessária, pois pode ser realizada em software posteriormente. A Figura 6.19 traz uma simulação do formato da banda de passagem para esta configuração.

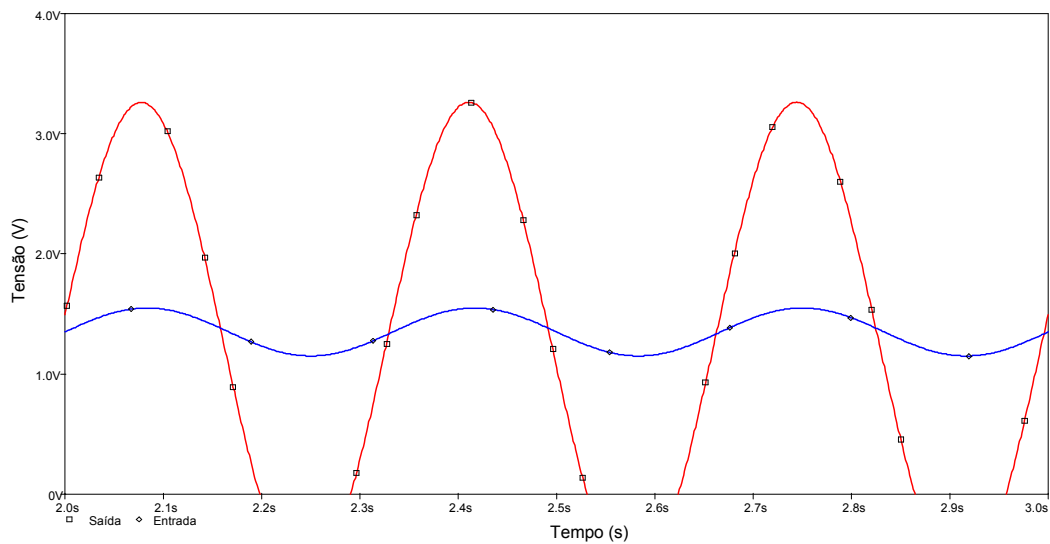


Figura 6.18 – Saída do amplificador quando a entrada excursiona entre seus possíveis extremos.

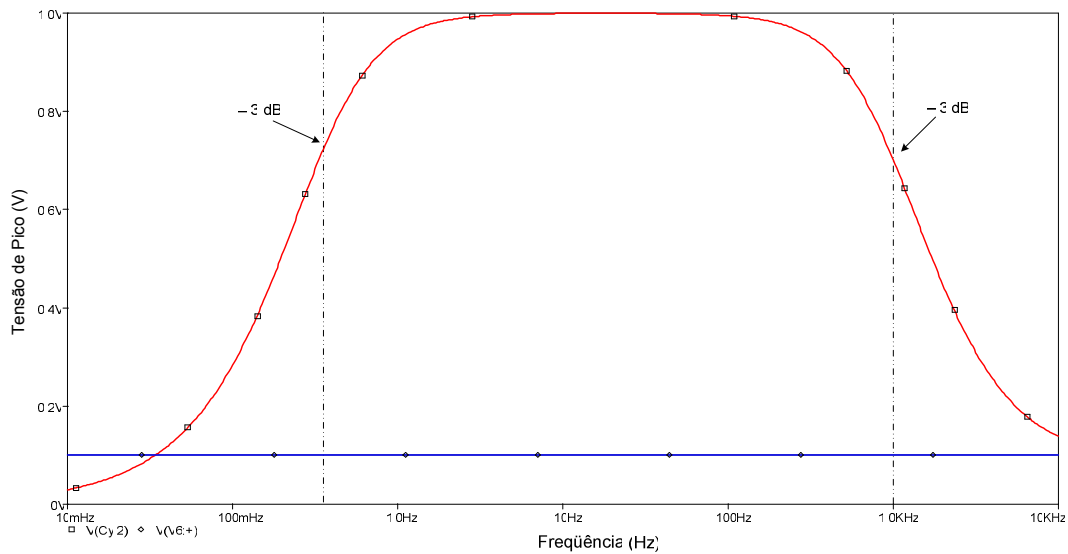


Figura 6.19 – Resposta em frequência do circuito amplificador passa-banda. Os pontos de corte estão em torno de 0,35 Hz e 1000 Hz.

Solução do Problema do *off-set*

O projeto da Figura 6.20 apresenta uma solução melhorada para o problema em questão. Utilizando apenas um amplificador o circuito ajusta o zero da saída em 2,5 V e amplifica adequadamente o sinal para que o mesmo excursiona na faixa desejada, como da Figura 6.14. A única modificação em relação ao circuito da Figura 6.20 foi a mudança na referência negativa e a retirada do capacitor passa-alta, dessa forma o efeito de saturação vista na Figura 6.18 não irá mais ocorrer pois o off-set de saída subirá para 2,5 V. O Resistor R_2 da Figura 6.20 pode ser substituído por um potenciômetro para que seja possível fazer ajustes a tensão de saída devido a tolerância dos componentes.

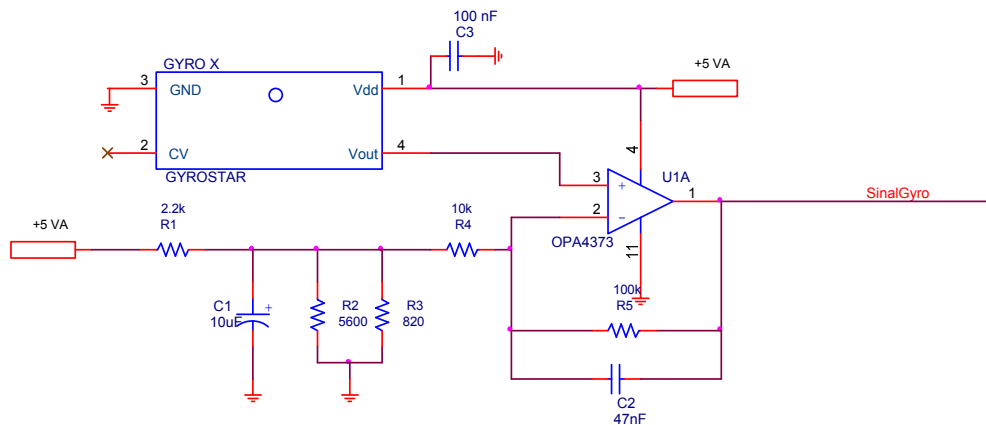


Figura 6.20 – Solução de amplificação com zero em 2,5 V.

A Figura 6.21 mostra o resultado da simulação que demonstra a correção do *off-set*.

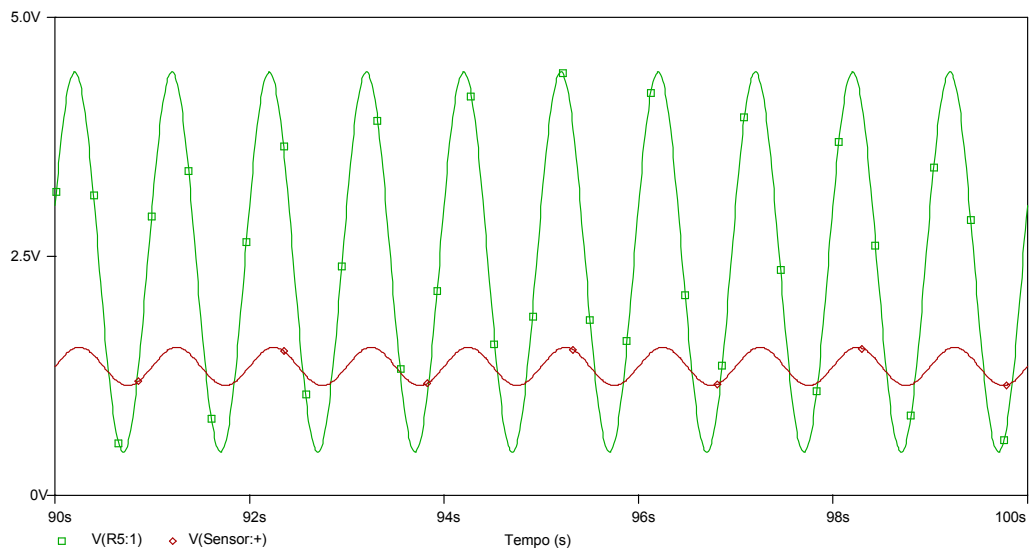


Figura 6.21 – Solução de amplificação com zero em 2,5 V.

D. Experimentação do girômetro com amplificador

Uma aquisição foi feita para verificar o funcionamento do amplificador. A Figura 6.22 mostra a aquisição. O girômetro foi movimentado aleatoriamente.

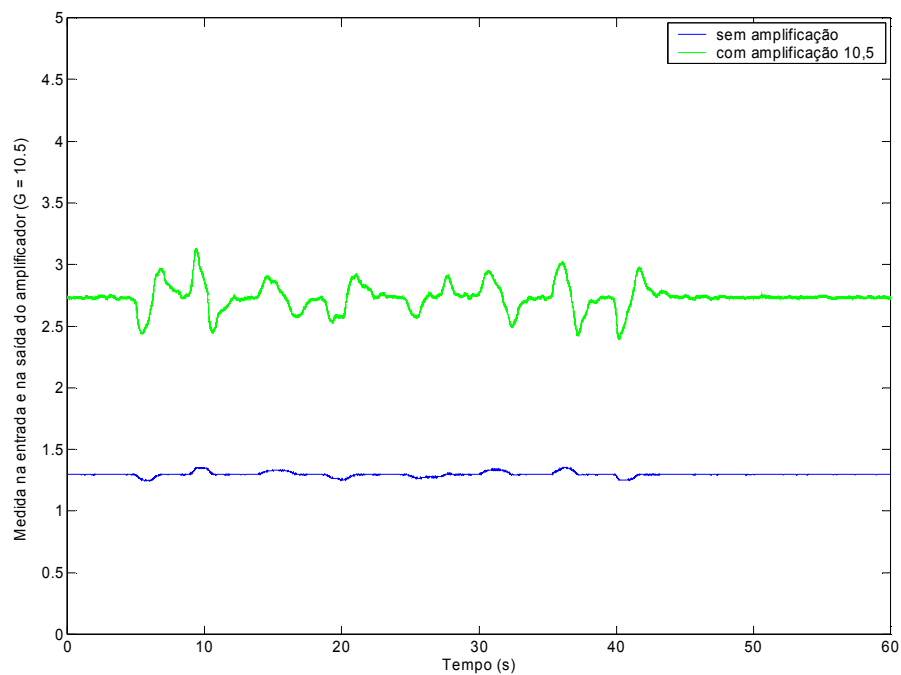


Figura 6.22 – Medida real do giro para o circuito da Figura 6.21. Taxa de 1 kHz.

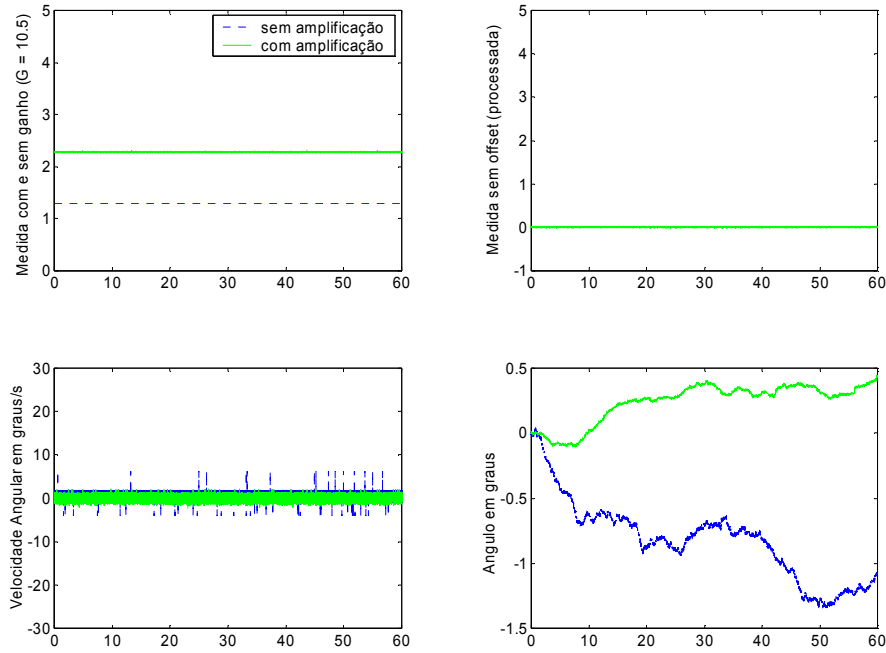


Figura 6.23 – Medida do Giro com e sem amplificação para estimação da deriva. Tempo de aquisição de 60 segundos. Taxa 1kHz.

Na Figura 6.23, o giro foi deixado em repouso e o sinal com e sem amplificação são adquiridos para fins de comparação das derivas. Como o sensor ficou em repouso o gráfico que indica ‘ângulos em graus’ deveria manter ambos sinais em zero constantemente. O sinal amplificado manteve o erro menor que 1° durante um período de integração de 60 segundos. Resultado melhor que o obtido com o sinal sem amplificação.

O ruído do sinal amplificado é um pouco maior que o ruído do sinal sem amplificação. Medindo o ruído RMS do sinal tem-se

$$\sigma = 0,5679 \text{ } ^\circ/\text{s} \quad (6.12)$$

Após a filtragem digital, para limitar a banda em 50 Hz, o desvio padrão do ruído torna-se

$$\sigma = 0,2051 \text{ } ^\circ/\text{s} \quad (6.13)$$

6.4.3 – Sensor magnético HMC2003

A. Descrição geral do sensor magnético

O sensor magnético HMC2003 é um sensor híbrido da Honeywell, ou seja, ele é formado por um sensor magnético no eixo Z, um sensor magnético duplo em X e Y e amplificadores, tudo isso montado sobre uma plataforma DIP 600 mils de 20 pinos.

Os sensores que compõem o HMC2003 são transdutores construídos com uma membrana permalloy magnetoresistiva. Essa membrana varia sua resistência de acordo como a força de alinhamento causada por um campo magnético externo que é aplicado a ela.

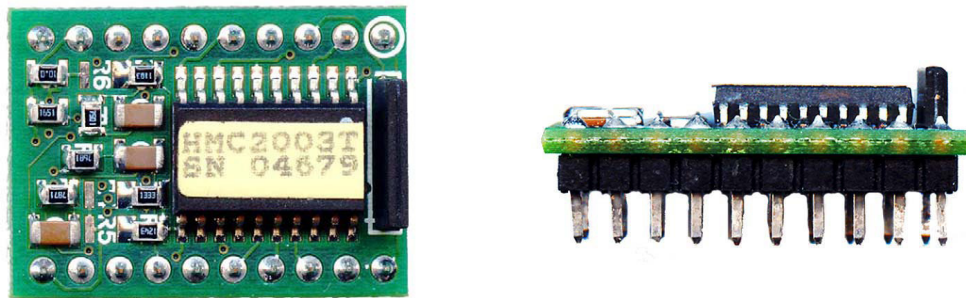


Figura 6.24 – Sensor Híbrido HMC2003 – Contém uma pastilha com eixos X e Y e um partilha vertical para leitura no eixo Z. No DIP também estão incluídos amplificadores que ajustam a escala para 1V/Gauss.

As especificações do sensor híbrido HMC2003 é apresentada na Tabela 6.3, a seguir.

Tabela 6.3 – Principais especificações do sensor HCM2003

Característica	MIN	MAX	TIP	Unidades
Tensão de alimentação	6	15		V _{DC}
Consumo		20		mA
Faixa do Campo magnético	- 2	+ 2		gauss
Tensão de saída	0,5	4,5		V
Resolução		40		μ gauss
Largura de banda		1		kHz
Fator de escala	0,98	1,02	1	V/gauss
Saída com campo nulo	2,3	2,7	2,5	V

Linearidade		2	0,5	%FS
Histerese		0,1	0,05	%FS
Resistência set/rst		6		Ω
Deriva do fator de escala		- 600		ppm/°C
Deriva da saída com campo nulo (com set/rst ligado)		± 100		ppm/°C

O sensor HCM2003 já foi preparado para que os sinais em sua saída sejam digitalizados e processados considerando medidas de campos magnéticos com intensidade de até ± 2 gauss com fator de escala de 1 V/gauss.

Como a aplicação para este sensor é medir o campo magnético terrestre, então o maior campo que irá medir, por exemplo em Londrina – PR, será 0,27 gauss, ou seja, pouco mais de 10% do fundo de escala. A faixa dinâmica para a aplicação será de aproximadamente

$$\Delta v = s \Delta B \quad (6.14)$$

$$\Delta v = 1 \times 0.54 = 540 \text{ mV}$$

indicando necessidade de amplificação por um fator entre 5 e 8. Este sensor, assim como ADXL202, apresenta saída de tensão de 2,5 V para entrada nula, esse é um parâmetro importante porque a referência da digitalização será 2,5 V.

Características como linearidade, histerese e resolução são muito boas, sendo a maioria menores que 1% do fundo de escala. As derivas são significativas só em ambientes onde a temperatura varia muito.

A resposta em frequência do sensor é de 1 kHz, bem mais que o necessário para a aplicação.

B. Amplificação do sinal dos Magnetômetros

A Figura 6.25 mostra o esquemático de um amplificador não-inversor de ganho 8 como frequência de corte em 50 Hz. Como o sensor fornece uma referência de 2,5 V não é preciso criá-la com resistores.

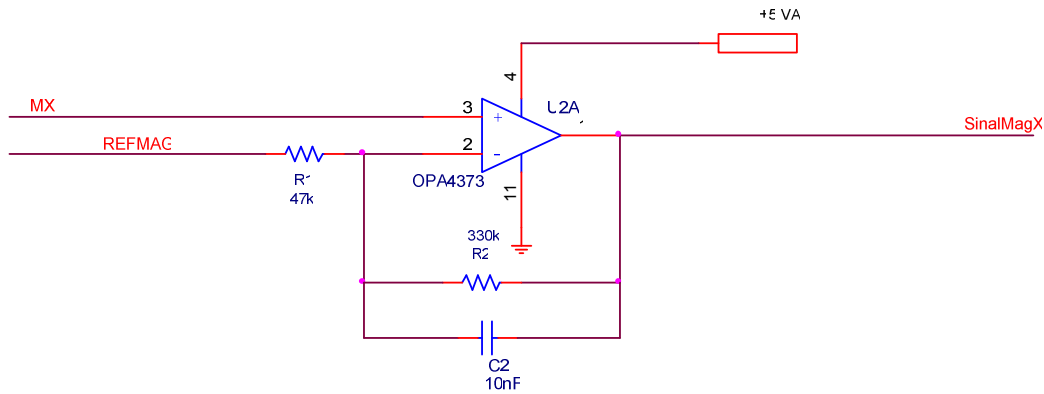


Figura 6.25 – Circuito amplificador de ganho 8, não-inversor com frequência de corte em 50 Hz usado para amplificação do sinal do canal X dos magnetômetro.

Como um ganho de 8 vezes a faixa de excursão do magnetômetro passa a ser

$$\Delta v = G_s \Delta B$$

$$\Delta v = 8 \times 1 \times 0,54 = 4,320 \text{ V} \quad (6.15)$$

6.5 – Set/Reset

Os sensores magnéticos da Honeywell permitem a aplicação de uma técnica de leitura diferencial que melhora significativamente o seu desempenho comparado a outros sensores magnéticos. Essa técnica é conhecida como Set/Reset [CAR04a]. A membrana permalloy interna está polarizada magneticamente em uma direção com um campo de aproximadamente 10 Gauss. A resistência da membrana varia proporcionalmente ao produto interno do campo interno próprio com o campo externo aplicado na membrana. O campo magnético interno se deteriora, perdendo intensidade com o passar do tempo, o que arruína as medidas. Então, para otimizar o processo, antes de cada leitura seria interessante *setar* o campo interno novamente para seu valor máximo sem deterioração, e depois, realizar a medida. Por isso, os sensores magnéticos da Honeywell possuem uma microbobina interna capaz de gerar tal campo para *setagem* da membrana *permalloy*, bastando passar por essa microbobina uma corrente de 3 a 4 Ampères, por não mais que 5 μ s. Além disso, é possível reverter a polaridade do campo interno, ou seja, aplicar um *reset* na membrana, passando agora uma corrente de 3 a 4 Ampères no sentido contrário. Com a possibilidade de *setar* e *resetar* o campo da

membrana, é possível realizar uma medida diferencial da tensão, eliminando assim todos os tipos de *off-sets* do sensor até o digitalizador.

A Figura 6.26 mostra o processo de set/reset descrito acima.

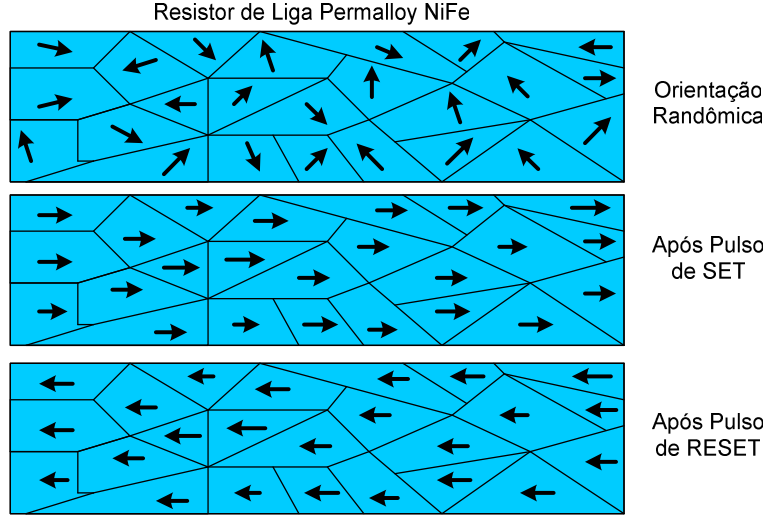


Figura 6.26 – Processo do set/reset na membrana *permalloy* dos sensores que compõe o híbrido HMC2003.

Verificando que o ruído e o *off-set* é aditivo a partir da saída do sensor

$$\begin{aligned} V_{\text{set}} &= +V_{\text{out}} + \xi \\ V_{\text{rst}} &= -V_{\text{out}} + \xi \end{aligned} \quad (6.16)$$

onde ξ é a combinação do *off-set* e do ruído.

Fazendo a diferença dos valores, a tensão de saída fica

$$\frac{V_{\text{set}} - V_{\text{rst}}}{2} = \frac{+V_{\text{out}} - (-V_{\text{out}} + \xi) + \xi}{2} = V_{\text{out}} \quad (6.17)$$

O circuito de set/reset

O circuito de Set/Reset melhora significativamente a performance do sensor magnético. Esse circuito deve prover uma corrente entre 3 e 4 Ampères sobre uma resistência de 5 Ohms por não mais que $5\mu\text{s}$ para *setar* a direção do campo magnético;

e gerar a mesma corrente reversamente para *resetar* o campo interno do sensor para a direção oposta. Como dito anteriormente, esse processo propicia uma leitura diferencial do sinal anulando assim o *off-set* propagado pelos amplificadores até o conversor AD.

Para executar essa função foi utilizado o circuito da Figura 6.27, que pode ser melhor visto no Anexo II. O fabricante sugere o uso de um pulsador *push-pull* sem fonte simétrica como [CAR04a].

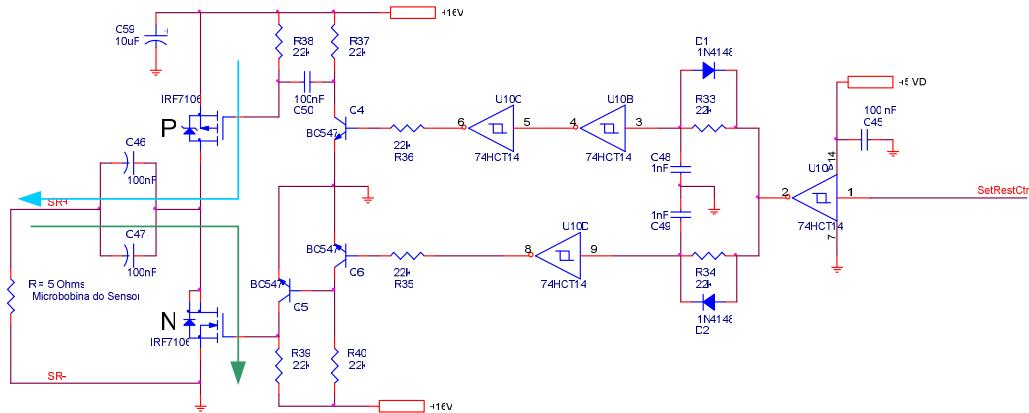


Figura 6.27 – Circuito set/reset sugerido pelo fabricante.

O fechamento da chave P é realizado na borda de subida do pulso de sincronismo enviado (SetRstCtrl). O circuito com o transistor antes da chave P, serve para ligar a chave por um tempo não maior que uma constante de tempo RC. Com a chave P fechada, flui um pico de corrente para a carga durante $5 \mu s$. Quando ocorre a borda de descida do pulso de sincronismo, a chave N é acionada e a tensão acumulada no capacitor gera uma corrente pela carga com o mesmo formato temporal no sentido contrário de antes.

A Figura 6.28 mostra os formatos dos pulsos adquiridos na carga em função da borda aplicada à entrada.

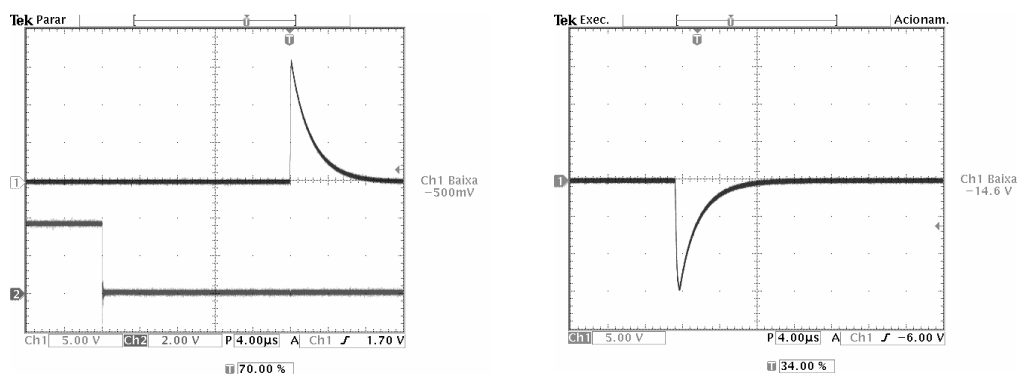


Figura 6.28 – Formato das ondas de Set/Reset adquiridas por um osciloscópio nos pontos de entrada e saída do circuito.

Experimentação do sensor magnético

Uma interface de testes foi arranjada para experimentar o funcionamento do sensor híbrido HMC2003. O esquemático é apresentado na Figura 6.29. Um outro cabo blindado envia sinais de sincronismo para gerar os pulsos de Set/Reset no sensor. A saída de tensão do sensor e os pulsos de sincronismo são enviados por um cabo blindado até uma placa digitalizadora no computador. Os sinais de sincronismo são necessários para reconstruir o sinal.

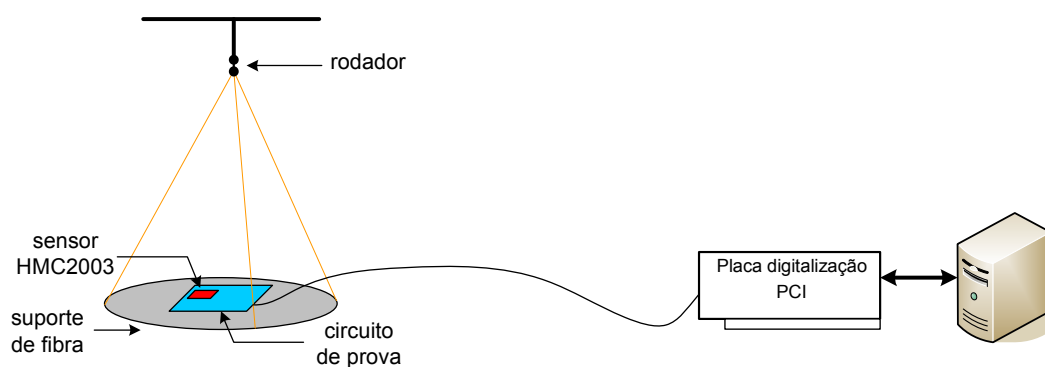


Figura 6.29 – Esquemático de ensaio do sensor HMC2003

Para o sensor magnético foram realizados para verificar a amplificação a performance do sensor sob o funcionamento do circuito se Set/Reset. Na falta de uma mesa de rotação, fora improvisada uma prancha rotativa livre anexada ao teto, como mostra o esboço na Figura 6.29. O objetivo desta montagem é ler o campo magnético

sem interferências de qualquer metal, fiação elétrica ou equipamentos elétricos em sua proximidade.

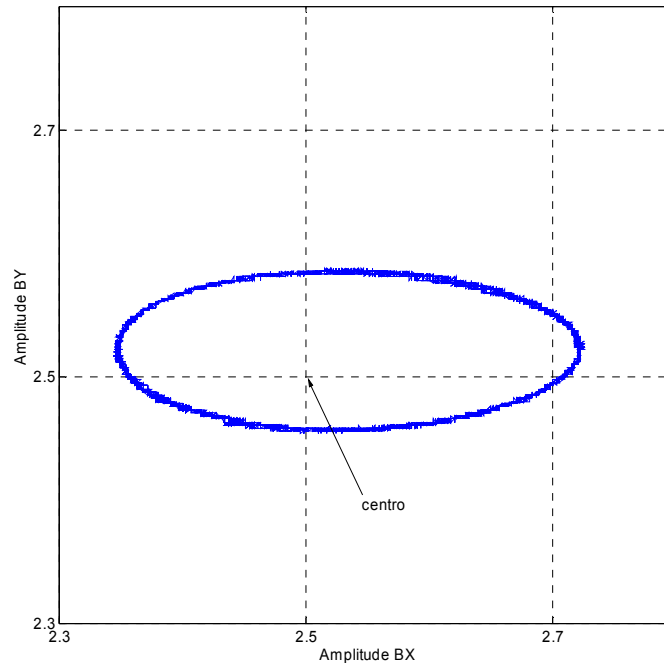


Figura 6.30 – Sinal adquirido do sensor HMC2003 sem amplificação e sem Set/Reset.

Durante a aquisição o sistema é colocado em rotação em torno do eixo da gravidade e então se adquire dados por 1 ou 2 minutos. A Figura 6.30 mostra uma plotagem XY dos dados. A elipse seria uma circunferência se não fosse pela diferença dos ganhos. Como citado em [CAR04b] a existência de uma rotação dos eixos da elipse, indicaria proximidade a objetos metálicos ou condutores. Visualmente esta rotação de eixos não é significativa.

A Figura 6.30 mostra os dados de saída dos canais XY sem amplificação e com o circuito de Set/Reset desligado. Note que o ganho é diferente em cada canal pelo fato do campo de referência interno (campo da membrana permalloy) ter se deteriorado. Após ser aplicado ao circuito o Set/Reset, a amplificação e fazendo novamente o mesmo procedimento de aquisição, obtém-se o resultado da Figura 6.31. Figura 6.32 apresenta o mesmo gráfico, mas em função do tempo no eixo Z.

Foram feitas também algumas medidas para verificar se o desempenho do ruído do sinal de saída do sensor é alterado pelo funcionamento do Set/Reset, e principalmente, quantificar essa melhora. O ruído sem amplificação e sem S/R é de

$$\sigma = 0,0013 \text{ Gauss} \quad (6.18)$$

Com amplificação e Set/Reset o ruído medido ficou cerca de

$$\sigma = 0,0026 \text{ Gauss} \quad (6.19)$$

O ruído e deriva de cada sensor são parâmetros importantes na determinação do erro total da atitude.

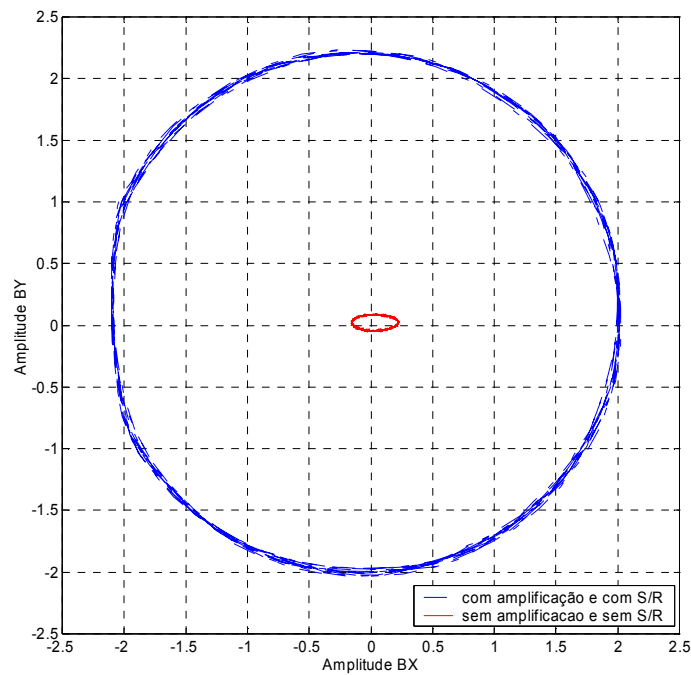


Figura 6.31 – Sinal adquirido do sensor HMC2003 com amplificação e com Set/Reset comparado ao Figura 6.30.

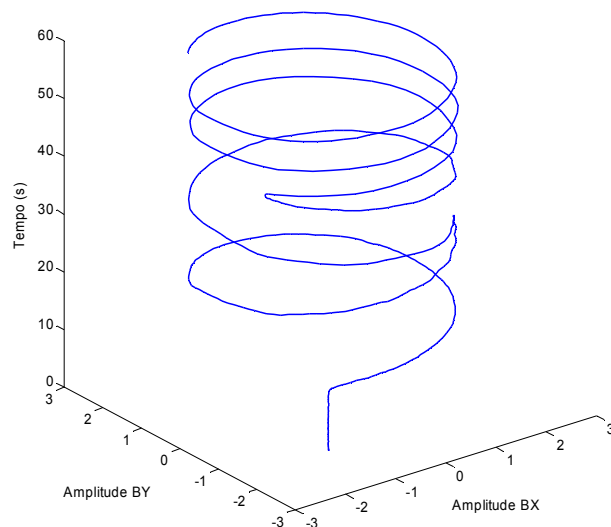


Figura 6.32 – Sinal adquirido do sensor HMC2003 com amplificação e com Set/Reset plotado em função do tempo no eixo Z.

6.6 – Conversores Analógico-Digital

Para fins de comparação foram selecionados dois conversores AD, ambos de 16 bits, para digitalizar os sinais dos sensores. Um utiliza a interface serial (SPI), o outro utiliza interface paralela multiplexada em 8 bits.

6.6.1 – Conversor analógico-digital ADS8344

O ADS8344 é um conversor AD de 8 canais multiplexados, 16 bits de resolução e interface SPI. O esquema de interface deste AD é como mostra a Figura 6.33. A folha de rosto da folha de dados deste componente se encontra em anexo assim como seu circuito de interface. Foram utilizados dois conversores para atender os 12 canais de sensores.

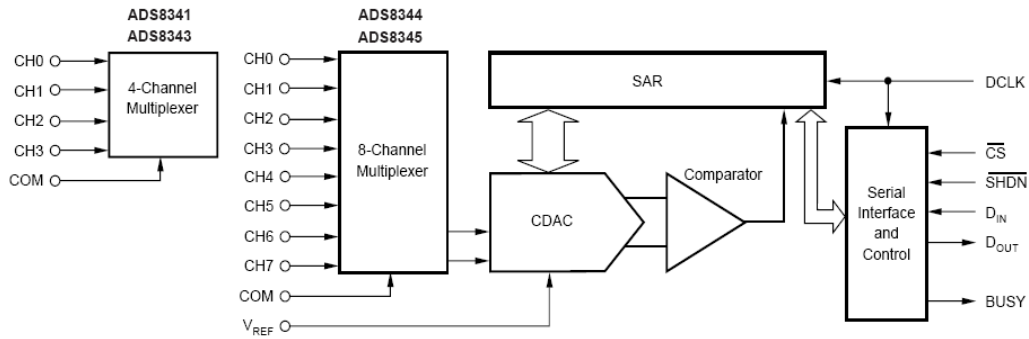


Figura 6.33 – Diagrama de blocos do conversor ADS8344

6.6.2 – Conversor analógico-digital ADS8364

O ADS8364 é um conversor AD de 6 canais não-multiplexados de amostragem e conversão simultânea. A amostragem simultânea dos canais foi o que levou a escolha deste AD. A Figura 6.34 apresenta seu diagrama lógico. O esquema de interface deste AD está no projeto em anexo. São utilizados dois destes conversores para atender os 12 canais de sensores.

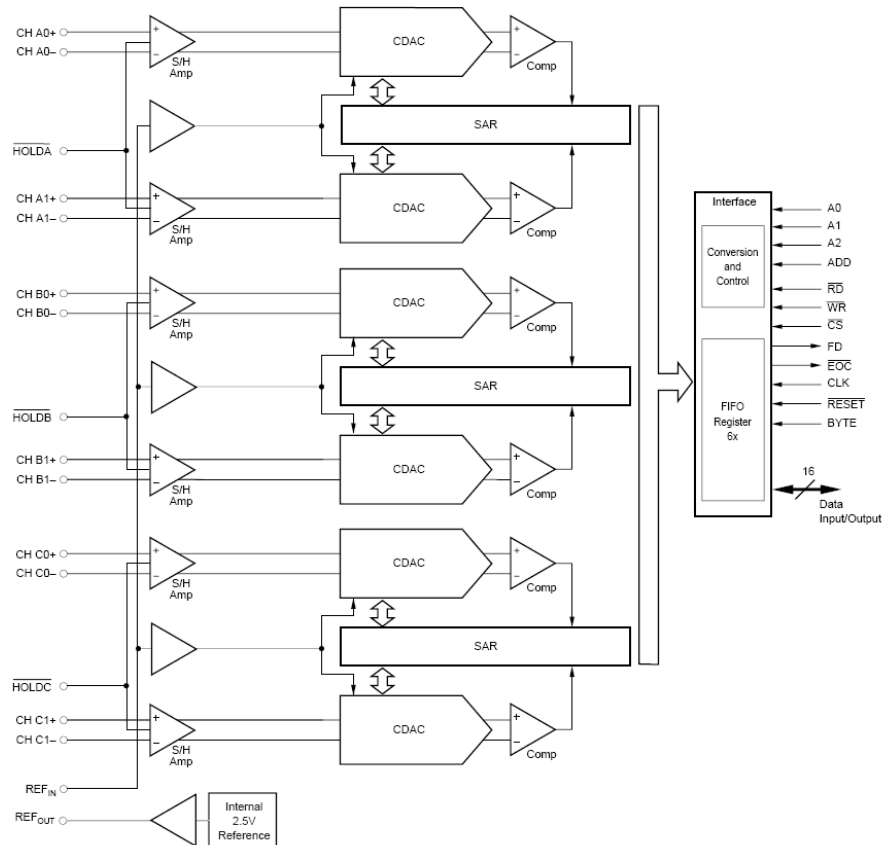


Figura 6.34 – Diagrama de blocos do conversor ADS8364

6.7 – O Microcontrolador

Para controlar os conversores AD e realizar o processamento da atitude fora utilizado o microcontrolador FS68HCS12DJ128, da família HCS12. Esse microcontrolador é uma evolução do *core* de instruções do MC68HC11 (projeto de 1985) mantendo compatibilidade. A arquitetura é de 16 bits CISC com 8 KB de memória RAM e 128 KB de memória FLASH, expansível até 4 MB. O ciclo de instrução mínimo chega a 40 ns. Existem diversos periféricos integrados a arquitetura como serial assíncrona (SCI com suporte a IrDA – protocolo *Wireless*), serial síncrona (SPI), barramento I2C e PWM (Modulação por largura de pulso – pode servir como conversor digital analógico). Esse microcontrolador também possui 16 entradas analógica para conversão AD em 10 bits e dezenas de GPIO (entradas e saídas digitais para uso genérico) tudo isso encapsulado em um invólucro de 112 pinos.

Para desenvolvimento e depuração existe a interface BDM (*Background Debug Module* – Módulo de Depuração em *Background*) e a Metrowerks® disponibiliza a

ferramenta de programação, o CodeWarrior® com limitação de código em 8KB em C/C++ gratuitamente e codificação ilimitada em assembly.

O esquemático das ligações do microcontrolador se encontram no projeto em anexo.

6.8 – Software básico

6.8.1 – Inicializações

```
#include "main.h"
#include <stdio.h>

/* ----- Defines */
#define  HOLDX      Regs.porte.bit.pte3
#define  RD         Regs.porte.bit.pte2
#define  SEL_ADC0   Pim.ptj.bit.ptj0
#define  SEL_ADC1   Pim.ptj.bit.ptj1
#define  SET_RESET  Page.portk.bit.ptk7

/* ----- Variáveis Globais e Tipos*/

typedef union{
    int word;
    struct {
        unsigned char byteh;
        unsigned char bytel;
    } part;
} aux8and16;

aux8and16 adc_0[6],adc_1[6],adcS_0[6],adcS_1[6];

int Mag[6],Acc[6];

long int accsoma0[6] = {0,0,0,0,0,0};
long int accsoma1[6] = {0,0,0,0,0,0};

unsigned char cntdec0 = 0;
unsigned char cntdec1 = 0;
unsigned char i,j;
```

```

char str[40];
const char StrIni[25] = "Conection OK!";
const char TabCan[8] = {0x86, 0xC6, 0x96, 0xD6, 0xA6, 0xE6, 0xB6,
0xF6};

/* ----- Função de Inicialização das Portas */

void PimConfig(void) {
    Page.ddrk.byte   = 0x83;    // Programa Enables dos ADsSPI com saídas
    Page.portk.byte  = 0x03;    // Mantém os ADs desabilitados
    Pim.ddrt.byte    = 0x30;    // Programa os Pinos dos LEDs como saídas
    Pim.ptt.byte     = 0x30;    // Mantém os LEDs apagados

    Regs.irqcr.byte  = 0x00;
    Regs.pear.byte   = 0x10;    // Desliga ECLK
    Regs.ddra.byte   = 0x00;    // Deixa a porta A como entrada
    Regs.porte.byte  = 0x0C;    // PE2 e PE3 = 1
    Regs.ddre.byte   |= 0x0C;   // Deixa PE2 PE3 como saída

    Pim.ptj.byte     |= 0x03;   // PJ0 e PJ1 = 1
    Pim.ddrj.byte    = 0x03;    // PJ0 e PJ1 são saídas
}

/* ----- Função de Configuração do Clock */

void ClockConfig(void){
    Crg.synr.byte    = 0x03;    // Multiplicador m+1
    Crg.refdv.byte    = 0x01;    // Divisor d+1
    while(Crg.crgflg.bit.lock == 0){ }
    Crg.clksel.byte  |= PLLSEL;
}

```

6.8.2 – Comunicação

O código C a seguir mostra as funções de comunicação pela serial (SCI).

```

// ----- Configurações da Serial
void SerialConfig(void){
    /* SCI baud rate = SCI clock / (16 x BR) */

```



```

    Sci0.scibd.word = 0x000D; // Seleciona taxa: 000D = 115200 bps
    Sci0.scicr1.byte = 0x00; // Configurações Default
    Sci0.scicr2.byte = 0x0C; // Habilita Transmissão e Recepção em Int
}

// ----- Função para Imprimir caractere
void PrintChar(char c){
    while (Sci0.scisr1.bit.tdre == 0) { }
    Sci0.scidr1.byte = c;
}

// ----- Função para Ler caractere (loop)
unsigned char GetChar(){
    while (Sci0.scisr1.bit.tdre == 0) { }
    return Sci0.scidr1.byte;
}

// ----- Função para enviar String
void PrintStr(const char *str){
    unsigned char i = 0;
    while (str[i] != '\0') {
        PrintChar(str[i++]);
    }
}

```

6.8.3 – Aquisição

As funções de aquisição são aquelas que controlam os conversores AD. Foram feitas duas funções. Uma para atender os conversores AD SPI e outra para atender os conversores AD paralelos.

A. Funções para acesso ao conversor AD SPI

```

// ----- Função para configura SPI
void SpiConfig(void){
    Spi0.spicr1.byte = 0x50;
    Spi0.spicr2.byte = 0x00;
    Spi0.spibr.byte = 0x03; /* BusClk/8 */
}

```

```

// ----- Função para enviar e receber um byte via SPI
unsigned char SpiTransfer(unsigned char c){
    Spi0.spidr.byte = c;
    while (Spi0.spisr.bit.spif == 0) { }
    return Spi0.spidr.byte;
}

// ----- Função Ler do AD, recebe canal e retorna inteiro lido
// O Chip select está na porta PK0
unsigned int Read_ADS8344_0(unsigned char c){
    aux8and16 aux;
    unsigned char lixo;
    Page.portk.bit.ptk0 = 0;           /* Ativa CS (chip select)*/
    aux.part.bytel = SpiTransfer(c);    /* Envia Seleção de Canal */
    Page.portk.bit.ptk0 = 1;           /* desAtiva CS (chip sel.)*/
    Delay_micro(20);
    Page.portk.bit.ptk0 = 0;           /* Ativa CS (chip select)*/
    aux.part.byteh = SpiTransfer(0x00); /* Recebe Hi byte */
    aux.part.bytel = SpiTransfer(0x00); /* Recebe Lo byte */
    lixo = SpiTransfer(0x00);          /* Recebe último bit */
    Page.portk.bit.ptk0 = 1;           /* desAtiva CS (chip sel.)*/
    return aux.word;
}

// ----- Função Ler do AD, recebe canal e retorna inteiro lido
// O Chip select está na porta PK1
unsigned int Read_ADS8344_1(unsigned char c){
    aux8and16 aux;
    unsigned char lixo;
    Page.portk.bit.ptk1 = 0;           /* Ativa CS (chip sel.)*/
    aux.part.bytel = SpiTransfer(c);    /* Envia Seleção de Canal */
    Page.portk.bit.ptk1 = 1;           /* desAtiva CS (chip sel.)*/
    Delay_micro(20);
    Page.portk.bit.ptk1 = 0;           /* Ativa CS (chip sel.)*/
    aux.part.byteh = SpiTransfer(0x00); /* Recebe Hi byte */
    aux.part.bytel = SpiTransfer(0x00); /* Recebe Lo byte */
    lixo = SpiTransfer(0x00);          /* Recebe último bit */
    Page.portk.bit.ptk1 = 1;           /* desAtiva CS (chip sel.)*/
    return aux.word;
}

```

B. Funções para acesso ao conversor AD paralelo

```
// Função para Ler AD paralelo, recebe ponteiro do vetor de canais
void Read_ADC8364_0(aux8and16 *c){
    unsigned char ii,ptr;
    SEL_ADC0 = 0;                                /* seleciona ADC0 */
    HOLDX = 0;                                    /* Manda converter */
    HOLDX = 1;
    SEL_ADC0 = 1;                                /* deseleciona ADC0 */
    for(ii = 0; ii < 10; ii++);                  /* Aguarda uns 4 us */
    SEL_ADC0 = 0;                                /* seleciona ADC0 */
    for(ii = 0; ii < 6; ii++){
        RD = 0;
        ptr = (Regs.porta.byte & 0x70) >> 4;
        RD = 1;
        RD = 0;
        c[ptr].part.bytel = Regs.porta.byte;
        RD = 1;
        RD = 0;
        c[ptr].part.byteh = Regs.porta.byte;
        RD = 1;
    }
    SEL_ADC0 = 1;                                /* deseleciona ADC0 */
}

// Função para Ler AD paralelo, recebe ponteiro do vetor de canais
void Read_ADC8364_1(aux8and16 *c){
    unsigned char ii,ptr;
    SEL_ADC1 = 0;                                /* seleciona ADC0 */
    HOLDX = 0;                                    /* Manda converter */
    HOLDX = 1;
    SEL_ADC1 = 1;                                /* seleciona ADC0 */
    for(ii = 0; ii < 10; ii++);                  /* Aguarda uns 4 us */

    SEL_ADC1 = 0;                                /* seleciona ADC0 */
    for(ii = 0; ii < 6; ii++){
        RD = 0;
        ptr = (Regs.porta.byte & 0x70) >> 4;
        RD = 1;
```

```

    RD = 0;
    c[ptr].part.bytel = Regs.porta.byte;
    RD = 1;
    RD = 0;
    c[ptr].part.byteh = Regs.porta.byte;
    RD = 1;
}
SEL_ADC1 = 1;                               /* seleciona ADC0 */
}

```

6.8.4 – Interrupção do timer

Uma interrupção do Timer ocorre a cada 500 microsegundos. O que ocorre dentro da rotina do Timer depende do estado do SET/RESET do sensor Magnético. Se o sensor estiver em estado de SET, apenas o sensor magnético é lido, em seguida é aplicado a ele um RESET. Na próxima interrupção o sensor magnético estará em RESET, então o valor é lido e subtraído daquele lido anteriormente. O valor resultante é considerado como uma leitura diferencial do magnetômetro. Após a leitura o sensor é novamente colocado em SET. Os demais sensores são lidos em seguida e então o decimador acumula as medidas. Após 20 ciclos de SET/RESET a função decimadora zera a *flag* de contagem indicando que há novos dados decimados no vetor. Com o decorrer desse processo a taxa efetiva de dados de saída é de 50 amostras/segundo.

O código do Timer é apresentado abaixo e em seguida a rotina de configuração e de decimação dos dados

```

// ----- Rotina de Interrupção do Timer
void TimerMod(void){
    static long int aux1[3],aux2[3];

    Tim0.mcflg.bit.mczf = 1;
    Pim.ptt.bit.ptt4 = 1;                               // Led de status

    if (SET_RESET == 1) {
        Read_ADC8364_0(adcS_0);                         // Lê AD Can0...Can5
        SET_RESET = 0;                                   // aplica RESET
        aux2[0] = (long int) adcS_0[0].word;              // Armazena valores
        aux2[1] = (long int) adcS_0[1].word;
        aux2[2] = (long int) adcS_0[2].word;
    }
}

```

```

    }
    else {
        Read_ADC8364_0(adcs_0);                // Lê AD Can0...Can5
        SET_RESET = 1;                        // aplica SET
        aux1[0] = (long int) adcs_0[0].word;
        aux1[1] = (long int) adcs_0[1].word;
        aux1[2] = (long int) adcs_0[2].word;

        adcs_0[0].word = (int)((aux1[0] - aux2[0])>>1);
        adcs_0[1].word = (int)((aux1[1] - aux2[1])>>1);
        adcs_0[2].word = (int)((aux1[2] - aux2[2])>>1);

        Read_ADC8364_1(adcs_1);                // Lê AD Can6...Can5
        Decimador0(adcs_0);
        Decimador1(adcs_1);

    }
    Pim.ptt.bit.ptt4 = 0;                    // Led de status
}

// ----Função que configura o Timer

void TimerModConfig(void){
    Tim0.mcflg.bit.mczf = 1;
    Tim0.mcctl.byte = 0xE4;
    Tim0.mccnt.word = 24000;
    Tim0.tscr1.byte = 0x80;
}

// ---- Decimador MAG faz a média de N dados e dá apenas uma saída

void Decimador0(aux8and16 *ptr){

#define    N    10

    unsigned char ii;

    for (ii = 0; ii < 6; ii++){
        accsoma0[ii] += (long int) ptr[ii].word;
    }
}

```

```

cntdec0++;

if (cntdec0 == N){
    for (ii = 0; ii < 6; ii++) {
        Mag[ii] = (int)(accsoma0[ii]/N);
        accsoma0[ii] = 0;
    }
    cntdec0 = 0;
}
}

// ---- Decimador ACC faz a média de N dados e dá apenas uma saída

void Decimador1(aux8and16 *ptr){

#define    N    10

    unsigned char ii;

    for (ii = 0; ii < 6; ii++){
        accsoma1[ii] += (long int) ptr[ii].word;
    }

    cntdec1++;

    if (cntdec1 == N){
        for (ii = 0; ii < 6; ii++) {
            Acc[ii] = (int)(accsoma1[ii]/N);
            accsoma1[ii] = 0;
        }
        cntdec1 = 0;
    }
}

```

6.8.5 – Rotina ler e enviar pela serial – rotina de teste

Para teste do funcionamento do hardware e das rotinas básicas foi utilizado o código a seguir. A única função é realizar configurações e aguardar chamada do Timer. Quando a variável `cntdec0` zero, são enviados a média dos últimos 20 valores lidos.

```

/* ----- Programa Principal */
void main()
{
    DisableInterrupts; /* Desabilita as Interrupções */
    PimConfig();        /* Configura as Portas */
    ClockConfig();      /* Configura PLL e Clock */
    SerialConfig();     /* Configura a Serial 0 */
    SpiConfig();        /* Configura a SPI */
    TimerModConfig();   /* Configura o Timer Modulos */
    PrintStr(StrIni);   /* Imprime String de Inicialização */
    EnableInterrupts;   /* Habilita Interrupções */

    for(;;) {
        if (cntdec0 == 0){
            sprintf(str, "%5d\t%5d\t%5d\t%5d\t%5d\t%5d\t",
                    Mag[0], Mag[1], Mag[2], Mag[3], Mag[4], Mag[5]);
            PrintStr(str);

            sprintf(str, "%5d\t%5d\t%5d\t%5d\t%5d\t%5d\r\n",
                    Acc[0], Acc[1], Acc[2], Acc[3], Acc[4], Acc[5]);
            PrintStr(str);
        }
    }
}

```

6.9 – Implementação de Algoritmos de Atitude

6.9.1 – Implementação do algoritmo TRIAD

Para implementar o algoritmo TRIAD que retorne o quatérnion de Atitude são necessárias funções intermediárias como o produto interno, produto vetorial e multiplicação de matrizes dentre outras. O principal disso tudo é que as divisões, multiplicações, somas e subtrações serão todas feitas em ponto flutuante (float), ou seja as variáveis serão do tipo `float` e portanto a velocidade das operações cairá muito e o sistema chegará ao seu limite.

Para manter o cálculo do quatérnion em tempo-real é imprescindível que todas as operações do TRIAD sejam concluídas antes que as funções de decimação resultem

novos dados filtrados. Assim é necessário rodar o algoritmo TRIAD em menos de 1/50 Hz (que é a taxa de dados filtrados), ou seja, menos de 0,02 segundos.

O algoritmo TRIAD necessita dos valores dos vetores lidos na referência, ou seja, os vetores $\hat{\mathbf{v}}_1$ e $\hat{\mathbf{v}}_2$. Esses vetores podem ser obtidos colocando os eixos dos sensores alinhados com o sistema de referência e resetando o sistema.

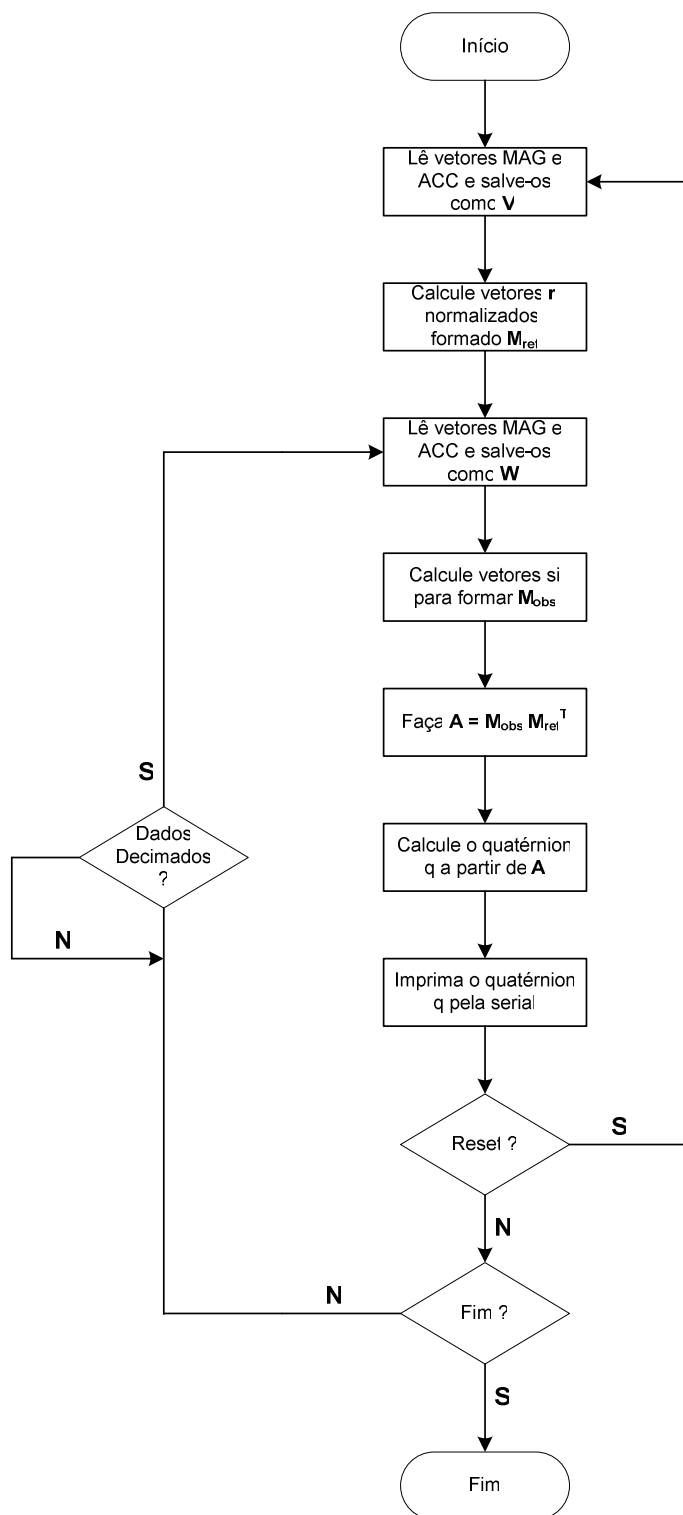


Figura 6.35 – Fluxograma implementado para algoritmo TRIAD

A partir do Reset, os primeiros vetores de dados lidos são considerados $\hat{\mathbf{v}}_i$ e todos os demais que serão lidos são considerados $\hat{\mathbf{w}}_i$. O fluxograma básico montado para o TRIAD é mostrado na Figura 6.35.

6.9.2 – Código C para o algoritmo TRIAD

A. Código de inicialização da matriz de vetores de referência

```

/* Inicialização do V1 e V2 */
V1[X] = MAG[X];
V1[Y] = MAG[Y];
V1[Z] = MAG[Z];

V2[X] = ACC[X];
V2[Y] = ACC[Y];
V2[Z] = ACC[Z];

/* Normalização do V1 e V2 */
N = norm(V1);          // função de normalização
V1[X] /= N;
V1[Y] /= N;
V1[Z] /= N;

N = norm(V2);
V2[X] /= N;
V2[Y] /= N;
V2[Z] /= N;

/* Construção da Matriz Mref com r */
r[0][X] = V1[X];
r[0][Y] = V1[Y];
r[0][Z] = V1[Z];
cross(r[1], V1, V2);   // Função produto vetorial
N = norm(r[1]);
r[1][X] /= N;
r[1][Y] /= N;
r[1][Z] /= N;
cross(r[2], r[0], r[1]);

```

B. Código para calcular quatérnion indefinidamente e enviá-lo para serial

```
for (;;) {  
    /* Aguarda decimação */  
    if (m == 0 && flag == 0) {  
  
        Pim.ptt.bit.ptt4 = 1;    // Pisca LED  
  
        /* Construção e normalização dos vetores W1 e W2 */  
        W1[X] = MAG[X];  
        W1[Y] = MAG[Y];  
        W1[Z] = MAG[Z];  
  
        W2[X] = V2[X];  
        W2[Y] = V2[Y];  
        W2[Z] = V2[Z];  
  
        N = norm(W1);  
  
        W1[X] /= N;  
        W1[Y] /= N;  
        W1[Z] /= N;  
  
        N = norm(W2);  
  
        W2[X] /= N;  
        W2[Y] /= N;  
        W2[Z] /= N;  
  
        /* Construção da Matriz Mobs com si */  
        s[0][X] = W1[X];  
        s[0][Y] = W1[Y];  
        s[0][Z] = W1[Z];  
  
        cross(s[1], W1, W2);  
  
        N = norm(s[1]);  
  
        s[1][X] /= N;  
        s[1][Y] /= N;  
        s[1][Z] /= N;  
    }
```

```

    cross(s[2], s[0], s[1]);

    /* Cálculo da Matriz de atitude */
    matmul(A,s,r);

    /* Obtenção do quatérnion de Atitude */
    q[0] = 0.5*sqrtf(A[0][0] + A[1][1] + A[2][2] + 1);
    q[1] = (A[1][2] - A[2][1])/(4*q[0]);
    q[2] = (A[2][0] - A[0][2])/(4*q[0]);
    q[3] = (A[0][1] - A[1][0])/(4*q[0]);

    sprintf(str, "q = [%f,%f,%f,%f]\r\n",q[0], q[1], q[2], q[3]);
    //if(GetChar() != 0) // sincronismo opcional
        PrintStr(str);
    }
}

```

C. Funções e Macros utilizadas

```

#define norm(arg) sqrtf(arg[X]*arg[X] + arg[Y]*arg[Y] +
arg[Z]*arg[Z]);

#define cross(res,arg1,arg2) \
    res[X] = arg1[Y]*arg2[Z] - arg2[Y]*arg1[Z]; \
    res[Y] = arg1[Z]*arg2[X] - arg2[Z]*arg1[X]; \
    res[Z] = arg1[X]*arg2[Y] - arg2[X]*arg1[Y];

void matmul(float res[][3], float s[][3], float r[][3]) {
    int ii, jj, kk;
    for (ii = 0; ii < 3; ii++) {
        for (jj = 0; jj < 3; jj++) {
            res[ii][jj] = 0;
            for (kk = 0; kk < 3; kk++) {
                res[ii][jj] += s[ii][kk]*r[jj][kk];
            }
        }
    }
}

```

6.10 – Ensaio e Testes

6.10.1 – Testes preliminares – Seguidor de Movimentos

Seguindo a idéia apresentada na Seção 5.11 foi implementado um seguidor de movimentos em tempo real para testar o determinador de atitude. Toda eletrônica do determinador de atitude está montada no interior de um cubo de acrílico. Uma figura cúbica é posicionada na tela e sua atitude é controlada pelo quatérnion que é recebido pela interface serial. Dessa maneira, o cubo virtual visto na tela do computador se movimenta de forma escrava ao movimento do cubo real que porta do determinador de atitude.

No programa mostrado em 6.9.2 (B) na penúltima linha é adicionado um controle de fluxo que envia o quatérnion somente se um caractere tiver sido recebido como um sinal de sincronismo.

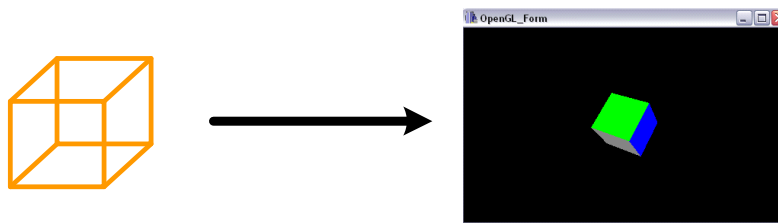


Figura 6.36 – Cubo real e virtual atuando de forma mestre-escravo respectivamente.

Essa implementação mestre-escravo é muito útil para observar se o determinador de atitude funciona. Como observado em testes preliminares, o sistema ainda necessita de alguns ajustes para funcionar perfeitamente. O cubo virtual na tela do computador não segue com fidelidade os movimentos do cubo real, sendo a causa mais provável a descalibração dos sensores.

O *off-set* e a diferença do ganho entre os eixos nos magnetômetros e nos acelerômetros estão prejudicando a leitura da direção dos vetores fonte. Outras causas apontam para distorção do campo magnético próximo ao sensor devido à proximidade a materiais ferromagnéticos.

O próximo passo é identificar os problemas e saná-los.

6.10.2 – Ensaio em mesa de Rotação

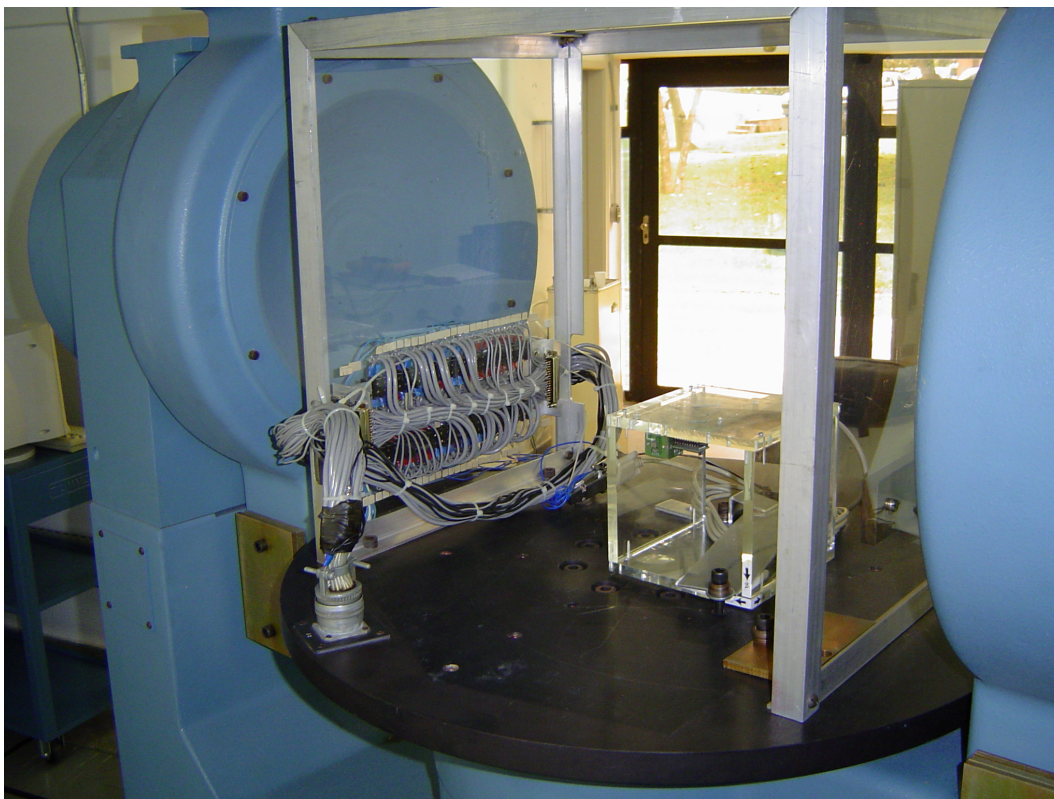


Figura 6.37 – Ensaio em mesas de rotação. Cubo no interior da mesa de simulação de três eixos.



Figura 6.38 – A segunda foto traz todo o sistema, inclusive o *notebook* de aquisição sobre o prato da mesa de mancal a ar de um eixo.

A. Introdução

Embora seja possível girar o hardware de determinação de atitude (cubo) sobre uma mesa qualquer para realização de alguns testes, quando é necessário aplicar rotações de ângulos em passos constantes ou então velocidades constantes, não é possível fazer isso sem uso de um equipamento adequado.

Para realização destes ensaios foram utilizados dois tipos de mesa de rotação: uma mesa Contraves® de três eixos com velocidade controlada e uma mesa de um eixo de mancal a ar (mancal flutuante sobre ar comprimido)⁴; ambas podem ser observadas nas Figuras 6.37 e 6.38.

Os ensaios sobre essas mesas foram divididos em quatro modos:

- Rotação em passos constantes com aquisição dos quatérnions;
- Rotação em passos constantes com aquisição dos dados dos sensores;
- Rotação contínua em velocidade constante com aquisição dos quatérnions; e
- Rotação contínua em velocidade constante com aquisição dos dados dos sensores.

A mesa Contraves® de três eixos controlados permite realizar qualquer um dos quatro ensaios acima expostos enquanto a mesa de mancal a ar de um eixo permite apenas os dois últimos sendo ainda a velocidade de rotação desconhecida. A vantagem da utilização da mesa de mancal a ar são pelo menos duas: não possui motores que geram interferências no sensor magnético e não é construída por grandes armações de aço, que também interferem no campo geomagnético. Essa comparação foi feita inserindo uma simples bússola no centro de cada prato rotativo.

B. Procedimento

Nos ensaios no simulador Contraves® de três eixos o determinador de atitude, inserido e alinhado a um cubo de acrílico de 14,5 cm de aresta, foi fixado na mesa o mais próximo possível do centro de rotação, com o eixo z do aparelho alinhado com o eixo do mancal interno do simulador; como visto na Figura 6.38. Um *notebook* permaneceu a pouco mais de um metro ao lado do simulador, e conectado ao aparelho

⁴ Ambas as mesas foram disponibilizadas pelo LabSim do INPE (Instituto Nacional de Pesquisas Espaciais) de S. J. dos Campos por meio do projeto UNIESPAÇO da AEB.

(interface SCI) por meio do sistema de aquisição de dados do simulador. O simulador foi comandado manualmente de modo a desempenhar os movimentos desejados em cada ensaio. Nos ensaios na mesa de mancal a ar, por limitações de espaço, o aparelho teve que ser acomodado sobre uma roda de reação que ali se encontrava desligada e ao lado do *notebook* (Figura 6.38). Em ambos os casos o aparelho foi posicionado de modo a que os magnetômetros ficassem distantes da base a fim de reduzir possíveis interferências. Os ensaios duraram cerca de um a cinco minutos cada e os dados foram adquiridos à taxa de amostragem de 50 Hz, com exceção de dois ensaios, os quais foram adquiridos à taxa variável por meio de comandos manuais via teclado do *notebook*.

C. Organização

Os dados adquiridos em cada ensaio foram armazenados em arquivos separados e em formato livre. Cada ensaio foi nomeado com uma letra maiúscula na sequência. A Tabela 6.4 traz essa organização. Os arquivos dos ensaios B a K contêm os elementos do quaternião de atitude: as componentes da parte vetorial q_1 , q_2 e q_3 e a componente escalar q_0 , nesta ordem, e com uma linha por instante de tempo. Os arquivos dos ensaios de L a Q contêm as leituras x , y e z do magnetômetros seguidas pelas dos acelerômetros e dos giros, uma linha por instante de tempo. As letras omissas correspondem a tomadas de dados interrompidas por razões operacionais. A conversão dos dados dos sensores para unidades de engenharia é mostrada na Tabela 6.5.

Tabela 6.4 – Configuração dos ensaios

#	Nome do arquivo	Descrição	Posição
B	labsim_aqs_quaternion2	passos de 30 graus em torno da vertical local	mancal interno
C	labsim_aqs_quaternion3	20 graus/s contínuo em torno da vertical local	mancal interno
E	labsim_quaternion_50hz_2	20 graus/s contínuo em torno da vertical local	mancal externo
F	labsim_quaternion_50hz_3	passos de 30 graus em torno da vertical local	mancal externo
G	labsim_quaternion_50hz_4	20 graus/s contínuo em	mancal interno

		torno de eixo inclinado (45 graus)	
I	labsim_quaternion_50hz_6	de 30 graus em torno de eixo inclinado (45 graus)	mancal interno
K	labsim_mancal_ar2	rotação contínua irregular, mesa de mancal a ar 1 eixo	
L	labsim_aqs_interno	na maior parte parado, trecho útil a 20 graus por segundo em torno da vertical local	–
N	labsim_aqs_interno3	20 graus/s contínuo em torno da vertical local	mancal interno
O	labsim_aqs_interno_passo	passos de 30 graus em torno da vertical local	mancal interno
P	labsim_aqs_externo_passo	passos de 30 graus em torno da vertical local	mancal externo
Q	labsim_aqs_externo	20 graus/s contínuo em torno da vertical local	mancal externo
R	labsim_gyro_x	de 0 a 30°/s ida e volta a passos de 5°/s	mancal externo, eixo do mancal interno na horizontal, eixo x do cubo na vertical local
S	labsim_gyro_y	de 0 a 30°/s ida e volta a passos de 5°/s, mancal externo	eixo do mancal interno na horizontal, eixo y do cubo na vertical
T	labsim_gyro_z	de 0 a 30°/s ida e volta a passos de 5°/s	Mancal interno na vertical
U	labsim_gyro_z_g	de 0 a 30°/s ida e volta, passos de 5°/s	mancal interno na horizontal

Tabela 6.5 – Conversão para unidades de engenharia

Sensor	Símbolo	Parâmetro	Valor
Magnetômetro	μ_1	Tensão de saída	1 V/Gauss

	μ_2	Ganho do amplificador	8
	μ_3	Fator de conversão A/D	5 V / 65536
	$\mu = \mu_3 / (\mu_2 \mu_1)$	Fator de conversão final	9,54 μ Gauss por nível de quantização
Acelerômetro	α_1	Tensão de saída	312 mV/g
	α_2	Fator de conversão A/D	5 V / 65536
	g	Aceleração da gravidade	9,8 m/s ²
	$\alpha = g \alpha_2 / \alpha_1$	Fator de conversão final	2,40 mm/s ² por nível de quantização
Giro	γ_1	Tensão de saída	0,67 mV/(°/s)
	γ_2	Ganho do amplificador	10
	γ_3	Fator de conversão A/D	5 V / 65536
	$\gamma = \gamma_3 / (\gamma_2 \gamma_1)$	Fator de conversão final	0,0114°/s por nível de quantização

D. Análise dos Dados

A seguir serão expostos dados de alguns ensaios de B a U (Tabela 6.4).

Ensaio B

Neste ensaio foram realizados 12 passos discretos de 30° em torno do eixo z (eixo da gravidade) de forma a dar uma volta completa. A Figura 6.39 mostra as componentes do quatérnion durante esse movimento. Como os passos são constantes entorno de um eixo fixo era esperado que a figura formasse um pirâmide. Na Figura 6.40 são plotadas as seqüências de deslocamento do ângulo. Neste caso os passos

deveriam ser de 30° entre um degrau e outro, chegando a 180° e depois retornando a 0° com o mesmo número de degraus entre um lado e o outro.

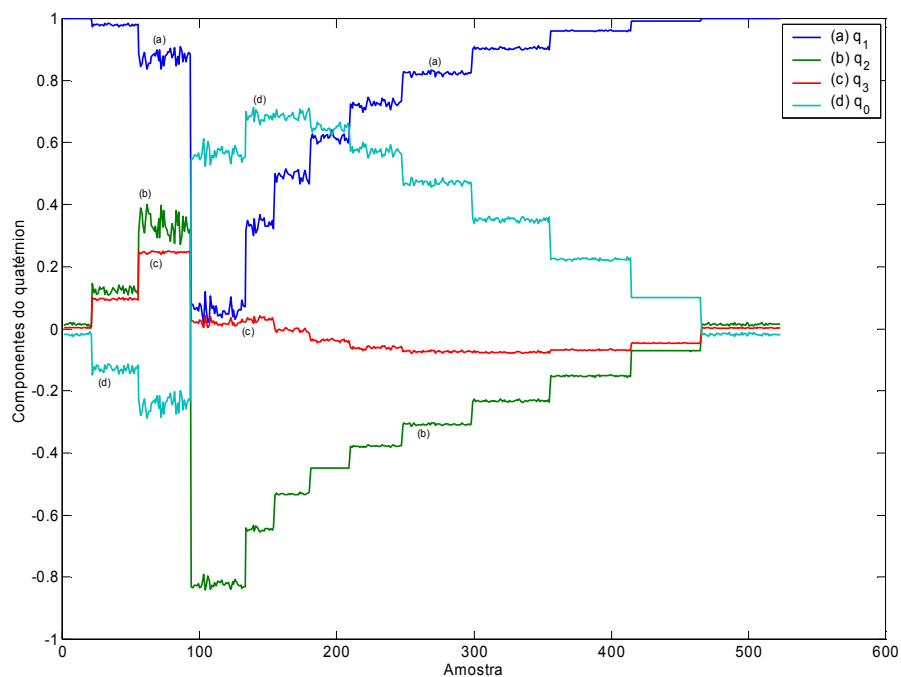


Figura 6.39 – Componentes do quatérnio adquiridas durante rotação em passos de 30° entorno da gravidade (eixo z).

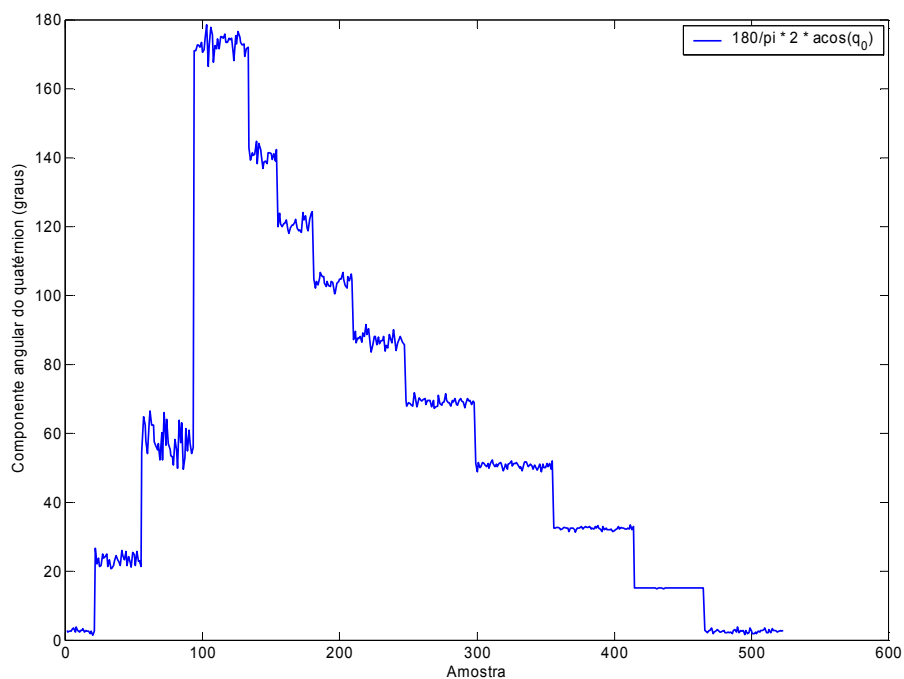


Figura 6.40 – Componente angular do quatérnio plotada em graus.

Ensaio C

Neste caso foi efetuada uma rotação contínua a 20°/s entorno do eixo z. A Figura 6.41 mostra as componentes do quatérnion neste caso. A Figura 6.42 mostra apenas a componente escalar do quatérnion convertida em graus.

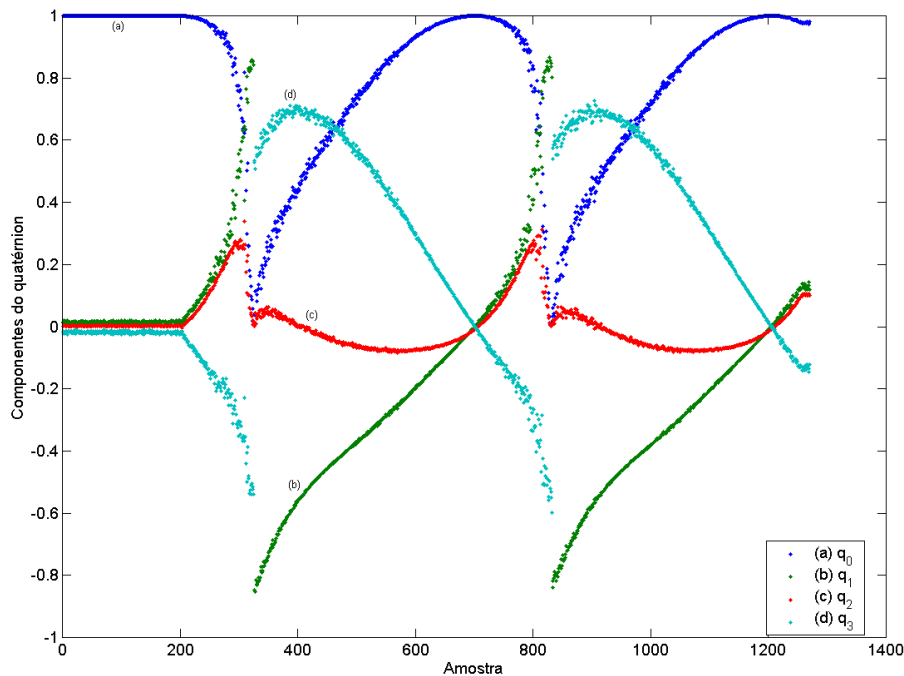


Figura 6.41 – Componentes do quatérnion durante giro contínuo a 20 °/s entorno do eixo z.

No caso da Figura 6.42 também era esperado uma aparência semelhante a um triângulo isóscele, já que a velocidade de rotação da mesa é absolutamente constante.

As outras três componentes do quatérnion podem ser analisadas plotando a direção do eixo de rotação normalizado sobre a superfície de uma esfera unitária, como visto na Figura 6.42. Neste caso também podemos observar que a direção do vetor de rotação, que a todo o momento é a direção do eixo z, não se mantém constante e nem está sobre o pólo como deveria.

Um possível motivo para a distorção do triângulo e para o deslocamento do eixo de rotação, é a interferência sofrida pelos eixos magnetômetros juntamente com a descalibração dos sensores. Fica comprovada a existência de *off-sets* nos canais e também a diferença nos ganhos em cada canal.

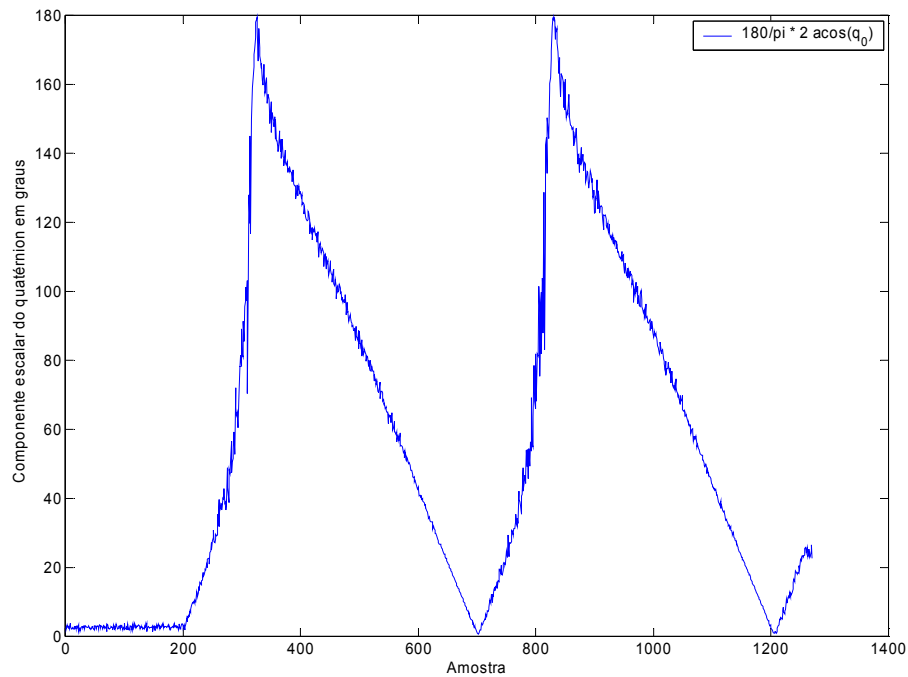


Figura 6.42 – Componente escalar do quatérnion convertida em graus para o ensaio de 20°/s contínuo entorno do eixo z.

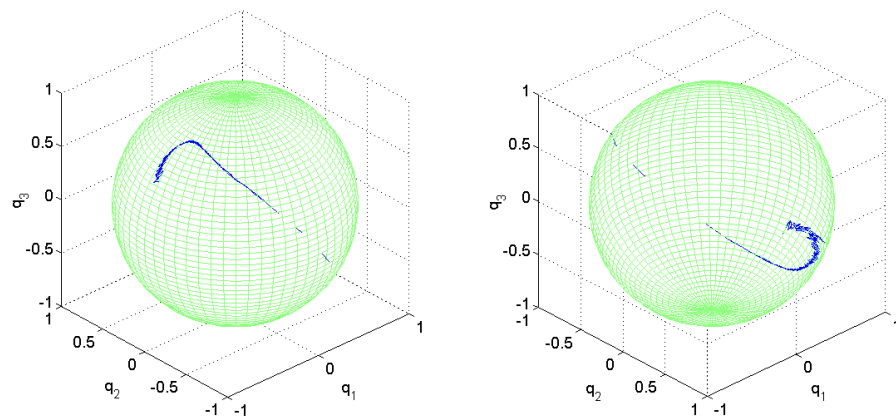


Figura 6.43 – Componente vetorial do quatérnion normalizada. A direção do vetor se movimenta.

Ensaio E

Neste ensaio o ensaio C é repetido, mas girando a mesa entorno de z movimentando o mancal externo. Como são motores diferentes e o tamanho do mancal de aço que se move é bem maior que o interno, será possível verificar o quanto o

mecanismo de ensaio está interferindo nos dados. A Figura 6.44 mostra as componentes do quatérnion.

A Figura 6.45 apresenta somente a componente escalar convertida em graus.

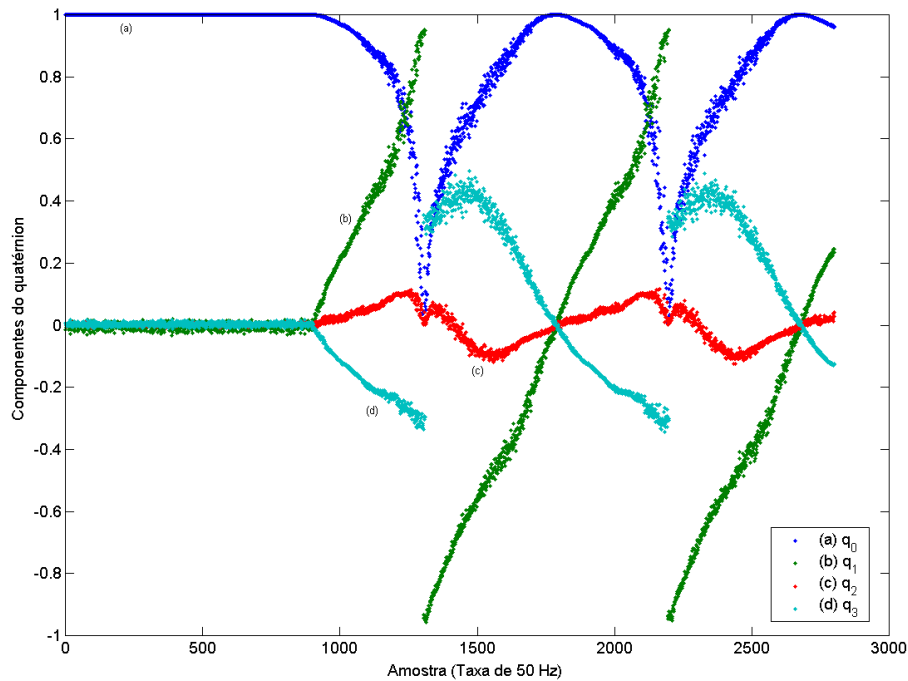


Figura 6.44 – Componentes do quatérnion adquiridas durante giro em velocidade constante entorno de z. Mecanismo movido por mancal externo.

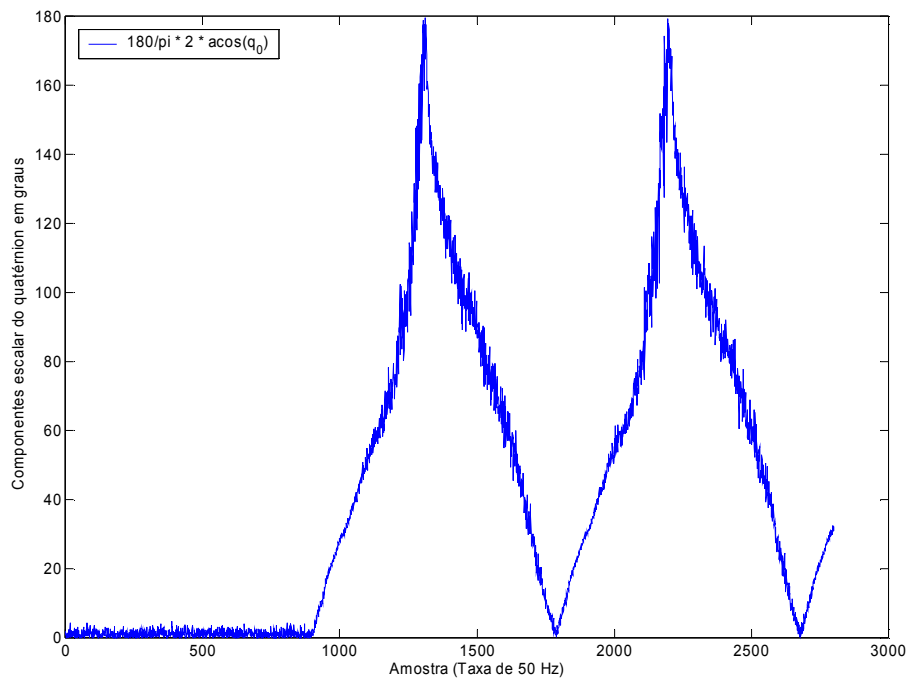


Figura 6.45 – Componente escalar do quatérnion convertida em graus. Movimento contínuo, a 20°/s, do mancal externo.

A Figura 6.46 mostra a componente vetorial do quatérnion normalizada plotada sobre uma esfera unitária.

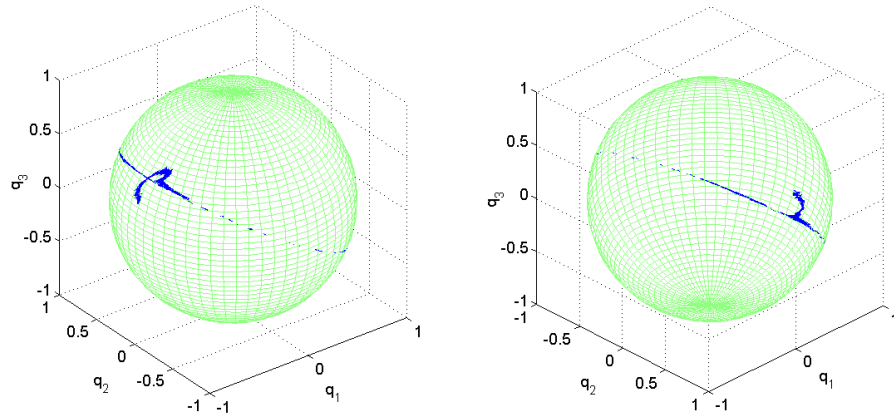


Figura 6.46 – Componente vetorial do quatérnion normalizada. Rotação mancal externo. A direção da componente vetorial do quatérnion se movimenta.

Comparando os ensaios C e E pouco pode-se concluir. O quatérnion manteve praticamente o mesmo desempenho. O ruído aumentou de C para E porque a taxa de aquisição praticamente dobrou.

Ensaio K

A Figura 6.47 traz as componentes do quatérnion adquirido durante rotação entorno do eixo z sobre a mesa de mancal a ar. Essa mesa não tem motores e o prato de suporte é de alumínio. Embora não seja possível impor à essa mesa uma velocidade de rotação constante, ela permite que a velocidade imposta ao prato, por meio de um impulso manual, se mantenha indefinidamente devido ao ínfimo atrito que a mesa de mancal a ar possui.

Observando a Figura 6.47 pode-se notar que as componentes estão com um formato mais senoidal do que em ensaios anteriores, como da Figura 6.41 e 6.44. A componente angular em graus mostrada na Figura 6.48 está muito próxima ao formato desejado de triângulo. Pouca distorção é notada, mas observando cuidadosamente, vê-se que o triângulo não é simétrico na base. Isto é evidenciado pela reta pontilhada traçada do cume à base, que não divide o triângulo em partes iguais.

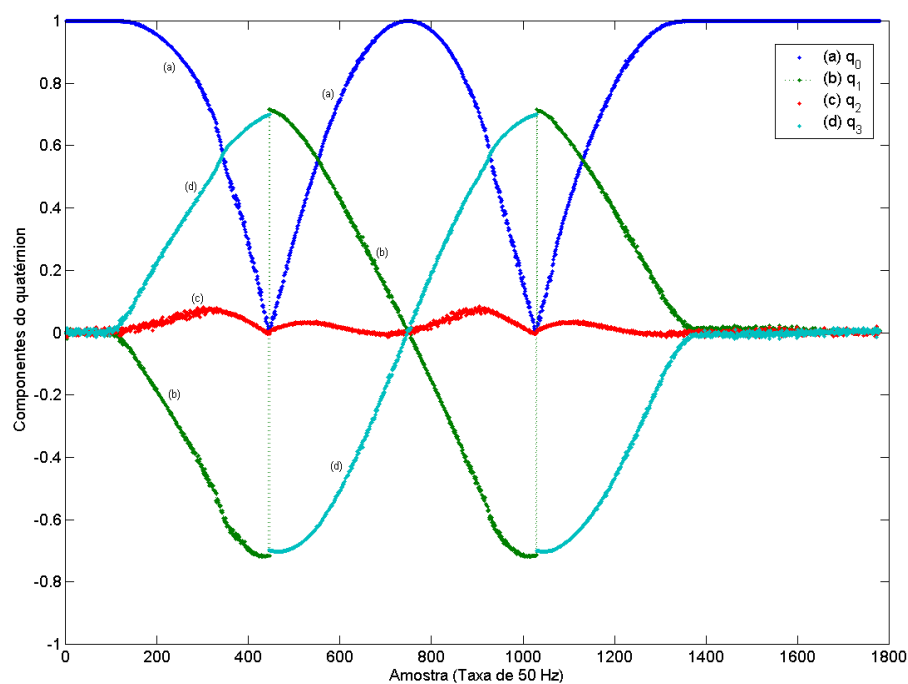


Figura 6.47 – Componentes do quaternião adquiridas em giro entorno do eixo z sobre a mesa de mancal ar. Velocidade de rotação constante.

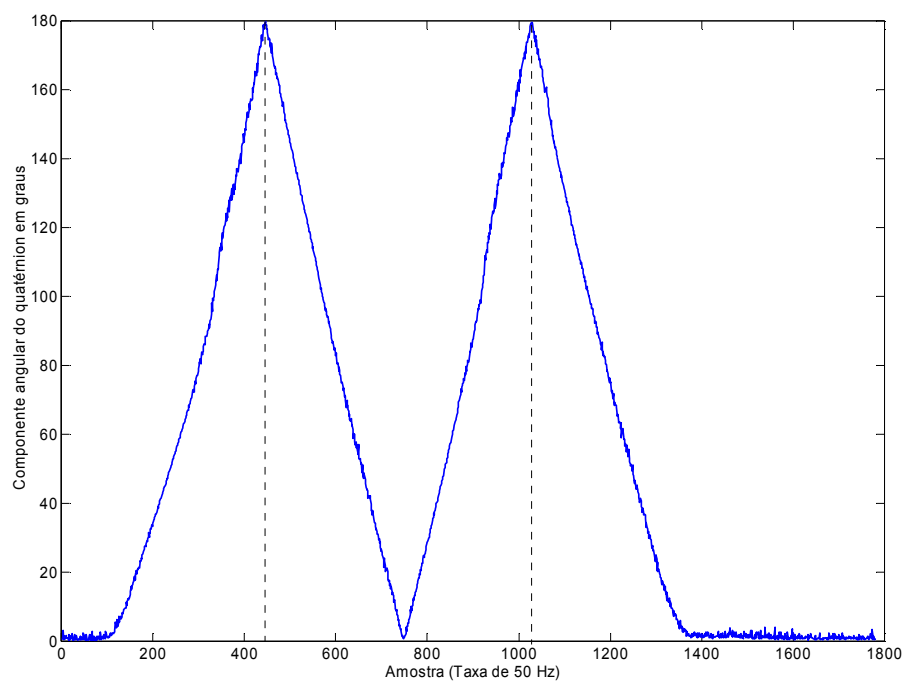


Figura 6.48 – Componente angular do quaternião adquirida durante rotação entorno do eixo z sobre a mesa de mancal a ar.

Pelo fato do quatérnion ser uma entidade unitária, um desvio em uma componente é absorvido pelas outras três. A Figura 6.49 trás a componente vetorial normalizada plotada sobre a superfície de uma esfera unitária, como já feito em outros ensaios. O eixo de rotação está deslocado, indicando que algo está errado e não é apenas distorção do campo magnético.

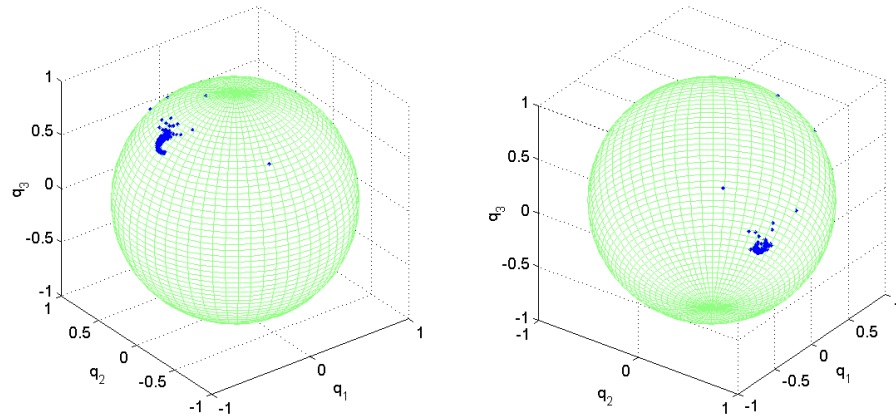


Figura 6.49 – Componente vetorial normalizada plotada sobre a superfície de uma esfera unitária.

Ensaio N

Neste ensaio foram adquiridos dados dos magnetômetros e acelerômetros para a partir deles observar o que pode estar afetando o quatérnion. Na Figura 6.50 tem-se as componentes lidas pelos magnetômetros em x, y e z. Como a rotação é entorno do eixo z, a componente de campo em z não deveria variar. Isto claramente é devido a uma forte interferência magnética causa provavelmente pelo motor da mesa. Outro fato que comprova a interferência é o *off-set* deslocado das senoides magnéticas.

A Figura 6.51 mostra as componentes de aceleração. Pelo fato da rotação ser entorno do eixo z, somente o acelerômetro z deve medir a gravidade. Os outro dois acelerômetros devem medir 0g. Embora o eixo x tenha um forte *off-set*, nota-se que não houve variação aparente nas medidas dos acelerômetros durante o movimento. O *off-set* do eixo x causaria erros no quatérnion se o movimento saísse do plano normal a z. A escala de aceleração está incorreta, pois o eixo z deveria estar próximo a -1 g e não $-0,27\text{ g}$.

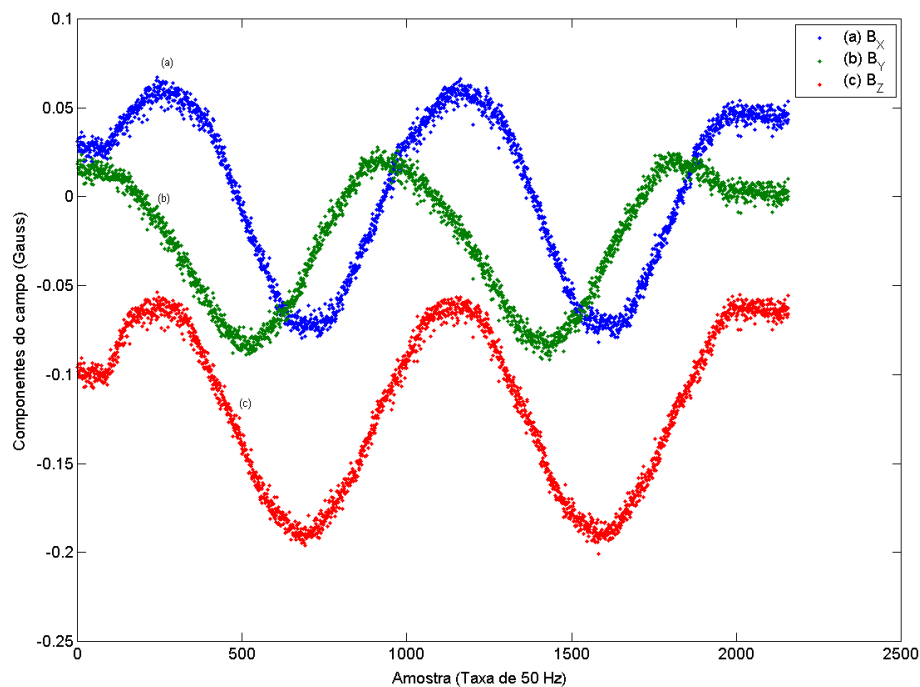


Figura 6.50 – Componentes do campo magnético medidas durante rotação entorno do eixo z pela movimentação do mancal interno. A componente z não deveria variar na realização desse movimento.

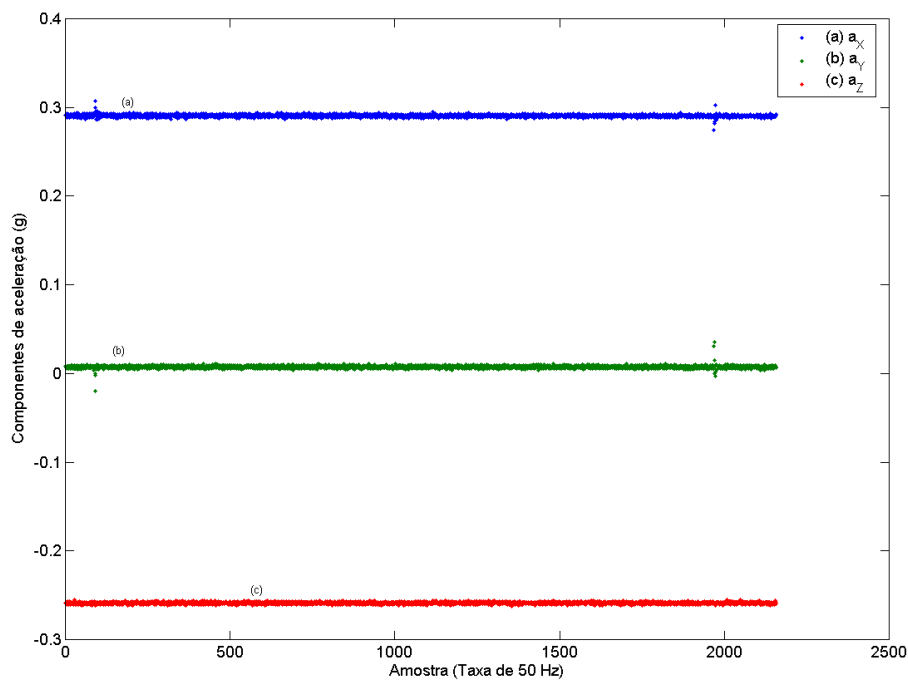


Figura 6.51 – Componentes de aceleração (gravidade). Somente o eixo z está medindo a gravidade. Os demais deveriam ser zero caso não houvesse *off-set*.

Ensaio Q

Este ensaio é idêntico ao ensaio N, mas o movimento da mesa é feito pelo mancal externo, que teoricamente produz menos interferência. Os gráfico 6.52 apresenta as componentes do campo magnético medidas. As componentes de aceleração obtidas foram idênticas às da Figura 6.51 e portanto serão omitidas. O campo magnético mudou completamente seu comportamento. Amplitude das componentes em x e em y aumentaram, enquanto a amplitude da componente z diminuiu substancialmente.

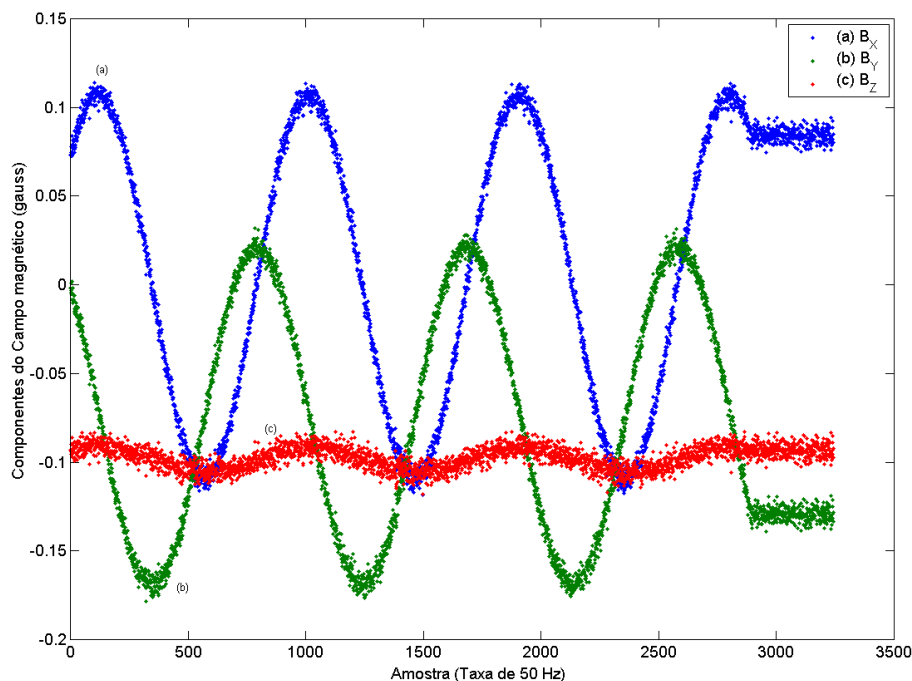


Figura 6.52 – Componentes do campo magnético medidas durante rotação do mancal externo em torno do eixo z.

A Figura 6.53 apresenta as componentes do ensaio N e Q plotadas tridimensionalmente em ângulos diversos. As Figuras 6.54 e 6.55 apresentam o quatérnio reconstituído dos ensaios N e Q respectivamente, onde nota-se altíssimas distorções causadas pela falta de calibração dos níveis de *off-set* e ganho de cada canal. Esperava-se observar nas Figura 6.53 duas circunferências perpendiculares e concêntricas.

A calibração dos sensores magnéticos e acelerométricos consiste em **zerar** o off-set e **igualar** os ganhos dos sensores que medem as mesmas grandezas.

Uma técnica de calibração será vista a seguir.

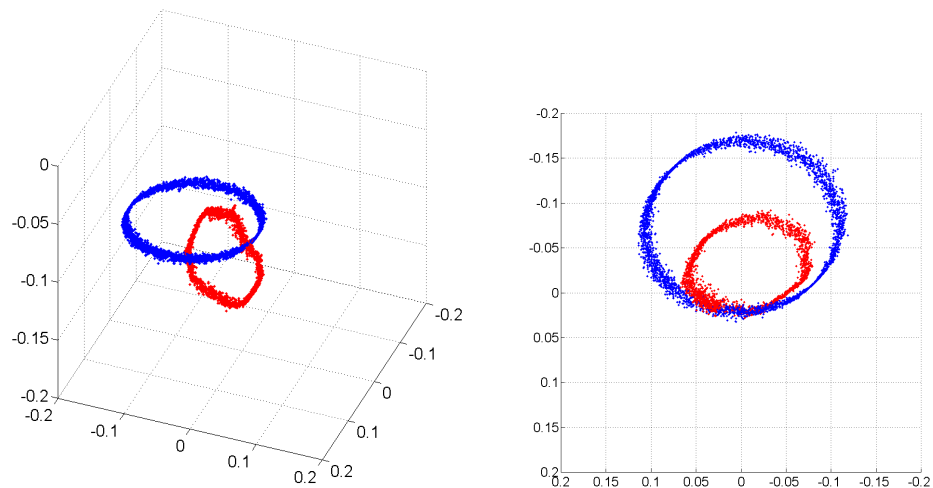


Figura 6.53 – Comparação entre as componentes do campo magnético medidas no ensaio N e Q. Dois pontos de vista.

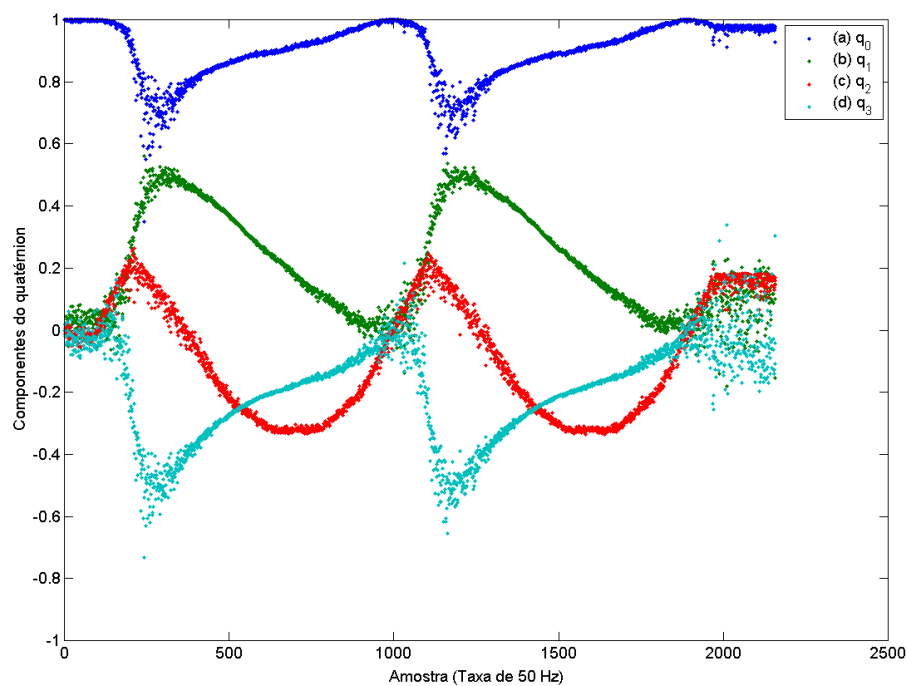


Figura 6.54 – Componentes do quatérnion obtidas a partir dos dados dos sensores magnéticos e de aceleração do ensaio N.

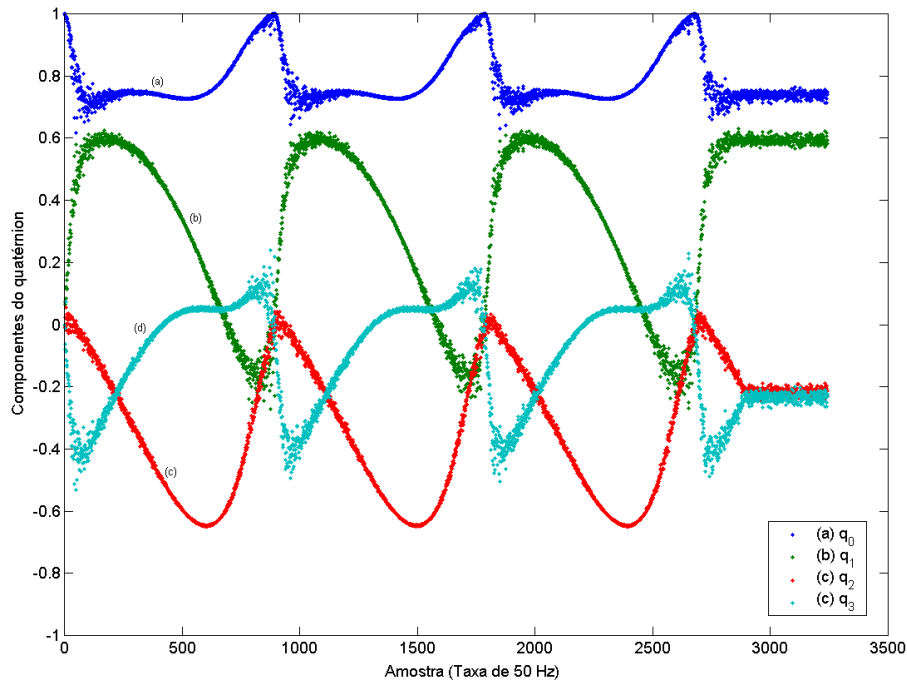


Figura 6.55 – Componentes do quaternião obtidas a partir dos dados dos sensores magnéticos e de aceleração do ensaio Q.

Técnica empregada na calibração

Para medir o *off-set* basta manter a entrada nula e ler o sinal de saída. Quando não é possível zerar a entrada (como no caso do sensor magnético), uma oscilação aplicada à entrada pode indicar o *off-set* através da seguinte relação

$$O_i = (\langle V_{i\max} \rangle + \langle V_{i\min} \rangle) / 2 \quad (6.20)$$

onde O_i é *off-set* e V_i o sinal oscilante no ponto em questão. Para o caso do acelerômetro, não é possível saber o *off-set* do canal z pois o sinal permaneceu constante durante o ensaio, mas sabe-se que os canais x e y deveriam estar em 0, portando neste caso

$$O_i = \langle V_i \rangle_{[\text{sem excitação}]} \quad (6.21)$$

Retirando o *off-set* dos sinais do ensaio Q, tem-se nas Figura 6.56 e 6.57 os sinais sem *off-set* dos magnetômetros e acelerômetros respectivamente. A Figura 6.58 mostra agora o quaternião calculado a partir destes dados processados, sem *off-set*.

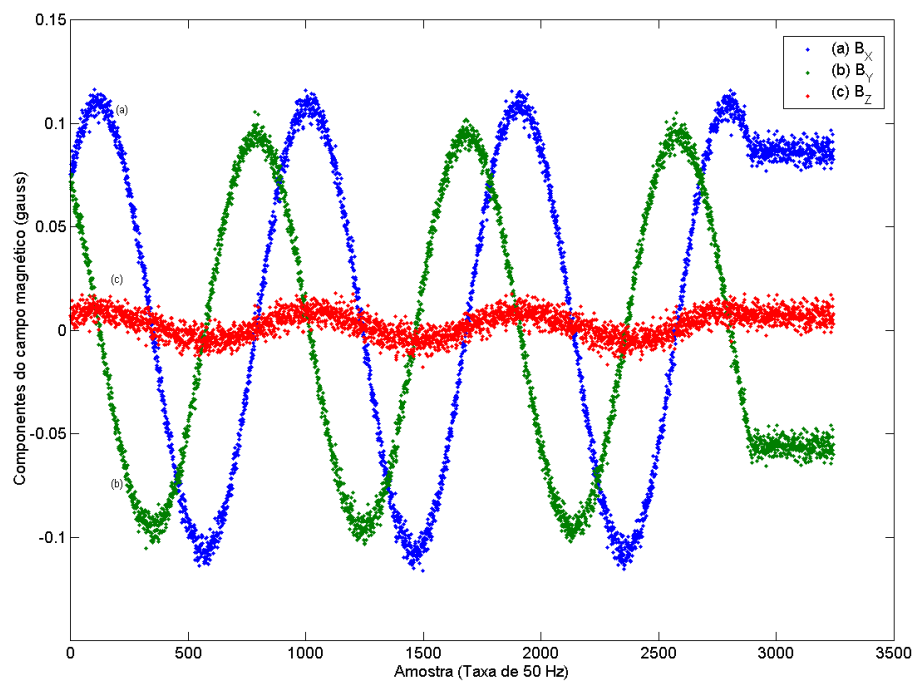


Figura 6.56 – Componentes do campo magnético sem *off-set*. Dados do ensaio Q.

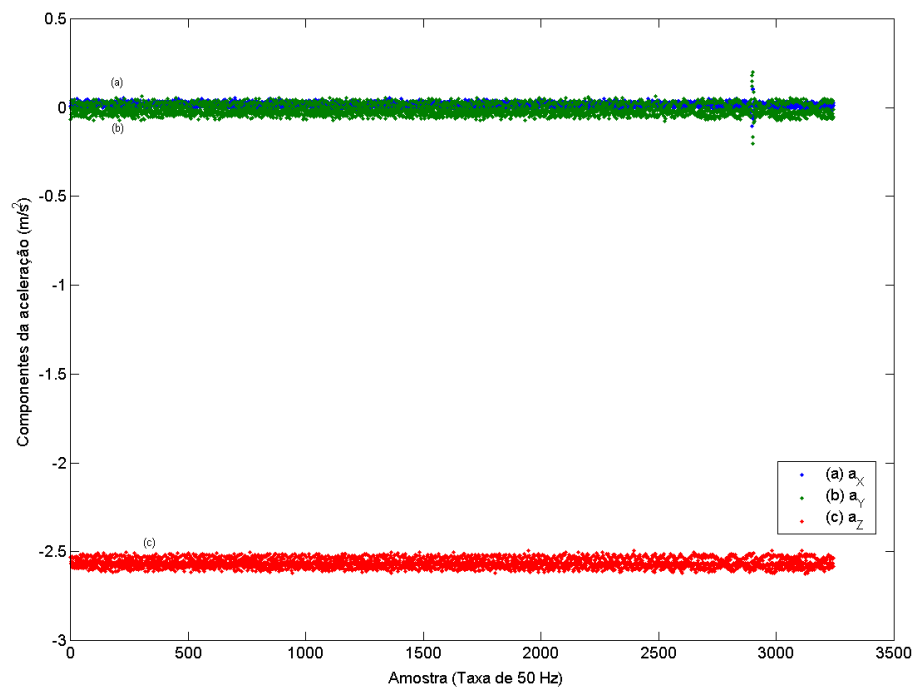


Figura 6.57 – Componentes de aceleração sem *off-set*. Dados do ensaio Q.

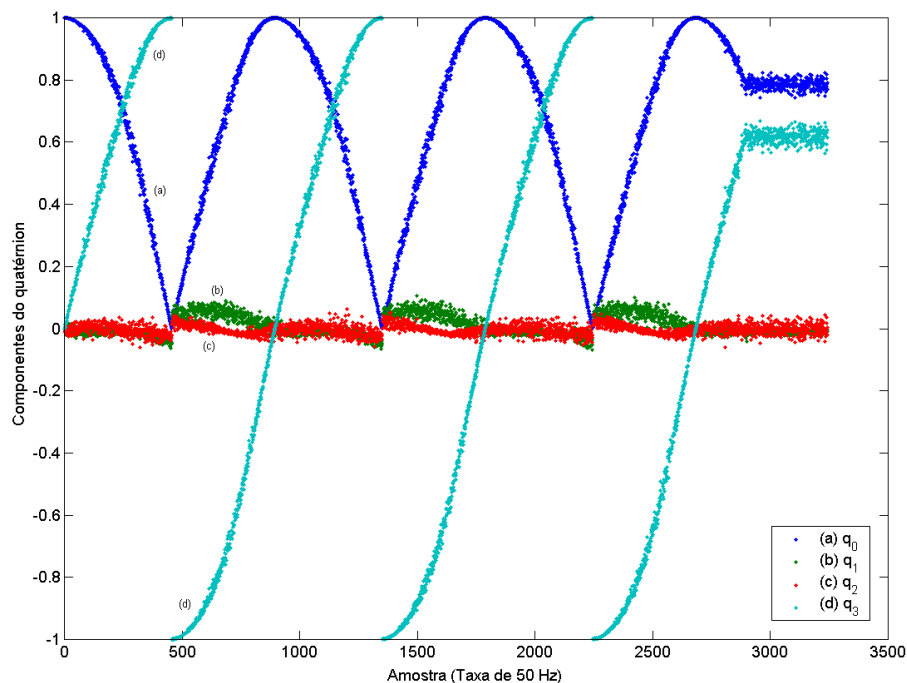


Figura 6.58 – Componentes do quatérnio recuperado com base nos dados do ensaio Q sem *off-set* nos sensores.

Para melhor visualização observe a Figura 6.59 onde a componente angular do quatérnio foi plotada em graus e a Figura 6.60 onde a componente vetorial normalizada do quatérnio fora plotada sobre a esfera unitária. Ambos os gráficos mostram que as seqüências de quatérnions calculados têm como média o vetor z como direção e o ângulo de rotação progride linearmente alternando a cada 180°, como era esperado em todos os ensaios.

Nota-se na comparação entre as Figuras 6.48 e 6.59 e na comparação de 6.49 com 6.60 que o *off-set* influi bastante na determinação do quatérnio de atitude. Um outro fator imprescindível é que o ganho seja idêntico em cada canal, em cada tríade de sensores. Observando a Figura 6.56 é possível notar que os canais x e y da tríade de magnetômetros possuem ganhos próximos, mas distintos. Como neste caso, sabe-se que os sensores estão no mesmo plano e sujeitos ao mesmo campo de excitação, pode-se encontrar o ganho dos canais x e y através da diferença do máximo e do mínimo em cada canal,

$$G_i = (\langle V_{i\max} \rangle - \langle V_{i\min} \rangle) / 2 \quad (6.22)$$

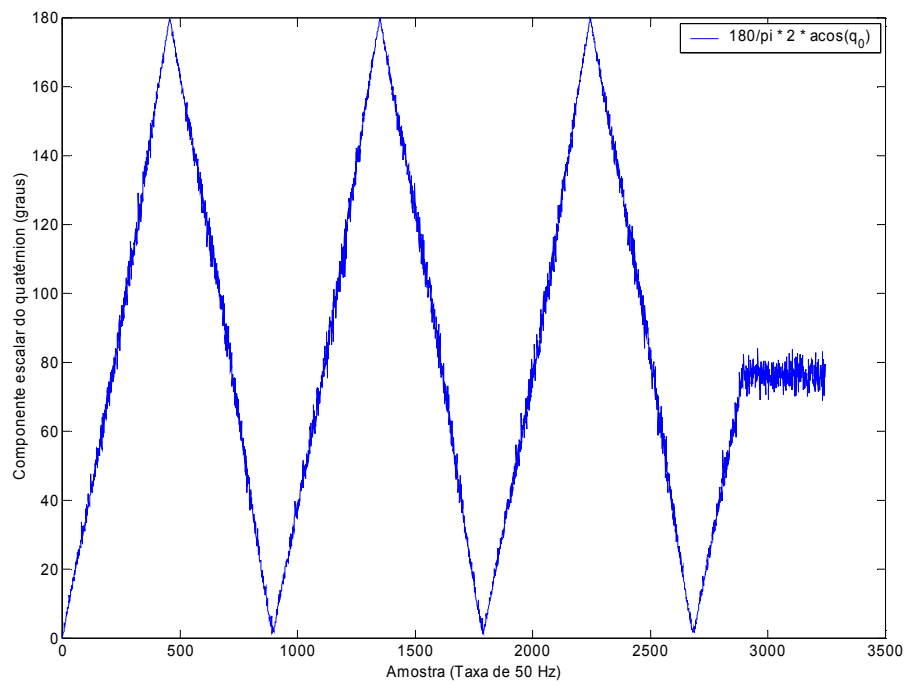


Figura 6.59 – Componente escalar do quatérnio recuperado dos dados do ensaio Q sem *off-set* nos sensores.

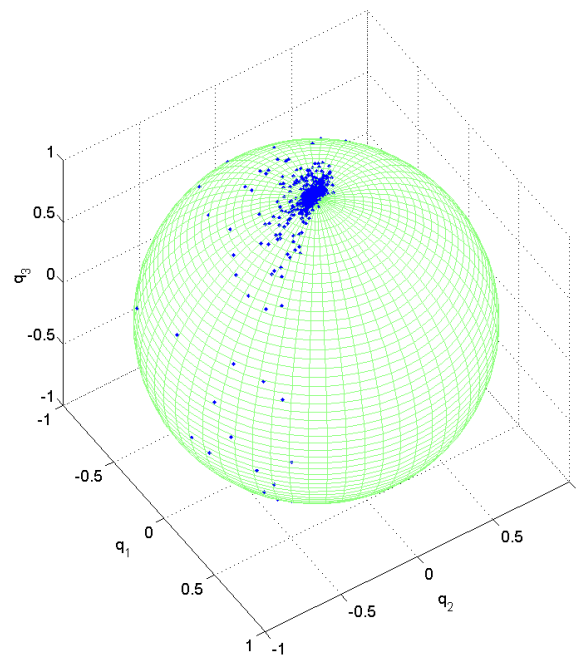


Figura 6.60 – Componente vetorial do quatérnio recuperado dos dados do ensaio Q sem *off-set* nos sensores.

Encontrando o ganho de cada canal e multiplicando x por G_y/G_x tem-se na Figura 6.61 mesma amplitude de excursão. Como a rotação foi entorno do eixo z, o mesmo não será calibrado em ganho, neste caso.

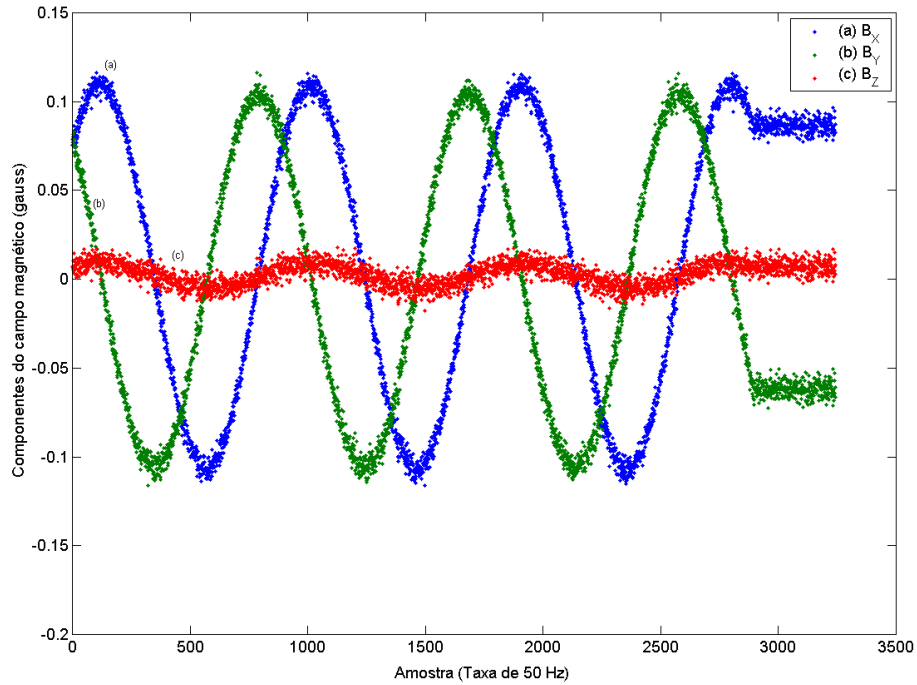


Figura 6.61 - Componentes do campo magnético sem *off-set*. Ganho do sensor em x igualado ao ganho do sensor em y. Dados do ensaio Q.

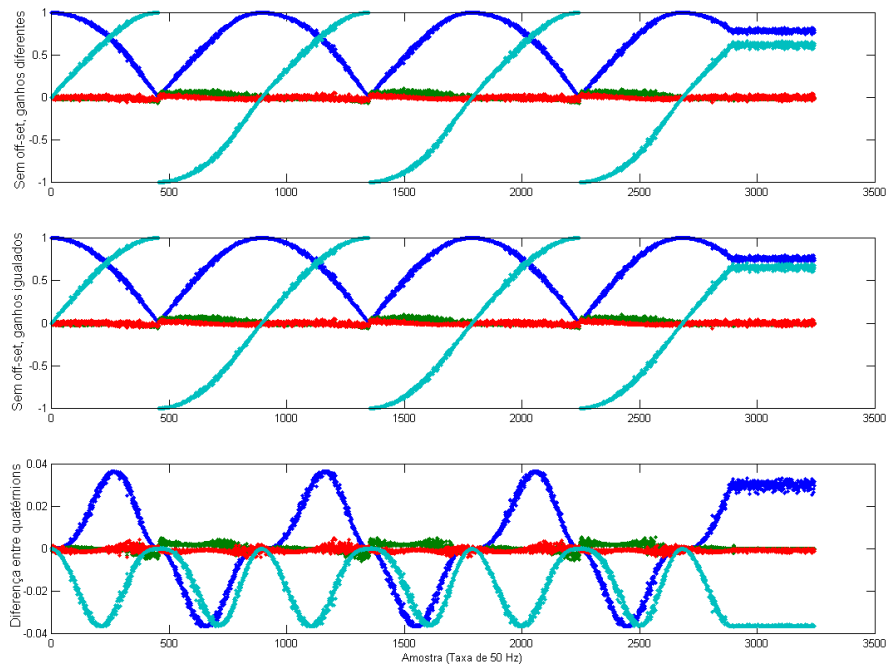


Figura 6.62 – Comparação entre quatérnion calculado a partir de dados de sensores com ganhos iguais e com ganhos diferentes. Dados sem *off-set* em ambos.

A Figura 6.62 traz o quatérnion dos dados do ensaio Q calculados de duas formas: com ganhos iguais nos canais dos magnetômetros e com ganhos diferentes. É possível observar pelo terceiro gráfico que a diferença entre os ganhos causam um erro significativo.

De forma geral, os erros observados em todos os ensaios têm ligação direta com o *off-set* e fator de escala dos sensores. Uma tríade de sensores calibrados sempre lê um vetor que quando normalizado resulta sempre o mesmo módulo, independente da atitude que a tríade esteja. Matematicamente, pode-se dizer que uma tríade calibrada lê pontos sobre a superfície de uma esfera unitária centrada na origem, cujas componentes do campo obedecem a relação

$$\frac{B_X^2}{B_{Total}^2} + \frac{B_Y^2}{B_{Total}^2} + \frac{B_Z^2}{B_{Total}^2} = 1 \quad (6.23)$$

enquanto uma tríade descalibrada lê pontos sobre a superfície de um elipsóide não centrado na origem, e portanto essas componentes se relacionam de acordo com a equação

$$\left(\frac{B_X - O_X}{G_X B_{Total}} \right)^2 + \left(\frac{B_Y - O_Y}{B_{Total} G_Y} \right)^2 + \left(\frac{B_Z - O_Z}{B_{Total} G_Z} \right)^2 = 1 \quad (6.22)$$

Onde G_i e O_i são os ganhos e *off-sets* para cada eixo da tríade.

Ensaio R, S e T

Os girometros só terão aplicação quando for implementado o Filtro de Kalman. Estes ensaios têm como objetivo levantar os ganhos dos giros, pois não faz sentido utilizá-los sem calibração. Para determinar o ganho de um giro, não há outra maneira, senão aplicar diversas velocidade angulares constantes e conhecidas em seu eixo sensitivo e medir suas respectivas tensões, e então, fazer uma regressão.

Os giros MEMS utilizados são muito ruins em questão de estabilidade térmica; aliás todos os MEMS são, salvo aqueles que possuem alguma compensação eficiente. O

off-set deste tipo de sensor é altamente dependente da temperatura. O equacionamento do Filtro prevê estados correspondentes aos *bias* (*off-set*) do giros pois estes são críticos para a aplicação, já que seriam integrados, causando erros absurdos na medida do deslocamento angular.

O gráficos das Figuras 6.63, 6.34 e 6.35 apresentam a saídas do giros dos canais x, y e z respectivamente.

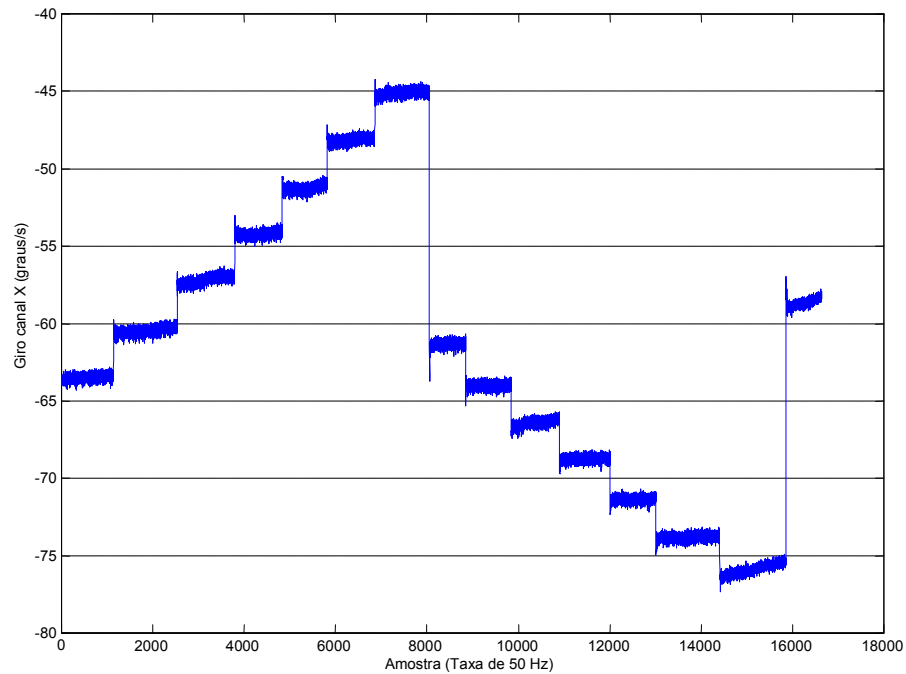


Figura 6.63 – Saída do giro do eixo x, quando girando entorno de x em passos incrementais de 5°/s e em seguida passos de – 5°/s .

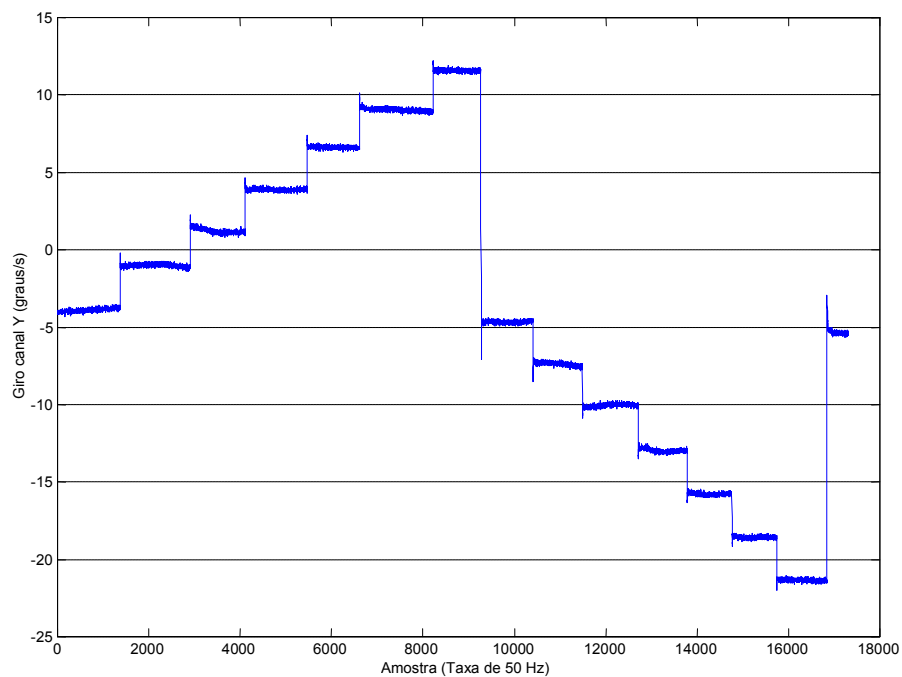


Figura 6.64 – Saída do giro do eixo y, quando girando entorno de y em passos incrementais de 5°/s e em seguida passos de – 5°/s .

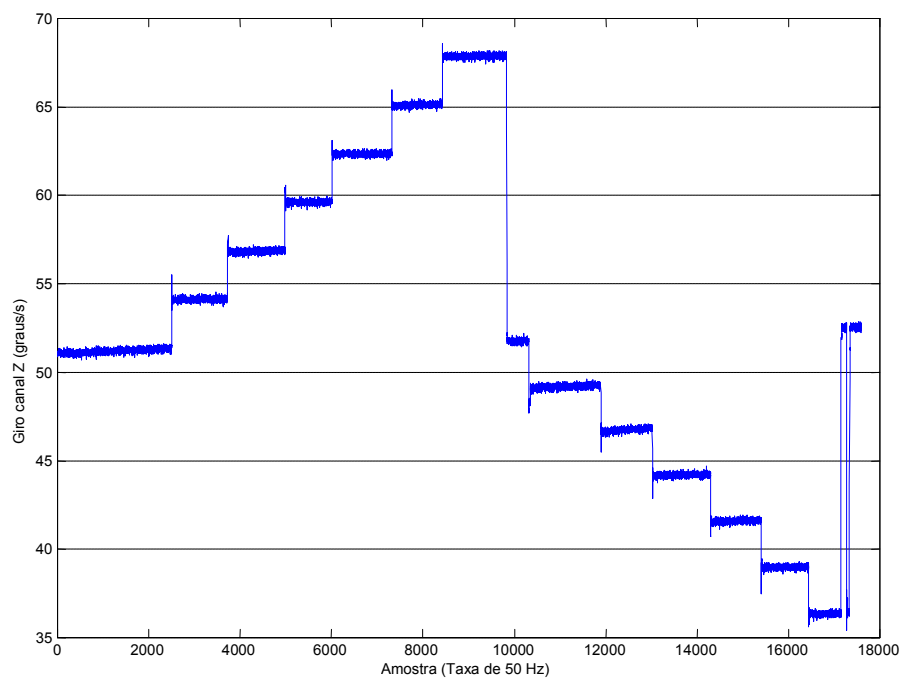


Figura 6.65 – Saída do giro do eixo z, quando girando entorno de z em passos incrementais de 5°/s e em seguida passos de – 5°/s .

6.12 – Considerações e Conclusões

Este capítulo visou expor o trabalho realizado na confecção, programação e testes de um determinador de atitude utilizando sensores MEMS de baixo custo.

A confecção do hardware demandou muito tempo de trabalho, enquanto que a programação dos algoritmos e os testes foram rapidamente explorados. Durante os testes fica evidente as limitações do uso de sensores magnéticos e também a necessidade da calibração contínua dos sensores.

Embora tenha sido implementada uma calibração para os magnetômetros e acelerômetros antes da rotina TRIAD, nota-se que essa calibração se perde completamente quando um campo magnético local constante desloca o *off-set* modificando o comportamento dos sensores. Neste caso, precisaria ser refeita a calibração sob as atuais circunstâncias. Quando o ensaio foi feito na mesa de mancal a ar, o campo magnético pouco afetou a calibração padrão, e por esse motivo, o quatérnio de atitude se mostrou quase como esperado.

Os giros, ainda que disponíveis, não foram utilizados, pois não foi feita a implementação do Filtro de Kalman no hardware. A carga computacional necessária para fazer os produtos matriciais para atualizar o vetor de estados (quatérnio e *bias* dos giros) e as matrizes de covariância, faria com que a taxa de quatérnio caísse para cerca de 1 Hz. Uma simplificação, como uso do ganho de Kalman constante, determinado em simulação, poderia viabilizar a implementação parcial do filtro, mas ainda a uma taxa do quatérnio não seria superior a 10 Hz.

Mesmo com todas as limitações, e o ruído elevado nos sensores, o hardware determinador de atitude apresentou bom desempenho em quesitos como a velocidade de conversão dos dados, a filtragem digital, em operações em ponto flutuante e também na taxa de saída dos quatérnions, que ficou em 50 Hz.

Capítulo 7 – Sumário e Conclusões

O objetivo deste trabalho foi estudar ferramentas que pudessem levar a construção de um determinador de atitude que utilizasse sensores MEMS, tais como acelerômetros, magnetômetros e girômetros.

Foram apresentadas revisões de conceitos importantes em determinação de atitude, como sistemas de coordenadas e representações de referências nestes sistemas. Três formas de representar atitude foram discutidas: matriz de atitude, quaternion e Ângulos de Euler e as transformações entre essas representações. Dois algoritmos de determinação de atitude foram descritos: o TRIAD, e o QUEST como solução para o método-q. Uma introdução a filtragem de Kalman aplicada às equações dinâmicas da atitude foram descritas passo a passo e simulados juntamente com os algoritmos de determinação de atitude. Um método que usa computação gráfica para visualização de resultados foi construído e utilizado como recurso de análise.

Um hardware de aquisição foi montado para adquirir dados de sensores MEMS e processar as informações para obter a atitude em tempo real. O hardware e o software foram descritos, assim como ensaios realizados com cada sensor.

Os ensaios realizados permitiram levantar alguns aspectos não previstos pela teoria apresentada, mas necessários ao bom funcionamento do equipamento. Na prática tratam-se de questões específicas dos sensores MEMS que poderiam ser evitados se os sensores utilizados fossem sensores de qualidade superior, ou seja, sensores intermediários àqueles de aplicação militar, mesmo custando de 5 a 10 vezes mais do que os sensores utilizados aqui.

Há ainda muitas tarefas que podem ser realizadas, principalmente na questão da implementação do sistema. Os algoritmos do Filtro de Kalman precisam ser escritos em C, simulados e implementados um a um em uma plataforma de processamento baseado em algum microcontrolador comercial que suporte a carga de processamento demandada. Algoritmos de calibração, alinhamento e ensaios em mesas rotativas para determinação do ganho dos Giros devem ser pesquisados, desenvolvidos e implementados. O desenvolvimento de processos eficazes de calibração são essenciais para o sucesso da implementação do determinador de atitude. Os modelos dos sensores

devem ser melhorados, pois deles depende um melhor funcionamento do Filtro de Kalman. Devem ser empregados modelos mais sofisticados para os sensores que simplesmente os de primeira ordem e também devem ser utilizadas compensações para variação de temperatura.

Fica a sugestão para que em trabalhos futuros sejam implementados e testados processos de calibração contínua. A simulação do Filtro de Kalman também pode ser explorada mais a fundo assim como sua implementação direta no hardware.

Referências Bibliográficas

- [ADX04] ADXL202, *Datasheet Analog Devices*: www.analog.com, visitado em: 29/12/2005.
- [APP04] APPEL, P., Appel, P. Attitude Estimation from Magnetometer and Earth-Albedo- Corrected Coarse Sun Sensor Measurements, Master Thesis, 2003.
- [CAR04a] CARUSO, M. J. *A New Perspective on Magnetic Field Sensing*, Honeywell: www.ssec.honeywell.com, visitado em: 29/12/2005.
- [CAR04b] CARUSO, M. J. *Applications of Magnetoresistive Sensors in Navigation Systems*, Honeywell: www.ssec.honeywell.com, visitado em: 29/12/2005.
- [CHI00] CHIANG, Y. T., et al. *Data Fusion of Three Attitude Sensors*. SICE 2001, Nagoya, pg 234-239
- [DUM99] DUMAN, I., *Design, implementation and testing of real-time software system for a quaternion-based attitude estimation filter*, Master Thesis, NPS March 1999.
- [ELE95] ELEMAG, *Modelo do Campo Magnetico*, software de 1995, disponível em: <http://paginas.terra.com.br/lazer/zeca/astronomia/download.htm#elemag>, visitado em 05/01/2006.
- [FAL01] FALBEL, G., Paluszek, M. A. *An Ultra Low Weight/Low Cost Three Axis Attitude Sensor Readout System for Nano-Satellites*. IEEE 2001 Aerospace Conference Proceedings, pg. 2469-2481 vol. 5.
- [GEB04] GEBRE-EGZIABHER, D., et al. *Design of Multi-Sensor Attitude Determination Systems*. IEEE Transactions on Aerospace and Electronics Systems, Vol. 40, Nº 2, Abril 2004, pg. 627-649.
- [GEL74] GELB, A. , Ed. *Applied Optimal Estimation*, MIT Press, Cambridge, Mass., 1974.
- [GYR04] GYROSTAR, *Datasheet Murata*: www.murata.com, visitado em: 29/12/2005.
- [HAY96] HAYKIN, S. *Adaptive Filter Theory*, 3º Ed. Prentice Hall Information and Systems Science Series , 1996
- [HAM49] HAMEL, G, *Theoretische Mechanik*, Spring-Verlag, Berlin, 1949
- [JAZ70] JAZWINSKI, A., *Stochastic Process and Filtering Theory*, Academic Press, NY, 1970.

- [KAL60] KALMAN, R.E., “*A New Approach to Linear Filtering and Prediction Problems*” Paper 59-IRD-11 presented at ASME Instruments and Regulators Conference, March 29-April 2, 1959 (also Transactions of ASME, Series D, Journal of Basic Engineering, Vol. 82, March 1960, pp. 35-45).
- [KUI02] KUIPERS, J. B., *Quaternions and Rotations Sequences*, Princeton University Press, 2002.
- [LEF82] LEFFERTS, E. J.; MARKLEY, F. L.; SHUSTER, M. D.; *Kalman filtering for spacecraft attitude estimation*. J. Guidance, Reston, v.5, n.5, p.417-429, 1982.
- [LIE04] LIEBE, C. C., *Solar Compass Chip*, IEEE Sensors Journal Vol. 4 No. 6, 2004 December.
- [MCL05] McLEAN S., Macmilan, S., et al., *The US/UK World Magnetic Model – WMM2005-2010* NO, December 2004, NOAA Technical Report NESDIS/NGDC-1.
- [MEI00] MEIJA, J., Villela, T., Braga, J. *The CCD Stellar Sensor of the Masco Telescope Pointing System*, 2000 , Advances in Space Res., v. 26, n. 9, p. 1407, 2000.
- [MAR00] MARKLEY, F. L.; MORTARI, D., *Quaternion Attitude Estimation Using Vector Observations*, The Journal of the Astronautical Science, Vol. 48, N° 2 e 3, April-September 2000, pp 359-380.
- [MAR01] MARINS, J. L., *An Extended Kalman Filtering for Quaternion-Based Orientation Estimation Using MARG Sensors*. Proceedings of the IEEE 2001. pg. 2003-2011.
- [SAN00] SANTONI, F., Bolotti, F., *Attitude Determination of Small Spinning Spacecraft Using Three Axis Magnetometer and Solar Panels*, Proceedings of the IEEE Aerospace Conference, Big Sky, USA, 2000, pg. 127-133.
- [SMI61] SMITH, G. L. and Schmidt, S.F., “*The Application of Statistical Filter Theory to Optimal Trajectory Determination Onboard a Circumlunar Vehicle*” Paper 61-92, presented at AAS Meeting August 1-3 1961.
- [SOR66] SORENSON, H. W., “*Kalman Filtering Techniques*” Advances in Control Systems, Vol. 3 Ed. C. T. Leondes, Academic Press, NY, 1966.
- [SHR04] SHREINER D., *OpenGL® Reference Manual: The Official Reference Document to OpenGL*. Addison Wesley Professional 2004.
- [SHU80] SHUSTER, M. D., and S. D. Oh. *Three-Axis Attitude Determination from Vector Observations*. AIAA Journal of Guidance and Control 1980. Vol. 4, No. 1. pg. 70-77.

- [SHU01] SHUSTER, M. D. *In QUEST of better attitudes*. (paper AAS 01-250) Advances in the Astronautical Sciences, Vol. 108, Part II, pp. 2089-2117, 2001.
- [TAP05] TAPLEY, B., J. Ries, S. Bettadpur, D. Chambers, M. Cheng, F. Condi, B. Gunter, Z. Kang, P. Nagel, R. Pastor, T. Pekker, S. Poole, F. Wang, "[GGM02 - An improved Earth gravity field model from GRACE](#)", Journal of Geodesy (2005), DOI 10.1007/s00190-005-0480-z [The original publication is available at www.springerlink.com](#)
- [WAH66] WAHBA, G. *A least squares estimate of satellite attitude*. Siam Review, Philadelphia, v.8, i.3, p.384-386, 1966.
- [WER78] WERTZ, R. D. *Spacecraft Attitude Determination and Control*, D. Reidel, Dordrecht, Holanda, 1978.
- [WIL04] WILLIAMS, A. J., *A Solid State Tilt Meter for Current Meter Attitude Determination*. IEEE Oceans '04, :1395-1398, 2004.
- [ZAR02] ZARCHAN, P., Musoff, H., *Fundamentals of Kalman Filtering – A Practical Approach*, Progress in Astronautics and Aeronautics Series, Vol. 190 Ed. AIAA, 2002

ANEXO I – Código do Programa para Apresentação da Atitude Graficamente

```
//-----  
// Programa para apresentação da Atitude graficamente  
// Creditos: Marília F. Chiozo e Francisco Granziera Júnior  
//-----  
// Borland C++Builder  
// Copyright (c) 1987, 1998-2002 Borland Corporation. All Rights  
Reserved.  
//-----  
#include <vcl.h>  
#include <stdio.h>  
#include <math.h>  
#pragma hdrstop  
#include "draw.h"  
//-----  
#pragma resource "*.dfm"  
TOpenGL_Form *OpenGL_Form;  
//-----  
__fastcall TOpenGL_Form::TOpenGL_Form(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
// Interrupção do Timer Ts = 20 milisegundos  
//-----  
void __fastcall TOpenGL_Form::Timer1Timer(TObject *Sender)  
{  
    Timer1->Enabled = false; /* Desabilita Interrupção do Timer */  
    DrawScene();             /* Função de atualização das imagens */  
    Yield();  
    Timer1->Interval = 20;    /* Intervalor de Interrupção fixo */  
    Timer1->Enabled = true;   /* Habilita Interrupção do Timer */  
}  
//-----  
// Função de Setup do modo Pixel (OpenGL)
```

```

//-----
BOOL bSetupPixelFormat(HDC hdc)
{
    PIXELFORMATDESCRIPTOR pfd, *ppfd;
    int pixelformat;

    ppfd = &pfd;

    ppfd->nSize = sizeof(PIXELFORMATDESCRIPTOR);
    ppfd->nVersion = 1;
    ppfd->dwFlags = PFD_DRAW_TO_WINDOW | PFD_SUPPORT_OPENGL |
                    PFD_DOUBLEBUFFER;
    ppfd->dwLayerMask = PFD_MAIN_PLANE;
    ppfd->iPixelFormat = PFD_TYPE_COLORINDEX;
    ppfd->cColorBits = 8;
    ppfd->cDepthBits = 16;

    ppfd->cAccumBits = 0;
    ppfd->cStencilBits = 0;

    if ( (pixelformat = ChoosePixelFormat(hdc, ppfd)) == 0 )
    {
        MessageBox(NULL, "ChoosePixelFormat failed", "Error", MB_OK);
        return FALSE;
    }

    if (SetPixelFormat(hdc, pixelformat, ppfd) == FALSE)
    {
        MessageBox(NULL, "SetPixelFormat failed", "Error", MB_OK);
        return FALSE;
    }
    return TRUE;
}

//-----
// Função De Atualização da Imagem (chamada pelo Timer)
//-----

void TOpenGL_Form::DrawScene()
{
    /* Inicializa variáveis só uma vez (static) */
    static float q[4];
    static int T = 1;

```

```

static FILE *ptr, *ptr2, *ptr3;

/* Abre arquivos se ainda não abriu */
if (T == 1)
{
    ptr = fopen("c:\\Documents and Settings\\Junior\\Desktop\\Tese
Mestrado\\FiltroKalmanSim_A\\NovosAlgs\\qtrue.txt", "r+");
    ptr2 = fopen("c:\\Documents and Settings\\Junior\\Desktop\\Tese
Mestrado\\FiltroKalmanSim_A\\NovosAlgs\\q_Tri.txt", "r+");
    ptr3 = fopen("c:\\Documents and Settings\\Junior\\Desktop\\Tese
Mestrado\\FiltroKalmanSim_A\\NovosAlgs\\q_est.txt", "r+");
}
T++;

/* limpa buffer (tela) */
glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );

/* Lê do arquivo qtrue.txt o quaternion verdadeiro
caso seja NULL, fecha arquivo */
if((fscanf(ptr, "%f", &q[0])) == NULL) fclose(ptr);
    fscanf(ptr, "%f", &q[1]);
    fscanf(ptr, "%f", &q[2]);
    fscanf(ptr, "%f", &q[3]);

/* Salva na pilha matriz de informação do objeto */
glPushMatrix();
/* roda objeto de acordo com quaternion e translada -1 (para
esquerda) */
Rodada(2*180/PI*acos(q[3]), q[0], q[1], q[2], -1.0);
glCallList(GLOBE);
/* Desempilha matriz de informação do objeto */
glPopMatrix();

/* Lê do arquivo q_Tri.txt o quaternion de atitude sem filtragem
caso seja NULL, fecha arquivo */
if((fscanf(ptr2, "%f", &q[0])) == NULL) fclose(ptr2);
    fscanf(ptr2, "%f", &q[1]);
    fscanf(ptr2, "%f", &q[2]);
    fscanf(ptr2, "%f", &q[3]);

```

```

/* Salva na pilha matriz de informação do objeto */
glPushMatrix();
/* roda objeto de acordo com quaternion e translada 0 (centro) */
Rodada(2*180/PI*acos(q[3]),q[0],q[1],q[2], 0.0);
glCallList(GLOBE);
/* Desempilha matriz de informação do objeto */
glPopMatrix();

/* Lê do arquivo q_est.txt o quaternion de atitude sem filtragem
   caso seja NULL, fecha arquivo */
if((fscanf(ptr3,"%f",&q[0])) == NULL) fclose(ptr3);
    fscanf(ptr3,"%f",&q[1]);
    fscanf(ptr3,"%f",&q[2]);
    fscanf(ptr3,"%f",&q[3]);

/* Salva na pilha matriz de informação do objeto */
glPushMatrix();
/* roda objeto de acordo com quaternion e translada +1 (para
direita) */
Rodada(2*180/PI*acos(q[3]),q[0],q[1],q[2], 1.0);
glCallList(GLOBE);
/* Desempilha matriz de informação do objeto */
glPopMatrix();

/* Atualiza buffer */
SwapBuffers(ghDC);
}

//-----
// Função chamada quando a um resize da TELA
//-----
GLvoid TOpenGL_Form::resize( GLsizei width, GLsizei height )
{
    GLfloat aspect;

    glViewport( 0, 0, width, height );

    aspect = (GLfloat) width / height;

    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();

```



```

gluPerspective(45.0, aspect, 3,7.0 );
glMatrixMode( GL_MODELVIEW );
}

//-----
// Função chamada para criar objeto, neste caso um CUBO que é
// construído vértice por vértice.
//-----

GLvoid TOpenGL_Form::createObjects()
{
    GLUquadricObj *quadObj;
    float x = .3;

    glNewList(GLOBE, GL_COMPILE);
        quadObj = gluNewQuadric ();
        gluQuadricDrawStyle (quadObj, GLU_FILL);
        glBegin(GL_QUADS);
            glColor3f(1,0,0);
            glVertex3f(-x,-x, x);
            glVertex3f( x,-x, x);
            glVertex3f( x, x, x);
            glVertex3f(-x, x, x);
        glEnd();
        glBegin(GL_QUADS);
            glColor3f(0,1,0);
            glVertex3f( x,-x,-x);
            glVertex3f( x, x,-x);
            glVertex3f( x, x, x);
            glVertex3f( x,-x, x);
        glEnd();
        glBegin(GL_QUADS);
            glColor3f(0,0,1);
            glVertex3f(-x, x,-x);
            glVertex3f( x, x,-x);
            glVertex3f( x, x, x);
            glVertex3f(-x, x, x);
        glEnd();
        glBegin(GL_QUADS);
            glColor3f(.5,.5,.5);
            glVertex3f(-x,-x,-x);
            glVertex3f( x,-x,-x);

```

```

        glVertex3f( x, x,-x);
        glVertex3f(-x, x,-x);
    glEnd();
    glBegin(GL_QUADS);
        glColor3f(.1,.1,.1);
        glVertex3f(-x,-x,-x);
        glVertex3f(-x, x,-x);
        glVertex3f(-x, x, x);
        glVertex3f(-x,-x, x);
    glEnd();
    glBegin(GL_QUADS);
        glColor3f(1,1,1);
        glVertex3f(-x,-x,-x);
        glVertex3f( x,-x,-x);
        glVertex3f( x,-x, x);
        glVertex3f(-x,-x, x);
    glEnd();
    glEndList();
}

//-----
// Função chamada para inicializar parâmetro da GL
//-----

GLvoid TOpenGL_Form::initializeGL(GLsizei width, GLsizei height)
{
    GLfloat      maxObjectSize, aspect;
    GLdouble     near_plane;

    glClearColor( (GLfloat)BLACK_INDEX);

    glClearDepth(1);

    glEnable(GL_DEPTH_TEST);
    glMatrixMode( GL_PROJECTION );
    aspect = (GLfloat) width / height;
    gluPerspective(45.0, aspect, 3, 7.0);
    glMatrixMode( GL_MODELVIEW );

    near_plane = 3.0;
    maxObjectSize = 1.0F;

```

```

        radius = near_plane + maxObjectSize/2.0;

        createObjects();
    }

//-----
// Função para vista polar. Não utilizada neste caso.
//-----
void TOpenGL_Form::polarView(GLdouble radius, GLdouble twist, GLdouble
latitude,

                                GLdouble longitude)
{

}

//-----
// Essa função roda o objeto de acordo com o quaternion fornecido
// Também desloca o objeto para posição fornecida
//-----
void TOpenGL_Form::Rodada(GLdouble angulo, GLdouble Xvec, GLdouble
Yvec,

                                GLdouble Zvec, GLdouble Deslocx)
{
    glTranslated(Deslocx, 0.0, -radius);
    glRotatef(angulo,Xvec,Yvec,Zvec);
}

//-----
// Chamada da função quando a janela é criada
// aqui são feitas as inicializações
//-----
void __fastcall TOpenGL_Form::FormCreate(TObject *Sender)
{
    ghDC = GetDC(Handle);
    if (!bSetupPixelFormat(ghDC))
        Close();
    ghRC = wglCreateContext(ghDC);
    wglMakeCurrent(ghDC, ghRC);
    initializeGL(ClientRect.Right, ClientRect.Bottom);
}
//-----

```

```

// Chamada do resize
//-----
void __fastcall TOpenGL_Form::FormResize(TObject *Sender)
{
    resize(ClientRect.Right, ClientRect.Bottom);
}

//-----
// Chamada da função quando a janela é Fechada
// aqui são feitas as inicializações
//-----
void __fastcall TOpenGL_Form::FormClose(TObject *Sender, TCloseAction
&Action)
{
    Timer1->Enabled = false;
    if (ghRC)
        wglDeleteContext(ghRC);
    if (ghDC)
        ReleaseDC(Handle, ghDC);
}

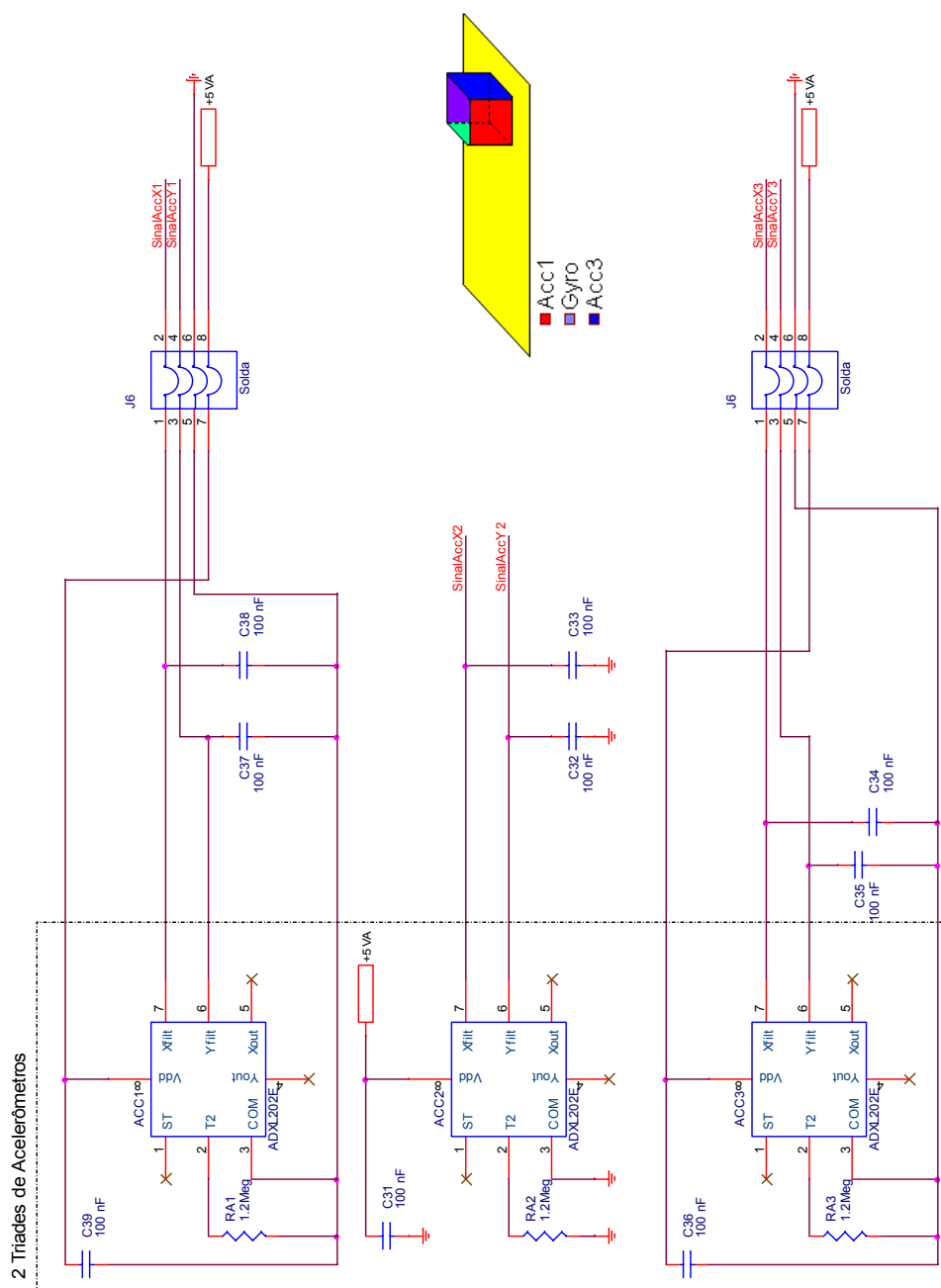
//-----
// Chamada quando alguma tecla é pressionada
//-----
void __fastcall TOpenGL_Form::FormKeyDown(TObject *Sender, WORD &Key,
    TShiftState Shift)
{
    // Fecha programa
    Close();
}

//-----

```

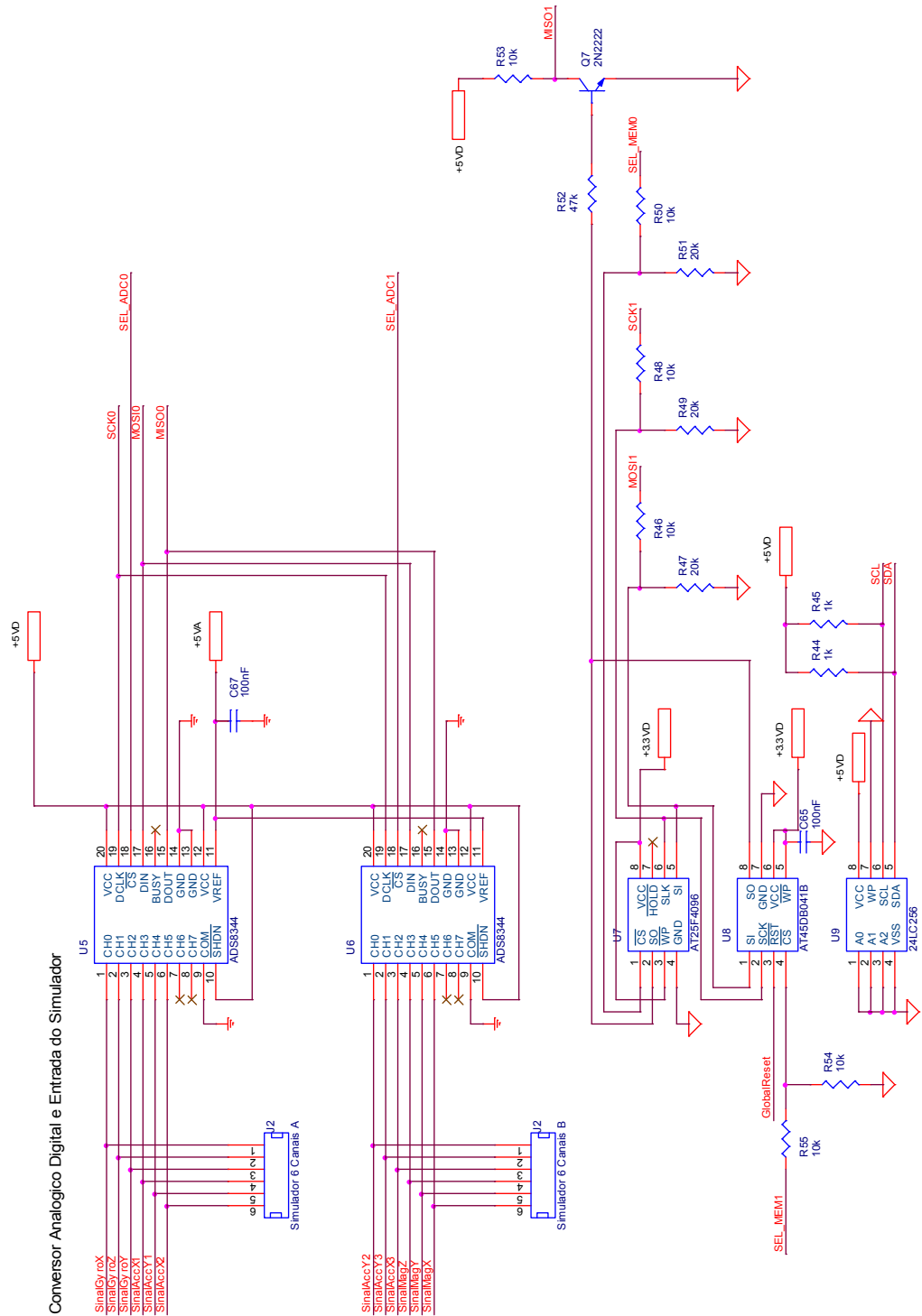
Anexo II – Esquemático do Hardware

II.1 – Esquemático dos Acelerômetros

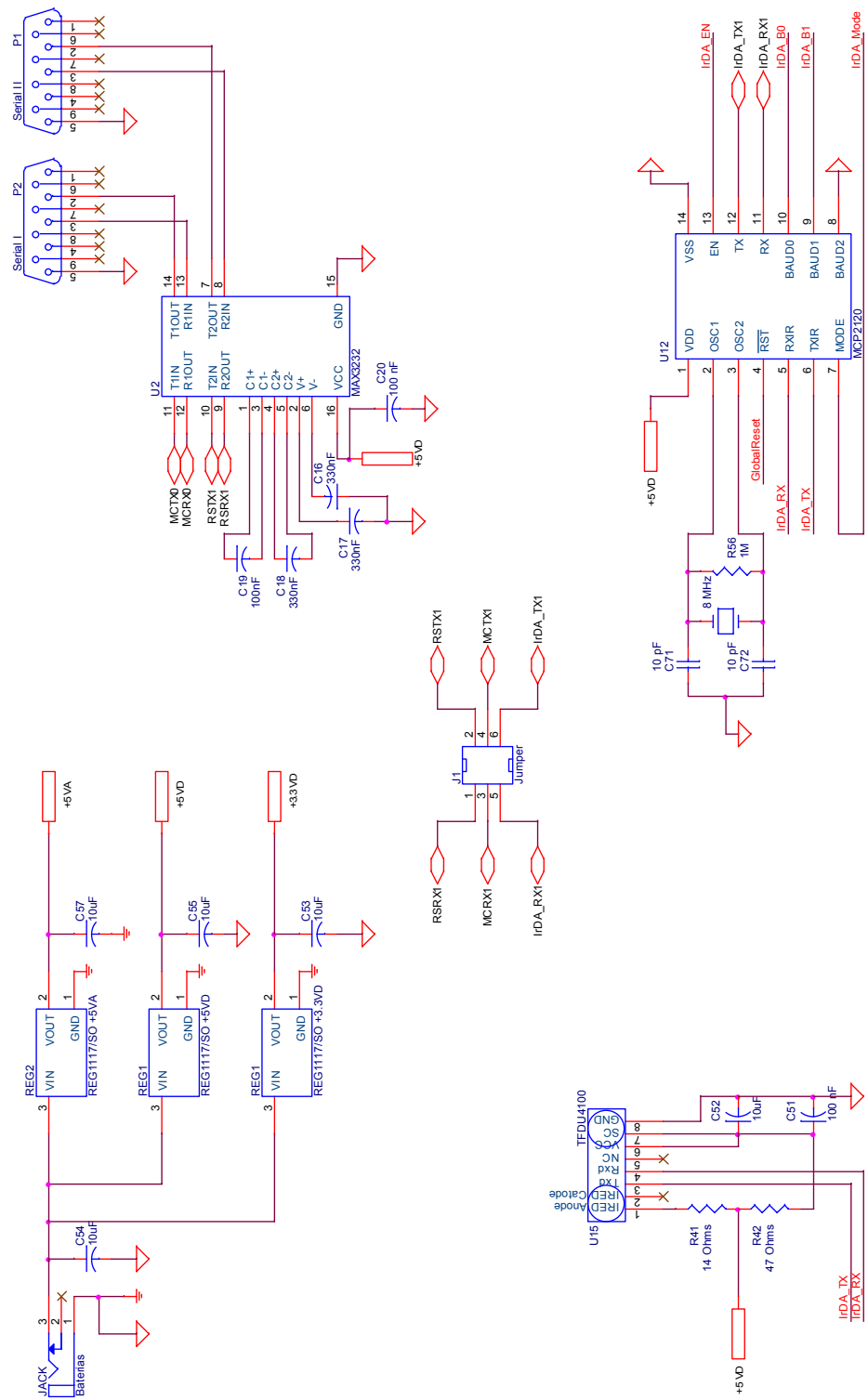


The schematic diagram illustrates the internal components and connections of the AD9364 evaluation board. It features two AD9364 ICs (U11 and U12) and two AD9364-1 ICs (U13 and U14). The AD9364 ICs are connected to a 5V supply and a 5MHz clock signal. The AD9364-1 ICs are connected to a 5V supply and a 5MHz clock signal. The diagram also shows various peripheral components, including capacitors (C60, C62, C63, C64, C65, C66, C67, C68, C69, C70, C71, C72, C73, C74, C75, C76, C77, C78, C79, C80, C81, C82, C83, C84, C85, C86, C87, C88, C89, C90, C91, C92, C93, C94, C95, C96, C97, C98, C99, C100), resistors (R43, R44, R45, R46, R47, R48, R49, R50, R51, R52, R53, R54, R55, R56, R57, R58, R59, R60, R61, R62, R63, R64, R65, R66, R67, R68, R69, R70, R71, R72, R73, R74, R75, R76, R77, R78, R79, R80, R81, R82, R83, R84, R85, R86, R87, R88, R89, R90, R91, R92, R93, R94, R95, R96, R97, R98, R99, R100), and connectors (X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22, X23, X24, X25, X26, X27, X28, X29, X30, X31, X32, X33, X34, X35, X36, X37, X38, X39, X40, X41, X42, X43, X44, X45, X46, X47, X48, X49, X50, X51, X52, X53, X54, X55, X56, X57, X58, X59, X60, X61, X62, X63, X64, X65, X66, X67, X68, X69, X70, X71, X72, X73, X74, X75, X76, X77, X78, X79, X80, X81, X82, X83, X84, X85, X86, X87, X88, X89, X90, X91, X92, X93, X94, X95, X96, X97, X98, X99, X100). The diagram is labeled with various signals and components, including +5V, -CLOCK_5MHz, GlobalReset, and various data and control lines.

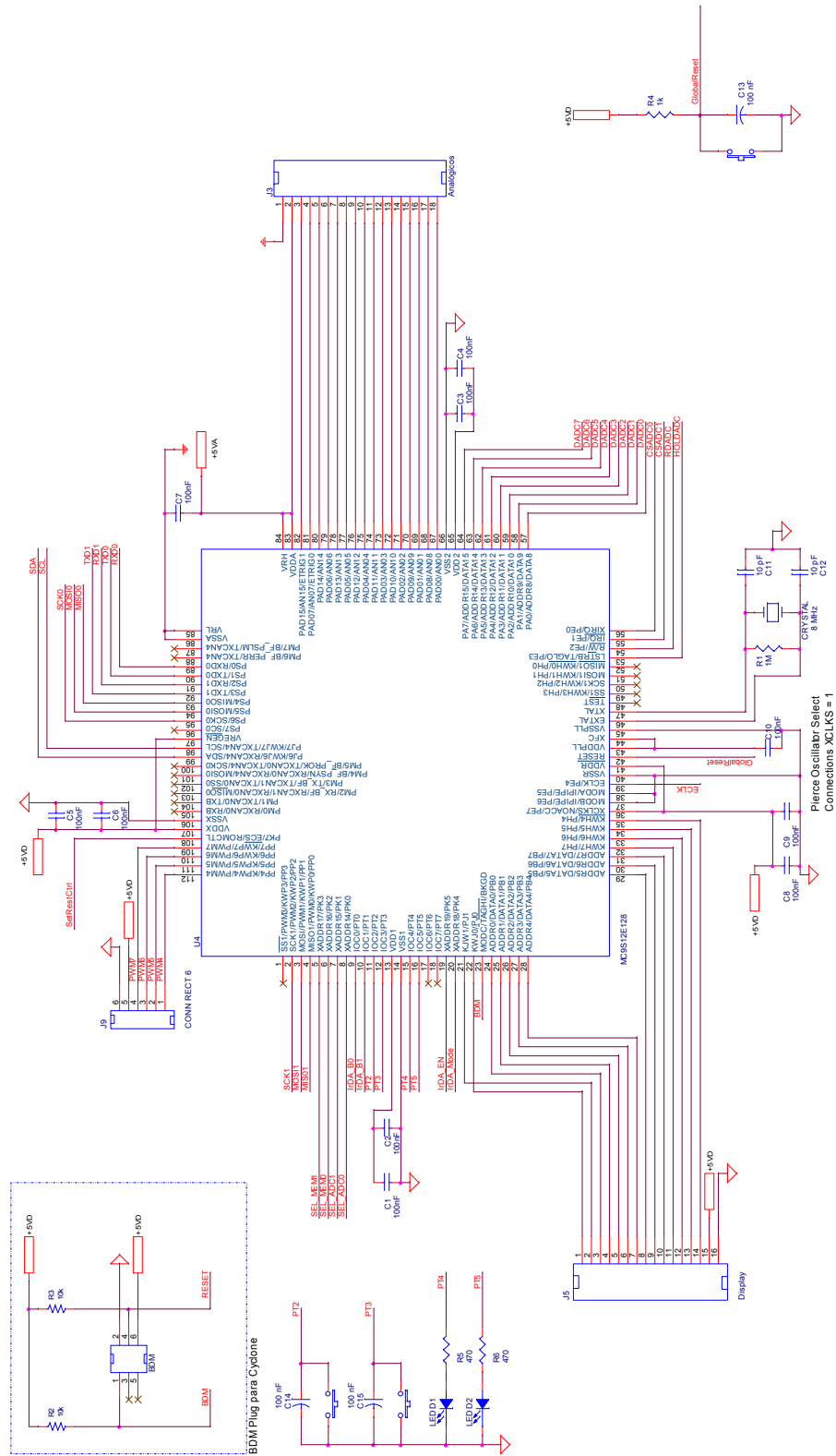
II.3 – Conversores AD e memórias (opcionais)



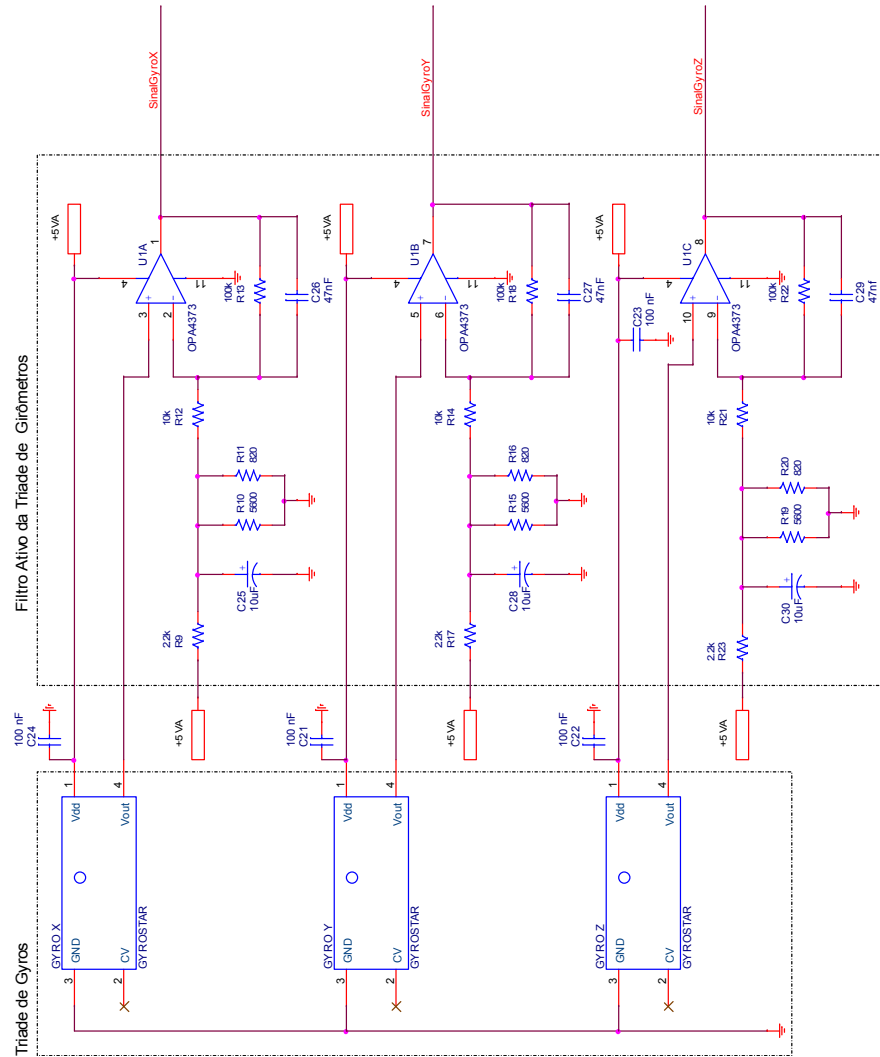
II.4 – Circuitos reguladores de Buffer serial e interface IrDA (opcional)



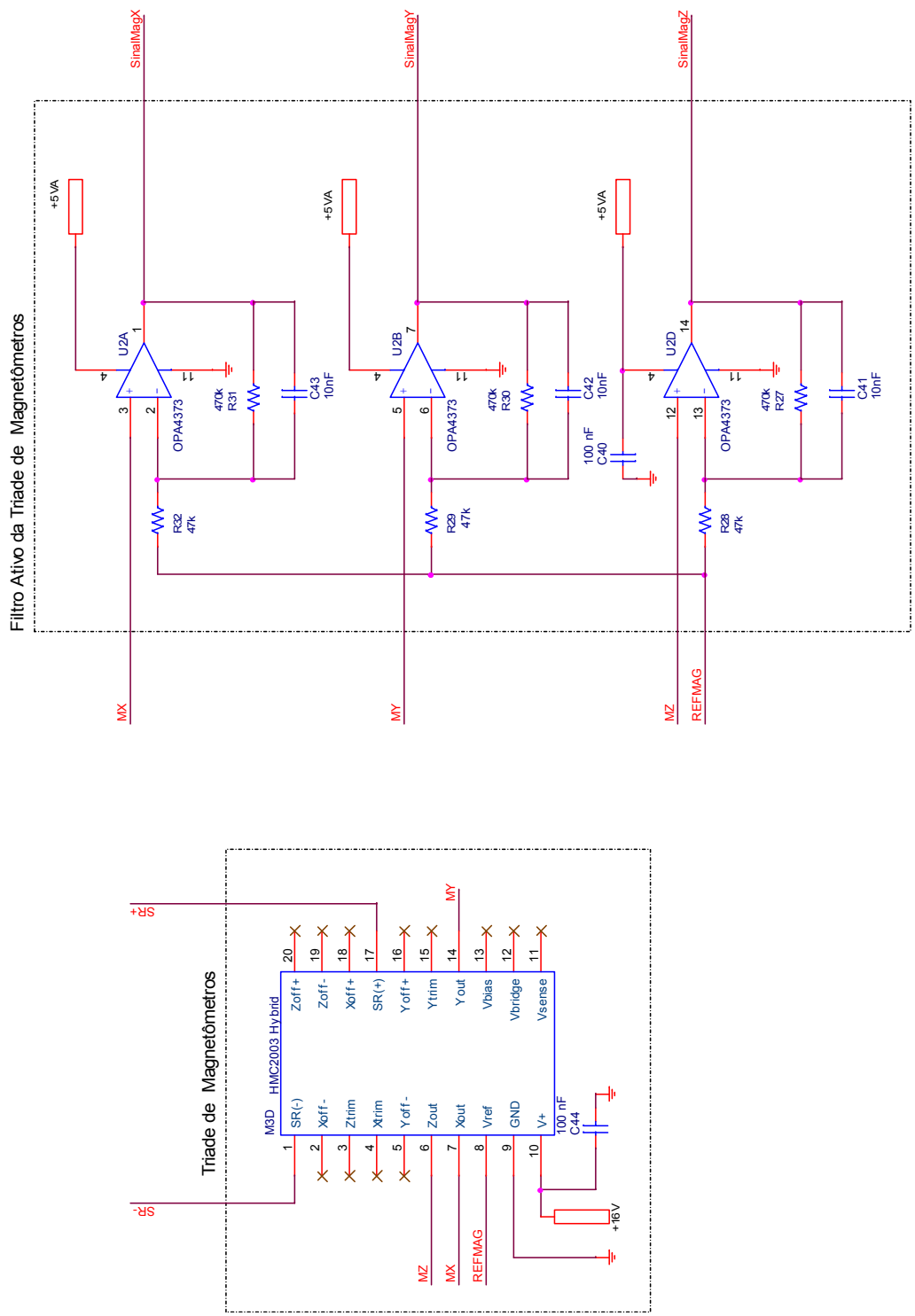
II.5 – Microcontrolador, clock, conexões (opcionais), reset, LED de status e botões (opcionais).



II.6 – Girômetros e amplificadores



II.7 – Magnetômetro e amplificadores



II.8 – Circuito de Set/Reset

