



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA MECÂNICA
CURSO DE ENGENHARIA MECÂNICA

GABRIEL GOES ARAGÃO SANTANA

**UMA METODOLOGIA DE INTEGRAÇÃO E FILTRAGEM DE DADOS INERCIAIS
PARA SISTEMAS LINEARES**

Recife
2019

GABRIEL GOES ARAGÃO SANTANA

**UMA METODOLOGIA DE INTEGRAÇÃO E FILTRAGEM DE DADOS INERCIAIS
PARA SISTEMAS LINEARES**

Trabalho de Conclusão de Curso
apresentado na Universidade Federal de
Pernambuco como requisito parcial para a
obtenção do grau de Bacharel em
Engenharia Mecânica.

Orientador: Prof. Dr. Guaraci Guimarães Bastos Júnior

Recife
2019

Catálogo na fonte
Bibliotecário Gabriel Luz, CRB-4 / 2222

S232m Santana, Gabriel Goes Aragão.
Uma metodologia de integração e filtragem de dados inerciais para
sistemas lineares / Gabriel Goes Aragão Santana – Recife, 2019.
102 f.: fig., gráfs., tabs., siglas.

Orientador: Prof. Dr. Guaraci Guimarães Bastos Júnior.
TCC (Graduação) – Universidade Federal de Pernambuco. CTG.
Departamento de Engenharia Mecânica, 2019.
Inclui referências e apêndices.

1. Engenharia mecânica. 2. Navegação. 3. Sensores inerciais. 4.
MEMS. 5. Fusão de dados. 6. Sistemas embarcados. I. Bastos Junior,
Guaraci Guimarães (Orientador). II. Título.

UFPE

621 CDD (22. ed.)

BCTG / 2020-18

GABRIEL GOES ARAGÃO SANTANA

**UMA METODOLOGIA DE INTEGRAÇÃO E FILTRAGEM DE DADOS INERCIAIS
PARA SISTEMAS LINEARES.**

Trabalho de Conclusão de Curso
apresentado na Universidade Federal de
Pernambuco como requisito parcial para a
obtenção do grau de Bacharel em
Engenharia Mecânica

Aprovada em: ____/____/____.

BANCA EXAMINADORA

Profº. Dr. Guaraci Guimarães Bastos Júnior (Orientador)
Universidade Federal de Pernambuco

Profº. Dr. Félix Christian Guimarães Santos (Examinador Interno)
Universidade Federal de Pernambuco

Profº. Dr. João Paulo Cerquinho Cajueiro (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

À minha querida mãe pelas palavras e suporte nos momentos mais difíceis, ao meu pai pela sabedoria e exemplo, à minha irmã pela inspiração.

Aos colegas de Mecânica pelos momentos e risadas. Aos professores pelas lições, em especial ao professor Guaraci pela contribuição neste trabalho. A Victor e Danilo, técnicos em LAC, pelos conselhos e paciência com minhas dúvidas. Ao colega Émerson pela ajuda no suporte.

À cada sala, andar e corredor do CTG pelas muitas memórias.

Aos amigos afastados pelo tempo, mas sempre presentes.

RESUMO

Navegação é a prática e ciência de se localizar a partir de algum tipo de medição. A navegação inercial, especificamente, diz respeito a localização a partir de medições internas de giroscópios e acelerômetros ao longo do tempo. Com os avanços da eletrônica e de sistemas micromecânicos, em conjunto denominados MEMS, sensores inerciais a preços acessíveis tornaram-se disponíveis ao grande público. Dentre os usos para essa tecnologia, estão aplicações na área de robótica, veículos não tripulados, estabilização de imagem e reconhecimento de gestos. Reconhecendo o potencial para navegação, o presente trabalho busca descrever e implementar um código genérico a partir de dados obtidos por um desses sensores, testando as capacidades e limitações da tecnologia atual, analisando os valores obtidos para orientação, posição e velocidade. Adicionalmente, a utilização de filtros, como o filtro de Kalman e complementar também foi discutida como forma de superar limitações de exatidão. Um microcontrolador Arduino foi utilizado e códigos no programa Octave foram escritos para o tratamento dos valores obtidos. Testes com rotações bidimensionais foram realizados a partir de um par de servomotores. Os resultados demonstraram uma boa acurácia para a orientação no espaço, principalmente pelos filtros e, para a posição, indicam a possibilidade de utilização desses sensores em conjunto com outras tecnologias mais precisas.

Palavras-chave: Navegação. Sensores inerciais. MEMS. Fusão de dados. Sistemas embarcados.

ABSTRACT

Navigation is the science and technique of finding one's location through some kind of measurement. Inertial navigation employs gyroscopes and accelerometers inside a moving vehicle in order to find its orientation, position and velocity as time progresses. Recently, cheap inertial sensors have become mainstream and easily available to the general public, thanks to advances in electronics and micro-mechanical systems, together named MEMS. Uses include robotic applications, unmanned vehicles, image stabilization and gesture recognition. Recognizing the potential for navigation applications, the present work aims to describe and implement a general code based on measurements taken by MEMS sensors, testing capabilities and limitations of the current technology and analyzing orientation, position and velocity results. Filtering by Kalman and Complementary filter - as a way to overcome sensors errors - is presented as well. An Arduino micro-controller is employed and code is written in a Octave script. Tests with two-axes rotations were performed employing servomotors. Good results were obtained for orientation - particularly with the filters - and positions results indicate that MEMS inertial sensors might be useful for navigation when combined with more accurate sensors.

Keywords: Navigation. Inertial sensors. MEMS. Data fusion. Embedded systems.

LISTA DE FIGURAS

Figura 1 –	Noção geométrica da astronavegação	18
Figura 2 –	Noção geométrica da navegação hiperbólica	19
Figura 3 –	Sistema de plataforma estável utilizado no foguete Saturno V	22
Figura 4 –	Esquema dos sistemas do tipo <i>strapdown</i>	23
Figura 5 –	Giroscópio mecânico	26
Figura 6 –	Efeito Coriolis em uma massa vibrante	26
Figura 7 –	Referencias ECEF e NED	31
Figura 8 –	Algoritmo para os sistemas do tipo <i>strapdown</i>	32
Figura 9 –	Convenção aeronáutica para os eixos	34
Figura 10 –	Diagrama de blocos do filtro complementar	47
Figura 11 –	Visão do algoritmo do filtro de Kalman	52
Figura 12 –	MEMS MPU-6050	53
Figura 13 –	Fluxo dos dados	57
Figura 14 –	Sequência do algoritmo inercial	58
Figura 15 –	Suporte Pan/Tilt	64

LISTA DE GRÁFICOS

Gráfico 1 –	Ângulo em função do tempo	67
Gráfico 2 –	Erro em função do tempo	67
Gráfico 3 –	Indicações dos eixos do giroscópio em repouso	68
Gráfico 4 –	Distribuição dos ruídos do giroscópio	69
Gráfico 5 –	Indicações médias do acelerômetro em diversas orientações....	70
Gráfico 6 –	Indicação do acelerômetro em repouso.....	70
Gráfico 7 –	Aproximação da função senoidal.....	72
Gráfico 8 –	Indicações do ângulo de pitch pelo giroscópio e acelerômetro...	72
Gráfico 9 –	Zoom do gráfico 8.....	73
Gráfico 10 –	Erros dos sensores.....	74
Gráfico 11 –	Indicações do filtro de Kalman e filtro complementar.....	75
Gráfico 12 –	Zoom do gráfico 11.....	75
Gráfico 13 –	Erros dos filtros.....	76
Gráfico 14 –	Ângulos de pitch e yaw medidos pelo giroscópio.....	77
Gráfico 15 –	Ângulo de pitch medido pelo acelerômetro e giroscópio.....	77
Gráfico 16 –	Zoom do gráfico 15.....	78
Gráfico 17 –	Erros dos sensores na rotação bidimensional.....	78
Gráfico 18 –	Ângulo de pitch pelo filtro de Kalman e filtro complementa.....	79
Gráfico 19 –	Zoom do gráfico 18.....	80
Gráfico 20 –	Erros dos filtros na rotação bidimensional.....	80
Gráfico 21 –	Ângulo de yaw medido pelo giroscópio.....	81
Gráfico 22 –	Posição no tratamento 1.....	82
Gráfico 23 –	Distância absoluta no tratamento 1.....	82
Gráfico 24 –	Orientação no tratamento 1.....	83
Gráfico 25 –	Posição no tratamento 2.....	84
Gráfico 26 –	Distância absoluta no tratamento 2.....	84
Gráfico 27 –	Velocidade no tratamento 2.....	85
Gráfico 28 –	Velocidade absoluta no tratamento 2.....	85
Gráfico 29 –	Orientação no tratamento 2.....	86

LISTA DE TABELAS

Tabela 1 –	Viés do giroscópio para diferentes aplicações	29
Tabela 2 –	Termos na equação de estados	48
Tabela 3 –	Parâmetros do giroscópio	54
Tabela 4 –	Parâmetros do acelerômetro	54
Tabela 5 –	Frequência de amostragem em função da taxa serial	55
Tabela 6 –	Vieses dos eixos dos giroscópio	68
Tabela 7 –	Desvio padrão dos eixos do giroscópio.....	68
Tabela 8 –	Vieses dos eixos do acelerômetro.....	70
Tabela 9 –	Ganhos dos eixos do acelerômetro.....	71
Tabela 10 –	Desvio padrão dos eixos do acelerômetro.....	71
Tabela 11 –	Parâmetros do filtro de Kalman.....	74

LISTA DE SIGLAS

ECEF	Referencial centrado e fixo na Terra
EKF	Filtro de Kalman Estendido
GNSS	Sistema de satélites de navegação global
GNU	GNU não é Unix
GPL	Licença público geral
GPS	Sistema de posição global
I ² C	Inter-Circuito Integrados
IMU	Unidade de medição inercial
INS	Sistema de navegação inercial
MEMS	Sistemas microeletromecânicos
NED	Referencial norte, leste, para baixo
PWM	Modulação por largura de pulso
UART	Transmissor/receptor universal e assíncrono
USART	Transmissor/receptor universal síncrono e assíncrono.

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	15
1.1.1 Objetivo Geral.....	15
1.1.2 Objetivo Específico.....	15
2 FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 NAVEGAÇÃO.....	17
2.2 NAVEGAÇÃO INERCIAL.....	20
2.2.1 Princípios.....	20
2.2.2 Arquitetura.....	21
2.2.3 Histórico e aplicações.....	23
2.3 SENSORES.....	25
2.3.1 Tecnologia.....	25
2.3.2 Erros.....	28
2.4 ALGORITMO PARA SISTEMAS STRAPDOWN.....	30
2.5 ORIENTAÇÃO.....	32
2.5.1 Matriz de rotação.....	32
2.5.2 Quatérnions.....	36
2.6 SOLUÇÃO DAS EQUAÇÕES DE NAVEGAÇÃO.....	39
2.6.1 Algoritmo dos cossenos diretos.....	39
2.6.2 Algoritmo dos quatérnions.....	41
2.6.3 Obtenção das velocidades e posições.....	42
2.7 CÁLCULO DO TILT.....	45
2.8 FILTRO COMPLEMENTAR.....	46
2.9 FILTRO DE KALMAN.....	48
3 METODOLOGIA.....	53
3.1 SENSORES.....	53
3.2 PROGRAMAÇÃO.....	55
3.3 CALIBRAÇÃO.....	58
3.4 FILTRAGEM.....	61
3.5 MODELAGEM DO FILTRO DE KALMAN.....	62
3.6 SUPORTE PARA OS TESTES.....	64
4 RESULTADOS.....	66

4.1 TESTE DO ALGORITMO.....	66
4.2 CALIBRAÇÃO.....	68
4.3 ROTAÇÃO UNIDIMENSIONAL.....	71
4.4 ROTAÇÃO BIDIMENSIONAL.....	76
4.5 ALGORITMO INERCIAL.....	81
5 CONCLUSÃO.....	87
REFERÊNCIAS.....	90
APÊNDICE A - CÓDIGO PARA A CALIBRAÇÃO DO ACELERÔMETRO.....	93
APÊNDICE B - INTEGRAÇÃO DO GIROSCÓPIO.....	96
APÊNDICE C - CÓDIGO DO FILTRO DE KALMAN.....	100
APÊNDICE D - CÓDIGO DO ALGORITMO INERCIAL.....	102

1 INTRODUÇÃO

Sistemas microeletromecânicos (MEMS) são considerados uma tecnologia bastante promissora. Esses sistemas têm o potencial de revolucionar tanto a indústria quanto os produtos para o consumidor, em virtude da combinação da microeletrônica à base de silício com a tecnologia de micro usinagem.

A capacidade tecnológica do MEMS pode ser vista no grande alcance de suas aplicações e mercado, como por exemplo, nas indústrias automotiva, médica, eletrônica, de comunicações e nas aplicações militares. Dispositivos MEMS incluem acelerômetros para sensores *airbag*, cabeçotes para impressoras de jato de tinta, cabeçotes para leitura/escrita para discos rígidos, biossensores, sensores de pressão sanguínea, dentre outros. O emprego de técnicas de fabricação em lote permite que os componentes e dispositivos MEMS alcancem um elevado desempenho e confiabilidade, além de reduções de tamanho (dimensões variando de micrômetros para milímetros), peso e custo (PRIME Faraday, 2002).

Em termos de sensores, uma das áreas mais consolidadas dos MEMS é a que abrange a parte de acelerômetros e giroscópios. Considerando esse fato, aliado ao baixo preço de mercado desses sensores, este trabalho foi concebido com o objetivo de analisar o desempenho de um sistema de navegação inercial baseado na tecnologia MEMS. Para tanto, um algoritmo para navegação inercial foi implementado em um *software* de análise numérica (GNU Octave) a partir de dados puros obtidos por um sensor inercial disponível comercialmente. Baseado nos resultados obtidos foram simulados os parâmetros orientação, posição e velocidade em função do tempo, validados com valores desejados (pré estabelecidos).

A motivação para o desenvolvimento desse trabalho surgiu a partir de uma curiosidade inicial sobre o tema de navegação inercial. O formato desse trabalho foi significativamente influenciado pela consulta da seguinte bibliografia especializada.

Brown (2004) utilizou sensores MEMS e um receptor GPS para navegação de um veículo terrestre não tripulado. Essas entradas foram combinadas a partir de um *software* comercial da fabricante Internav, a partir de um filtro de Kalman com 32 componentes. O conjunto produziu um erro de posição inferior a 1 m, contudo, a perda do sinal de GPS produziu um erro de 10 m após 20 s.

Eskin (2006) desenvolveu uma metodologia geral para um sistema inercial baseado em MEMS, bem como um modelo adequado para sua calibração. O

modelo foi implementado em um microcontrolador e os resultados obtidos foram de um erro de posição 0,17 m após 1 s e um erro de velocidade de 0,3 m/s após o mesmo período. O autor concluiu afirmando a viabilidade da tecnologia MEMS e que uma combinação desse sistema com GPS permitiria ganhos de precisão.

Fairfax (2012) utilizou um filtro de Kalman estendido (EKF) a partir de sensores MEMS e do sinal de GPS para estimação da posição de projéteis disparados por morteiros. Em particular, esse trabalho também utilizou o conhecimento das características da dinâmica de voo do projétil para aperfeiçoar o modelo. Testes em campo também foram realizados e, de acordo com o autor, erros de posição inferiores a 40 m foram alcançados no tempo de voo de 30 s, usando apenas sensores MEMS.

O tratamento de Romaniuk (2014) é abrangente e intuitivo. Nesse trabalho, o autor utilizou um filtro de Kalman relativamente simples para a fusão dos dados de giroscópios, acelerômetros e um magnetômetro para a orientação e de um acelerômetro e *receiver* de GNSS para a posição. Os sensores utilizados são disponíveis comercialmente e de baixo custo, enquanto a implementação ocorreu pelo uso de um microcontrolador.

Os trabalhos citados a seguir não tratam da navegação inercial em sua forma mais ampla, mas ainda assim utilizam dispositivos MEMS para estimativa da orientação e se valem da fusão de sensores.

Paina (2011) comparou os resultados fornecidos por um filtro complementar e um filtro de Kalman utilizados para a orientação de um drone quadrotor, a partir de acelerômetros e giroscópios. Embora os resultados do filtro de Kalman fossem livres de viés, foram mais ruidosos que o filtro complementar e exigem uma implementação mais complexa. A determinação das covariâncias de maneira experimental, essencial para a aplicação do filtro de Kalman, também é apresentada.

Wang (2014) utilizou uma formulação com filtros complementares para o desenvolvimento de um sistema de orientação a partir de acelerômetros, giroscópios e um magnetômetro. Os resultados obtidos são comparados com os valores fornecidos por um sensor comercial e mostram uma significativa concordância.

Petterson (2015) desenvolveu e implementou um filtro de Kalman estendido (EKF) a partir da formulação por quatérnions. Em geral, a rotação tridimensional não pode ser precisamente descrita pelo filtro de Kalman linear. O modelo conta um

vetor de estado com 7 componentes – 4 componentes do quatérnion e 3 vieses dos giroscópios. Os dados de acelerômetros, giroscópios e magnetômetros são combinados. Ademais, dados de GNSS são utilizados para a determinação da aceleração do veículo, o que permitiu que o acelerômetro fornecesse dados mais precisos mesmo durante as fases de aceleração. O modelo foi testado em tempo real para uma aplicação de um veículo não tripulado.

1.1 OBJETIVOS

Essa seção descreve os objetivos gerais e específicos do presente trabalho.

1.1.1 Objetivo Geral

O presente trabalho tem como objetivo desenvolver um código para a navegação inercial, que permita a obtenção da orientação, velocidade e posição de um corpo a partir de medições internas realizadas por giroscópios e acelerômetros solidários ao corpo em rotações bi-dimensionais.

Igualmente, o autor pretende mostrar a possibilidade de integração de dados a partir de sensores distintos, permitindo uma melhor estimativa dos valores. Como os sensores usados possuem preço acessível ao público, não é razoável esperar que seu desempenho seja semelhante aos sensores do estado da arte. Portanto, a integração torna-se uma ferramenta interessante para superar as eventuais limitações.

Por fim, mas tão importante quanto, espera-se expor o tema da navegação inercial de uma maneira clara e precisa, estabelecendo uma base teórica de maneira que trabalhos futuros aprofundem-se ainda mais nesse campo.

1.1.2 Objetivo Específico

Como objetivos específicos, buscar-se-á testar as possibilidades de navegação inercial com sistemas MEMS, pensando na viabilidade de aplicações futuras de baixo custo, tais como o uso em veículos não tripulados.

A tecnologia MEMS possui limitações atuais e a calibração dos sensores é de natural importância. Espera-se expor uma maneira confiável e simples de calibrar

esses sensores. Outras considerações práticas, e que se tornarem evidentes com a experiência, também serão expostas.

2 FUNDAMENTAÇÃO TEÓRICA

De maneira a fornecer uma exposição adequada do tema, a presente seção inicia-se com uma breve descrição sobre o conceito de navegação, para então definir a navegação inercial de maneira mais formal, a partir de seus princípios e tecnologias construtivas.

A definição e representação da orientação por ângulos de Bryan e quatérnions é apresentada, servindo de base para a resolução das equações dinâmicas da propagação da orientação e da posição.

Por fim, as possibilidades de fusão de sensores são discutidas a partir de duas ferramentas matemáticas de uso geral: o filtro complementar e o filtro de Kalman.

2.1 NAVEGAÇÃO

Navegação é arte de descobrir sua posição atual, sua velocidade e orientação e tomar decisões de forma a maximizar sua chance de sobrevivência (STOVALL, 1997).

Uma definição alternativa e mais universal é dada pela NASA:

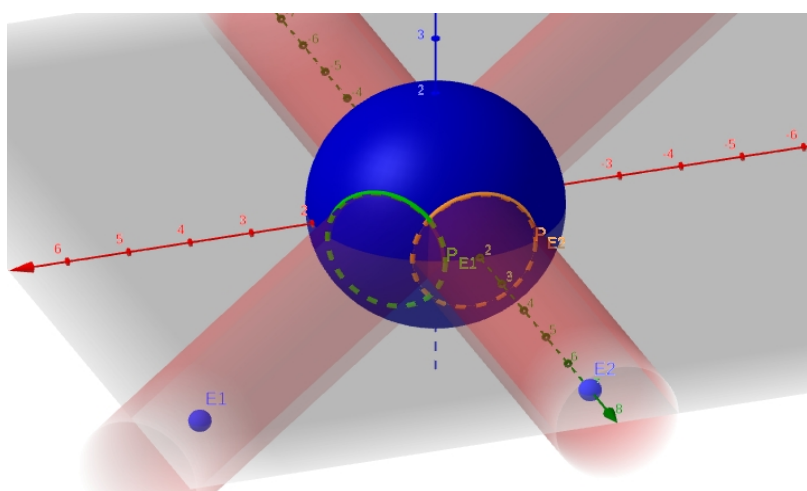
Navegação é a determinação da posição atual a partir de medições feitas dentro do veículo;
Guiagem é a computação das manobras necessárias para alcançar uma trajetória desejada
Controle é a execução das manobras de guiagem pelo controle apropriado do hardware. (NASA, 1968, p. 157)

A navegação é tão antiga quanto a humanidade e, de fato, é empregada por diversas espécies migratórias. Variados tipos de navegação foram desenvolvidos ao longo da história, dentre eles, podemos citar (GREWAL, 2004):

- a) pilotagem, essencialmente baseada no reconhecimento de paisagens e referências para a definição da localização e orientação. Referências frequentemente usadas incluem montanhas, rios, lagos, praias ou construções humanas como rodovias, represas, faróis;
- b) *dead reckoning* ou navegação estimada, baseada numa localização inicial conhecida, propaga a posição pelo conhecimento da direção – que pode ser obtida por uma bússola – e uma estimativa da distância percorrida, obtida pelo produto da velocidade e do tempo;

c) navegação celeste ou astronavegação (figura 1), baseada em medições de tempo e ângulos entre o horizonte local e objetos celestes conhecidos (Sol, Lua, planetas, estrelas) para a estimativa da orientação, latitude e longitude. O ângulo formado determina um círculo específico na superfície da Terra, centrado na projeção do corpo celeste em um ponto na Terra. A interseção de dois desses círculos, possível por uma medição adicional, permite a estimativa da localização;

Figura 1 - Noção geométrica da astronavegação



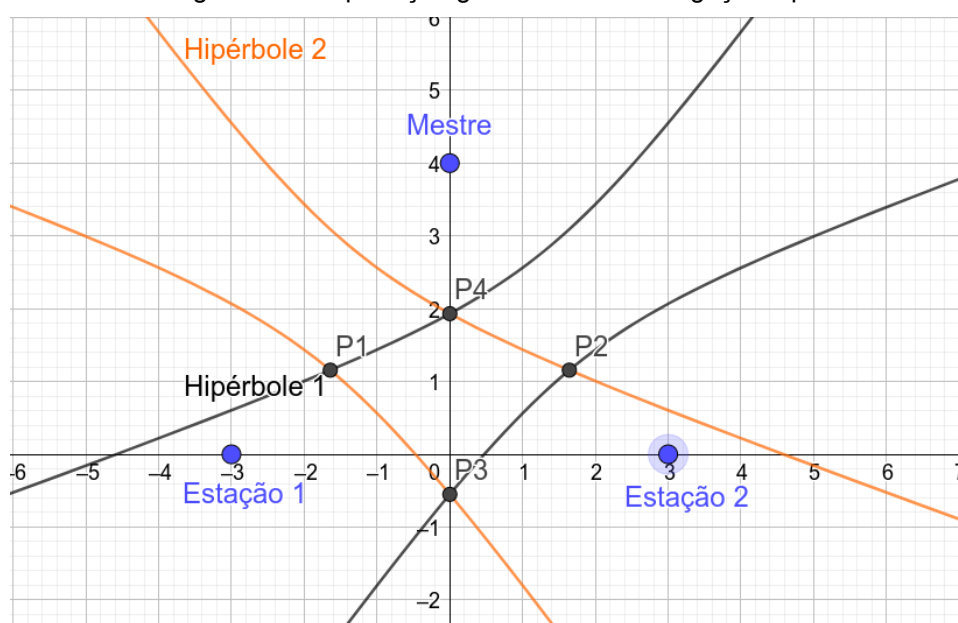
Fonte: o autor (2019)

d) radio navegação, consiste na utilização de radiofrequência para a determinação da posição. Na navegação hiperbólica (Figura 2), a recepção de dois sinais fornece a diferença da distância do veículo a estações de localização conhecida. Uma estação mestre sincroniza o relógio de outras estações, chamadas escravas. A interseção de duas dessas hipérboles geradas pelas estações como focos determina a posição do corpo. Sistemas como o americano LORAN e o internacional Ômega foram introduzidos nas décadas após a Segunda Guerra Mundial;

e) mais recentemente, sistemas de satélite de navegação global (GNSS), como o GPS (Estados Unidos), GLONASS (Rússia), Galileo (União Europeia) e BeiDou (China), tem-se tornado onipresentes e amplamente disponíveis ao grande público. Nesses sistemas, constelações de satélites em altas órbitas terrestres permitem a localização global através da medição do tempo entre a

emissão do sinal e a recepção pelo *receiver*. O sinal de 4 satélites é necessário para a determinação da posição e a presença de satélites adicionais, permite uma melhor estimativa. A acurácia, de maneira geral, é da ordem de metros para usos públicos e de algumas dezenas de centímetros para sinais criptografados, empregados para usos militares.

Figura 2 - Interpretação geométrica da navegação hiperbólica



Fonte: o autor (2019)

O presente trabalho tem como foco um outro tipo de navegação: a navegação inercial. Navegação inercial é uma técnica de navegação na qual medições de giroscópios e acelerômetros são utilizadas para a determinação da posição, velocidade e orientação de um corpo a partir de condições iniciais conhecidas. Os acelerômetros fornecem os valores da aceleração, geralmente em três eixos perpendiculares e os giroscópios, os valores da velocidade angular, também em três eixos perpendiculares, embora alguns giroscópios forneçam o valor do ângulo diretamente. Os valores medidos pelos sensores fornecem a aceleração e a rotação do corpo medido no seu referencial em relação ao referencial inercial.

2.2 NAVEGAÇÃO INERCIAL

Nessa seção, busca-se estabelecer os fundamentos gerais e apresentar uma breve história da aplicação da navegação inercial, de maneira a facilitar o seu entendimento.

2.2.1 Princípios

A navegação inercial é uma consequência direta das leis da mecânica e da gravitação enunciadas por Isaac Newton (1642-1727). A primeira lei de Newton ou lei da inércia diz respeito a tendência de um corpo a permanecer parado ou em movimento retilíneo uniforme, a menos que atuado por uma força resultante não nula. A segunda lei, por sua vez, estabelece a proporcionalidade entre força e aceleração, correspondente à derivada temporal da velocidade, ou segunda derivada da posição.

Portanto, caso os valores da aceleração estejam disponíveis, bem como as condições iniciais do corpo, uma integração em função do tempo fornecerá os valores da velocidade para qualquer instante e uma segunda integração fornecerá a posição. As integrações descritas podem ser expressas por:

$$\vec{v}(t) = \int_{t_0}^t \vec{a}(t') dt' + \vec{v}(t_0) \quad (1)$$

$$\vec{r}(t) = \int_{t_0}^t \vec{v}(t') dt' + \vec{r}(t_0) \quad (2)$$

onde $\vec{a}$ é a aceleração, $\vec{v}$ a velocidade e $\vec{r}$ a posição.

Embora seja menos evidente, um processo análogo pode ser aplicado para a orientação. Partindo-se de uma orientação inicial conhecida, a orientação para qualquer instante posterior pode ser determinada caso a velocidade angular do corpo esteja disponível para integração.

As leis de Newton aplicam-se a referenciais inerciais, portanto, as integrais dadas acima fornecem a posição de um corpo em relação a um referencial inercial, como as estrelas distantes, originalmente citadas por Newton. Contudo, em muitas aplicações, a descrição do movimento em relação a outros referenciais não inerciais

pode ser mais conveniente. No caso da navegação terrestre, por exemplo, é de natural interesse a obtenção dos vetores em um referencial terrestre – a partir de latitude, longitude e altitude em relação à superfície terrestre. Nesse caso, correções adicionais no algoritmo são necessárias para a correção das pseudoforças centrífuga e de Coriolis, efeitos decorrentes da rotação da Terra.

Os pontos apresentados acima estabelecem os princípios da navegação inercial. A aceleração é detectada por meio de acelerômetros, sensores que são essencialmente constituídos por um tipo de conjunto massa-mola amortecido: uma aceleração causa uma deformação na mola, que é então processada para o cálculo da aceleração correspondente. Rotações são detectadas por giroscópios, que medem a velocidade angular ao longo de um dado eixo. Uma discussão mais ampla da tecnologia construtiva desses dispositivos é apresentada posteriormente.

A principal vantagem da navegação inercial, como pode ser concluído comparando com as demais formas de navegação, é a independência dessa técnica em relação a medições externas. Diferente da pilotagem, por exemplo, a navegação inercial não depende de condições visuais favoráveis, nem está sujeita a possibilidade de interferência presentes na rádio navegação.

2.2.2 Arquitetura

Utilizando-se 3 giroscópios e 3 acelerômetros, geralmente dispostos em 3 eixos perpendiculares entre si, é possível o cálculo da posição e orientação de um corpo ao longo do tempo. Os giroscópios e acelerômetros são montados em um mesmo conjunto, o qual é genericamente chamado de Unidade de Medição Inercial (IMU – *Inertial Measurement Unit*). A junção dos instrumentos do IMU e da implementação do algoritmo de navegação com alguma unidade de processamento é chamada de sistema de navegação inercial (INS, em inglês *Inertial Navigation System*). Elementos adicionais não mencionados incluem fontes de alimentação, unidades de controle de temperatura e outros elementos de suporte.

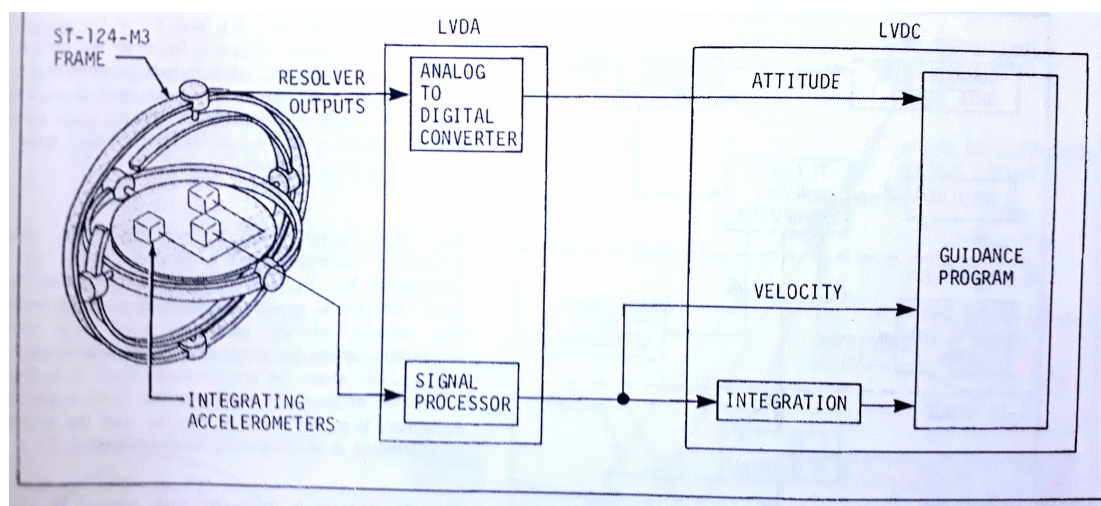
Os sistemas de navegação inercial podem ser divididos em duas classes principais: sistemas em uma plataforma estável e sistemas fixados (em inglês, *strapdown system*). Os sistemas em plataforma estável foram os primeiros a surgir. Nos primórdios, a ausência de computadores rápidos o bastante e a incapacidade

dos sensores de cobrirem todo o intervalo da velocidade angular impediu o desenvolvimento de sistemas *strapdown*.

Nos sistemas de plataforma estável (Figura 3), os acelerômetros são montados numa plataforma apoiada por *gimbals* - suporte pivotados que permitem rotação em único eixo - acionados por motores, de forma a manter a plataforma sempre alinhada com o referencial inicial. Dessa forma, há um isolamento da rotação do veículo, que não influencia a plataforma. (WALCKO, 2002). Com essa configuração, os três ângulos dos *gimbals* estão associados diretamente aos ângulos de *roll*, *pitch* e *yaw*, descritos mais adiante.

Como o conjunto se mantém fixo, a estimativa da posição é possível a partir da simples soma (integração) dos valores fornecidos pelos acelerômetros, corrigidos pelo valor da gravidade local. O sistema é inicialmente alinhado em relação a um referencial conhecido, geralmente o referencial geográfico: norte, leste e para baixo.

Figura 3 - Sistema de plataforma estável utilizado no foguete Saturno V



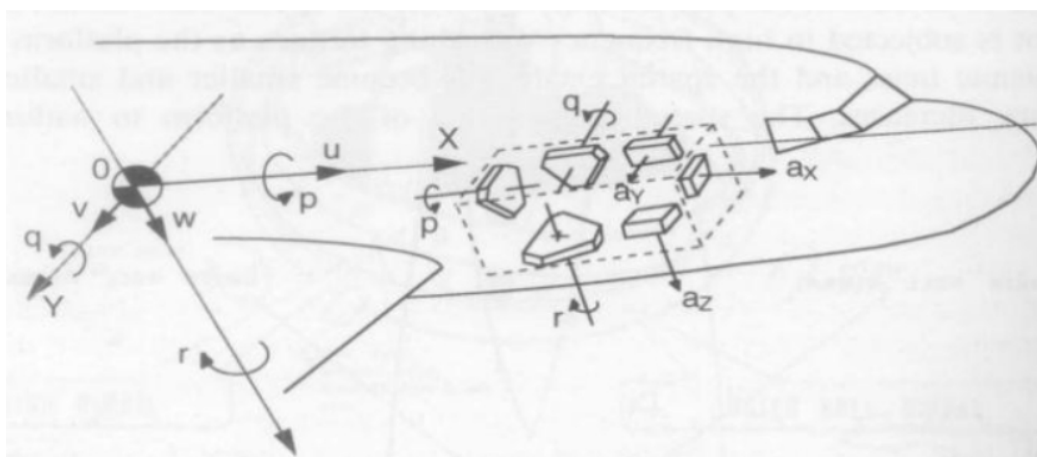
Fonte: NASA (1968)

Dentre as desvantagens apresentadas pelos sistemas em plataforma estável estão: maior volume, custo maior e, devido aos componentes mecânicos, apresentam maior complexidade construtiva que os sistemas *strapdown*. Contudo, ainda são utilizados em submarinos e navios, aplicações que geralmente requerem maior precisão em um largo intervalo de tempo. A vantagem desses sistemas é a maior facilidade de integração.

Nos sistemas *strapdown*, os sensores inerciais são rigidamente presos ao corpo, de maneira que a aceleração e velocidade angular medidas são descritas no referencial do corpo. A figura 4 ilustra essa configuração.

Os sistemas do tipo *strapdown* possuem menor custo, maior confiabilidade e menor tamanho - devido aos avanços na área de sistemas microeletromecânicos (MEMS), os sensores podem ser contidos numa placa de circuito impressa. A desvantagem para essa aplicação é a maior complexidade das equações a serem resolvidas, resultando em uma maior complexidade computacional, além da necessidade de se trabalhar com maiores faixas de velocidade angular. Os avanços recentes experimentados pelos processadores e sistemas embarcados atenuaram muito essas restrições.

Figura 4 - Esquema dos sistemas do tipo strapdown



Fonte: PRAVEENA (2013)

2.2.3 Histórico e aplicações

Historicamente a navegação inercial teve sua primeira aplicação durante a Segunda Guerra Mundial (1939-1945), com o desenvolvimento pela Alemanha nazista do míssil balístico de curto alcance V-2. O V-2 possuía um alcance de aproximadamente 300 km; acelerômetros eram utilizados para o controle do tempo preciso do desligamento do motor foguete, que por sua vez era determinante para o ponto de impacto da carga. Um computador analógico era responsável pela dupla integração dos valores da aceleração.

Avanços significativos na tecnologia e implementação de sistemas inerciais ocorreram no final dos anos 1940 e começo dos anos 1950, com as primeiras

implementações de sistemas de navegação em aeronaves e navios mercantes. O engenheiro e cientista americano Charles Stark Draper é frequentemente chamado de pai da navegação inercial, graças às suas contribuições na teoria e invenções na área de navegação inercial. Draper ainda contribuiria, anos depois, com o desenvolvimento dos sistemas de navegação usados no programa espacial Apollo.

A navegação inercial difundiu-se para diversos usos, permitindo uma capacidade de navegação global, atendendo, ao mesmo tempo, requisitos elevados de desempenho. Alguns exemplos são citados a seguir.

Em particular, a navegação inercial é fundamental para os submarinos modernos, visto que a camada de água do mar é condutora e impede a propagação efetiva de ondas eletromagnéticas da atmosfera. Ademais, os requisitos de furtividade exigem um tipo de navegação que não dependa de observações externas. Embora os números oficiais sejam confidenciais, parece razoável estimar a acurácia desses sistemas como próxima a 1 milha náutica por dia – 1,85 km por dia ou um ângulo equivalente de 1 minuto na superfície terrestre.

Aeronaves também possuem requerimentos para navegação inercial: condições de baixa visibilidade, ausência de referenciais terrestres, travessia de grandes distâncias, dentre outras. De fato, muito possivelmente a maior aplicação para navegação inercial é na área aeronáutica. O sistema é iniciado com a aeronave em solo, em repouso, e permite que a trajetória percorrida passe por certos pontos intermediários, chamados de *waypoints*. A taxa de amostragem, em geral, está entre 100 a 1.000 Hz. Uma estimativa razoável para o erro desses sistemas é de 1 km/hora.

Por fim, mísseis de cruzeiro e balísticos devem operar em um possível ambiente no qual outras formas de comunicação e navegação sejam degradadas ou destruídas por ação inimiga. Nesse caso, a navegação inercial fornece um sistema independente de interferências externas. Os requisitos de acurácia também são críticos, visto que estão associados a destruição do alvo. Como um exemplo, o míssil balístico lançado por submarinos Trident II, operado pela marinha americana e britânica e fabricado nos Estados Unidos, possui alcance máximo de 12.000 km e pode carregar até 8 ogivas nucleares. Seu sistema de navegação é do tipo inercial combinado com a astronavegação pela medição estelar, resultando em um erro circular provável de 90 m, o que implica que metade dos disparos deve cair dentro de um raio de 90 m centrado no alvo (Missile Defense Project, 2018).

2.3 SENSORES

A presente seção tem como objetivo expor as tecnologias presentes e princípios de funcionamento nos dispositivos de medição inercial, assim como uma definição dos erros inerentes em suas medições.

2.3.1 Tecnologia

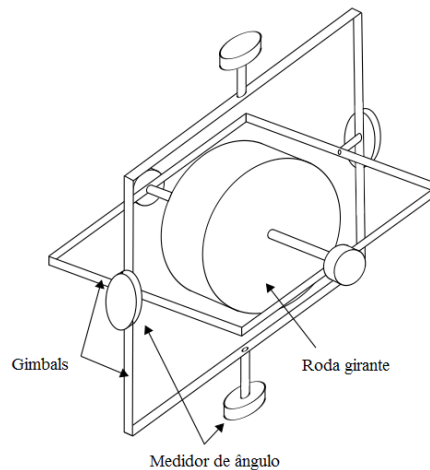
Os giroscópios podem ser genericamente divididos nas seguintes classes: giroscópios girantes, baseados no princípio da roda girante, giroscópios óticos e giroscópios vibrantes.

Os giroscópios girantes consistem essencialmente em uma massa em rotação ao redor de um eixo (Figura 5). Graças a longa experiência e uso histórico, são extremamente precisos, embora ocupem grande espaço e sejam mecanicamente complexos. O princípio de funcionamento é a conservação do momento angular: os giroscópios tendem a manter um mesmo eixo de rotação, fixo em relação ao referencial inercial. A presença de um torque externo causa o movimento de precessão numa direção perpendicular a força e ao braço de alavanca. Caso uma mola possa contrabalancear esse torque, a velocidade angular pode ser calculada.

O funcionamento dos giroscópios óticos é baseado no efeito Sagnac, observado em um referencial em rotação. Um feixe de luz é dividido em dois feixes, que percorrem a mesma distância mas em sentidos diferentes. A posterior recombinação dos feixes produz um fenômeno da interferência, cuja intensidade é função da velocidade angular. O efeito Sagnac é associado à diferença da distância percorrida pelos feixes: devido a rotação, um feixe percorre uma distância superior e, portanto, ao ser recombinado, apresenta uma diferença de fase.

Dentre os giroscópios dessa classe, estão os giroscópios de anel de laser (*ring laser gyro* – RLG), frequentemente empregados na aviação, e os giroscópios de fibra ótica (*fiber optic gyro* – FOG).

Figura 5 - Giroscópio mecânico

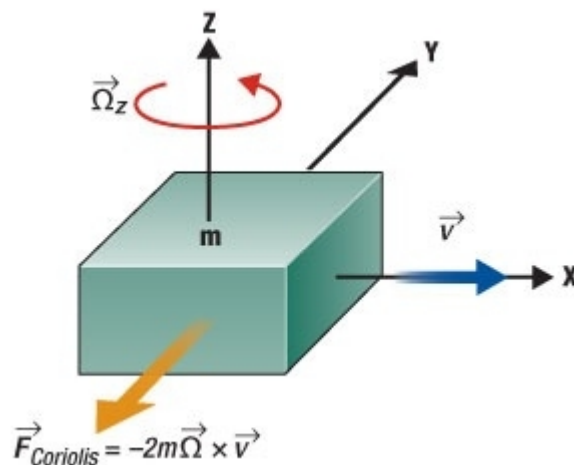


Fonte: WOODMAN (2007)

Os giroscópios do tipo MEMS detectam a rotação a partir do Efeito Coriolis atuante sob um cristal vibrante de quartzo. Possuem pequeno tamanho, baixo custo e baixo consumo de energia.

Quando uma massa m move-se com uma velocidade linear v e é submetida a uma rotação angular Ω , a massa é sujeita a uma pseudoforça advinda do efeito Coriolis (Figura 6). O deslocamento físico causado pelo efeito Coriolis é detectado através da mudança da capacitância de um circuito e, com isso, a velocidade angular pode ser determinada (SCHEWEBER, 2018).

Figura 6 - Efeito Coriolis em uma massa vibrante



Fonte: SCHEWEBER (2018)

Os acelerômetros de uma maneira geral operam a partir do princípio do sistema massa-mola amortecido. Ao submeter o sistema a uma força externa, a mola é deformada e, conhecendo-se a constante da mola, a força pode ser determinada.

Devido ao princípio da equivalência enunciado por Albert Einstein em sua teoria da relatividade, o acelerômetro é incapaz de detectar a ação da gravidade. Por exemplo, em um objeto em queda livre, isto é, em queda vertical sob a única influência da gravidade, o acelerômetro indicará uma aceleração nula. Da mesma maneira, um acelerômetro deixado em cima de uma mesa indicará a aceleração da gravidade local, embora permaneça imóvel em relação a Terra. De maneira geral, pelo princípio da equivalência, é impossível construir um instrumento capaz de ser sensível somente aceleração, mas não gravidade, ou somente gravidade, e não aceleração (CANNON, 1967).

Essa reflexão sugere que o acelerômetro, na verdade, mede a diferença da aceleração de um corpo em relação a um corpo local em queda livre, sendo essa aceleração comumente chamada de aceleração própria. O termo força específica também é usado para denotar as medições do acelerômetro. A discussão acima referida indica que a aceleração medida pelo acelerômetro deve ser corrigida pela gravidade, isto é (TITTERTON, 2004):

$$\vec{f}_b = \frac{d^2 \vec{r}}{dt^2} - \vec{g} \quad (3)$$

em que f_b denota a força específica que o acelerômetro mede no referencial do corpo.

As tecnologias mais modernas, chamadas de estado sólido, encontradas nos sensores do tipo MEMS se baseiam no efeito piezoelétrico ou a partir de alterações na capacitância de duas placas paralelas. Nos acelerômetros de efeito piezoelétrico, a aceleração advinda de uma força externa faz com que um cristal microscópico adquira uma carga elétrica. A carga é detectada por um circuito externo e convertida para a respectiva aceleração.

No caso da capacitância, a aceleração provoca o afastamento ou aproximação de duas placas paralelas. Essa distância cria uma variação da capacitância, que medida pelo circuito, permite a estimativa da aceleração.

Os avanços nas últimas décadas na área de sistemas microeletromecânicos permitiram o barateamento e ampla difusão de acelerômetros e giroscópios para a indústria e consumidores em geral. Atualmente, esses dispositivos são encontrados em diversas aplicações, tais como aparelhos celulares, *airbags*, estabilização de imagem em câmeras digitais, consoles de videogame, discos rígidos para notebooks, veículos não tripulados, dentre outros.

2.3.2 Erros

Os erros para os acelerômetros e giroscópios podem ser expressos de uma maneira geral da seguinte forma: dado um valor verdadeiro η para a grandeza que se deseja medir, o valor $\tilde{\eta}$ indicado pelo sensor terá um formato dado por:

$$\tilde{\eta} = \eta + S\eta + N\eta + b + v \quad (4)$$

O termo b é chamado de viés e representa uma constante adicionada às medições, independente do valor de entrada. Apresenta dependência com a temperatura e pode mudar com o tempo.

O viés pode ser medido considerando o valor médio para o caso do giroscópio parado ou a partir da diferença fornecida entre a aceleração e a gravidade local, para o caso do acelerômetro. É expresso em $^{\circ}/h$ para o giroscópio e m/s^2 ou em μg para o acelerômetro. O viés costuma ser a fonte dominante do erro. Na Tabela 1 são apresentados vieses do giroscópio típicos para diferentes tipos de aplicações.

A dupla integração na posição, implica que o viés causa um erro quadrático com o tempo na posição e um erro linear nos ângulos de orientação e da velocidade.

Tabela 1 - Viés do giroscópio para diferentes aplicações

Tipo	Viés (°/h)	Aplicação
Consumidor em geral	30 - 1000	Interface de movimento
Industrial	1 - 30	Guiagem de munições e projéteis
Tático	0.1 - 30	Estabilização de plataforma
Tático avançado	0.1 - 1	Navegação de mísseis
Navegação	0.01 - 0.1	Navegação aérea
Estratégico	0.0001 – 0.01	Navegação submarina

Fonte: PASSARO (2017)

O termo v é um ruído normal de média zero e variância finita, advindo de fontes diversas de ruído. Caso um sensor esteja medindo uma grandeza constante, um ruído aleatório estará presente nas medições. A integração desse ruído no tempo – necessária para a obtenção dos ângulos e da velocidade - produz uma caminhada aleatória em tempo discreto, também chamada de processo de Wiener no domínio contínuo. Esse processo é dito não estacionário, visto que suas propriedades estatísticas variam com o tempo. Em particular, o desvio padrão em função do tempo $\sigma_\theta(t)$ obedece a:

$$\sigma_\theta(t) = \sigma \cdot \sqrt{\Delta t \cdot t} \quad (5)$$

onde Δt é o tempo de amostragem. Do resultado acima, o número $\sigma \cdot \sqrt{\Delta t}$ é uma referência do quanto o ruído da medição faz os valores calculados se afastarem do valor real. É frequentemente dado em $^\circ/\sqrt{h}$ ou $^\circ/\sqrt{s}$ para giroscópios e em $\mu g/\sqrt{Hz}$ ou $\mu g/\sqrt{h}$ para os acelerômetros. Esses valores são geralmente fornecidos pelo fabricante, a uma frequência específica, e tendem a diminuir com o aumento da frequência de amostragem, como pode ser visto pela equação acima.

O fator de escala S está associado à linearidade entre a entrada e a saída. Por exemplo, supondo o exemplo simples de uma massa e uma mola, a deformação da mola está diretamente relacionada à força aplicada e, portanto, à aceleração. Caso uma força de 100% seja aplicada, pode-se esperar uma deformação de 100% do valor calibrado. O fator de escala está ligado a variações nessa relação e assim, uma força de 100% pode produzir uma deformação de 105%. Esse fator pode ser pensado como o desvio da reta do gráfico entrada x saída em relação à função

identidade, sendo ele denotado em partes por milhão (ppm). É representado por uma matriz com apenas os elementos da diagonal principal como não nulos.

A matriz N diz respeito a erros de alinhamento entre os eixos dos sensores, isso é, à não ortogonalidade. Os termos de N não nulos são os componentes fora da diagonal principal. Esses erros dão origem a uma correlação entre as medições. Por exemplo, um acelerômetro orientado na horizontal medirá o valor da gravidade no eixo z , quando estiver em repouso, mas o desalinhamento dos eixos pode fazer com que o eixo x ou y também meçam uma componente não nula da gravidade.

Modelos mais complexos para o erro podem considerar efeitos não lineares, tais como termos proporcionais a η^2 .

2.4 ALGORITMO PARA SISTEMAS STRAPDOWN

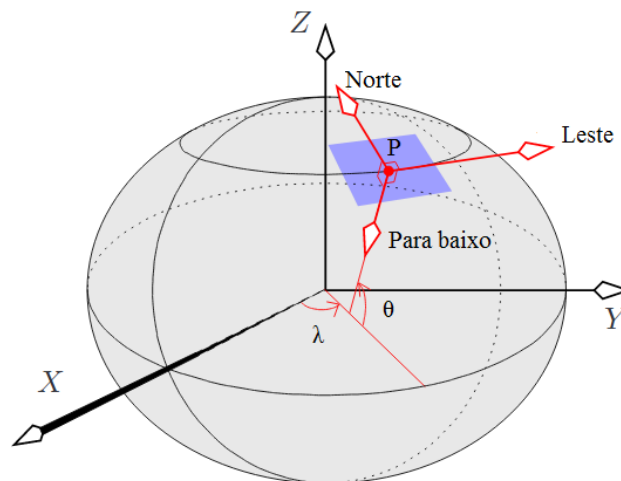
Dada a disponibilidade e fácil acesso a sensores MEMS, o presente trabalho tem como foco sistemas do tipo *strapdown*. Nesse sistema, as medições fornecem valores de força específica e rotação no referencial do corpo em relação ao referencial inercial, como já exposto anteriormente. Essas medições, de alguma maneira, precisam ser integradas em função do tempo para a determinação da orientação, posição e velocidade.

Esses valores são calculados em relação a um referencial de navegação, que geralmente não é inercial. Na superfície terrestre, por exemplo, uma maneira de se localizar um corpo é através da utilização dos ângulos de latitude θ , longitude λ e altitude (distância vertical em relação ao nível do mar). Os ângulos são calculados em relação ao Equador e ao meridiano de Greenwich, respectivamente. A origem desse sistema está no centro da Terra, de maneira que o eixo z aponta para o eixo de rotação da Terra (pólo norte) e o eixo x e y rotacionam com a Terra. Esse sistema é denominado na literatura como ECEF (*Earth Centered Earth Fixed* – Centrado e fixo na Terra).

Outro referencial é o sistema cartesiano NED, cujas letras representam, respectivamente, norte (*north*), leste (*east*) e a vertical para baixo (*down*). Nesse sistema, o eixo x aponta para o norte geográfico, o eixo y para leste geográfico e o eixo z para o centro da Terra. A figura 7 a seguir indica a latitude e longitude de um ponto P , medidos no ECEF, bem como o referencial NED definido em P .

A figura 8 ilustra o algoritmo de navegação do corpo para um referencial inercial. Como os acelerômetros estão presos ao veículo, a força específica f_b precisa ser transformada para o referencial inercial a partir do conhecimento da orientação do corpo. A orientação, por sua vez, é possível a partir da medição das velocidades angulares ω_{bi}^b e deve igualmente ser integrada, como será visto adiante. Com isso, a força específica f_i é determinada e, com a adicional correção da gravidade, que depende da posição do corpo, a aceleração no referencial inercial é encontrada. Por fim, a aceleração é integrada numericamente e a velocidade é encontrada. Outra integração fornece a posição.

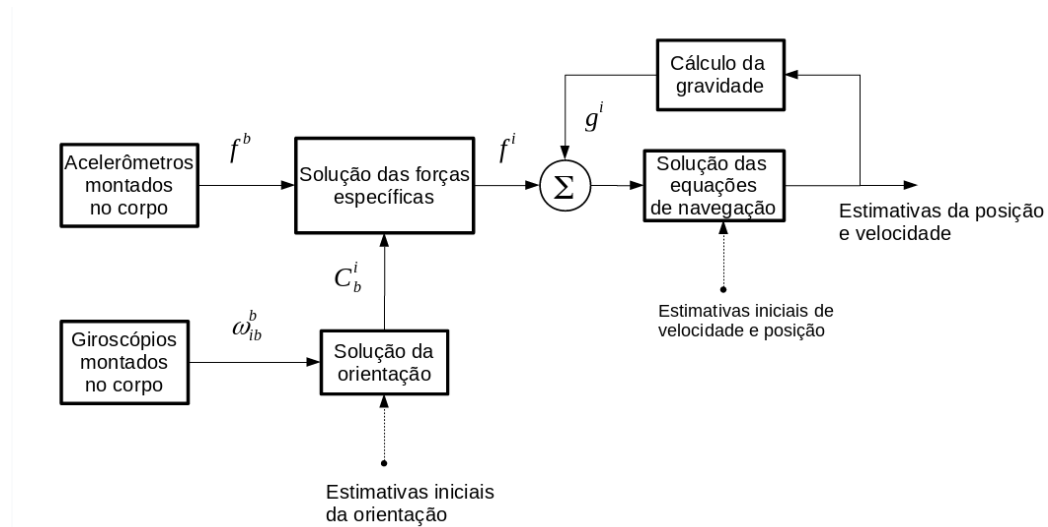
Figura 7 - Referenciais ECEF e NED



Fonte: Adaptado de Koks (2008)

O referencial NED é frequentemente usado, devido a sua conveniência e intuitividade. Contudo, a rotação da Terra, uma revolução a cada 24 horas ou $15^\circ/\text{h}$, dá origem a componentes não inerciais, tais como a força de Coriolis e a aceleração centrífuga. Ademais, a curvatura da Terra implica que os eixos desse referencial para dois pontos distintos não são paralelos. Por conveniência, essas dificuldades serão desprezadas e será assumido que o referencial NED é um referencial inercial.

A hipótese utilizada acima implicitamente assume uma Terra de raio infinito (plana), desprovida de movimento de rotação, e campo gravitacional constante. Evidentemente, para aplicações locais, que não envolvam grandes intervalos de tempo, amplas áreas ou grandes velocidades, essa hipótese deve ser considerada satisfatória.

Figura 8 - Algoritmo para os sistemas do tipo *strapdown*

Fonte: o autor (2019)

2.5 ORIENTAÇÃO

No espaço tridimensional, um corpo rígido livre possui seis graus de liberdade. Três graus de liberdade estão associados a translação do seu centro de massa, medida nos eixos x , y e z de um referencial fixo. Os outros três graus de liberdades estão associados a orientação de um conjunto de eixos presos ao corpo em relação ao referencial fixo. Assim, por exemplo, uma aeronave em voo em relação a um referencial, como um aeroporto, pode ser descrita pela sua altitude e distância radial, porém a descrição completa do seu estado exige o conhecimento da inclinação das asas, elevação do nariz e direção de voo. Duas formas de representar a orientação serão apresentadas a seguir: o uso de matrizes de rotação, combinando com ângulos de *yaw*, *pitch* e *roll*, e a representação por quatérnions.

2.5.1 Matriz de rotação

Em geral, a orientação de um corpo pode ser descrita como uma sequência definida de rotações por ângulos θ_1 , θ_2 e θ_3 ao redor de eixos específicos, contanto que duas sequências consecutivas não envolvam o mesmo eixo. A

sequência de rotação permite a transformação de coordenadas de um referencial do corpo b para o referencial global de navegação n , e vice-versa. Essas transformações podem ser expressas por uma matriz genérica R quadrada de dimensão 3.

A primeira rotação pode ocorrer ao longo de qualquer dos três eixos, a segunda rotação ao longo dos dois eixos não rotacionados e a terceira rotação ao longo de outros dois eixos. Portanto, há $3 \times 2 \times 2 = 12$ sequências possíveis para descrever uma rotação.

É importante ressaltar aqui a diferença entre rotações intrínsecas e extrínsecas. No primeiro caso, as sucessivas rotações ocorrem ao redor dos novos eixos gerados pela rotação, que são solidários ao corpo. Nas rotações extrínsecas, a rotação ocorre ao redor dos eixos globais, que permanecem fixos. As rotações aqui apresentadas são todas do tipo intrínseco.

Duas convenções existem para a representação da rotação no espaço: os ângulos de Euler e os ângulos de Tait-Bryan, cada uma representando 6 sequências possíveis. Os ângulos de Euler representam uma sequência de rotação na qual o primeiro e o terceiro eixos são iguais, tal como ZYZ, ZXZ ou YZY. Os ângulos de Tait-Bryan, geralmente utilizados em aplicações aeroespaciais, representam a rotação de um corpo ao redor de três eixos distintos x-y-z, ou qualquer permutação deles, tal como XYZ, ZXY ou YZX.

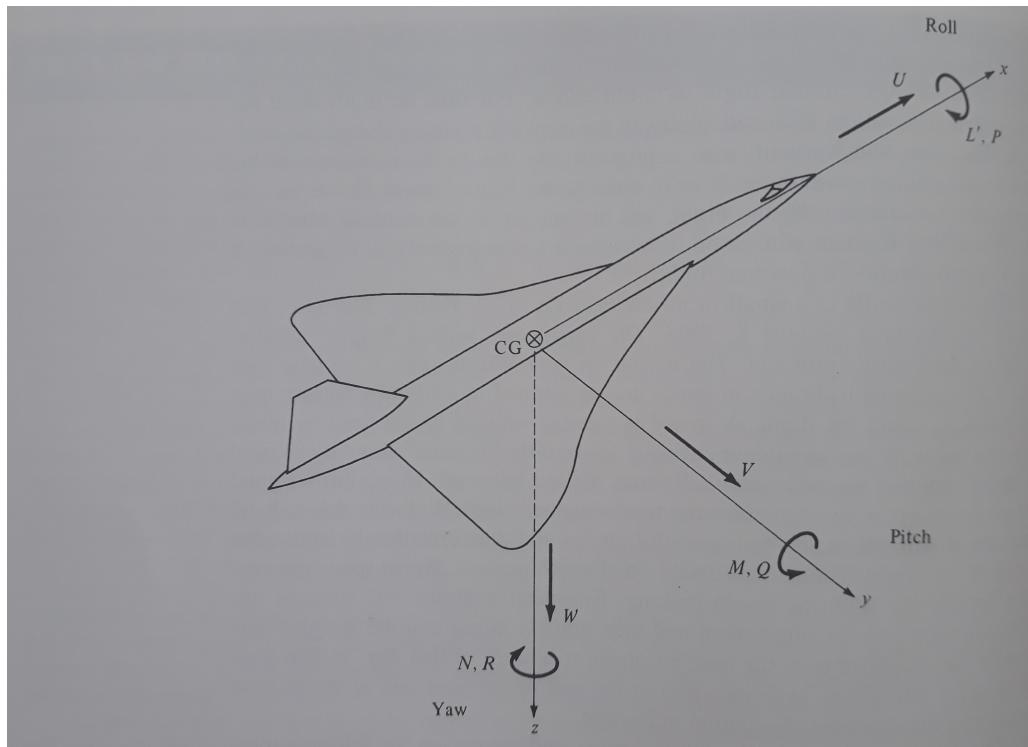
A convenção adotada na aviação é de uma rotação ao redor de z, seguida por rotações em y e x. O ângulo de rotação em z é chamado de ângulo de guinada ou, em inglês, *yaw*, o ângulo em y de arfagem ou *pitch* e o ângulo em x de ângulo de rolamento ou *roll* (figura 9). Notar que o eixo z é definido como apontando para baixo, de maneira que a elevação do nariz (*pitch*) seja representada por ângulos positivos.

Dado o exposto acima, um vetor v^n , expresso no referencial de navegação, é transformado para o as coordenadas do referencial do corpo, obtendo-se v^b

$$v^b = R_n^b v^n \quad (6)$$

onde R_n^b é a matriz de transformação do referencial de navegação para o corpo, que é representado pelo uso dos ângulos de Tait-Bryan.

Figura 9 - Convenção aeronáutica para os eixos



Fonte: ANDERSON (2005)

$$R_z(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Caso a rotação seja em x ou y, então:

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (8)$$

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (9)$$

As matrizes acima podem ser combinadas sucessivamente para obter-se a matriz R_n^b :

$$R_n^b = R_x(\phi) R_y(\theta) R_z(\psi) \quad (10)$$

$$R_n^b = R(\psi, \theta, \phi) = \begin{bmatrix} c\psi \cdot c\theta & s\psi \cdot c\theta & -s\theta \\ c\psi \cdot s\theta \cdot s\phi - s\psi \cdot c\phi & s\psi \cdot s\theta \cdot s\phi + c\psi \cdot c\phi & c\theta \cdot s\phi \\ c\psi \cdot s\theta \cdot c\phi + s\psi \cdot s\phi & s\psi \cdot s\theta \cdot c\phi - c\psi \cdot s\phi & c\theta \cdot c\phi \end{bmatrix} \quad (11)$$

Onde as abreviações adotadas indicam cosseno para c e seno para s.

Deve-se observar que a rotação, assim como a multiplicação de matrizes em geral, não é uma operação comutativa, ou seja, a permutação da sequência não produz o mesmo resultado; sequências de rotação diferentes fornecerão matrizes diferentes.

A matriz de rotação possui as seguintes propriedades:

$$R \cdot R^T = I \quad (12)$$

$$\det(R) = 1 \quad (13)$$

A relação acima implica que a matriz inversa, R_b^n , que permite a transformação das coordenadas de um vetor dado no referencial do corpo para o referencial de navegação, obedece:

$$R_b^n = (R_n^b)^{-1} = (R_n^b)^T = R_z^T R_y^T R_x^T \quad (14)$$

Dada uma matriz de rotação genérica $C_{3 \times 3}$, representando o cosseno dos ângulos entre os eixos de dois referenciais, é possível extrair os ângulos de *yaw*, *pitch* e *roll* comparando-a com a matriz já obtida:

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (15)$$

$$\psi = \tan^{-1} \left(\frac{c_{12}}{c_{11}} \right) \quad (16)$$

$$\theta = \tan^{-1} \left(\frac{-c_{13}}{\sqrt{c_{23}^2 + c_{33}^2}} \right) \quad (17)$$

$$\phi = \tan^{-1} \left(\frac{c_{23}}{c_{33}} \right) \quad (18)$$

O quadrante do ângulo deve ser selecionado considerando o sinal do numerador e do denominador, o que pode ser facilmente realizado pelo uso de uma função atan2 .

Os resultados acima expõem uma singularidade presente na representação matricial; caso o ângulo de *pitch* θ alcance o valor de 90 graus, c_{11} , c_{12} , c_{23} e c_{33} se tornarão nulos e os ângulos ψ e ϕ se tornarão indeterminados.

Essa situação é conhecida em inglês como *gimbal lock* ou trava gimbal, em referência ao sistema de plataforma estável e a perda de um grau de liberdade do sistema. Esse problema pode ser evitado a partir da instalação de um quarto *gimbal* ou da utilização da representação por quatérnions, que por usarem 4 (quatro) parâmetros em vez de 3 (três), não estão sujeitos a essa singularidade.

2.5.2 Quatérnions

Quatérnions podem ser vistos como uma generalização dos números complexos. Propostos em 1843 pelo matemático e físico irlandês William Hamilton, um quatérnion q é um número definido por uma parte escalar a e por uma parte vetorial b, c, d da seguinte forma:

$$q = a + b \cdot i + c \cdot j + d \cdot k \quad (19)$$

i , j e k são versores no espaço tridimensional que obedecem a:

$$i^2 = j^2 = k^2 = i \cdot j \cdot k = -1 \quad (20)$$

$$\begin{aligned} i \cdot j &= k, j \cdot i = -k \\ j \cdot k &= i, k \cdot j = -i \\ k \cdot i &= j, i \cdot k = -j \end{aligned} \quad (21)$$

A norma de um quatérnion q é definida por:

$$|q| = \sqrt{a^2 + b^2 + c^2 + d^2} \quad (22)$$

Um quatérnion é dito unitário caso sua norma seja igual a 1. Utilizando-se as relações acima, é possível demonstrar que o resultado da multiplicação de dois quatérnions q_1 e q_2 é um terceiro quatérnion q_3 , tal que:

$$q_1 \cdot q_2 = q_3 = \begin{bmatrix} a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 \\ a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2 \\ a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2 \\ a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2 \end{bmatrix} \quad (23)$$

Como um quatérnion conta com uma parte escalar s e uma parte vetorial $\vec{v}$, o produto de dois quatérnions pode igualmente ser descrito por:

$$q_1 \cdot q_2 = q_3 = \begin{bmatrix} s_1 \cdot s_2 - \langle \vec{v}_1, \vec{v}_2 \rangle \\ s_1 \vec{v}_2 + s_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2 \end{bmatrix} \quad (24)$$

É importante observar que a multiplicação de quatérnions não é comutativa – devido à presença do produto vetorial - ou seja:

$$q_1 \cdot q_2 \neq q_2 \cdot q_1 \quad (25)$$

O conjugado de um quatérnion q , denotado aqui por $\hat{q}$, é um quatérnion com a mesma parte escalar de q , mas parte vetorial de sinal oposto:

$$\hat{q} = \begin{bmatrix} a \\ -b \\ -c \\ -d \end{bmatrix} \quad (26)$$

Em geral, de acordo com o teorema de Euler, uma rotação qualquer no espaço pode ser descrita como a rotação de um dado vetor por um ângulo θ . Esse vetor corresponde ao autovetor da matriz de rotação. Um quatérnion pode representar essa rotação através do seguinte formato (TITTERTON, 2004):

$$\mathbf{q} = \begin{bmatrix} \cos \frac{\theta}{2} \\ \vec{u} \cdot \sin \frac{\theta}{2} \end{bmatrix} \quad (27)$$

$$\|\vec{u}\| = 1 \quad (28)$$

Pode-se provar que a rotação de um vetor qualquer $\vec{p} = (p_x, p_y, p_z)$ ao redor de $\vec{u}$ pelo ângulo θ produz um vetor $\vec{p}'$, tal que:

$$\mathbf{p}' = \mathbf{q} \mathbf{p} \mathbf{q}^* \quad (29)$$

A notação $\mathbf{p}'$ e $\mathbf{p}$ representa os vetores escritos em sua corresponde forma de quatérnion, com parte escalar nula, isto é:

$$\mathbf{p} = \begin{pmatrix} 0 \\ p_x \\ p_y \\ p_z \end{pmatrix} \quad (30)$$

$$\mathbf{p}' = \begin{pmatrix} 0 \\ p'_x \\ p'_y \\ p'_z \end{pmatrix} \quad (31)$$

A partir dos componentes do quatérnion, a respectiva matriz de rotação associada é:

$$R_b^n = (R_n^b)^T = \begin{bmatrix} (a^2 + b^2 - c^2 - d^2) & 2(bc - ad) & 2(bd + ac) \\ 2(bc + ad) & (a^2 - b^2 + c^2 - d^2) & 2(cd - ab) \\ 2(bd - ac) & 2(cd + ab) & (a^2 - b^2 - c^2 + d^2) \end{bmatrix} \quad (32)$$

Igualmente, a partir dos ângulos de *roll*, *pitch* e *yaw*, é possível obter o quatérnion equivalente a rotação acima com os seguintes componentes (HENDERSON, 1977, p. 25):

$$a = \cos \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} \quad (33)$$

$$b = \cos \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} - \sin \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} \quad (34)$$

$$c = \cos \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} \quad (35)$$

$$d = \sin \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} - \cos \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} \quad (36)$$

A implementação da orientação por quatérnions é frequente em aplicações aeroespaciais. A vantagem dessa representação é a não existência de singularidades e, como são usados 4 variáveis apenas para representar uma dada orientação, a complexidade numérica é menor que o tratamento análogo com matrizes.

2.6 SOLUÇÃO DAS EQUAÇÕES DE NAVEGAÇÃO

As equações dinâmicas são agora resolvidas matematicamente, de maneira a permitir a orientação e posição a partir das entradas do giroscópio e do acelerômetro.

2.6.1 Algoritmo dos cossenos diretos

A rotação do veículo ao redor dos seus próprios eixos faz com que a matriz de rotação se torne uma função do tempo. A propagação da matriz de rotação é dada pela equação:

$$\dot{R}_b^n(t) = R_b^n(t) \Omega(t) \quad (37)$$

onde define-se a matriz rotação angular:

$$\Omega(t) = \begin{bmatrix} 0 & -\omega_{bz} & \omega_{by} \\ \omega_{bz} & 0 & -\omega_{bx} \\ -\omega_{by} & \omega_{bx} & 0 \end{bmatrix} \quad (38)$$

O vetor $\omega_b = [\omega_{bx} \ \omega_{by} \ \omega_{bz}]^T$ é a velocidade angular fornecida pelas indicações dos giroscópios presos ao corpo.

A equação acima é uma equação diferencial ordinária de primeira ordem. Dentro de um intervalo, a equação acima tem a seguinte solução exponencial:

$$R(t_{k+1}) = R(t_k) \exp \int_{t_k}^{t_{k+1}} \Omega(t) dt \quad (39)$$

Os subscritos e sobrescritos não são escritos de agora em diante por conveniência Definindo a matriz dos deslocamentos angulares como:

$$B = \Delta t \Omega = \begin{bmatrix} 0 & -\sigma_z & \sigma_y \\ \sigma_z & 0 & -\sigma_x \\ -\sigma_y & \sigma_x & 0 \end{bmatrix} \quad (40)$$

Assume-se que a velocidade angular é constante durante o intervalo, ou seja, o modelo é de ordem zero. Uma escolha natural para a velocidade angular é o valor obtido em t_k , contudo uma média dos valores de t_k e t_{k+1} também é possível.

Pela definição da série da exponencial, pode-se expressar a integral como:

$$R(t_{k+1}) = R(t_k) \left(I + B + \frac{B^2}{2!} + \frac{B^3}{3!} + \frac{B^4}{4!} + \dots \right) \quad (41)$$

$$R(t_{k+1}) = R(t_k) \left(I + B + \frac{B^2}{2!} - \frac{\sigma^2 B}{3!} - \frac{\sigma^2 B^2}{4!} + \dots \right) \quad (42)$$

Utilizando-se as séries de Taylor para o seno e cosseno, a expressão acima é reduzida a:

$$R(t_{k+1}) = R(t_k) \left(I + \frac{\sin \sigma}{\sigma} B + \frac{1 - \cos \sigma}{\sigma^2} B^2 \right) \quad (43)$$

$$\sigma^2 = \sigma_x^2 + \sigma_y^2 + \sigma_z^2 \quad (44)$$

O desenvolvimento acima é chamado na literatura de algoritmo dos cossenos diretos. Em geral, os coeficientes das matrizes B e B^2 , que envolvem termos trigonométricos, são aproximados pelo cálculo de uma série usando um número finito de termos, isto é:

$$R(t_{k+1}) = R(t_k) (I + a_1 B + a_2 B^2) \quad (45)$$

$$a_1 = 1 - \frac{\sigma^2}{3!} + \frac{\sigma^4}{5!} - \dots \quad (46)$$

$$a_2 = \frac{1}{2} - \frac{\sigma^2}{4!} + \frac{\sigma^4}{6!} - \dots \quad (47)$$

2.6.2 Algoritmo dos quatérnions

De maneira semelhante ao exposto acima, a equação de propagação para o quatérnion $\mathbf{q}$ ao longo do tempo, que representa R_b^n , é dada por:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{a} \\ \dot{b} \\ \dot{c} \\ \dot{d} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} * \begin{bmatrix} 0 \\ \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} \quad (48)$$

A equação pode ser reescrita para uma equação de primeira ordem reordenando os termos das matrizes:

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_{bx} & -\omega_{by} & -\omega_{bz} \\ \omega_{bx} & 0 & \omega_{bz} & -\omega_{by} \\ \omega_{by} & -\omega_{bz} & 0 & \omega_{bx} \\ \omega_{bz} & \omega_{by} & -\omega_{bx} & 0 \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \frac{1}{2} S(\omega) * \mathbf{q} \quad (49)$$

De maneira análoga aos cossenos diretos, a equação acima sugere uma solução exponencial. Para o caso em que o vetor $\vec{\omega}$ permanece constante ao longo do intervalo de integração Δt , a equação tem solução (TITTERTON, 2004):

$$\mathbf{q}_{k+1} = \left[\exp \frac{1}{2} \int_{t_k}^{t_{k+1}} S(\omega) dt \right] \mathbf{q}_k \quad (50)$$

$$\int_{t_k}^{t_{k+1}} S(\omega) dt = \Sigma = \begin{bmatrix} 0 & -\sigma_x & -\sigma_y & -\sigma_z \\ \sigma_x & 0 & \sigma_z & -\sigma_y \\ \sigma_y & -\sigma_z & 0 & \sigma_x \\ \sigma_z & \sigma_y & -\sigma_x & 0 \end{bmatrix} \quad (51)$$

Onde a seguinte notação é usada:

$$\sigma_i = \omega_i \Delta t \quad (52)$$

Logo, de maneira semelhante ao desenvolvimento anterior, a solução é:

$$\mathbf{q}_{k+1} = \exp\left(\frac{\Sigma}{2}\right) \mathbf{q}_k \quad (53)$$

onde $\mathbf{q}_{k+1}$ denota o quatérnions no instante de tempo t_{k+1}

A exponenciação da matriz pode ser expandida através do uso da série de Taylor para o número de Euler e do uso de multiplicação de matrizes, isso é:

$$\exp\left(\frac{\Sigma}{2}\right) = I + \frac{\Sigma}{2} + \frac{(\Sigma \times \Sigma)}{2^2 2!} + \frac{(\Sigma \times \Sigma \times \Sigma)}{2^3 3!} + \dots \quad (54)$$

É trivial demonstrar que:

$$\Sigma \times \Sigma = -\sigma^2 I \quad (55)$$

$$\Sigma \times \Sigma \times \Sigma = (\Sigma \times \Sigma) \times \Sigma = -\sigma^2 \Sigma \quad (56)$$

$$\sigma^2 = \sigma_x^2 + \sigma_y^2 + \sigma_z^2 \quad (57)$$

$$\mathbf{q}_{k+1} = \left[I \cos(\sigma/2) + \Sigma \frac{\text{sen}(\sigma/2)}{\sigma} \right] \mathbf{q}_k \quad (58)$$

Por fim, usando as definições de produto de quatérnion, chega-se ao resultado:

$$\mathbf{q}_{k+1} = \mathbf{q}_k \mathbf{r}_k \quad (59)$$

onde $\mathbf{r}_k$ é um quatérnion definido por:

$$\mathbf{r}_k = \begin{bmatrix} \cos \sigma/2 \\ \left(\frac{\sigma_x}{\sigma}\right) \text{sen} \sigma/2 \\ \left(\frac{\sigma_y}{\sigma}\right) \text{sen} \sigma/2 \\ \left(\frac{\sigma_z}{\sigma}\right) \text{sen} \sigma/2 \end{bmatrix} \quad (60)$$

2.6.3 Obtenção das velocidade e posições

Os métodos apresentados acima fornecem uma solução para a determinação da orientação de um veículo em função do tempo, dadas as medições ω_b dos giroscópios e de uma orientação inicial. Com a obtenção da orientação do corpo

para um tempo qualquer, o vetor forças específicas $f_b = [f_{bx} \ f_{by} \ f_{bz}]^T$ obtido pelos acelerômetros pode ser projetado no referencial de navegação:

$$f_n(t) = R_b^n(t) f_b(t) \quad (61)$$

Por fim, a velocidade e posição são obtida pela integração dos valores da aceleração, compensados pelo valor da gravidade:

$$v_n(t) = v_n(t_0) + \int_{t_0}^t a_n(t') dt' + g dt' \quad (62)$$

$$s_n(t) = s_n(t_0) + \int_{t_0}^t v_n(t') dt' \quad (63)$$

A integração pode ser feita considerando a variação de velocidade u_n medida pelos acelerômetros em um intervalo de amostragem (TITTERTON, 2003):

$$u_n = R_k \int_{t_k}^{t_{k+1}} A f_b(t) dt \quad (64)$$

onde R_k é utilizado para simplificar a notação, representando a matriz de rotação no tempo t_k . Expandindo a matriz A :

$$u_n = R_k \int_{t_k}^{t_{k+1}} \left(I + B + \frac{B^2}{2!} + \frac{B^3}{3!} + \dots \right) f_b(t) dt \quad (65)$$

Desprezando os termos de ordem igual ou maior que 2 e reconhecendo o produto vetorial do segundo termo, visto que a matriz B acima é antissimétrica, obtém-se:

$$u_n = R_k \left[\int_{t_k}^{t_{k+1}} f_b(t) dt + \int_{t_k}^{t_{k+1}} \alpha \times f_b(t) dt \right] \quad (66)$$

onde α é o vetor dado pelo incremento do ângulo:

$$\alpha = \int_{t_k}^t \omega_b(t) dt \quad (67)$$

Realizando a integração por partes do produto vetorial e definindo o vetor v como:

$$v = \int_{t_k}^t f_b(t) dt \quad (68)$$

Chega-se, finalmente, a expressão:

$$u_n = R_k \left(v_{k+1} + \frac{1}{2} \alpha_{k+1} \times v_{k+1} + \int_{t_k}^{t_{k+1}} (\alpha \times f_b - \omega_b \times v) dt \right) \quad (69)$$

A expressão acima é composta por três termos: o primeiro, representa a integral das medições do acelerômetro; o segundo é o produto vetorial do incremento do ângulo com o incremento da velocidade – que é chamado de correção da rotação, e o terceiro é um termo dinâmico. Caso os valores de f_b e ω_b sejam constantes durante o intervalo, o terceiro termo será nulo, chegando-se a (TITTERTON, 2003):

$$u_n = R_k \left(v_{k+1} + \frac{1}{2} \alpha_{k+1} \times v_{k+1} \right) \quad (70)$$

A partir dos valores acima do incremento, é possível calcular a velocidade no tempo $k+1$ como:

$$v_{n(k+1)} = v_{n(k)} + u_n + g \Delta t \quad (71)$$

Por fim, um método numérico é empregado para o cálculo do vetor posição x_{k+1} , tal como o método de Euler, dos trapézios ou a regra de Simpson, descritos pelas seguintes fórmulas

$$x_{k+1} = x_k + v_k \Delta t \quad (72)$$

$$x_{k+1} = x_k + \left(\frac{v_k + v_{k+1}}{2} \right) \Delta t \quad (73)$$

$$x_{k+1} = x_{k-1} + \left(\frac{v_{k-1} + 4v_k + v_{k+1}}{3} \right) \Delta t \quad (74)$$

O método apresentado baseia-se no algoritmo dos cossenos diretos. De maneira semelhante, o desenvolvimento pode ser expandido para a representação de quatérnions. O incremento da velocidade num intervalo torna-se:

$$\mathbf{u}_n = \mathbf{q}_k \left(\mathbf{v}_{k+1} + \frac{1}{2} \alpha_{k+1} \times \mathbf{v}_{k+1} + \int_{t_k}^{t_{k+1}} (\alpha \times \mathbf{f}_b - \omega_b \times \mathbf{v}) dt \right) \dot{\mathbf{q}}_k \quad (75)$$

A partir desse resultado, os mesmos métodos de integração para a posição e velocidade são aplicáveis.

2.7 CÁLCULO DO TILT

A detecção da gravidade pelo acelerômetro, mesmo quando imóvel, oferece a oportunidade da detecção dos ângulos de *pitch* e de *roll*, que em conjunto são chamados de *tilt*. Usando a matriz de rotação já definida, pode-se provar que o vetor gravidade, expresso como $[0 \ 0 \ -g]^T$ no referencial NED, será visto no referencial do corpo como:

$$\mathbf{g}_b = -g \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (76)$$

Portanto, o ângulo de *pitch* θ e de *roll* ϕ podem ser expressos em função dos valores indicados pelo acelerômetro a partir de:

$$\theta = \tan^{-1} \left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}} \right) \quad (77)$$

$$\phi = \tan^{-1} \left(\frac{-a_y}{-a_z} \right) \quad (78)$$

Para o ângulo de *roll*, em particular, os sinais negativos são mantidos de maneira a permitir a utilização da função atan2.

Deve-se mencionar que, em regime de operação, os acelerômetros também detectarão as forças específicas acarretadas por outras forças que não a gravidade. Portanto, as fórmulas acima devem ser vistas com ressalvas: fornecem uma boa estimativa do *pitch* e *roll* mas serão distorcidas pela presença de outras forças. Naturalmente, nenhuma informação a respeito do *yaw* pode ser obtida por esse

método, visto que o acelerômetro não detecta nenhuma variação em suas medições quando for rotacionado ao longo do eixo z global.

O algoritmo de navegação também permite a determinação desses ângulos pelas integrações das medições dos giroscópios. No caso dos acelerômetros, há um grande ruído, mas essas medições não sofrem com o *drift* advindo da integração. No caso dos giroscópios, embora exista esse *drift*, as medições são mais confiáveis durante um intervalo de tempo menor.

A utilização de medições auxiliares é de interesse na área de navegação, visto que confere maior acurácia ao sistema ao combinar técnicas com características diferentes. A fusão desses dados pode ser implementada pelas técnicas descritas a seguir: o filtro complementar, de fácil entendimento e aplicação, e o filtro de Kalman, uma forma mais sofisticada e mais geral.

2.8 FILTRO COMPLEMENTAR

O filtro complementar busca utilizar os componentes de alta frequência de uma medição e os de baixa frequência de outra medição, otimizando assim os resultados fornecidos pelos sensores. O filtro complementar pode ser visto como o filtro de Kalman para o estado estacionário, a partir de um ganho fixo, sem considerações sobre as propriedades estatísticas do ruído (HIGGINS JR, 1975).

Em geral, dadas duas medições x e y , a estimativa $\hat{z}$ é dada por:

$$\hat{z} = (1 - G(s))x + G(s)y \quad (79)$$

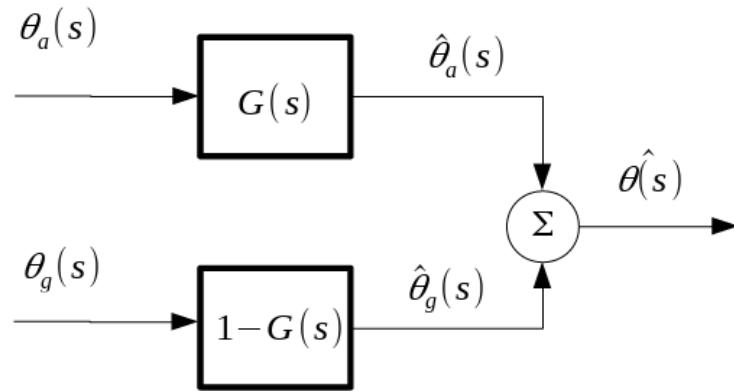
O aspecto complementar é decorrente da soma dos coeficientes serem iguais a 1. Em particular, é interessante observar que se $x = y$, então $\hat{z} = x$.

Uma forma fácil de se implementar o filtro complementar com as medições θ_a do acelerômetro e θ_g do giroscópio para a estimação $\hat{\theta}$ é (REDOLF, 2011):

$$\hat{\theta} = G(s)\theta_a + (1 - G(s))\theta_g \quad (80)$$

O diagrama de blocos para o filtro complementar é mostrado na figura 10

Figura 10 - Diagrama de blocos do filtro complementar



Fonte: Adaptado (HIGGINS, 1975)

Assumindo uma função de transferência passa baixo de primeira ordem $G(s)$ tal que:

$$G(s) = \frac{\alpha}{s + \alpha} \quad (81)$$

Notar que o termo $1 - G(s)$, por sua vez, será um filtro passa alto. O ângulo pode ser calculado por:

$$\hat{\theta}_k = \hat{\theta}_{a(k)} + \hat{\theta}_{g(k)} \quad (82)$$

onde $\theta_{a(k)}$ e $\theta_{g(k)}$ são obtidas a partir da transformada z usando um retentor de ordem zero :

$$\hat{\theta}_{a(k)} = e^{-\Delta t/\tau} \hat{\theta}_{a(k-1)} + (1 - e^{-\Delta t/\tau}) \theta_{a(k)} \quad (83)$$

$$\hat{\theta}_{g(k)} = e^{-\Delta t/\tau} \hat{\theta}_{g(k-1)} + \theta_{g(k)} - \theta_{g(k-1)} \quad (84)$$

τ representa a constante do filtro e está associada a frequência de corte. Maiores valores de τ fazem o filtro seguir mais de perto as indicações do giroscópio; menores valores tornam os valores do filtro mais próximo ao acelerômetro.

2.9 FILTRO DE KALMAN

O filtro de Kalman leva o nome do seu idealizador, o matemático e engenheiro húngaro Rudolf E. Kalman, que o desenvolveu no final da década de 1950. A ideia de Kalman foi expandir a formulação do filtro de Wiener do domínio da frequência para o domínio de estados.

Na prática, o filtro de Kalman combina o conhecimento existente da dinâmica de um sistema – linear ou não linear - com medições realizadas por algum tipo de sensor, que geralmente são uma função linear do vetor de espaço de estados. O seu grande valor está na possibilidade de diferentes implementações e a fusão de sensores com características diferentes. Em particular, o filtro de Kalman se tornou a arquitetura *de facto* para a integração de sensores inerciais com sistemas GNSS.

Em geral, um sistema linear descrito em espaço de estados em tempo contínuo tem o formato genérico:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (85)$$

$$y(t) = Cx(t) + Du(t) \quad (86)$$

A primeira equação é denominada “equação de estados do sistema” enquanto a segunda, “equação de saída”. Os componentes das equações, bem como as suas dimensões, são listados na Tabela 2, para o caso de um sistema genérico, onde n é o tamanho do vetor estado (vetor coluna), p é o número de saídas e m é o número de entradas (NISE, 2004)

Tabela 2 - Termos na equação de estados

Símbolo	Significado	Dimensões
x	Vetor espaço de estados	$n \times 1$
$\dot{x}$	Derivada do vetor espaço de estados	$n \times 1$
y	Valor de saída	$p \times 1$
u	Vetor de entrada ou de controle	$m \times 1$
A	Matriz do sistema ou de transição	$n \times n$
B	Matriz de entrada	$n \times m$
C	Matriz de saída	$p \times n$
D	Matriz <i>feedforward</i>	$p \times m$

Fonte: O autor (2019)

As equações acima representam o sistema em tempo contínuo. No caso discreto, as equações podem ser rescritas como equações de diferença de 1ª ordem em função do tempo, em um formato genérico (assumindo D como nula):

$$x_k = F_k x_{k-1} + G_k u_{k-1} \quad (87)$$

$$z_k = H_k x_k \quad (88)$$

Onde a seguinte notação é adotada para o tempo:

$$x_k = x(t_k) \quad (89)$$

$$t_1 < t_2 < t_3 \dots t_k < t_{k+1} \dots < t_m \quad (90)$$

$$t_{k+1} - t_k = \Delta t \quad (91)$$

Idealmente, as equações da dinâmica são determinísticas, isto é, para uma dada condição inicial $x(t_0)$ e determinadas entradas $u(t)$, é possível estimar o único estado possível $x(t)$ para qualquer tempo posterior. Contudo, em aplicações reais, tanto o sistema quanto a medição estão sujeitas a algum tipo de ruído. Portanto, as equações do sistema precisam ser rescritas considerando o ruído do processo w_k e o ruído da medição v_k :

$$x_k = F x_{k-1} + G u_{k-1} + w_{k-1} \quad (92)$$

$$z_k = H x_k + v_k \quad (93)$$

Assume-se que o ruído do processo w_k e o ruído da medição v_k são ruídos brancos gaussianos de média zero, tal que o valor esperado $E\langle \rangle$ seja:

$$E\langle w_k w_j^T \rangle = Q \Delta_{ij} \quad (94)$$

$$E\langle v_i v_j^T \rangle = R \Delta_{ij} \quad (95)$$

Onde o símbolo Δ_{ij} é o delta de Kronecker, igual a 1 se $i=j$ e 0, caso contrário. Q e R são definidas como as matrizes de covariância.

Assume-se, também, que não há nenhuma relação entre esses ruídos:

$$E\langle w_k v_j^T \rangle = 0 \quad (96)$$

As condições iniciais para o sistema no tempo t_0 agora incluem a covariância associada ao valor:

$$E\langle x_0 \rangle = \hat{x}_0 \quad (97)$$

$$E\langle \tilde{x}_0 \tilde{x}_0^T \rangle = P_0 \quad (98)$$

A notação utilizada daqui em diante tem o formato genérico $x_{a|b}$, que significa a estimativa do estado no tempo a com os dados da medição até o tempo b . Se a for superior a b , o problema é denominado como estimação; se ambos forem iguais, filtragem e caso b seja superior a a , utiliza-se o termo suavização ou interpolação (KALMAN, 1960).

Utilizando-se o estado previsto pela equação dos estados, $x_{k|k-1}$, e a posterior medição z_k , - duas estimativas independentes -, calcula-se um estimador para o estado a partir da média ponderada com ganhos $\bar{K}_k$ e K_k^1 de forma a obter um estado ótimo, isto é, um estado com a menor variância possível em relação ao valor verdadeiro. Sob esse aspecto, os filtros de Kalman podem ser vistos como uma técnica mais sofisticada, mas relacionada ao método dos mínimos quadráticos, concebido pelo matemático alemão Carl F. Gauss. A relação descrita pode ser expressa como:

$$x_{k|k} = K_k^1 x_{k|k-1} + \bar{K}_k z_k \quad (99)$$

O algoritmo é dividido em duas fases. Na fase de previsão, o filtro utiliza a equação de estados do sistema para prever o próximo estado. Essa estimativa é comumente denominada de *a priori*, visto que ocorre sem os valores da medição. A covariância $P_{k|k-1}$ dos estados também é atualizada, de acordo com a propagação gaussiana de uma variável aleatória.

$$\hat{x}_{k|k-1} = F \hat{x}_{k-1|k-1} + G u_{k-1} \quad (100)$$

$$P_{k|k-1} = F P_{k-1|k-1} F^T + Q \quad (101)$$

Posteriormente, na fase de atualização, com a medição disponível, o resíduo $\tilde{y}_k$, definido como a diferença entre o valor previsto da medição, dado a dinâmica do sensor e a medição real z_k é calculado:

$$\tilde{y}_k = z_k - H x_{k|k-1} \quad (102)$$

A covariância do resíduo S_k é calculada, seguindo a mesma lógica da covariância dos estados:

$$S_k = H P_{k|k-1} H^T + R \quad (103)$$

O ganho ótimo $\bar{K}_k$ é calculado por:

$$\bar{K}_k = P_{k|k-1} H^T S^{-1} \quad (104)$$

O ganho $\bar{K}_k$ é calculado de maneira a minimizar o valor esperado da diferença quadrática do estado real e do valor $x_{k|k}$, isso é:

$$\frac{\partial E \langle (x_{k|k} - x_k)^2 \rangle}{\partial \bar{K}_k} = 0 \quad (105)$$

A estimativa do estado é atualizada, portanto, com o estado $x_{k|k}$, também chamado de *a posteriori*:

$$x_{k|k} = x_{k|k-1} + \bar{K}_k \tilde{y}_k \quad (106)$$

Por fim, a covariância dessa nova estimativa também deve ser calculada, $P_{k|k}$:

$$P_{k|k} = (I - \bar{K}_k H) P_{k|k-1} \quad (107)$$

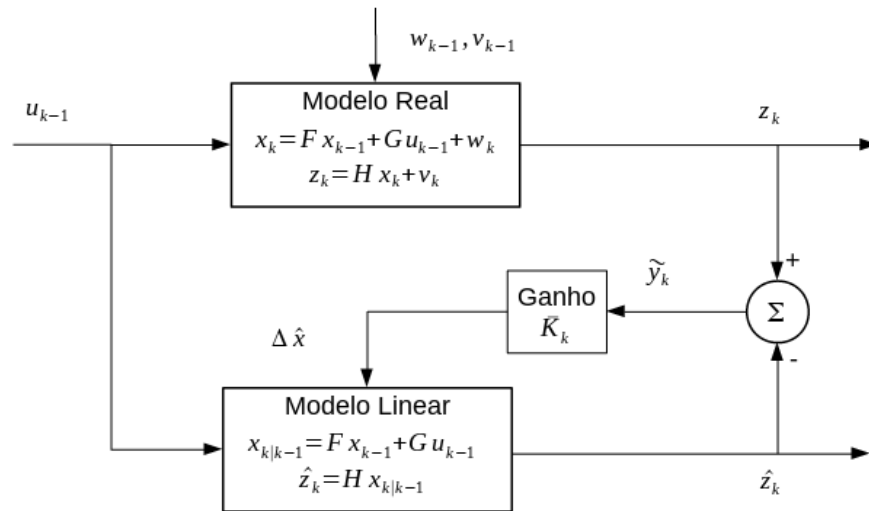
Os valores atualizados do estado e da covariância são alimentados para o próximo ciclo e o algoritmo prossegue, conforme pode ser visto na figura 11.

Uma vantagem oferecida pelos filtros de Kalman é a não necessidade de coincidência dos ciclos de previsão e atualização, ou seja, as medições podem ocorrer de forma periódica ou mesmo irregular. Durante o intervalo sem medições, o estado estimado do sistema é propagado somente pela dinâmica do sistema, até ser atualizado por uma nova medição. Matematicamente, isso corresponde a definir a matriz H como nula, de maneira que seguindo o algoritmo, tem-se:

$$x_{k|k} = x_{k|k-1} \quad (108)$$

$$P_{k|k} = P_{k-1|k} \quad (109)$$

Figura 11 - Visão do algoritmo do filtro de Kalman



Fonte: Adaptado (TITTERTON, 2004)

Para sistemas não-lineares, uma abordagem frequentemente utilizada é o filtro de Kalman estendido (EKF), que lineariza as equações através de derivadas parciais em relação ao estado estimado. No EKF, as equações de estado são representadas na sua forma não linear:

$$\dot{x}(t) = f(t, x(t), u(t)) \quad (110)$$

$$y(t) = h(t, x(t), u(t)) \quad (111)$$

Em relação ao filtro de Kalman no caso linear, o EKF não é em geral o estimador ótimo e a convergência do estado não é garantida se o sistema tiver uma forte não linearidade. Contudo, esses são capazes de lidar com sistemas não lineares. Essa é uma grande vantagem em relação ao filtro mais simplificado.

Os filtros de Kalman – no formato do EKF - tiveram uma importante aplicação nos sistemas de navegação para as missões lunares Apollo, lançadas entre 1968 e 1972. Os requisitos de navegação podem ser melhor compreendidos pelos números a seguir: a distância entre a Terra e Lua é de aproximadamente 400.000 km, e a trajetória da espaçonave devia passar a uma distância de 100 km da superfície lunar, para que o conjunto pudesse desacelerar e ser transferido para a órbita lunar. Ao mesmo tempo, a Lua é um alvo móvel em relação a Terra, orbitando a uma velocidade relativa de 1 km/s (MCGEE, 1985).

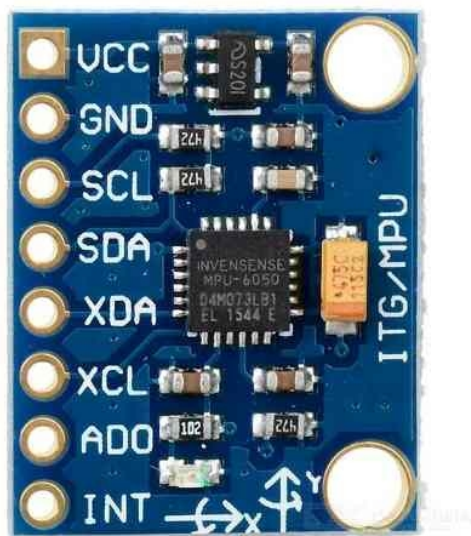
3 METODOLOGIA

Essa seção considera certos aspectos práticos e específicos em relação tanto aos sensores como aos códigos escritos, que se encontram nos anexos. Ademais, a maneira de testar o conjunto a partir do movimento simulado dos servomotores é descrita.

3.1 SENSORES

Para aquisição dos dados e teste do algoritmo, optou-se pelo sensor MPU-6050, um dispositivo de placa impressa, com dimensões de 21,2 mm x 16,4 mm x 3,3 mm, fabricado pela empresa InvenSense. O MPU-6050 conta com um acelerômetro MEMS de 3 eixos e um giroscópio MEMS de 3 eixos, além de um sensor de temperatura interno, e é mostrado na figura 12. Notar o sentido dos eixos X e Y; o eixo Z aponta para fora da página, conforme a regra da mão direita.

Figura 12 - MEMS MPU-6050



Fonte: MERCADO LIVRE (2019)

O sensor é capaz de detectar uma aceleração e rotação de até 2 g e 250 °/s, respectivamente, embora seja possível configurá-lo para trabalhar com intervalos maiores, alterando-se os registradores internos do sensor. Evidentemente, ao

aumentar a escala, a sensibilidade é diminuída. Os valores do giroscópio e do acelerômetro são convertidos por um conversor analógico-digital de 16 bits, fornecendo valores entre -32.768 e 32.677. Os valores reais podem ser obtidos a partir da seguinte fórmula:

$$\text{Valor Métrico} = \frac{\text{Valor Bruto}}{2^{15}} \text{Máximo Valor} \quad (112)$$

O valor adotado para a gravidade é a gravidade padrão conforme definida pelo Comitê Internacional de Pesos e Medidas (ICPM) de 9,80665 m/s² (BIPM, 2006).

As tabelas 3 e 4 a seguir fornecem alguns dos parâmetros principais dos sensores, de acordo com as informações disponibilizadas pelo fabricante no *datasheet* do produto.

Tabela 3 - Parâmetros do giroscópio

Parâmetro	Valor	Unidade
Intervalo de medição	±250	°/s
Comprimento de palavra	16	bits
Fator de sensibilidade	131	LSB/(°/s)
Frequência de amostragem	4-8000	Hz
Não-linearidade (25 °C)	0.2	%
Ruído total (RMS, filtragem interna)	0.05	°/s
Densidade espectral de ruído (10Hz)	0.005	°/s/√Hz

Fonte: InvenSense Inc (2013)

Tabela 4 - Parâmetros do acelerômetro

Parâmetro	Valor	Unidade
Intervalo de medição	±2	g
Comprimento de palavra	16	bits
Fator de sensibilidade	16384	LSB/g
Frequência de amostragem	4-1000	Hz
Não-linearidade (25 °C)	0.5	%
Densidade espectral de ruído (10Hz)	400	μg/√Hz

Fonte: InvenSense Inc (2013)

A plataforma de prototipagem Arduino foi escolhida para estabelecer a comunicação com o sensor. O modelo escolhido foi o Arduino UNO, que conta com um microcontrolador ATmega328, 8 bits de largura de dados e 16 MHz de clock.

A comunicação entre o sensor e o microcontrolador é realizada a partir do protocolo I²C, para *Inter-Integrated Circuit* ou Inter-Circuitos Integrados. Esse protocolo estabelece um dispositivo mestre (no caso, o microcontrolador) e um ou

mais dispositivos escravos (sensores) em paralelo. Os escravos são identificados por endereços de 7 bits preestabelecidos, no caso do MPU-6050 por exemplo, o endereço é 0x68 (0b1101000).

O I²C permite a comunicação em ambos os sentidos a partir da utilização de somente dois fios: SCL para *clock serial*, e SDA para *data serial*. No Arduino, essa comunicação é possível pela biblioteca Wire.h, com as entradas analógicas A4 para SDA e A5 para SCL na placa Uno.

A comunicação entre o microcontrolador e o computador é então realizada a partir da comunicação serial, também chamada de UART ou USART. No Arduino, a taxa de transferência pelo padrão serial é definida pela função *Serial.begin()*, que é chamada no bloco de *setup* do código. Essa taxa é dada em bauds, ou bits por segundo.

Os valores disponíveis, e que se encontram dentro da faixa de interesse para o projeto, são de 9600, 19200, 38400, 57600 e 115200 bauds. A tabela a seguir descreve a frequência de amostragem aproximada para cada variável de interesse – as 3 medições do giroscópio, 3 medições do acelerômetro e o tempo da medição – em função da taxa de Bauds selecionada.

Tabela 5 - Frequência de amostragem em função da taxa serial

Bauds	Frequência (Hz)
9.600	19,2
19.200	38,5
38.400	76,9
57.600	117,6
115.200	235,1

Fonte: o autor (2019)

Para a obtenção dos dados, os valores dos registradores internos do MPU-6050 (0x3B a 0x48) são lidos pelo Arduino, a partir do protocolo I²C, e então impressos no monitor serial pela função *Serial.print()*. O intervalo de tempo entre medições consecutivas é calculada a partir da função *micros()* do Arduino, que retorna o tempo em microssegundos.

3.2 PROGRAMAÇÃO

Os programas e simulações foram realizadas no *software* de análise numérica GNU Octave, versão 4.2.2. O Octave é parte do projeto GNU e, portanto, é um

software livre, sendo disponibilizado pela licença GNU GPL. A escolha do Octave foi pautada em razão de suas capacidades e funcionalidades. Julgou-se que a linguagem de alto nível permitiria mais facilmente a implementação das funções e da lógica do programa, facilitando a visualização dos resultados e reduzindo os problemas encontrados.

A linguagem utilizada no Octave é uma linguagem de programação interpretada baseada no paradigma estruturado, semelhante ao da linguagem C. A sintaxe é baseada em operações com matrizes (EATON, 2019).

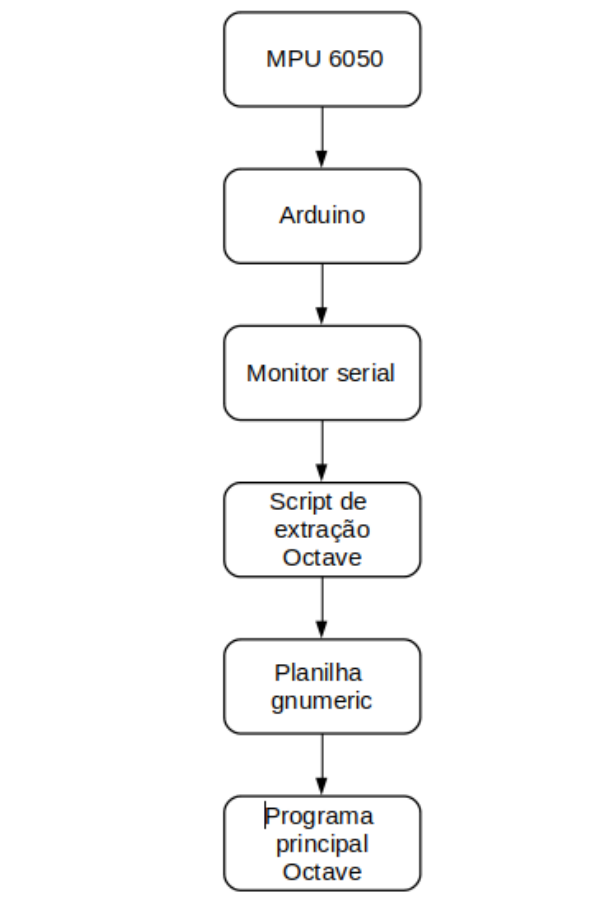
Em termos de sintaxe e de funções nativas, a linguagem utilizada pelo Octave é bastante similar com a do MATLAB, de maneira que a compatibilidade do código entre essas duas plataformas é bastante elevada. Os arquivos de *script* e funções do Octave possuem um formato .m que é executável no MATLAB. As funções, em particular, devem ser salvas em um arquivo contendo somente o seu código e o nome do arquivo deve ser igual a função. As funções são salvas no mesmo diretório, de maneira a serem chamadas pelo programa principal.

Um *script* foi desenvolvido no Octave para a extração dos valores obtidos no monitor serial do Arduino, a partir do uso dos pacotes do Octave *input-output* e *instrument-control*. Esse *script* permite salvar os dados para um arquivo do tipo planilha eletrônica, -tal como .gnumeric -, ou analisar diretamente os dados para obtenção dos resultados de interesse..

Os dados salvos na planilha podem ser posteriormente lidos e executados normalmente. Essa escolha permite maior flexibilidade e uma análise posterior dos resultados caso se deseje alterar algum parâmetro do algoritmo. A figura 13 ilustra o fluxo geral dos dados.

Os códigos consideram o caso genérico de navegação inercial do tipo *strapdown*, convertendo os valores obtidos de aceleração e velocidade angular no referencial do corpo para o referencial de navegação no formato norte, leste e para baixo (NED), fornecendo velocidade, posição e orientação. As condições iniciais dos ângulos de *roll* ϕ_0 e *pitch* θ_0 são dadas pelas leituras iniciais dos acelerômetros, a partir do cálculo do valor médio no primeiro segundo. O ângulo de *yaw* ψ_0 pode ser assumido, sem perdas de generalidade, como sendo nulo, embora possa ser definido a partir do conhecimento do Norte geográfico.

Figura 13 - Fluxo dos dados



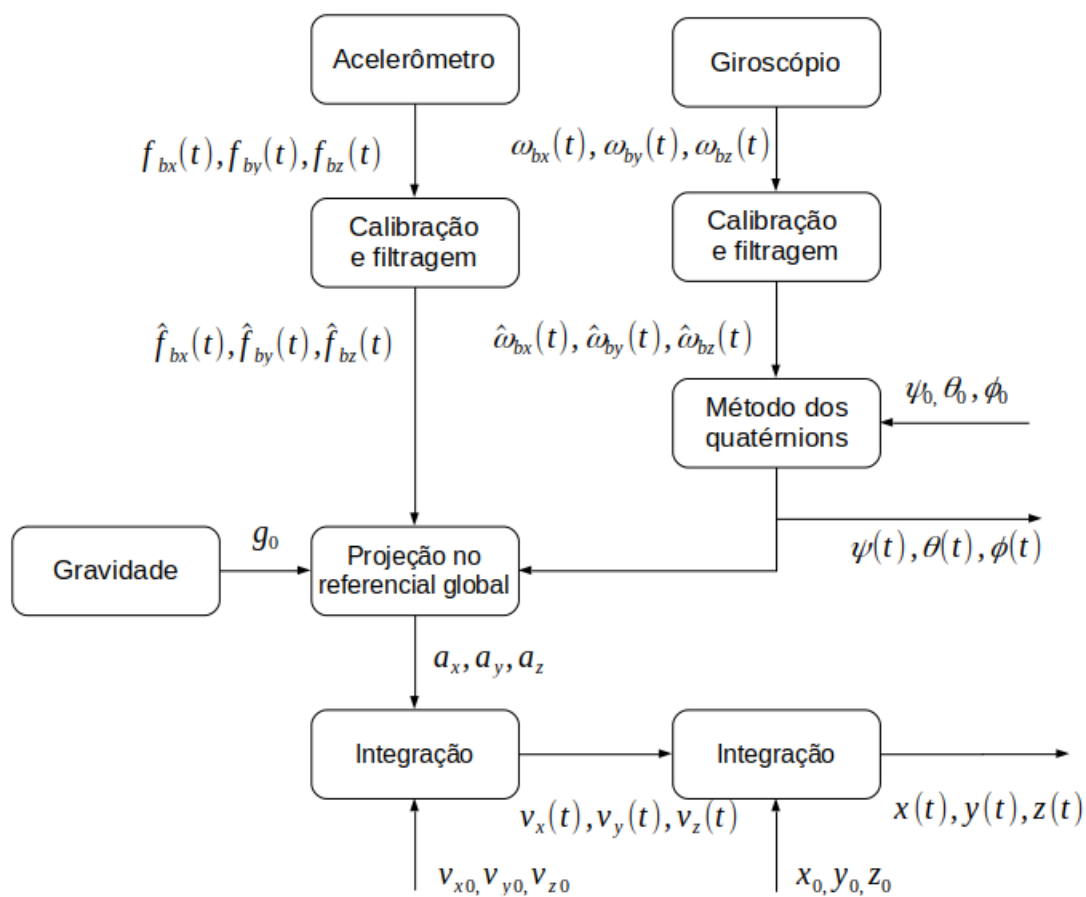
Fonte: o autor (2019)

Pensando nas possíveis aplicações e dada a complexidade do assunto, optou-se por desprezar a rotação da Terra, as variações da gravidade com altitude e o raio finito da Terra, que faz com que as direções dos eixos de referência não sejam sempre paralelos.

O código é baseado na representação em quatérnions, visto que essa representação não possui singularidades e requer apenas 4 componentes para expressar a orientação. Ao todo, somando os *scripts* principais com as funções auxiliares, o código possui um valor próximo a 1000 linhas. Uma visão geral do tratamento dos dados obtidos pelos acelerômetros e giroscópios é mostrado na figura 14.

Os programas foram escritos e testados com o sistema operacional Ubuntu Linux, em um notebook Lenovo Ideapad 310, com processador Intel i5-6200U e 8 GB de RAM. Deve-se notar que a aplicação opera de maneira *offline*, de maneira que os valores dos sensores são capturados por um intervalo de tempo pré-definido e então são processados em lote. Uma visão ampla do algoritmo inercial – do cálculo da orientação ao da posição – é dada na figura 14.

Figura 14 - Sequência do algoritmo inercial



Fonte: o autor (2019)

3.3 CALIBRAÇÃO

Os sensores inerciais possuem erros de fábrica que devem ser corrigidos de maneira a permitir melhores resultados. Os sensores devem ser comparados com algum sinal conhecido para que os parâmetros do erro possam ser encontrados.

Metodologias foram desenvolvidas para a calibração de sensores MEMS, permitindo o ajuste de parâmetros como viés, fator de escala e fatores de não ortogonalidade, a partir do método de mínimos quadráticos (SYED, 2007).

Para os dispositivos do tipo MEMS, caminhada aleatória pelo ruído e vieses não compensados tendem a ser a fonte principal do erro (WOODMAN, 2007).

Observações iniciais com os giroscópios e acelerômetros evidenciaram a observação acima. Naturalmente, o ruído tem uma natureza estocástica e deve ser tratado com algum tipo de filtragem. No presente trabalho, optou-se por estimar o viés do giroscópio e o viés e ganho para o acelerômetro.

Em repouso, o giroscópio teoricamente é capaz de detectar a própria rotação da Terra. Contudo, a rotação da Terra de 4 milésimos de grau por segundo é ainda assim inferior aos 7 milésimos de grau por segundo correspondentes ao menor intervalo detectável. Portanto, o valor do viés para os giroscópios é considerado como o valor médio indicado com os sensores em repouso.

No caso do acelerômetro o sinal de referência, naturalmente, é a gravidade. O viés pode ser obtido facilmente medindo-se os valores médios indicados ao se alinhar o eixo do acelerômetro na direção vertical para cima e para baixo. O viés é então o valor médio dessas medições. Notar que dessa maneira não é necessário o conhecimento da gravidade local.

Uma forma ainda mais geral e poderosa para a determinação dos vieses é o uso do clássico método dos mínimos quadráticos. Em particular, comparado ao método anterior, essa técnica tem a vantagem de ser menos sensível a eventuais erros de alinhamento com a vertical.

Como dito anteriormente, em repouso o acelerômetro é capaz de detectar a gravidade. Isso é, dado um conjunto de medições de média $(\bar{a}_x, \bar{a}_y, \bar{a}_z)$ nos eixos x, y e z do acelerômetro, a magnitude medida do vetor gravidade $|\vec{g}_{med}|$ será dada por:

$$|\vec{g}_{med}| = \sqrt{(\bar{a}_x - b_x)^2 + (\bar{a}_y - b_y)^2 + (\bar{a}_z - b_z)^2} \quad (113)$$

onde (b_x, b_y, b_z) são os vieses dos acelerômetros.

A partir do exposto acima, a abordagem torna-se clara: os vieses devem ser definidos de maneira que o erro quadrático S entre a magnitude da gravidade local

$|\vec{g}|$ e a calculada $|\vec{g}_{med}|$ das N amostras seja mínimo. Mais formalmente, tem-se:

$$S = \sum_{i=1}^N (g - f_i(\bar{a}_x, \bar{a}_y, \bar{a}_z))^2 \quad (114)$$

$$\frac{\partial S}{\partial b_x} = 0 \quad (115)$$

$$\frac{\partial S}{\partial b_y} = 0 \quad (116)$$

$$\frac{\partial S}{\partial b_z} = 0 \quad (117)$$

Onde:

$$f(\bar{a}_x, \bar{a}_y, \bar{a}_z) = \sqrt{(\bar{a}_x - b_x)^2 + (\bar{a}_y - b_y)^2 + (\bar{a}_z - b_z)^2} \quad (118)$$

Esse problema é de natureza não linear. Sua solução pode ser encontrada a partir da linearização da função f ao redor de um ponto inicial (b_{x0}, b_{y0}, b_{z0}) , isso é:

$$b_x = b_{x0} + \Delta b_x \quad (119)$$

$$b_y = b_{y0} + \Delta b_y \quad (120)$$

$$b_z = b_{z0} + \Delta b_z \quad (121)$$

$$f_i = f_{i0} + \frac{\partial f}{\partial b_x} \Delta b_x + \frac{\partial f}{\partial b_y} \Delta b_y + \frac{\partial f}{\partial b_z} \Delta b_z \quad (122)$$

onde as derivadas são calculadas nos pontos $(\bar{a}_{xi}, \bar{a}_{yi}, \bar{a}_{zi})$

Os vieses são então obtidos de maneira iterativa, resolvendo a seguinte equação linear:

$$\begin{bmatrix} \left[\frac{\partial f}{\partial b_x} \frac{\partial f}{\partial b_x} \right] & \left[\frac{\partial f}{\partial b_x} \frac{\partial f}{\partial b_y} \right] & \left[\frac{\partial f}{\partial b_x} \frac{\partial f}{\partial b_z} \right] \\ \left[\frac{\partial f}{\partial b_y} \frac{\partial f}{\partial b_x} \right] & \left[\frac{\partial f}{\partial b_y} \frac{\partial f}{\partial b_y} \right] & \left[\frac{\partial f}{\partial b_y} \frac{\partial f}{\partial b_z} \right] \\ \left[\frac{\partial f}{\partial b_z} \frac{\partial f}{\partial b_x} \right] & \left[\frac{\partial f}{\partial b_z} \frac{\partial f}{\partial b_y} \right] & \left[\frac{\partial f}{\partial b_z} \frac{\partial f}{\partial b_z} \right] \end{bmatrix} \begin{bmatrix} \Delta b_x \\ \Delta b_y \\ \Delta b_z \end{bmatrix} = \begin{bmatrix} (g-f) \frac{\partial f}{\partial b_x} \\ (g-f) \frac{\partial f}{\partial b_y} \\ (g-f) \frac{\partial f}{\partial b_z} \end{bmatrix} \quad (123)$$

onde a seguinte notação é utilizada:

$$\left[\frac{\partial f}{\partial b_i} \frac{\partial f}{\partial b_j} \right] = \sum_{k=1}^N \frac{\partial f_k}{\partial b_i} \frac{\partial f_k}{\partial b_j} \quad (124)$$

Ou seja, o problema é reduzido ao método de mínimos quadráticos linear, onde os parâmetros Δb_k devem ser encontrados.

De maneira análoga, o método descrito também permite a inclusão de três ganhos G_x, G_y, G_z como parâmetros, isso é, a função f pode ser definida como:

$$f(\bar{a}_x, \bar{a}_y, \bar{a}_z) = \sqrt{G_x^2(\bar{a}_x - b_x)^2 + G_y^2(\bar{a}_y - b_y)^2 + G_z^2(\bar{a}_z - b_z)^2} \quad (125)$$

Naturalmente, a matriz de derivadas terá tamanho 6 x 6.

Evidentemente, um critério de parada deve ser definido, a partir da convergência dos valores ou do número de iterações. O algoritmo apresentado acima é conhecido como o algoritmo de Gauss-Newton.

3.4 FILTRAGEM

Os dados do MPU-6050 foram obtidos de maneira pura, isso é, sem o uso do pré-processamento possibilitado pelo processador do dispositivo. Julgou-se que essa decisão permitiria o maior manuseio dos dados e uma maior consciência do sistema.

Após a aquisição, os dados podem ser facilmente filtrados por uma filtro passa-baixo, de maneira a reduzir os ruídos de maior frequência. Um exemplo de filtragem é o filtro de primeira ordem para um sinal discreto $x[k]$, produzindo a sequência discreta $y[k]$, tal que:

$$y[k] = \alpha x[k] + (1 - \alpha) y[k-1] \quad (126)$$

onde o parâmetro α deve estar entre 0 e 1. A aplicação desse filtro suaviza o sinal, deixando-o com um aspecto mais contínuo.

Uma outra forma simples de implementar um filtro passa-baixo é a partir do filtro média móvel, que retorna o valor $y[n]$ de um conjunto de medições $x[k]$ pela seguinte equação:

$$y[n] = \text{med}\left(\sum_{i=-a}^{i=b} x[n+i]\right) \quad (127)$$

Ou seja, o filtro média móvel retorna a média da vizinhança de um elemento n do sinal não filtrado.

Deve-se notar que embora ambos os filtros permitam uma redução do ruído, a atenuação dos componentes de maior frequência leva a uma atraso na resposta. Ou seja, quanto menor for o α ou maior for o comprimento da vizinhança nos casos acima, maior será o tempo que as entradas filtradas levarão para corresponder ao estado do sistema real. Considerações de largura de banda e frequência de Nyquist também deve ser feitas.

3.5 MODELAGEM DO FILTRO DE KALMAN

Uma implementação para o filtro de Kalman na rotação vale-se da linearização para o ângulo θ_k , a partir da velocidade angular ω_k e do intervalo de amostragem Δt , obtendo as equações de estado (ROMANIUK, 2014):

$$\theta_k = \theta_{k-1} + (\omega_{k-1} - b_{k-1}) \Delta t \quad (128)$$

$$b_k = b_{k-1} \quad (129)$$

O vetor espaço de estados x_k agora é composto pelo ângulo θ_k e pelo viés b_k , que também precisa ser compensado e torna-se função do tempo:

$$x_k = \begin{bmatrix} \theta_k \\ b_k \end{bmatrix} \quad (130)$$

A partir da equação acima, as matrizes para o caso discreto podem ser definidas como:

$$F = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \quad (131)$$

$$G = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \quad (132)$$

$$H = [1 \quad 0] \quad (133)$$

onde segue-se a convenção adotada na fundamentação teórica. A entrada e saída do sistema são definidas como:

$$u = [\omega_k] \quad (134)$$

$$z = [\theta_k] \quad (135)$$

A entrada em função do tempo ω_k é definida pelos valores fornecidos pelo giroscópio, ω_y para o *pitch* e ω_x para o *roll*. A medição z do ângulo é advinda de alguma outro sensor, como um acelerômetro ou magnetômetro. No presente trabalho, utiliza-se os dados do acelerômetro para estimativa dos ângulos de *roll* e *pitch*.

Como exposto anteriormente, o filtro de Kalman pode prosseguir no seu ciclo caso a medição não esteja disponível ou seja considerada inválida. O acelerômetro, em geral, detectará a ação de outras forças que não a gravidade, o que tenderá a prejudicar a medição. Portanto, pode-se definir uma medição como válida apenas se valor da aceleração escalar estiver dentro do intervalo $g[1 - erro_{max}, 1 + erro_{max}]$.

Os ruídos do processo Q e da medição R são definidos como:

$$Q = \Delta t^2 \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix} \quad (136)$$

$$R = [\sigma_w^2] \quad (137)$$

A metodologia descrita acima para o filtro de Kalman pode ser aplicada para cada eixo individualmente. O viés tende a ser a principal fonte de erro para os giroscópios do tipo MEMS, portanto a possibilidade de calcular o viés de maneira dinâmica deve ser vista como uma importante vantagem.

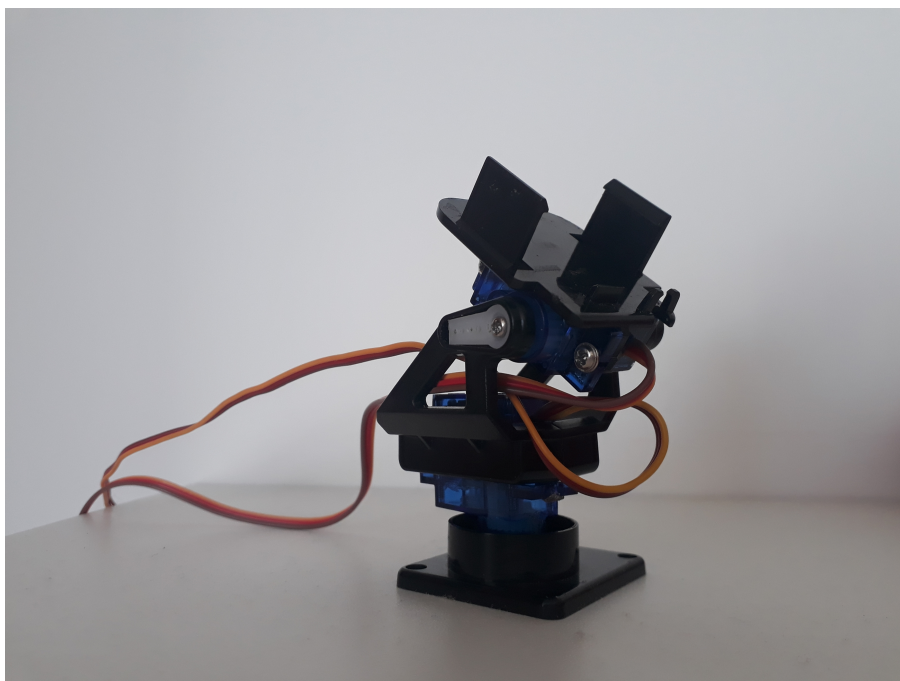
Por outro lado, é importante enfatizar que a rotação no espaço não é identicamente igual à mera integração dos valores do giroscópio no tempo, portanto o próprio modelo incorre em erros teóricos. Contudo, dada a presença de uma medição auxiliar no filtro, esses erros podem ser julgados adequados. A solução descrita acima é comum na literatura, dada sua simplicidade e facilidade de implementação pela utilização de matrizes 2 x 2. Uma solução mais exata deve-se valer do uso do EKF.

3.6 SUPORTE PARA OS TESTES

De maneira a validar os valores fornecidos pelo algoritmo de orientação, utilizou-se uma estrutura móvel controlado por servos motores, fornecendo 2 graus de liberdade. O suporte utilizado foi do tipo *pan/tilt*, usado para câmeras fotográficas e facilmente encontrado na internet. O suporte é mostrado na figura 15.

Deve-se ressaltar que na área de fotografia, o termo *pan* é frequentemente usado para referir-se ao ângulo de *yaw* enquanto o termo *tilt* refere-se ao ângulo de *pitch*. Dessa maneira, para a presente aplicação, é mais correto dizer que o suporte é do tipo *pitch/yaw*.

Figura 15 - Suporte *Pan/Tilt*



Fonte: o autor (2019)

O sensor foi montado na parte superior, com o eixo z apontando para baixo e o eixo y na direção oposta ao sentido de rotação do servo de *pitch*, de maneira que a rotação deve produzir resultados negativos para esse ângulo ou “nariz para baixo”.

O servo utilizado foi um Micro Servo 9g SG90 TowerPro. Esse servo possui uma massa de 9g, tensão de operação de 3 a 7,2 V, fornecendo um torque máximo de 1,2 kg cm a 4.8 V e alcançando um ângulo máximo de 180 graus. Ambos os servos foram conectados a um segundo microcontrolador Arduino.

O servo possui três fios; o fio vermelho deve ser conectando a fonte do Arduino, o fio marrom ao terra (GND) e o fio laranja a alguma entrada de controle da placa. Essa entrada deve ser do tipo PWM (modulação de largura de pulso). O controle pelo Arduino é realizado a partir da biblioteca *Servo.h*, que permite mover o servo para um ângulo específico a partir da função *servo.write(ângulo)*

4 RESULTADOS

Essa seção descreve as várias formas pelas quais os sensores e algoritmos foram testados e os resultados obtidos.

4.1 TESTE DO ALGORITMO

De maneira a validar o algoritmo dos quatérnions para a rotação, buscou-se testá-lo numericamente simulando uma entrada conhecida e analisando o resultado da integração. Em particular, uma entrada atrativa para o teste é uma entrada para a rotação em uma dimensão $\theta(t)$ do tipo senoidal, tal que:

$$\theta(t) = \sin(\omega t) \quad (138)$$

, que seria lida pelo giroscópio como:

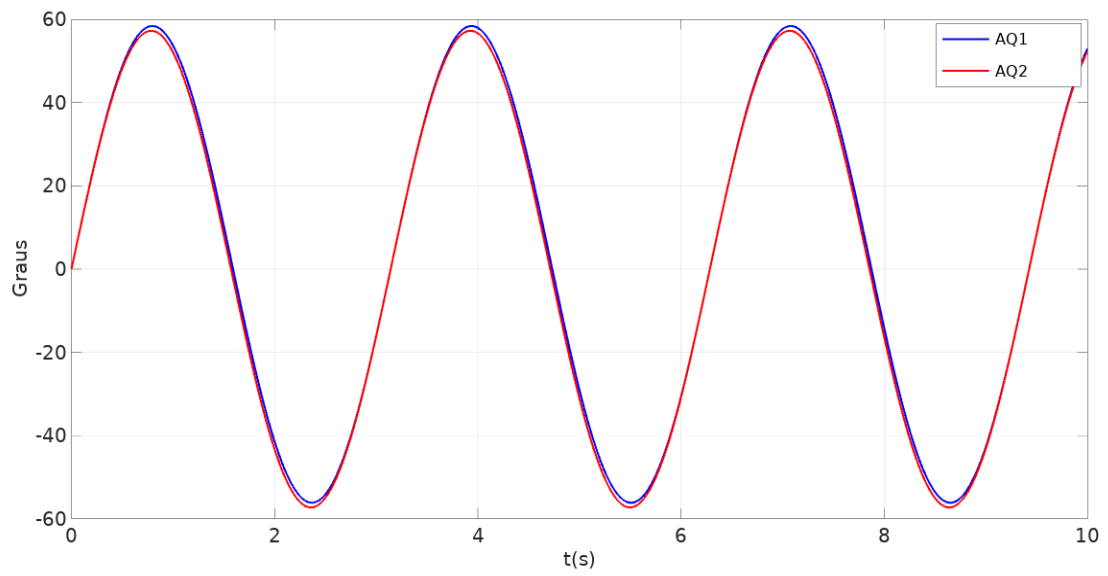
$$\Omega = \frac{d\theta}{dt} = \omega \cos(\omega t) \quad (139)$$

Além do próprio desempenho do algoritmo, a simulação permitiria obter uma ideia rudimentar da influência da frequência de amostragem no resultado. Estabelecendo a frequência como 50 Hz e o valor de $\omega = 2$, obtém-se o gráfico 1 para o ângulo e o gráfico 2 para a orientação.

O algoritmo dos quatérnions foi testado de duas formas: AQ1 e AQ2, como pode ser visto nos gráficos. Em geral, o ângulo no k-ésimo tempo é calculada a partir do ângulo no tempo k-1, utilizando-se uma velocidade angular constante. Essa velocidade pode ser assumida como a velocidade em k-1 (AQ1) ou como a média da velocidade em k-1 e k (AQ2).

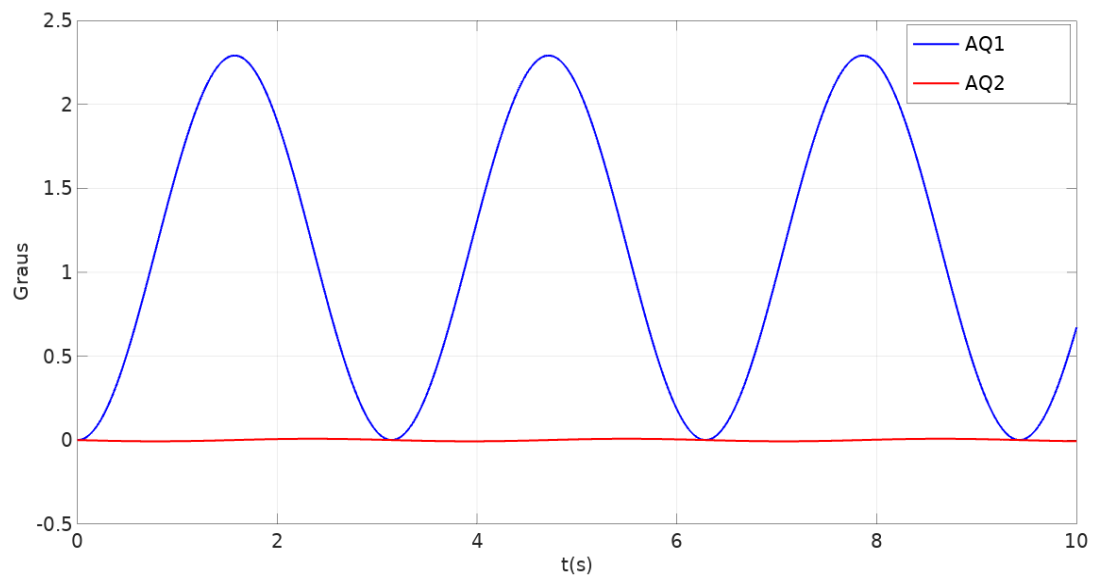
Como é razoável esperar, o algoritmo AQ2 possui maior exatidão: o erro máximo é de 0,04°, frente 2,29° de AQ1. Embora não mostrado, simulações posteriores revelam que o erro diminui com a frequência de amostragem. Também é importante observar que o erro do algoritmo não diverge com o tempo, isso é, oscila entre um valor máximo e mínimo.

Gráfico 1 - Ângulo em função do tempo



Fonte: o autor (2019)

Gráfico 2 - Erro em função do tempo



Fonte: o autor (2019)

4.2 CALIBRAÇÃO

Os sensores foram inicialmente testados para análise de seu comportamento. O gráfico 3 mostra a amostragem dos dados dos giroscópios por um período de aproximadamente 100 s, a 117,6 Hz.



Com o sensor em repouso, o valor médio das indicações corresponde ao viés presente, que é mostrado na tabela a seguir.

Tabela 6 - Vieses dos eixos dos giroscópio

Eixo	Viés (°/s)
X	-2,453
Y	-0,720
Z	1,263

Fonte: o autor (2019)

De maneira semelhante, a variância das medições está associado ao ruído presente e é um parâmetro de interesse. Os valores encontrados são mostrados na tabela seguinte.

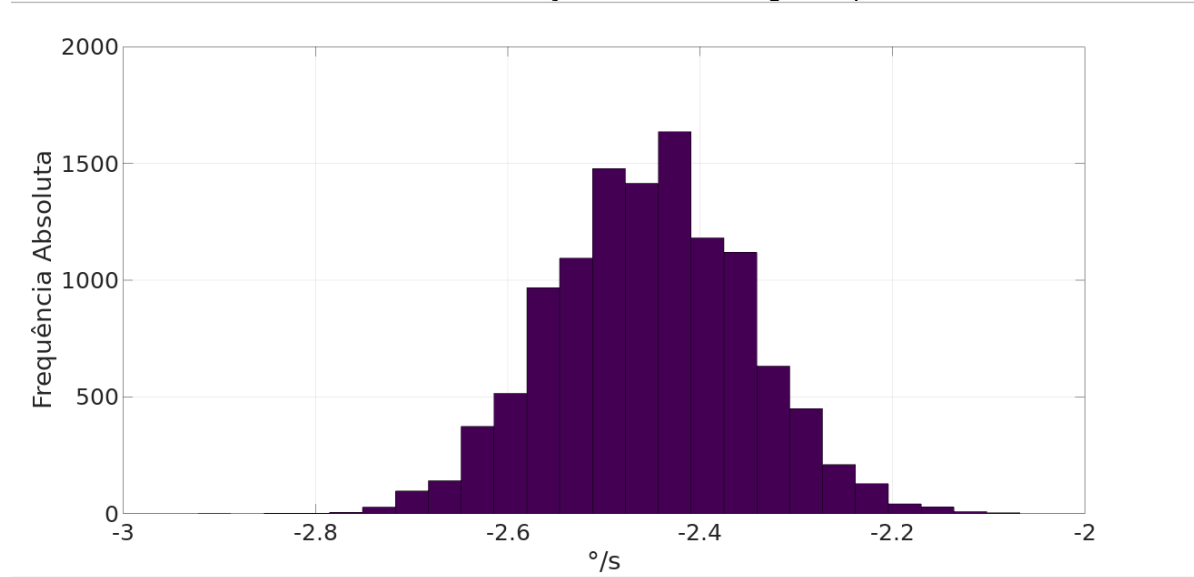
Tabela 7 - Desvio padrão dos eixos do giroscópio

Eixo	Desvio padrão (°/s)
X	0,101
Y	0,108
Z	0,085

Fonte: o autor (2019)

A natureza gaussiana do ruído é evidenciado pela curva de sino do histograma a seguir, obtido das medições no eixo X.

Gráfico 4 - Distribuição dos ruídos do giroscópio

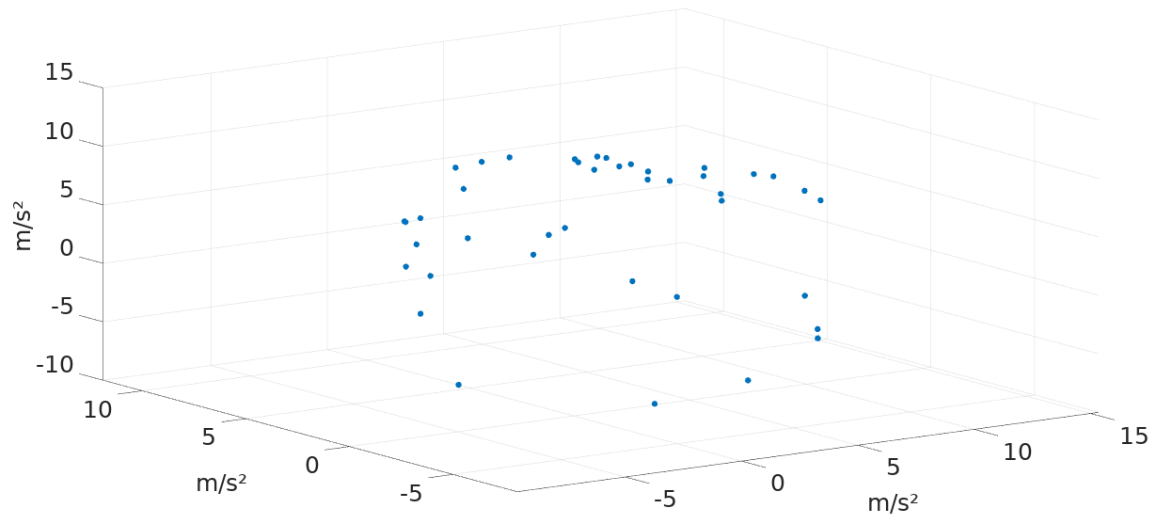


Fonte: o autor (2019)

Para a calibração do acelerômetro, o método dos mínimos quadrados foi aplicado utilizando os vieses e os ganhos como variáveis a serem determinadas, totalizando 6 parâmetros. Foram tomadas 40 medições do acelerômetro, com amostragem de 60 s cada. O valor médio é plotado no gráfico 5 a seguir. O gráfico 6 indica as medições do acelerômetro quando deixado em repouso, próximo a horizontal.

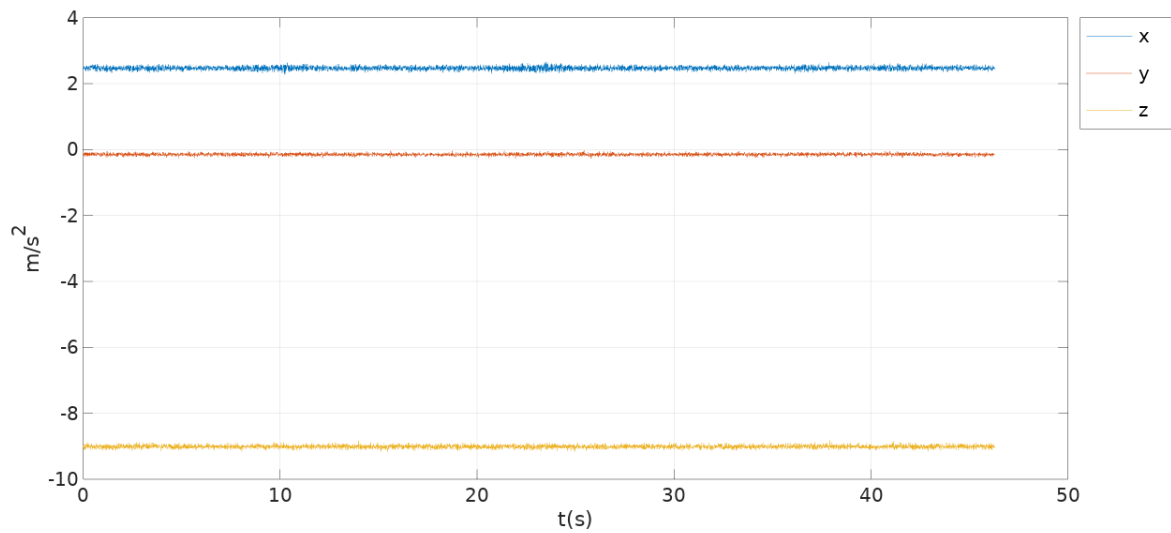
Para o cálculo da gravidade local, o modelo utilizado foi a Fórmula Internacional da Gravidade (IGF 1980), que considera a latitude e altitude do local. O local de experimentação possui latitude de $-9,58^{\circ}$ (S $9^{\circ} 35'$) e altitude de 100 m, resultando num valor de $9,78145 \text{ m/s}^2$ (SensorsONE, 2019).

Gráfico 5 - Indicações médias do acelerômetro em diversas orientações



Fonte: o autor (2019)

Gráfico 6 - Indicação do acelerômetro em repouso



Fonte: o autor (2019)

Os valores dos vieses e dos ganhos obtidos são mostrados nas tabelas.

Tabela 8 - Vieses dos eixos do acelerômetro

Eixo	Viés (m/s ²)
X	1,977
Y	0,029
Z	0,825

Fonte: o autor (2019)

Tabela 9 - Ganho dos eixos do acelerômetro

Eixo	Ganho
X	1,0033
Y	1,0038
Z	0,9835

Fonte: o autor (2019)

Em particular, é interessante destacar a convergência do algoritmo; usando o critério de parada de 10^{-6} como valor do passo, apenas 4 iterações foram necessárias.

O desvio padrão para os três eixos do acelerômetro, correspondente ao gráfico 6 é mostrado na tabela a seguir.

Tabela 10 - Desvio padrão dos eixos do acelerômetro

Eixo	Desvio padrão (m/s ²)
X	0,055
Y	0,035
Z	0,048

Fonte: o autor (2019)

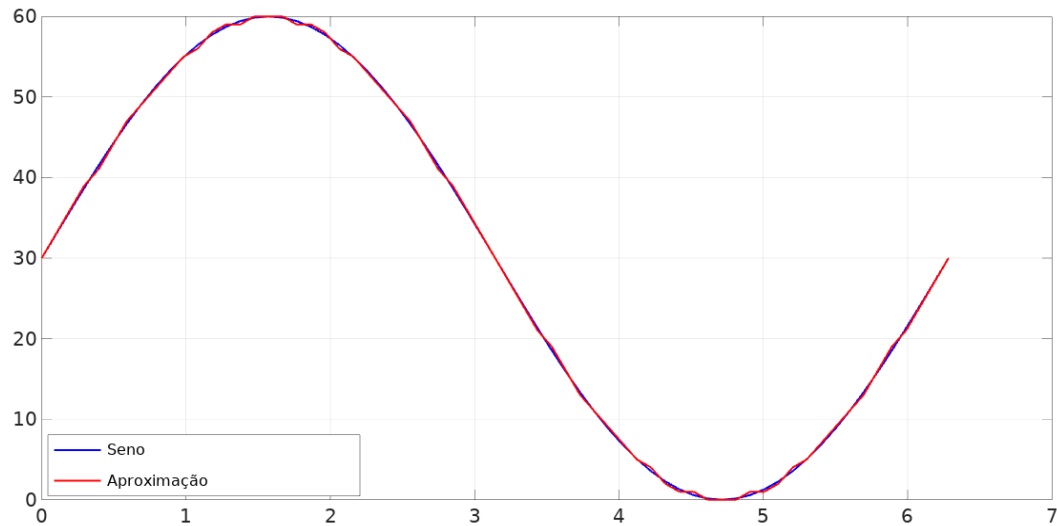
4.3 ROTAÇÃO UNIDIMENSIONAL

Os códigos foram testados inicialmente para a rotação em apenas 1 dimensão, utilizando uma entrada aproximadamente senoidal para o servo de *pitch*, escrita em C como um vetor de 32 componentes:

```
Int seno60_32[]={0,1,2,5,9,13,19,24,30,36,41,47,51,55,58,59,60,59,58,55,51,
47,41,36,30, 24,19,13,9,5,2,1}
```

A aproximação acima é comparada com uma função senoidal de mesma amplitude e frequência no gráfico 7.

Gráfico 7 - Aproximação da função senoidal

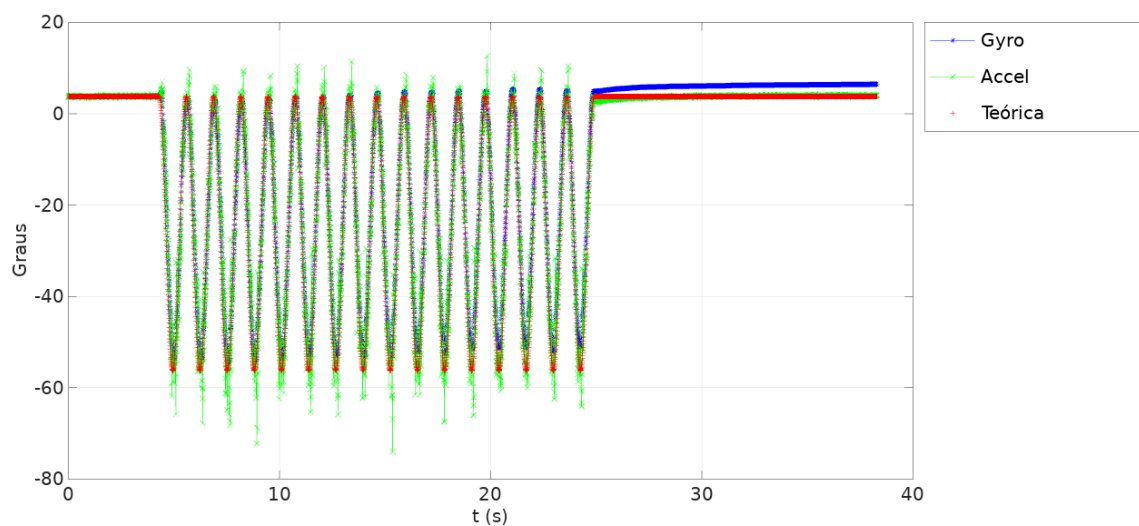


Fonte: o autor (2019)

A senoide oscila entre 0 e 60 graus de amplitude e é interrompida ao final de um número de ciclos arbitrário. As indicações dos giroscópios e dos acelerômetros são mostradas no gráfico 8, bem como uma curva senoidal teórica (em vermelho) associada ao movimento periódico do servo. O gráfico 9 fornece uma ampliação das indicações em um intervalo de tempo menor.

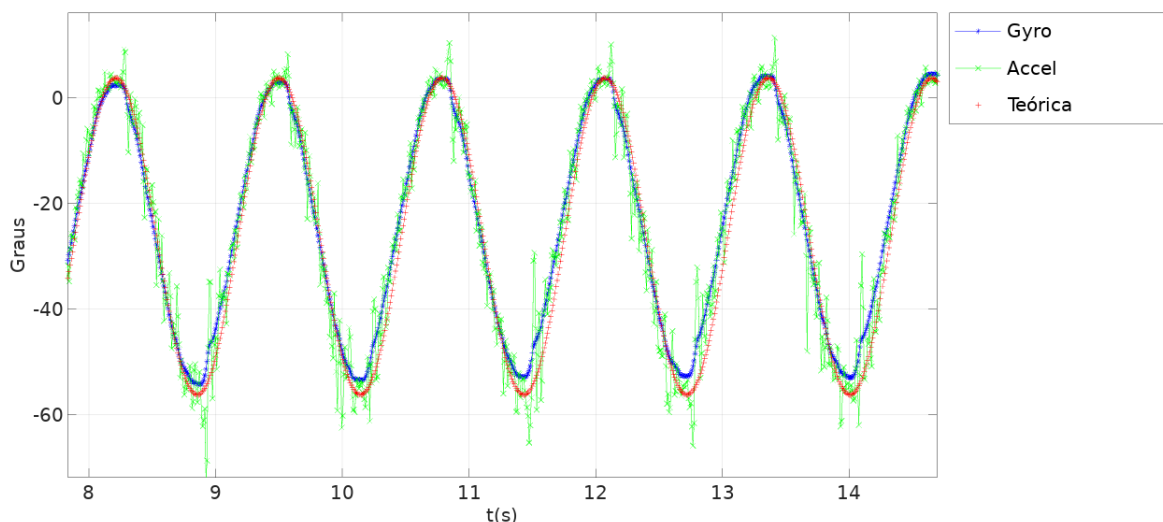
Os dados foram obtidos a 57.600 bauds e filtrados com um filtro passa-baixo de primeira ordem com $\alpha=0,5$ para o acelerômetro e $\alpha=0,6$ para o giroscópio.

Gráfico 8 - Indicações do ângulo de pitch pelo giroscópio e acelerômetro



Fonte: o autor

Gráfico 9 - Zoom do gráfico 8



Fonte: o autor (2019)

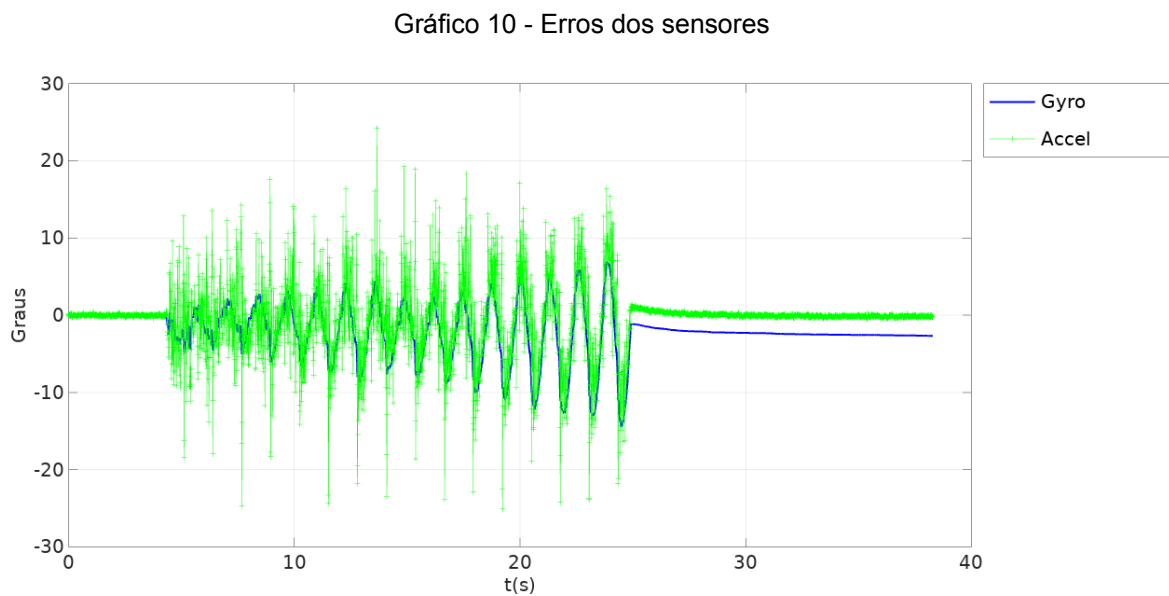
Analisando-se aspecto qualitativo dos gráficos acima, é possível notar uma elevada concordância entre os valores do giroscópio e do acelerômetro, ao menos na parte intermediária das oscilações. É nítido que as medições do acelerômetro são mais ruidosas que a integração correspondente do giroscópio. Em particular, os dados do acelerômetro são significativamente distorcidos ao se atingir os valores mínimos e máximos do ciclo. Isso se deve ao próprio aspecto do movimento do servo, advindo das acelerações bruscas e da vibração do conjunto. Muito possivelmente, caso o movimento fosse mais “contínuo”, o acelerômetro se aproximaria ainda mais do giroscópio.

Igualmente, é possível identificar que quando o conjunto é trazido ao repouso novamente, as indicações do giroscópio diferem das do acelerômetro por um valor próximo a 3 graus. Esse erro, como era de se esperar, pode ser atribuído a um viés que foi integrado. Deve-se notar que o ângulo inicial de *pitch* não é exatamente nulo, embora seja muito próximo disso. A orientação inicial é calculada pelas leituras iniciais do acelerômetro durante 1 segundo.

Embora a curva teórica do servo sirva como referencial para análise, problemas nítidos de exatidão do servo foram observados ao longo de toda sua faixa de operação. Por exemplo, ao se utilizar a função `servo.write()` para escrever um dado ângulo, o valor mostrado pelos sensores era significativamente diferente, geralmente por um valor próximo de 2 a 3 graus. Ainda que menos significativos, problemas também foram encontrados na repetitividade do ângulo.

Consultando o *datasheet* da fabricante, o autor não conseguiu encontrar nenhum valor específico para a exatidão do servo (Imperial College London, 2019). Pela experiência, o autor acredita que tais erros são inerentes aos servos 9g, causados por eventuais folgas nas engrenagens e defeitos de fabricação. De qualquer forma, optou-se por continuar utilizando o conjunto, visto que qualitativamente as entradas são bem próximas de uma onda senoidal e fornecem uma boa referência para os resultados.

O gráfico 10 mostra o erro das indicações do acelerômetro e do giroscópio em relação a curva teórica. Novamente, erros inerentes ao servo limitam a representatividade das curvas.



Fonte: o autor (2019)

O gráfico 11 apresenta as indicações dos filtros de Kalman e o filtro complementar para o mesmo caso, além da curva teórica já descrita. Para o filtro complementar, usou-se $\tau=3$. Os parâmetros usados para o filtro de Kalman foram estimados a partir de tentativa e erro, obtendo-se:

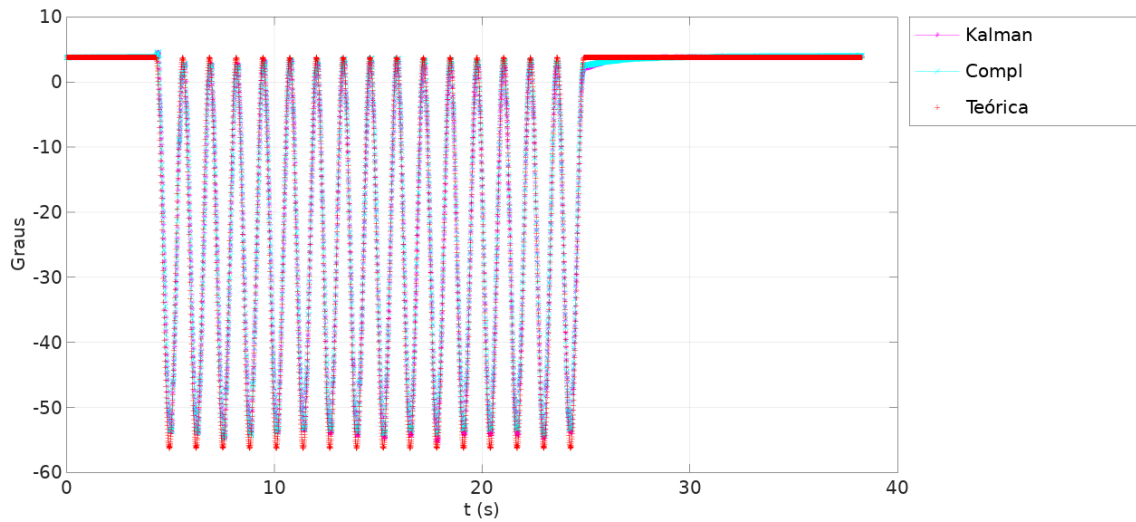
Tabela 11 - Parâmetros do filtro de Kalman

Parâmetro	Valor	Unidade
σ_θ	2,0	°
σ_b	0,5	°/s
σ_ω	2,0	°
$erro_{max}$	5,0	%

Fonte: o autor (2019)

Onde o $erro_{max}$ determina a diferença máxima permitida entre o módulo da aceleração e a magnitude da gravidade para que a leitura seja considerada válida.

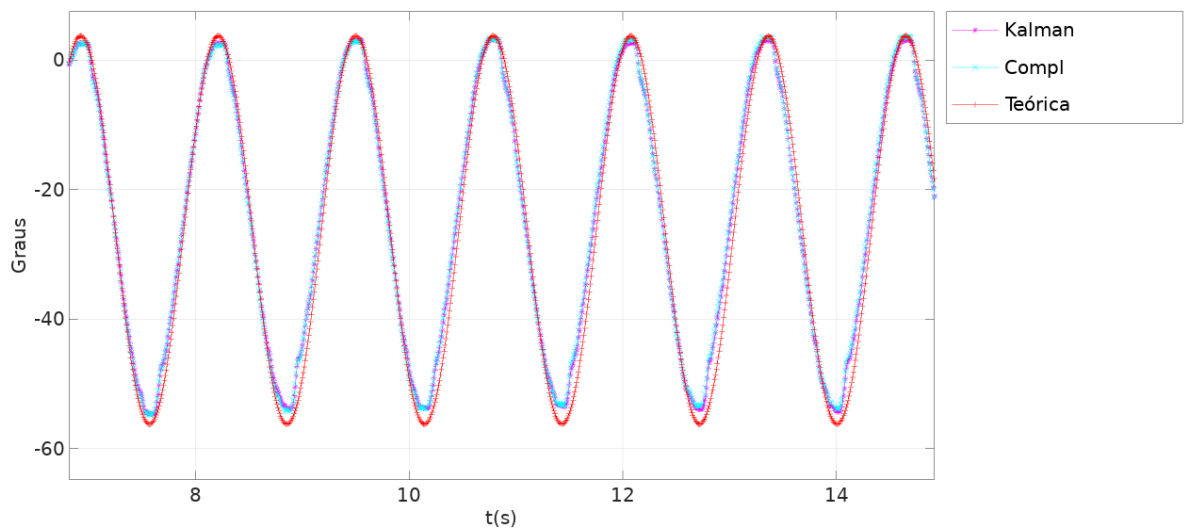
Gráfico 11 - Indicações do filtro de Kalman e filtro complementar



Fonte: o autor (2019)

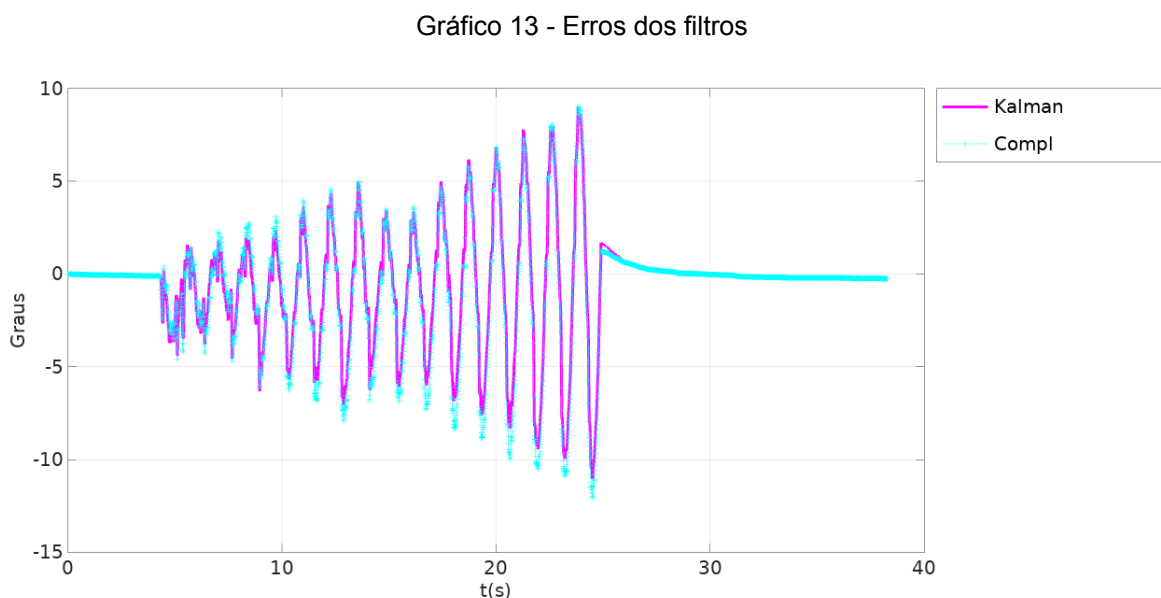
O gráfico 12 fornece uma visão do gráfico acima por um intervalo de tempo menor.

Gráfico 12 - Zoom do gráfico 11



Fonte: o autor (2019)

Os filtros tiveram um comportamento bem semelhante e foram capazes de convergir para a leitura do acelerômetro após o fim do movimento, além de apresentarem um aspecto mais suave característico do giroscópio. O gráfico 13 mostra o erros dos desvios em relação a curva teórica. Em particular, torna-se nítido que os erros foram muito próximos.



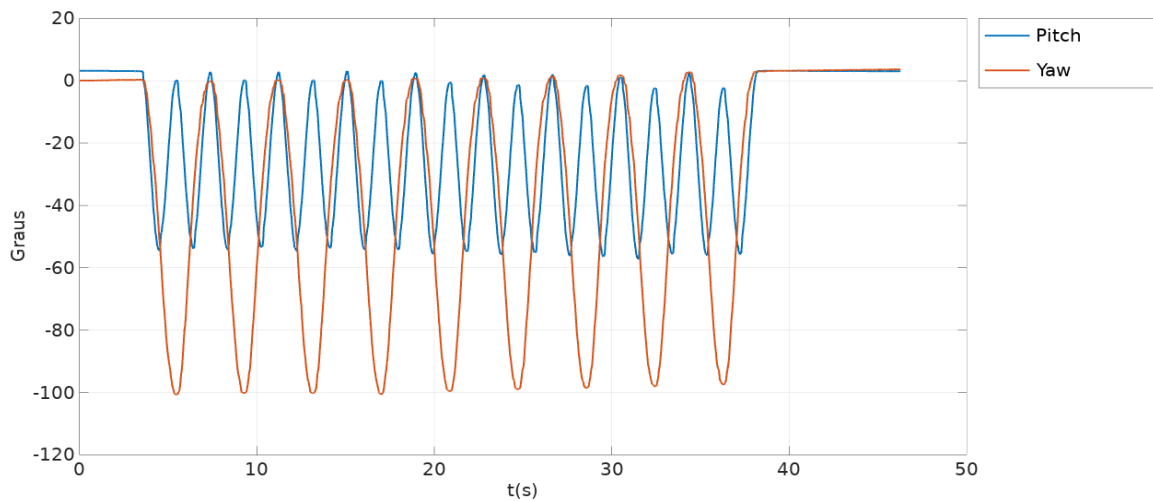
Fonte: o autor (2019)

4.4 ROTAÇÃO BIDIMENSIONAL

O conjunto também foi simulado usando movimento simultâneo nos dois eixos. Optou-se por fornecer aos dois servos 2 entradas aproximadamente senoidais – como no caso anterior, mas com amplitudes e frequências distintas. Isso foi possível usando *arrays* com comprimentos diferentes, percorrendo-as em um laço *while*.

A amplitude do ângulo de *yaw* foi definida como 100 graus e a do *pitch* permaneceu em 60 graus. Adicionalmente, a frequência do *pitch* foi escolhida como o dobro da de *yaw*. Prosseguindo de maneira similar ao caso anterior, o gráfico 14 indica as medições do giroscópio para os dois ângulos. Em particular, torna-se nítido a razão de frequências entre as funções de entrada.

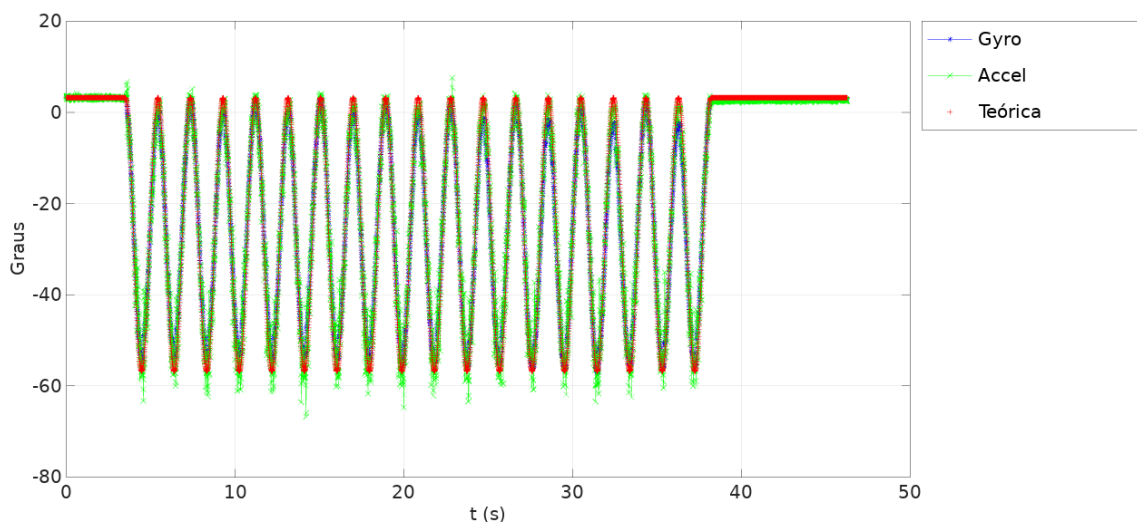
Gráfico 14 - Ângulos de pitch e yaw medidos pelo giroscópio



Fonte: o autor (2019)

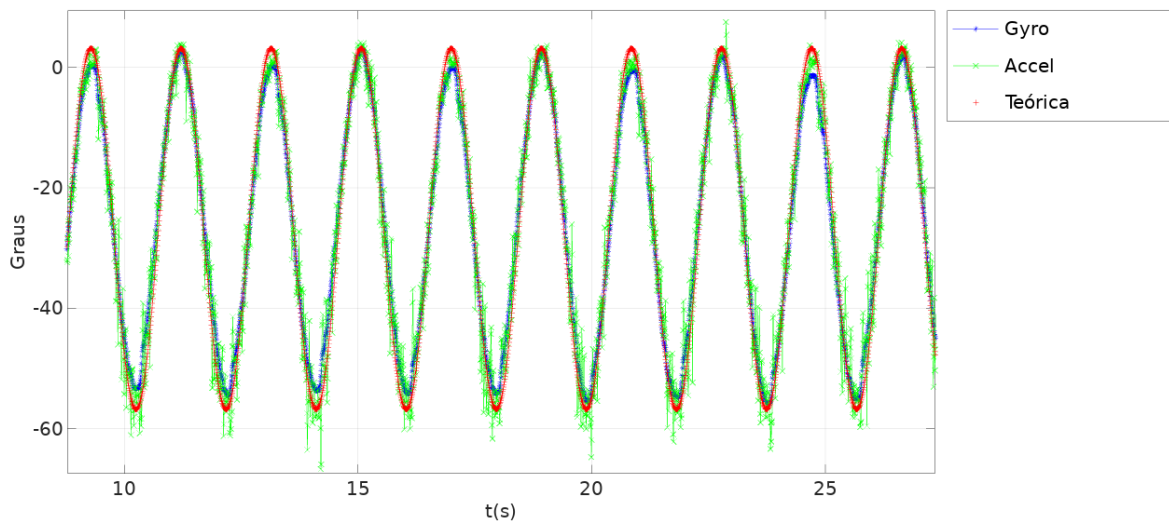
O gráfico 15 indica as medições do acelerômetro e do giroscópio, assim como a curva teórica. O comportamento é bem semelhante ao primeiro caso, mas dessa vez os dados do giroscópio foram mais próximos aos do acelerômetro e o ângulo final é praticamente igual. Como os dois testes foram realizados em um intervalo de tempo pequeno – questão de minutos – e os mesmos procedimentos foram tomados, o autor atribui esse comportamento a variações aleatórias do viés. O gráfico 16 fornece uma visão do gráfico 15 por um intervalo de tempo menor.

Gráfico 15 - Ângulo de pitch medido pelo acelerômetro e giroscópio



Fonte: o autor (2019)

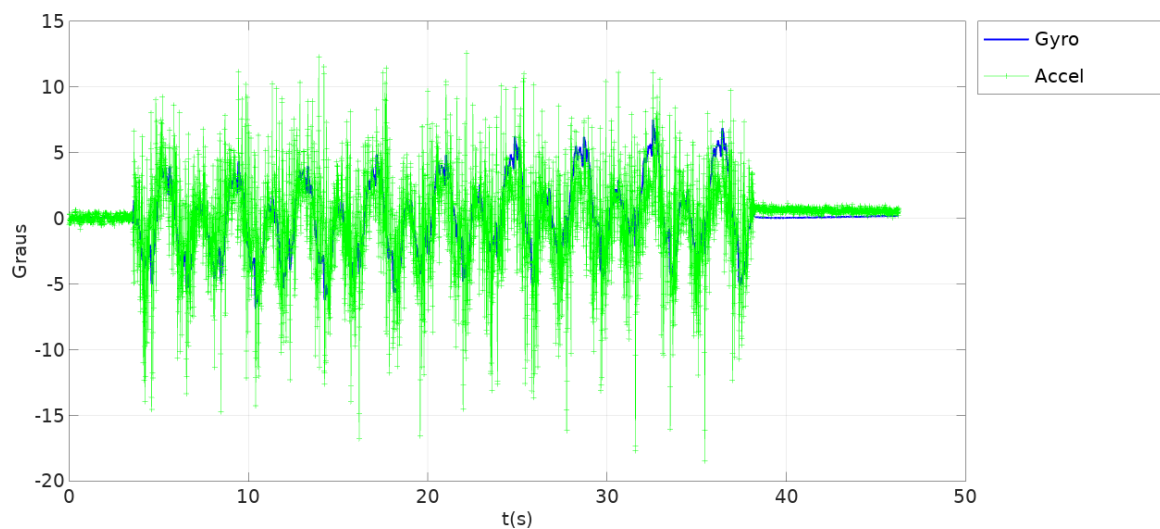
Gráfico 16 - Zoom do gráfico 15



Fonte: o autor (2019)

O gráfico 17 mostra, novamente, o erro do giroscópio e acelerômetro em relação à entrada teórica.

Gráfico 17 - Erros dos sensores na rotação bidimensional

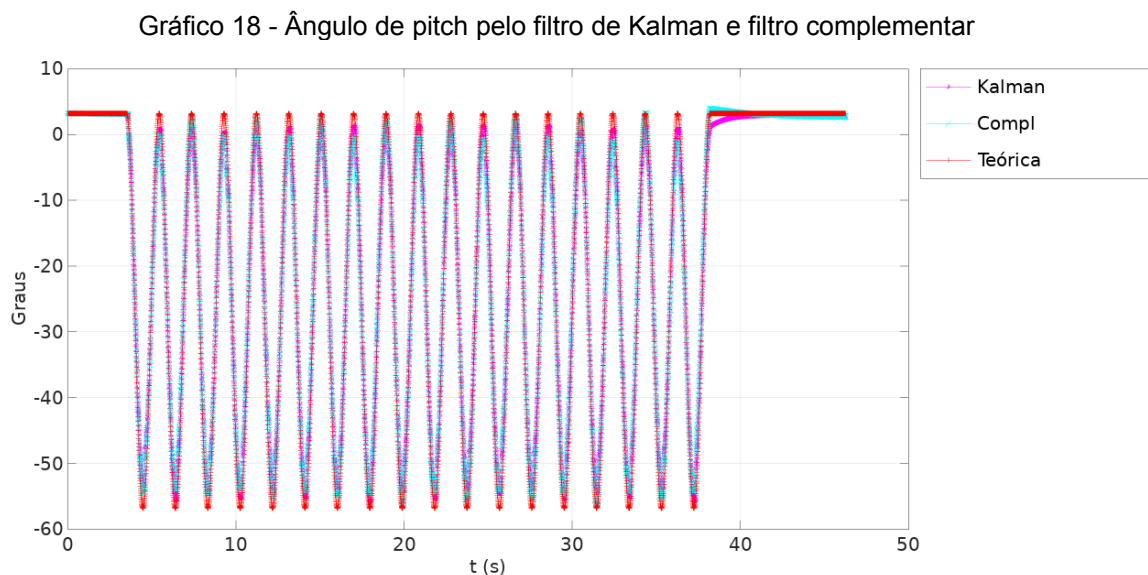


Fonte: o autor (2019)

O gráfico 18 apresenta os resultados do filtro de Kalman e do filtro complementar, comparados com a curva teórica em vermelho. O gráfico 19 amplia a figura para um menor número de ciclos. Os resultados, novamente, foram bem

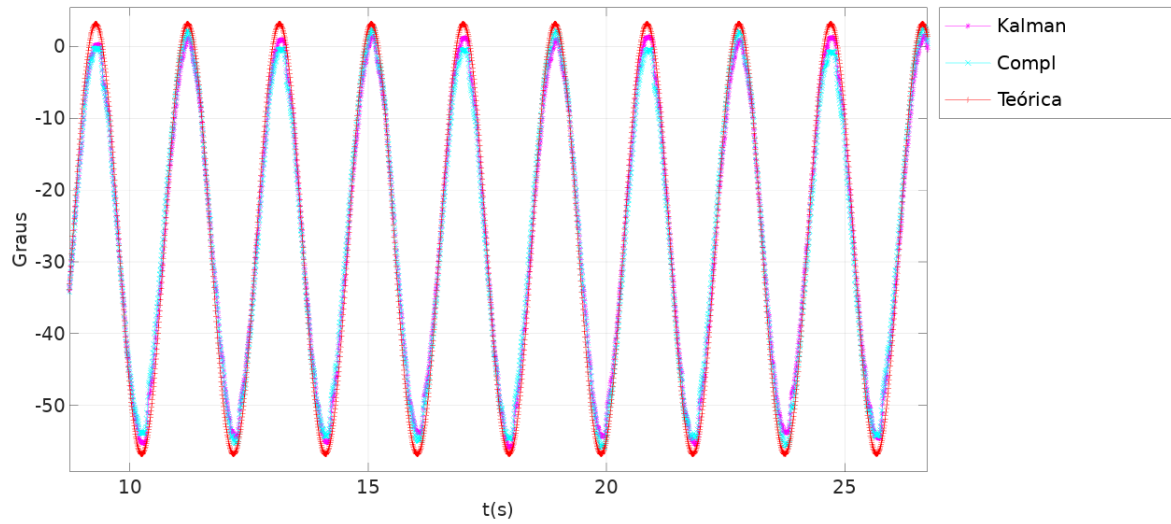
semelhantes. Em particular, o fato do filtro de Kalman se comportar de maneira próxima ao filtro complementar é um tanto surpreendente, visto que o modelo adotado parte de uma linearização da rotação. Esse resultado indica a robustez do filtro de Kalman e sugere que aplicações reais -possivelmente envolvendo rotações ao redor de um eixo principal ou pequenas oscilações em torno de uma dada orientação - seriam bem sucedidas.

Ainda assim, deve-se ressaltar que o modelo do filtro de Kalman é um modelo linear e a rotação foi realizada em apenas 2 eixos. A rotação ao redor de 3 eixos – equivalente ao caso mais geral – não foi testada por limitações de equipamento e tempo. Nesse último caso, é de se esperar que o desempenho do filtro de Kalman ,conforme aqui descrito, seja significativamente degradado.



Fonte: o autor (2019)

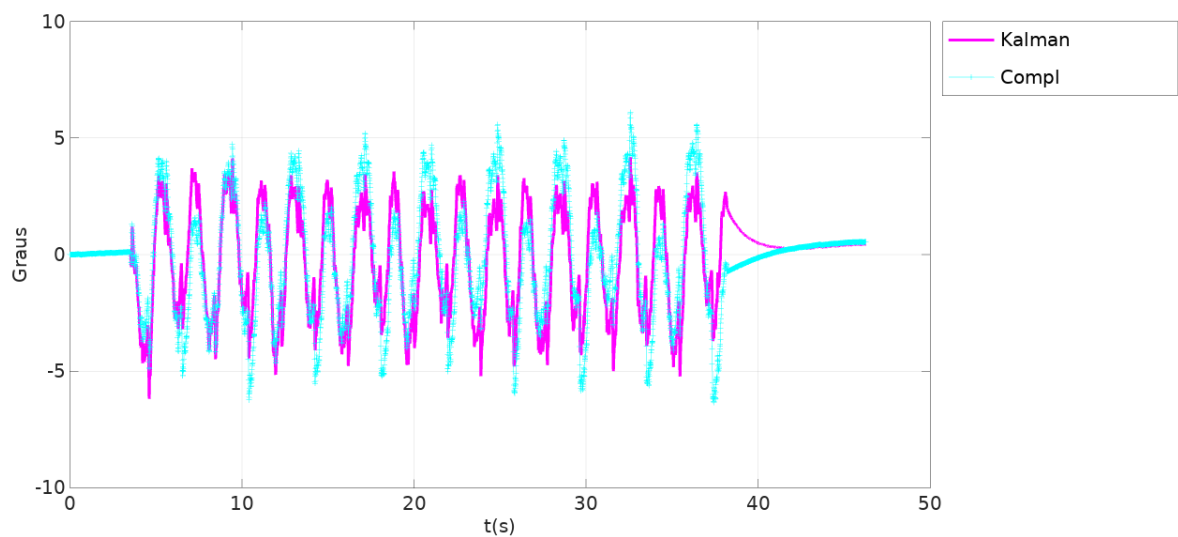
Gráfico 19 - Zoom do gráfico 18



Fonte: o autor (2019)

O gráfico 20 indica o erro teórico dos filtro de Kalman e complementar.

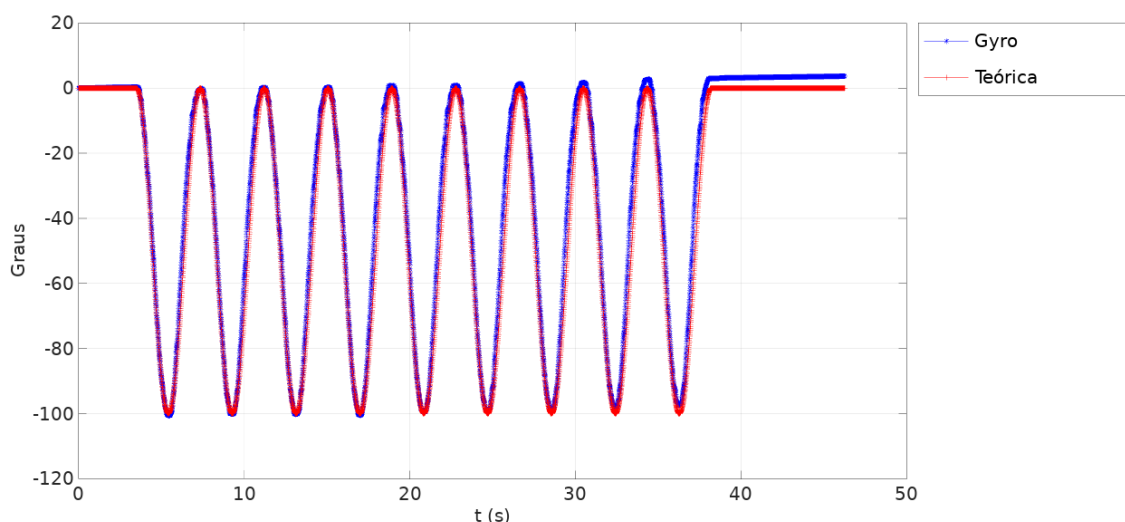
Gráfico 20 - Erros dos filtros na rotação bidimensional



Fonte: o autor (2019)

Por fim, o gráfico 21 fornece uma visão isolada do ângulo de yaw, medido apenas pelo giroscópio, bem como a curva teórica associada.

Gráfico 21 - Ângulo de yaw medido pelo giroscópio



Fonte: o autor (2019)

4.5 ALGORITMO INERCIAL

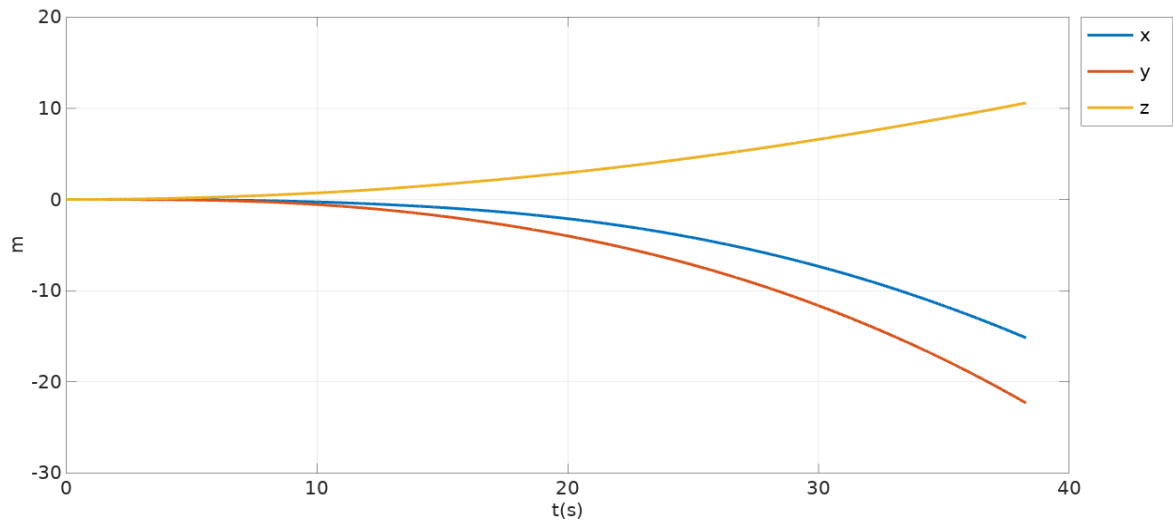
Depois dos testes com rotações, o código de navegação inercial foi testado em sua totalidade: os dados do acelerômetro foram utilizados para calcular a posição do corpo a partir da orientação fornecida. Da mesma maneira que trabalhos semelhantes, o MPU-6050 foi deixado em repouso e a posição resultante calculada. Nesse caso, os valores da posição e velocidade reais, fornecidos no referencial NED, são identicamente nulos e qualquer desvio desse comportamento corresponde ao erro.

Como já dito anteriormente, o algoritmo da posição depende do conhecimento da orientação do corpo – principalmente para a correção da gravidade - de maneira que erros na orientação devem influenciar a exatidão da posição. Reconhecendo-se esse conceito, optou-se por testar o algoritmo de 2 maneiras; no tratamento 1, a orientação obtida diretamente do giroscópio é usada, da mesma maneira que é feito nos sistemas de navegação inercial no estado da arte; e no tratamento 2, a orientação do filtro complementar é usada. O mesmo conjunto de dados foi utilizado para os dois tratamentos, de maneira a poder compará-los posteriormente.

Para o tratamento 1, o gráfico 22 mostra a posição do corpo nas coordenadas NED e o gráfico 23 a distância absoluta $d(t)$ - corresponde ao erro -, calculada como:

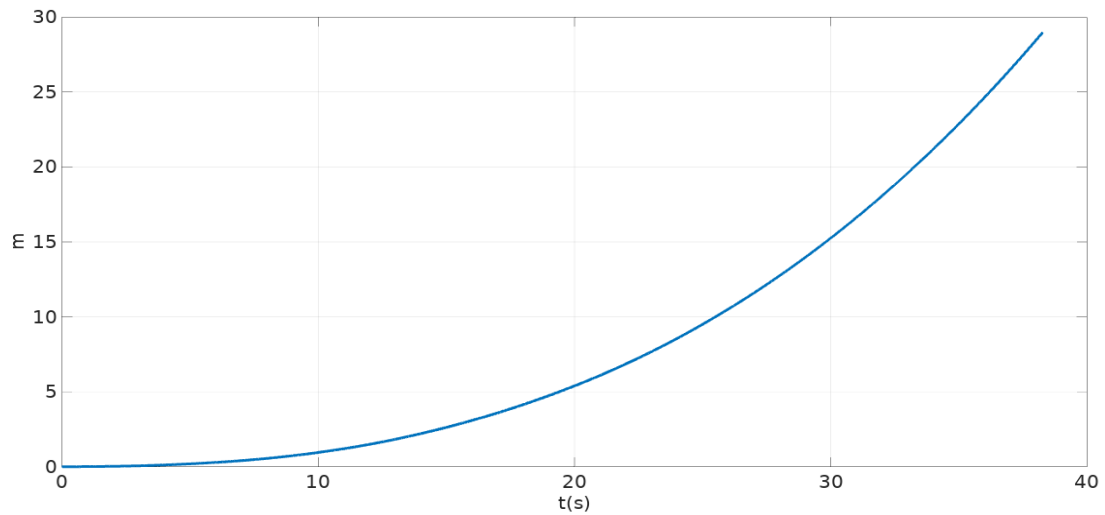
$$d(t) = \sqrt{x(t)^2 + y(t)^2 + z(t)^2} \quad (140)$$

Gráfico 22 - Posição no tratamento 1



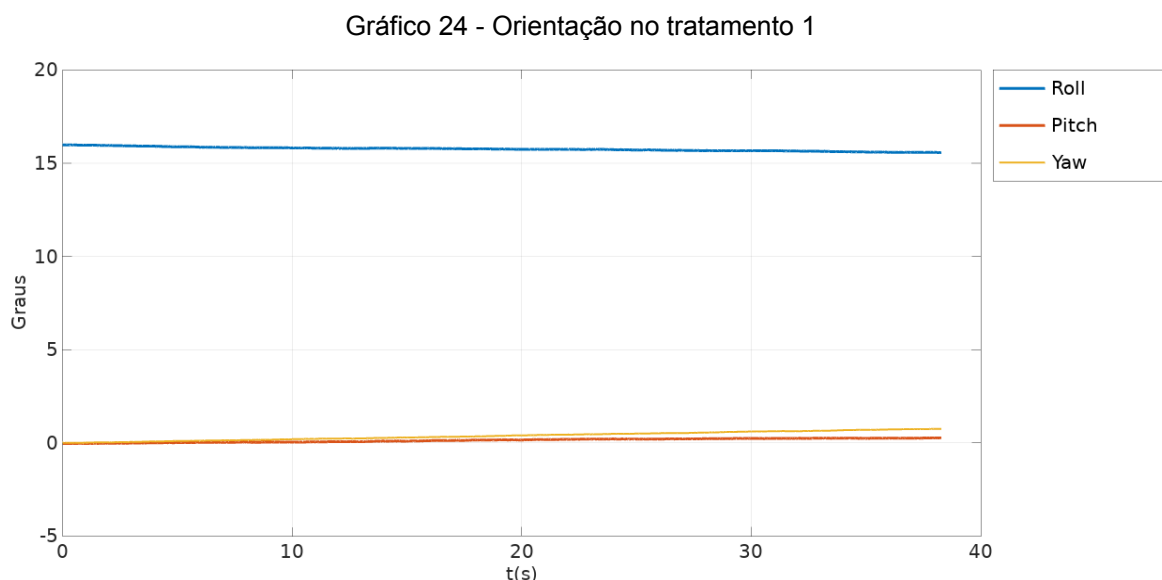
Fonte: o autor (2019)

Gráfico 23 - Distância absoluta no tratamento 1



Fonte: o autor (2019)

Para a distância absoluta, o erro chega a 15 m após 30 s. A orientação em função do tempo é mostrada no gráfico 24. Notar que o algoritmo interpreta que a orientação está mudando no tempo, principalmente no *pitch* e *yaw*, o que contribui para que os erros se acumulem na posição.



Fonte: o autor (2019)

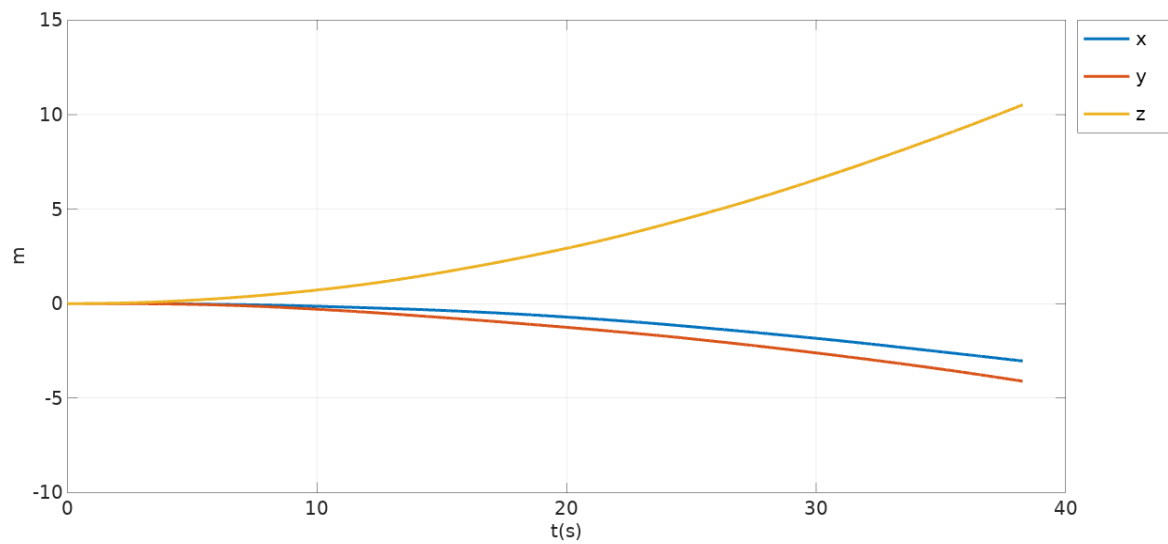
No tratamento 2, como já dito anteriormente, a orientação fornecida pelo filtro complementar é usada. O gráfico 25 descreve a posição do corpo, usando os mesmos dados do caso anterior, enquanto, de maneira semelhante, o gráfico 26 descreve a distância absoluta.

A melhora dos resultados é nítida; após 30 s, o erro é de 7,5 m, ou duas vezes inferior ao erro do tratamento 1. Embora os gráficos apresentem um comportamento não linear, de maneira que não é adequado atribuir um número definido para a exatidão, os resultados do tratamento 2 são significativamente superiores aos do tratamento 1.

O gráfico 27 mostra a velocidade estimada do corpo. O comportamento linear da velocidade indica um possível resíduo no viés calculado do acelerômetro. O gráfico 28 mostra o módulo da velocidade.

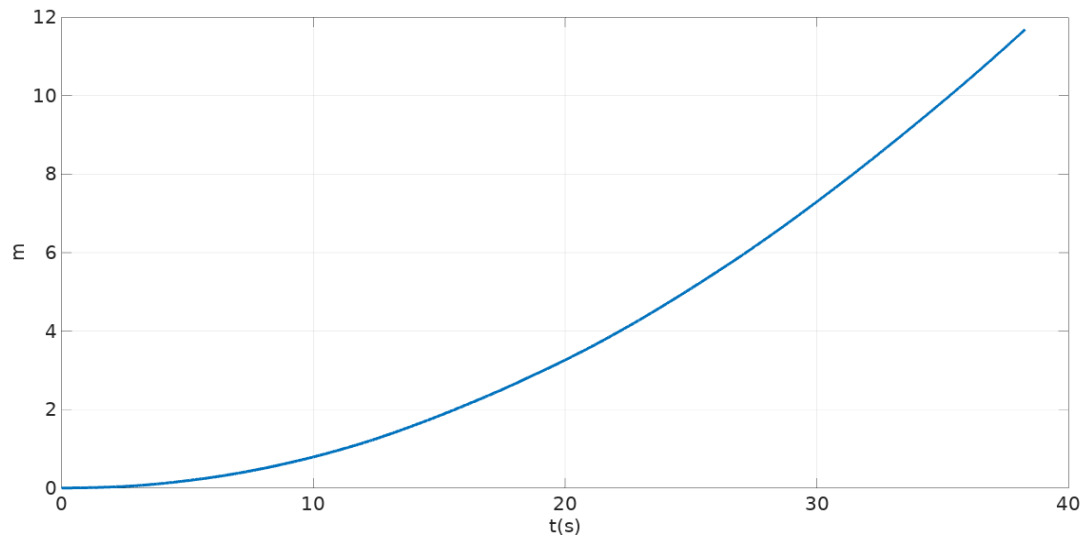
O gráfico 29 fornece a orientação do corpo. A principal melhoria trazida pela integração de sensores é a maior estabilidade dos valores de *pitch* e *roll*, que permanecem praticamente constantes, permitindo melhores resultados para o algoritmo inercial. Visto que o ângulo de *yaw* não possui uma medição auxiliar, seu valor não é compensado. Contudo, esse fato não é tão crítico nesse caso, visto que, como já apresentado, a rotação da gravidade para o referencial local não depende do *yaw*.

Gráfico 25 - Posição no tratamento 2



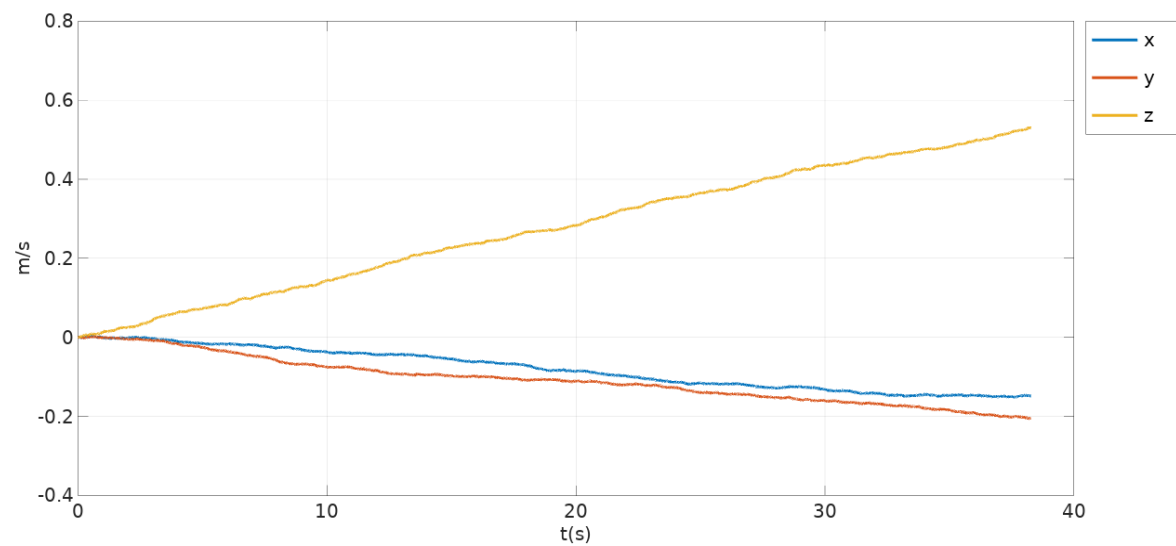
Fonte: Elaborado pelo autor (2019)

Gráfico 26 - Distância absoluta no tratamento 2



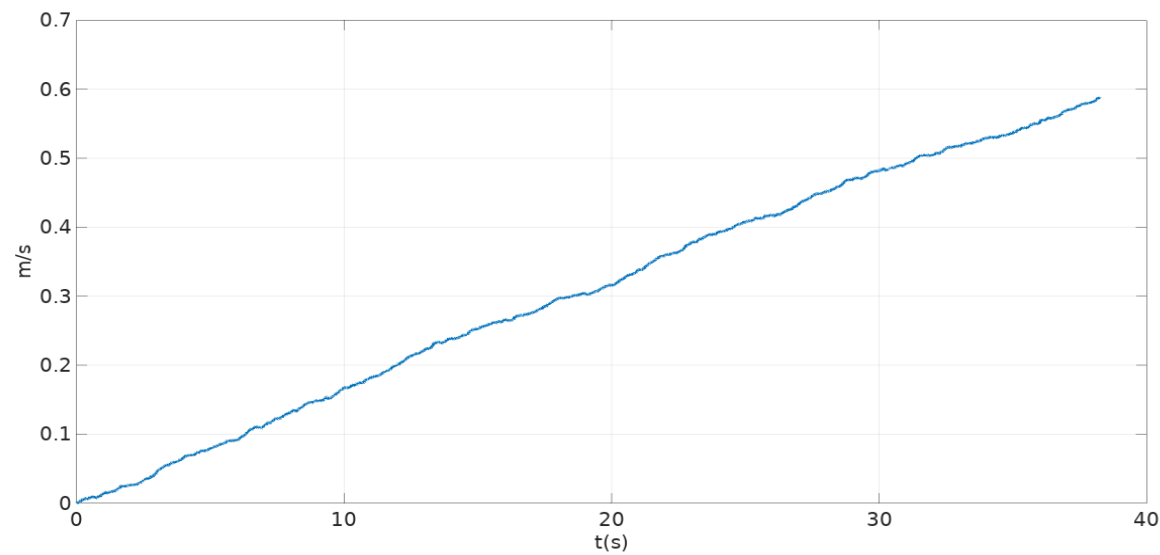
Fonte: o autor (2019)

Gráfico 27 - Velocidade no tratamento 2



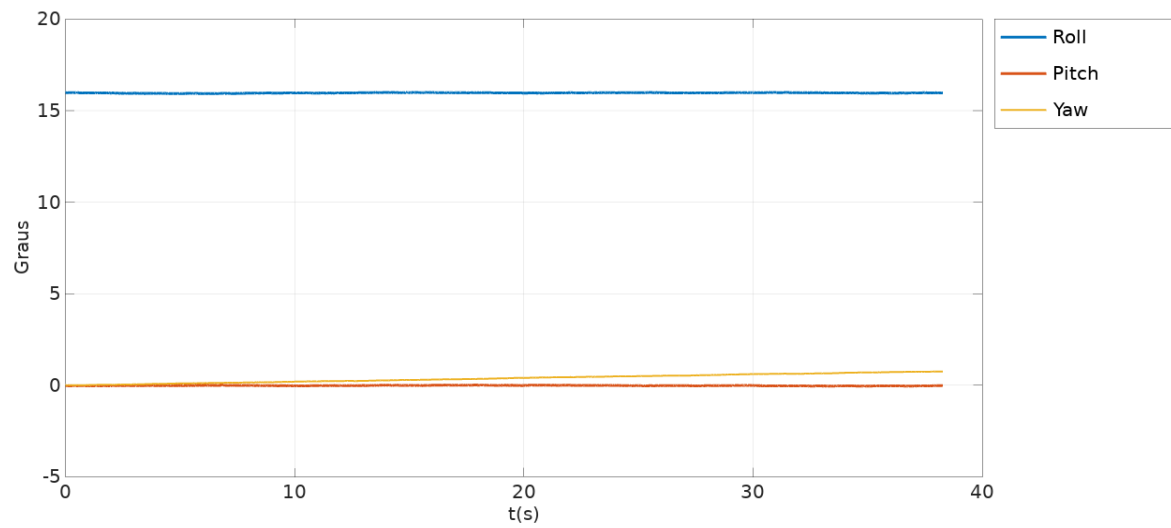
Fonte: o autor (2019)

Gráfico 28 - Velocidade absoluta no tratamento 2



Fonte: o autor

Gráfico 29 - Orientação no tratamento 2



Fonte: o autor (2019)

5 CONCLUSÃO

Os códigos escritos ao longo desse trabalho foram testados extensivamente e seus resultados demonstrados foram compatíveis com o esperado e com trabalhos semelhantes. Contudo, certas dificuldades foram encontradas que merecem ser mencionadas. Em particular, a diversidade de representações encontrada na literatura especializada é, *a priori*, um tanto confusa. De maneira semelhante, é interessante notar que muitas das fontes encontradas são referenciais que não se encaixam no perfil acadêmico, tais como institutos militares de pesquisa, agências aeroespaciais e fabricantes do ramo.

Como tornado evidente pela experiência do autor, o tema de navegação envolve considerações teóricas e práticas, muitas das quais só são trazidas à tona pelo próprio contato com os sensores. O autor acredita que, a partir do material disponível, a fundamentação e o desenvolvimento do tema foi apresentada de maneira clara e precisa. Igualmente, o código descrito e implementado mostrou-se coerente com o esperado, de maneira que capacidades adicionais podem ser futuramente acrescentadas.

Os programas foram implementados em uma linguagem de alto nível e os dados dos sensores foram processados de maneira *offline*, isso é, após serem todos armazenados. Para uma aplicação em tempo real em sistemas embarcados, os códigos e funções devem ser portados para uma outra linguagem propícia ao emprego em microcontroladores, tal como C ou C++. Embora o autor tenha tido contato com a placa Arduino para obtenção dos dados, sua opinião é que seria necessário uma placa com processador mais potente, tal como um STM32, que possui maior *clock* e maior largura de dados.

No geral, os resultados encontrados para a orientação foram julgados como ótimos. O viés é uma fonte predominante do erro em sensores MEMS e sua correção requer alguma fusão de sensores. Ou seja, os dados do giroscópio devem ser auxiliados por outra medição. A integração pelos filtro de Kalman e pelo filtro complementar foram descritas e apresentaram resultados praticamente idênticos.

Contudo, é preciso ressaltar a diferença entre essas duas técnicas: o filtro complementar é intuitivamente mais fácil de se entender, possui uma implementação menos complexa e considera a integração ao longo do tempo propícia dos dados do giroscópio. Entretanto, possui apenas um parâmetro para ser modificado. O filtro de

Kalman aqui implementado é linear, enquanto o problema em si é não linear, requer parâmetros estatísticos de difícil estimação e, no geral, é mais difícil de ser compreendido. Ainda assim, sua complexidade permite o desenvolvimento de modelos mais refinados. A notação em espaço de estados permite que o viés seja considerado explicitamente no modelo. Em outras palavras, é uma arma mais poderosa e geral, mas que requer o manuseio correto.

Por fim, o algoritmo para navegação inercial foi testado e os erros da posição com o sensor em repouso foram da ordem de uma dezena de metros após meio minuto. Os erros foram reduzidos com o melhor conhecimento da orientação do corpo. Embora é de se esperar que os erros sejam significativamente maiores caso o movimento seja complexo, os resultados provam a viabilidade de usar sensores inerciais MEMS com dados de GNSS, visto que esse último possui baixa taxa de atualização (próximo de 1 Hz), mas elevada exatidão. Nesse caso, os sensores MEMS poderiam fornecer a posição por um intervalo de tempo de alguns segundos, até que um eventual sinal do GNSS permitisse a estimativa da posição correta.

O desenvolvimento contínuo da tecnologia MEMS observado nas últimas décadas tende a produzir sensores cada vez mais capazes e acessíveis. Embora seja difícil realizar previsões, o progresso é notável e novas tecnologias como o MOEMS (Sistemas microopto-eletromecânicos), que combinam MEMS com a micro ótica, fornecem horizontes promissores.

Embora os sensores MEMS comercialmente disponíveis atualmente apresentem limitações, inerentes ao preço e à própria tecnologia disponíveis, bons resultados podem ser obtidos a partir de um desenvolvimento inteligente do algoritmo, que tanto permita se beneficiar de um conhecimento acerca da dinâmica do sistema, bem como integrar diferentes sensores para a obtenção de resultados mais exatos. De maneira geral, os erros advindos dos sensores MEMS podem ser limitados a partir do uso apropriado da capacidade computacional disponível, que atualmente é barata e abundante.

Como sugestões para trabalhos futuros, o autor propõe testar a maior integração de sensores e o desenvolvimento de algoritmos mais sofisticados que permitam essa fusão de sensores. Essa integração poderia ser realizada, por exemplo, usando um magnetômetro para o cálculo do ângulo de *yaw*, ou com algum sensor de reconhecimento de imagem, fornecendo uma solução completa para a orientação. Igualmente, um receptor GNSS poderia fornecer uma atualização

periódica da posição e, dado o baixo custo e a precisão, fornecer um bom desempenho ao sistema. O filtro de Kalman estendido (EKF) é a ferramenta padrão para a integração, mas modelos ainda mais refinados, como o *Unscented Kalman Filter* (UKF) prometem resultados ainda mais impressionantes.

Por último, é importante não perder de vista que a navegação não é um fim em si próprio. O conhecimento a respeito da orientação e posição é fundamental para veículos na execução de manobras e desempenho de sua maneira mais geral. Portanto, dominando-se a navegação, o autor imagina que os pontos seguintes seriam a integração com guiagem e o controle do sistema, de maneira a obter soluções mais amplas e complexas para situações reais.

REFERÊNCIAS

ANDERSON, J.D. **Introduction to Flight**. 5.ed. New York: McGraw-Hill, 2005.

BASTOS JR., G. G. **Integração de dados de navegação de um robô bípede**. 2008. 94 f. Dissertação (Mestrado em Engenharia Mecânica) - Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco, Recife, 2008.

BUREAU INTERNATIONAL DES POIDS ET MESURES. **The International System of Units (SI)**. Disponível em: <https://www.bipm.org/utis/common/pdf/si_brochure_8_en.pdf>. Acesso em: 30 ago. 2019.

CANNON, R. H. **Dynamics of Physical Systems**. 1.ed. McGraw-Hill: Nova Iorque, 1967.

EATON, J.W, et al. **GNU Octave Manual Version 5**. Disponível em: <<https://octave.org/octave.pdf>>. Acesso em: 15 jul. 2019.

ESKIN, M. **Design of an inertial navigation unit using MEMS sensors**. 2006. Dissertação (Mestrado em Engenharia Elétrica) - Cornell University, Ithaca, 2006.

FAIRFAX, L.D., FRESCONI, F.E., **Position Estimation for Projectiles Using Low-cost Sensors and Flight Dynamics**, Army Research Laboratory, Adelphi, 2012.

GREWAL, M.S, ANDREWS, A.P. **Kalman filtering: theory and practice using MATLAB**. 3.ed. Hoboken: John Wiley & Sons, 2008.

GREWAL, M. S., L. R. Weill, ANDREWS, A.P. **Global Positioning Systems, Inertial Navigation and Integration**. 2 ed. Nova Iorque: Wiley, 2007.

International Space Hall of Fame. Disponível em: < <http://www.nmspacemuseum.org/halloffame/detail.php?id=6>>. Acesso em: 16 mai. 2019.

HENDERSON, D. M., **Euler Angles, Quaternions and Transformation Matrices**, NASA Mission Planning and Analysis Division, Houston, 1977. Disponível em: <<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19770024290.pdf>> Acesso em: 8 dez. 2019.

HIGGINS JR, W.T. **A Comparison of Complementary and Kalman Filtering**. IEEE Transactions on Aerospace and Electronic Systems, v. 11, n. 3, p. 312-325, 1975.

HOAG, D.G. **Apollo Guidance and Navigation Vol 1**. Cambridge: MIT, 1965.

Imperial College London. **Servo Motor SG90 Datasheet**. Disponível em: <http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf> Acesso em: 23 out. 2019.

InvenSense Inc. **MPU-6000 and MPU-6050 Product Specification Revision 3.4**, Disponível em: <<https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>> Acesso em: 07 jun. 2019.

ISLAM, T., et al, **Comparison of complementary and Kalman filter based data fusion for attitude heading reference system**, AIP Conference Proceedings 1919, 020002, 2017.

KOKS, D., **Using Rotations to Build Aerospace Coordinate Systems**, Defense Science and Technology Organisation, Electronic Warfare and Radar Division, Edinburgo, 2008.

LIU, Z., **Introduction to Inertial Navigation and Pointing Control**, Northrop Grumman, San Diego, 2011. Disponível em: <http://www.zhang-liu.com/tutorials/Intro_Nav_Control.pdf?i=1> Acesso em: 27 mai. 2019.

KALMAN, R.E. **A new approach to linear filtering and prediction problems**. ASME – Journal of Basic Engineering, v.82, n.1, p. 35-45, 1960.

MCGEE, L.A., SCHMIDT, S.F., **Discovery of Kalman Filter as a Practical Tool for Aerospace and Industry**, AMES Research Center, 1985. Disponível em: <<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19860003843.pdf>>. Acesso em: 8 dez. 2019.

MERCADO LIVRE, **Acelerometro E Giroscopio 3 Eixos Mpu-6050**, Disponível em: <https://produto.mercadolivre.com.br/MLB-1149320947-15-x-acelerometro-e-giroscopio-3-eixos-mpu-6050-mpu6050-_JM?quantity=1> Acesso em: 30 mai. 2019.

NASA, **Saturn V Flight Manual SA 503**. 1968. Disponível em: <<https://history.nasa.gov/afj/ap08fj/pdf/sa503-flightmanual.pdf>>. Acesso em: 13 de dez. de 2019.

NISE, N.S. **Control system engineering**. 4.Ed. Jefferson City: John Willey & Sons, 2004.

NOVATEL INC. **IMU erros and their effects**. Disponível em: <<https://www.novatel.com/assets/Documents/Bulletins/APN064.pdf>> Acesso em: 29 mai. 2019.

PASSARO, V.M.N et al. **Gyroscope Technology and Applications: A Review in the Industrial Perspective**, Sensors (Basel). 2017 Oct; 17(10): 2284.

PETTERSSON, M. **Extended Kalman Filter for UAV Attitude Estimation**. 2015. Tese (Doutorado) - Departamento de Engenharia Elétrica, Universidade de Linköpings, Linköping, Suécia, 2015.

PRAVEENA, K., RAVIKUMAR, A. **Design of Inertial Navigation Systems using Kalman Filter**. International Journal of Engineering Inovation, v. 2, n.4, p 76-82, 2013.

ROMANIUK, S., GOSIEWSKI, Z. **Kalman filter realization for orientation and position estimation on dedicated processor**. Acta Mechanica et Automatica, Bialystok, v. 8, n. 2, p. 88-94, 2014.

REDOLF, J., GAYDOU, D.A., **Filtro Complementario para estimacion de actitud aplicado al controlador embebido de un cuatrirrotor**. National University of Technology, Congreso Argentino de Sistemas Embebidos, Buenos Aires, 2011.

SCHWEBER, B. **Gyroscopes, Part 2: Optical and MEMS implementations**, Analogic Tips, 2019 Disponível em: <<https://www.analogictips.com/gyroscopes-part-2-optical-and-mems-implementations-faq/>> Acesso em: 1 jul. 2019.

SensorsOne. **Local Gravity Calculator**. Disponível em: <<https://www.sensorsone.com/local-gravity-calculator/>> Acesso em: 14 out. 2019.

STOVALL, S.H. **Basic Inertial Navigation, Naval Air Warfare Center Division**, China Lake, 1997

TAYLOR, J.R. **Mecânica Clássica**. 1.ed. Porto Alegre: Bookman, 2013.

TITTERTON, D.H. WESTON, J.L. **Strapdown inertial navigation technology**. 2.ed. Stevenage: IEEE, 2004.

WALCKO, K.J, MASON, P.A.C. **Inertial Navigation**. University of Florida, 2002 Florida Conference on Recent Advances in Robotics.

WANG, Y. , et al. **Design and Implementation of an AHRS Based on MEMS Sensors and Complementary Filtering**. Advances in Mechanical Engineering [s. l.] , v. 2014, n.1, p. 1-11, Fev. 2015. Disponível em: <<https://journals.sagepub.com/doi/10.1155/2014/214726>>. Acesso em: 8 dez. 2019.

WETZSTEIN, G., **Inertial Measuring Units IEE 267 Virtual Reality**, Stanford University, 2019. Disponível em: <<https://stanford.edu/class/ee267/lectures/lecture9.pdf>> Acesso em: 30 mai. 2019.

WOODMAN, J.O., **An Introduction to Inertial Navigation**, University of Cambridge, Technical Report, N 696, 2007. Disponível em: <<https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>>. Acesso em: 8 dez. 2019.

APÊNDICE A - CÓDIGO PARA A CALIBRAÇÃO DO ACELERÔMETRO

#Estimação dos Erros pelo Métodos Mínimos Quadráticos não Linear

```
function [viesx,viesy,viesz,gx,gy,gz] = estimarviesfs (ax,ay,az,g)
```

```
n=length(ax);
```

```
k=0;
```

```
#valores iniciais
```

```
viesx=2;
```

```
viesy=0;
```

```
viesz=0;
```

```
gx=1;
```

```
gy=1;
```

```
gz=1;
```

```
while(1)
```

```
k++;
```

```
a11=0;
```

```
a22=0;
```

```
a33=0;
```

```
a44=0;
```

```
a55=0;
```

```
a66=0;
```

```
a12=0;
```

```
a13=0;
```

```
a14=0;
```

```
a15=0;
```

```
a16=0;
```

```
a23=0;
```

```
a24=0;
```

```
a25=0;
```

```
a26=0;
```

```
a34=0;
```

```
a35=0;
```

```

a36=0;
a45=0;
a46=0;
a56=0;
b1=0;
b2=0;
b3=0;
b4=0;
b5=0;
b6=0;
for i=1:n
    accg=sqrt(gx**2*(ax(i)-viesx)**2+gy**2*(ay(i)-viesy)**2+gz**2*(az(i)-viesz)**2);
    a11+=gx**4*(ax(i)-viesx)**2/accg**2;
    a22+=gy**4*(ay(i)-viesy)**2/accg**2;
    a33+=gz**4*(az(i)-viesz)**2/accg**2;
    a12+=gx**2*gy**2*(ax(i)-viesx)*(ay(i)-viesy)/accg**2;
    a23+=gy**2*gz**2*(ay(i)-viesy)*(az(i)-viesz)/accg**2;
    a13+=gx**2*gz**2*(ax(i)-viesx)*(az(i)-viesz)/accg**2;
    a14+=-gx**2*(ax(i)-viesx)*gx*(ax(i)-viesx)**2/accg**2;
    a15+=-gx**2*(ax(i)-viesx)*gy*(ay(i)-viesy)**2/accg**2;
    a16+=-gx**2*(ax(i)-viesx)*gz*(az(i)-viesz)**2/accg**2;
    a24+=-gy**2*(ay(i)-viesy)*gx*(ax(i)-viesx)**2/accg**2;
    a25+=-gy**2*(ay(i)-viesy)*gy*(ay(i)-viesy)**2/accg**2;
    a26+=-gy**2*(ay(i)-viesy)*gz*(az(i)-viesz)**2/accg**2;
    a34+=-gz**2*(az(i)-viesz)*gx*(ax(i)-viesx)**2/accg**2;
    a35+=-gz**2*(az(i)-viesz)*gy*(ay(i)-viesy)**2/accg**2;
    a36+=-gz**2*(az(i)-viesz)*gz*(az(i)-viesz)**2/accg**2;
    a44+=gx**2*(ax(i)-viesx)**4/accg**2;
    a55+=gy**2*(ay(i)-viesy)**4/accg**2;
    a66+=gz**2*(az(i)-viesz)**4/accg**2;
    a45+=gx*gy*(ax(i)-viesx)**2*(ay(i)-viesy)**2/accg**2;

```

```

a46+=gx*gz*(ax(i)-viesx)**2*(az(i)-viesz)**2/accg**2;
a56+=gy*gz*(ay(i)-viesy)**2*(az(i)-viesz)**2/accg**2;
b1+=(g-accg)*-gx**2*(ax(i)-viesx)/accg;
b2+=(g-accg)*-gy**2*(ay(i)-viesy)/accg;
b3+=(g-accg)*-gz**2*(az(i)-viesz)/accg;
b4+=(g-accg)*gx*(ax(i)-viesx)**2/accg;
b5+=(g-accg)*gy*(ay(i)-viesy)**2/accg;
b6+=(g-accg)*gz*(az(i)-viesz)**2/accg;

endfor
A=[a11,a12,a13,a14,a15,a16;a12,a22,a23,a24,a25,a26;a13,a23,a33,a34,a35,a36;a1
4,a24,a34,a44,a45,a46;a15,a25,a35,a45,a55,a56;a16,a26,a36,a46,a56,a66];

b=[b1,b2,b3,b4,b5,b6]';

x=linsolve(A,b);

viesx+=x(1);
viesy+=x(2);
viesz+=x(3);
gx+=x(4);
gy+=x(5);
gz+=x(6);

if((k>=50)||((norm(x)<0.00001))
printf("Convergência após %d iterações\n",k);
disp(x)
break;
endif
endwhile
endfunction

```

APÊNDICE B - INTEGRAÇÃO DO GIROSCÓPIO

```
function z=MultQuat(x,y)
    #multiplicação de dois quatérnions
    if((size(x)(1)==size(y)(1))&&(size(x)(2)==size(y)(2)))
        z=zeros(size(x)(1),size(x)(2));
        z(1)=x(1)*y(1)-x(2)*y(2)-x(3)*y(3)-x(4)*y(4);
        z(2)=x(1)*y(2)+x(2)*y(1)+x(3)*y(4)-x(4)*y(3);
        z(3)=x(1)*y(3)-x(2)*y(4)+x(3)*y(1)+x(4)*y(2);
        z(4)=x(1)*y(4)+x(2)*y(3)-x(3)*y(2)+x(4)*y(1);
    else
        disp("Os argumentos fornecidos tem o tamanho errado")
    end
endfunction
```

```
Function z=Qrot2(q,a1,a2,b1,b2,c1,c2,deltat,n=4)
    %algoritmo dos quaternions, velocidade angular é o valor médio do intervalo
    if(((size(q)==[4,1])==[true,true]))
        ang1=deltat*(a1+a2)/2;
        ang2=deltat*(b1+b2)/2;
        ang3=deltat*(c1+c2)/2;
        ang=sqrt(ang1**2+ang2**2+ang3**2);
        s=0.25*ang**2;
        switch n %ordem do algoritmo
            case 1
                ac=1;
                as=0.5;
            case 2
                ac=1-s/2;
                as=0.5;
            case 3
```

```

    ac=1-s/2;
    as=0.5*(1-s/6);
case 4
    ac=1-s/2+s**2/24;
    as=0.5*(1-s/6);
case 5
    ac=1-s/2+s**2/24;
    as=0.5*(1-s/6+s**2/120);
case 6
    ac=1-s/2+s**2/24-s**3/720;
    as=0.5*(1-s/6+s**2/120);
otherwise
    ac=cos(ang/2);
    as=sin(ang/2)/ang;
endswitch
r=zeros(size(q)(1),size(q)(2));
r(1)=ac;
r(2)=as*ang1;
r(3)=as*ang2;
r(4)=as*ang3;
z=MultQuat(q,r);
#caso se deseje normalizar, inserir z=(1+0.5*(1-(z*invquat(z))))*z
else
    disp("O argumento tem o tamanho errado")
end
endfunction

function [roll,pitch,yaw,q]=orient2(wx,wy,wz,t,a=0,b=0,c=0,n=6)
    %chama Qrot2, isso é, integra pela velocidade angular média no intervalo deltat
    q=zeros(4,length(t));
    q(:,1)=__ang2quater1__(a,b,c); #orientação inicial

```

```

deltat=intervalo(t);
roll=zeros(length(t),1);
pitch=zeros(length(t),1);
yaw=zeros(length(t),1);
for i=2:length(t)
    q(:,i)=Qrot2(q(:,i-1),wx(i-1),wx(i),wy(i-1),wy(i),wz(i-1),wz(i),deltat(i-1),n);
endfor
for k=1:length(t)
    [roll(k) pitch(k) yaw(k)]=quater2ang(q(:,k));
endfor
endfunction

function z=filtrocompl(accel,giro,t,tau=5)
    #Implementação do filtro Complementar. O parâmetro opcional tau é assumido
    como 5. Mudar
    #de acordo com a necessidade, maiores valores favorecem o giroscópio (maior
    frequência)
    dt=intervalo(t);
    T=(t(length(t))-t(1))/length(t);
    z(1)=giro(1);
    for k=2:length(t)
        z(k)=z(k-1)*exp(-dt(k-1)/tau)+(1-exp(-dt(k-1)/tau))*accel(k)+giro(k)-giro(k-1);
    endfor
    z=z';
endfunction

function [roll,pitch,yaw]=quater2ang(q)
    #retorna os ângulos equivalente em bryan1 de um quaternion q
    if((max(size(q))==4)&&(min(size(q))==1))
        roll=atan2(2*(q(3)*q(4)+q(1)*q(2)),1-2*(q(2)**2+q(3)**2));
        a=1-2*(q(2)**2+q(3)**2);

```

```

b=2*(q(3)*q(4)+q(1)*q(2));
r=sqrt(a^2+b^2);
pitch=asin(2*(q(1)*q(3)-q(4)*q(2)));
yaw=atan2(2*(q(1)*q(4)+q(2)*q(3)),1-2*(q(3)**2+q(4)**2));
else
    disp("O argumento fornecido tem o tamanho errado")
endif
endfunction

function q=__ang2quater1__(roll,pitch,yaw)
q=zeros(4,length(roll));
for i=1:length(roll)
    cp=cos(pitch(i)/2);
    sp=sin(pitch(i)/2);
    cr=cos(roll(i)/2);
    sr=sin(roll(i)/2);
    cy=cos(yaw(i)/2);
    sy=sin(yaw(i)/2);
    q(1,i)=cr*cp*cy+sr*sp*sy;
    q(2,i)=sr*cp*cy-cr*sp*sy;
    q(3,i)=cr*sp*cy+sr*cp*sy;
    q(4,i)=cr*cp*sy-sr*sp*cy;
endfor
endfunction

```

APÊNDICE C - CÓDIGO DO FILTRO DE KALMAN

```
function z=__kalman__(a,b,t,V,str,anginicial=0)

    #Implementação do filtro de Kalman, espaço de estados com 2 componentes

    #a corresponde a velocidade angular, b a medição auxiliar, o vetor V indica se as
    medições são válidas

    global n_bits;

    global wmax;

    deltat=intervalo(t);

    f=1/mean(deltat);

    H=[1 0];

    if(strcmpi(str,"roll")==1)

        #elementos de Q

        sigmaalfa=5; #desvio padrão

        sigmavies=0.5; #desvio padrão do viés em graus/s

        sigmaomega=2; #desvio padrão da medição do ângulo, em graus

    endif

    if(strcmpi(str,"pitch")==1)

        #elementos de Q

        sigmaalfa=1; #desvio padrão

        sigmavies=0.5; #desvio padrão do viés em graus/s

        sigmaomega=2; #desvio padrão da medição do ângulo, em graus

    endif

    if(strcmpi(str,"yaw")==1)

        #elementos de Q

        sigmaalfa=1; #desvio padrão

        sigmavies=0.5; #desvio padrão do viés em graus/s

        sigmaomega=2; #desvio padrão da medição do ângulo, em graus

    endif
```

```

sigmaalfa=pi/180*sigmaalfa; #converte para rad
sigmavies=pi/180*sigmavies;
Q=(1/f)^2*[sigmaalfa^2,0;0,sigmavies^2];
R=(pi/180*sigmaomega)^2;
x(:,1)=[anginicial 0]';
x1(:,1)=x(:,1);
x2(:,1)=x(:,1);
P2=zeros(2,2);
P1=zeros(2,2);

for k=2:length(t)
    #previsão
    A=[1 -deltat(k-1);0 1];
    B=[deltat(k-1);0];
    x1(:,k)=A*x2(:,k-1)+B*a(k-1);
    P1=A*P2*A'+Q;
    #atualização
    if(V(k)==0)
        x2(:,k)=x1(:,k);
        P2=P1;
        continue;
    endif
    K=P1*H'*inv(H*P1*H'+R);
    x2(:,k)=x1(:,k)+K*(b(k)-H*x1(:,k));
    P2=(eye(2,2)-K*H)*P1;
endfor
z=x2(1,:);
endfunction

```

APÊNDICE D - CÓDIGO DO ALGORITMO INERCIAL

```

function [pos,vel]=inercial(fx,fy,fz,q,wx,wy,wz,t,vx0=0,vy0=0,vz0=0,px0=0,py0=0,pz0=
0)
    global glocal;
    intnum=1;
    deltat=intervalo(t);
    pos=zeros(length(t),3);
    vel=zeros(length(t),3);
    pos(1,:)= [px0 py0 pz0]; vel(1,:)= [vx0 vy0 vz0];
    for k=2:length(t)
        fb=[fx(k-1) fy(k-1) fz(k-1)];
        w=[wx(k-1) wy(k-1) wz(k-1)];
        u=MultQuat(MultQuat(q(:,k)',vet2quat(deltat(k-1)**2*cross(w,fb)/2+fb*deltat(k-
1))),invquat(q(:,k))');
        vel(k,:)=vel(k-1,:)+quat2vet(u)+deltat(k-1)*[0 0 glocal];
        switch intnum
            case 1
                pos(k,:)=pos(k-1,:)+deltat(k-1)*vel(k-1,:);
            case 2
                pos(k,:)=pos(k-1,:)+deltat(k-1)*(vel(k-1,:)+vel(k,:))/2;
            case 3
                if(k==1)
                    pos(2,:)=pos(1,:)+deltat(1)/2*(vel(1,:)+vel(2,:));
                else
                    pos(k,:)=pos(k-2,:)+deltat(k)/3*(vel(k-2,:)+4*vel(k-1,:)+vel(k,:));
                endif
            endswitch
        endfor
    endfunction

```