



Centro de Tecnologia e Urbanismo
Departamento de Engenharia Elétrica

Rhelimuthe Elísio Dias Campos

Auto-Alinhamento para Sistemas de Navegação Inercial usando Algoritmo REQUEST com Estimador de Bias e Redes Neurais Artificiais.

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual de Londrina para obtenção do Título de Mestre em Engenharia Elétrica.

Londrina, PR



Rhelimuthe Elísio Dias Campos

Auto-Alinhamento para Sistemas de Navegação Inercial usando Algoritmo REQUEST com Estimador de Bias e Redes Neurais Artificiais.

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual de Londrina para obtenção do Título de Mestre em Engenharia Elétrica.

Área de concentração: Sistemas Eletrônicos
Especialidade: Otimização de Sistemas, Processamento de Sinais, Inteligência Artificial e Navegação Inercial

Orientador:
Prof. Dr. Leonimer Flávio de Melo

Ficha Catalográfica

Dias Campos, Rhelmuthe Elísio

Auto-Alinhamento para Sistemas de Navegação Inercial usando Algoritmo REQUEST com Estimador de Bias e Redes Neurais Artificiais.. Londrina, PR, . 73 p.

Dissertação (Mestrado) – Universidade Estadual de Londrina, PR. Departamento de Engenharia Elétrica

1. Reconhecimento de Padrões. 2. Inteligência Artificial. 3. Processamento de Sinais. 4. Navegação Inercial 5. Otimização de Sistemas I. Universidade Estadual de Londrina. Departamento de Engenharia Elétrica.

Rhelimuthe Elísio Dias Campos

Auto-Alinhamento para Sistemas de Navegação Inercial usando Algoritmo REQUEST com Estimador de Bias e Redes Neurais Artificiais.

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual de Londrina para obtenção do Título de Mestre em Engenharia Elétrica.

Área de concentração: Sistemas Eletrônicos
Especialidade: Otimização de Sistemas, Processamento de Sinais, Inteligência Artificial e Navegação Inercial

Comissão Examinadora

Prof. Dr. Leonimer Flávio de Melo
Depto. de Engenharia Elétrica
Orientador

Prof. Dr. José Alexandre de França
Depto. de Engenharia Elétrica
Universidade Estadual de Londrina

Prof. Dr. Alessandro Goedtel
Depto. de Engenharia Elétrica
Universidade Tecnológica Federal do Paraná

Prof. Dr. Francisco Granziera Junior
Depto. de Engenharia Elétrica
Universidade Estadual de Londrina

Prof. Dr. Marcelo Carvalho Tosin
Depto. de Engenharia Elétrica
Universidade Estadual de Londrina

4 de junho de 2018

Agradecimentos

Ao meu orientador Prof. Dr. Leonimer Flávio de Melo pela confiança atribuída a mim e por não medir esforços para ajudar em momentos difíceis.

Deixo registrada a minha gratidão, ao Prof. Dr. Marcelo Tosin, membro da banca examinadora, pela Co-Orientação, pelo esforço despendido em diversos momentos para desenvolver, contribuir e supervisionar o trabalho.

Ao Prof. Dr. Francisco Granziera jr, membro da banca examinadora, pela Co-Orientação, pelas contribuições, por compartilhar o seu conhecimento sobre o tema e principalmente pelas dicas fundamentais para produção do trabalho.

Ao Prof. Me. Daniel Strufaldi Batista pelas contribuições e sugestões ao trabalho.

Ao meu pai e a minha mãe pelo amor, comprometimento e esforço necessário para assegurar, sem preocupações extras, a minha dedicação ao curso.

Aos meus colegas que em momentos de descontração foram fundamentais.

A Agência Espacial Brasileira (AEB), a Coordenação Brasileira para o Aperfeiçoamento do Pessoal de Ensino Superior (CAPES), ao Instituto Brasileiro de Aeronáutica e Espaço (IAE), a Fundação Araucária Brasileira (FA), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) e também a Universidade Estadual de Londrina por apoiar a implementação deste projeto.

Resumo

O problema do alinhamento de Sistemas de navegação inercial (SNI) é encontrar uma matriz de atitude inicial. Métodos tradicionalmente aplicados para realizar o auto-alinhamento utilizam a gravidade, a taxa de rotação da Terra e as medidas inerciais para isso. No entanto, o ruído presente em sensores inerciais diminui a precisão e a rapidez deste processo. Normalmente, as soluções convencionais implementam um processo de duas etapas, primeiro realizam um cálculo analítico da atitude, seguido da estimação da atitude usando filtros. Para investigar uma alternativa a tais métodos, este trabalho propõe a solução grosseira através do uso do algoritmo REQUEST. A fim de otimizar e buscar uma solução ainda mais rápida, também é aplicado o uso de uma Rede Neural Artificial (RNA e um estimador de *bias* para melhorar o algoritmo REQUEST. Para avaliar os métodos propostos, o trabalho utiliza dados simulados e dados de um SNI comercial, e os resultados obtidos são comparados com uma solução tradicionalmente empregada na literatura, através do algoritmo TRIAD. Os resultados obtidos mostram que o algoritmo REQUEST apresentou melhor estimativa do que a solução usando TRIAD. Ademais, a inclusão da rede neural e do estimador de bias no sistema foi promissora, já que melhorou significativamente seu desempenho.

Abstract

Inertial Navigation Systems (INS) alignment problem is the need to find an initial attitude matrix. Methods traditionally applied in such alignment use gravity, Earth's rotation rate, and inertial measurements. However, the noise present in inertial sensors decreases the accuracy and speed of this process. Normally, conventional solutions implement a two-step process, first it is performed an analytical attitude calculation, followed by attitude estimation using filters. To investigate an alternative to such methods, this work proposes a solution through the use of the REQUEST algorithm. In order to optimize and seeking an even faster solution, the use of a multi-layer artificial neural network (ANN) and a bias estimator to improve the REQUEST algorithm is also applied. To evaluate the proposed methods, the work uses simulated data and data from a commercial INS, and the results obtained are compared with a traditional solution employed in the literature, using the TRIAD algorithm. The results obtained showed that the increased REQUEST method employed has better output data than TRIAD method. In addition, the inclusion of the neural network and bias estimator in the system was promising, significantly improving its performance.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

Lista de Símbolos

1	Introdução	1
1.1	Objetivo Geral	3
1.2	Objetivo Especifico	4
1.3	Justificativas	4
1.4	Metodologias	4
1.5	Organização da dissertação	5
2	Revisão Bibliográfica	6
2.1	Sistemas de Navegação Inercial	6
2.1.1	Unidade de Medida Inercial	7
2.1.2	Sistema de plataformas Estáveis	8
2.1.3	Sistema <i>Strapdown</i>	9
2.1.4	Ruídos	11
2.2	<i>Dead Reckoning</i> e Estrelas fixas	12
2.3	Sistemas de Coordenadas	12
2.3.1	Triedro de Referencia Inercial	13
2.3.2	Triedro do Corpo	13
2.3.3	Triedro de Local e Global	13

2.4	Alinhamento de Sensores Inerciais	14
2.5	Análise de Erro: Modelo Quasi-Estacionário	15
2.5.1	Ângulos de Euler	18
2.5.2	Quatérnions	20
2.5.3	Desvio Padrão de Allan	21
2.6	Alinhamento	21
2.6.1	Alinhamento Plataforma Fixa	22
2.7	Auto-Alinhamento	23
2.7.1	Alinhamento Grosseiro: Abordagem Tradicional	24
2.7.2	Alinhamento Grosseiro: Abordagem Alternativa	26
2.7.3	Alinhamento Fino: Abordagem FKD	28
2.7.4	Alinhamento Fino por Filtragem Indireta	29
2.7.5	Alinhamento Fino: Abordagem Clássica	29
2.7.6	Alinhamento Fino: Abordagem FKI	32
2.8	Estado da Arte	34
3	Auto-Alinhamento Tradicional	36
3.1	Metodologia Experimental	36
3.2	Dados Reais	36
3.3	Simulation Data	37
3.4	Alinhamento Grosseiro	37
3.5	Alinhamento Fino	40
3.6	Resultados	41
3.6.1	Dados reais	41
3.6.2	Dados simulados	42
4	Alinhamento Grosseiro usando REQUEST	44
4.1	Resultados	47
4.1.1	Dados reais	48

4.1.2	Dados simulados	49
5	REQUEST com estimador de <i>bias</i>	52
5.1	Estimador	52
5.2	Inicialização do Algoritmo	53
5.3	Resultados	54
5.3.1	Dados simulados	54
6	Método proposto <i>Neural REQUEST</i>	56
6.1	Esquema geral	56
6.2	Algoritmo de aprendizagem	58
6.2.1	Algoritmo geral baseado em gradiente	59
6.2.2	Extreme learning machine	60
6.3	Número de neurônios	60
6.4	Função de ativação	61
6.5	Resultados	61
6.5.1	Dados Simulados	61
6.6	Validação da rede neural	63
7	Análise dos resultados	65
8	Conclusões	68
8.1	Sugestões para Trabalho Futuros	68
	Referências	70
	Apêndice A – Produção Científica	73
A.1	Artigos Publicados	73
A.2	Artigos Submetidos	73

Lista de Figuras

1.1	Estrutura de referência global (Local) e do corpo.	3
2.1	Diagrama de blocos da estrutura de uma plataforma inercial	7
2.2	Sistema de plataformas Estáveis.	8
2.3	Algoritmo do SNI: Sistema de Plataforma Estável	9
2.4	<i>Gimbal Lock</i>	10
2.5	Sistema de plataforma <i>Strapdown</i>	10
2.6	O algoritmo de navegação inercial pelo sistema <i>Strapdown</i>	10
2.7	Conjunto de eixos de um veículo	14
2.8	Eixos de referencia Local	15
2.9	Alinhamento do vetor gravidade é um único plano.	22
3.1	Ângulos TRIAD com dados reais.	42
3.2	Ângulos de alinhamento grosseiro TRIAD.	42
3.3	Acurácia alinhamento grosseiro TRIAD	43
3.4	Precisão alinhamento grosseiro TRIAD.	43
4.1	Diagrama de Bloco REQUEST	47
4.2	Ângulos REQUEST com dados reais.	48
4.3	Ângulos REQUEST com dados reais em regime transitório.	48
4.4	Ângulos de alinhamento grosseiro REQUEST.	49
4.5	Acurácia alinhamento grosseiro REQUEST	50
4.6	Precisão alinhamento grosseiro REQUEST	51
5.1	Função $E(b_x, b_y, b_z)$ em 1 segundo.	53
5.2	Função $E(b_x, b_y, b_z)$ em 1 segundo.	54
5.3	Ângulos de alinhamento grosseiro REQUEST usando estimador usando dados simulados.	54

5.4	Acurácia alinhamento grosseiro REQUEST usando estimador usando dados simulados.	55
5.5	Precisão alinhamento grosseiro REQUEST usando estimador usando dados simulados.	55
6.1	Neuron Model.	56
6.2	Diagrama de blocos do NREQUEST	58
6.3	Ângulos de alinhamento grosseiro NREQUEST.	62
6.4	Acurácia alinhamento grosseiro NREQUEST	62
6.5	Precisão alinhamento grosseiro NREQUEST	63
7.1	Erro em quatro dimensões.	66

Lista de Tabelas

6.1	Validação	64
7.1	Erro médio quadrático de estimação	66

Lista de Abreviaturas

SNIS Sistemas de Navegação Inercial *Strapdown*

RNA Redes Neurais Artificiais

UMI Unidade de Medida de Aceleração

IAE Instituto de Aeronautica e Espaço

ELM *Extreme learning machine*

Lista de Símbolos

Os seguintes símbolos serão utilizados:

símbolo	descrição
ϵ_b	Erro de bias estimado pelo filtro de kalman
L	Latitude
g	Gravidade
Ω	Aceleração angular da Terra
∇	Bias do acelerômetro
σ_{f_i}	Desvio padrão do ruído branco para o acelerômetro no eixo i
σ_{ω_i}	Desvio padrão do ruído branco para o giroscópio do eixo i
ω_i	Taxa angular no eixo i do corpo
f_i	Força específica no eixo i do corpo
C_B^N	Matriz de atitude do eixo do corpo pro eixo de navegação
P_k	Matriz de covariância do erro
Q_k	matriz de covariância do ruído do processo
b	Bias
E	Erro modelado entre bias estimado e real a priori.
K	Autovalor do sistema REQUEST
λ	Autovalor do quatérnion
q	Quatérnion
ψ	Roll
θ	Pitch
ϕ	Yaw

1 Introdução

Um sistema de navegação inercial *strapdown* (SNIS) é frequentemente usado em aviões, mísseis e sistemas de lançamento espacial (TITTERTON; WESTON, 2004a).

O SNIS é um sistema Dead Reckoning, que integra dados de velocidade angulares aceleração obtidos de sensores inerciais em tempo hábil, fornecendo a posição, velocidade e atitude de um veículo durante o voo.

O alinhamento inicial preciso é um procedimento essencial para garantir a estabilidade dos SNISs.

O auto-alinhamento pode ser definido como a inicialização antes da navegação sem assistência externa (SAVAGE, 2007a). Em outras palavras, o sistema de inicialização deve encontrar automaticamente a atitude inicial do veículo e, também, os parâmetros iniciais de seus sensores.

No entanto, erros intrínsecos de sensores inerciais geram um erro significativo no alinhamento inicial. Portanto, os SNISs requerem alinhamento inicial efetivo para navegação precisa. Em várias aplicações, como as militares, esse processo deve ocorrer em um tempo muito curto.

O problema de auto-alinhamento é tradicionalmente resolvido em dois estágios, ou seja, um estágio de alinhamento grosso seguido por um estágio de alinhamento fino (JEKELI, 2001). O primeiro calcula a atitude analiticamente usando a gravidade e a taxa de rotação da Terra, vistas como medidas de seus sensores inerciais. O alinhamento fino é então usado para compensar os erros dos sensores. A atitude inicial precisa, o bias do sensor e a velocidade inicial são alcançados pelo alinhamento fino (LI et al., 2015; LIU et al., 2014).

O passo de alinhamento grosso é tipicamente baseado no algoritmo de determinação de atitude de três eixos (TRIAD) (SHUSTER; OH, 1981), enquanto que para o alinhamento fino, o filtro de Kalman tem sido tipicamente empregado (SAVAGE, 2007a).

A desvantagem desse procedimento tradicional é o tempo necessário para que

o alinhamento seja efetivo, normalmente em torno de 20 minutos (GAO et al., 2011).

As metodologias tradicionais levam até 20 minutos para atingir o alinhamento completo (GAO et al., 2011). Apesar disso, ao melhorar a acurácia do alinhamento grosseiro o alinhamento completo pode diminuir significativamente.

O desempenho do alinhamento grosseiro é crucial para alcançar um alinhamento preciso. Uma estimativa precisa do alinhamento aproximado aumentará a taxa de convergência do alinhamento fino e o tempo de alinhamento completo diminuirá (SHEN et al., 2016; LÜ; XIE; CHEN, 2009; XU et al., 2017a).

Portanto, se houver um método que possa ser implementado melhor que o alinhamento tradicional em termos de taxa de convergência, precisão de alinhamento e estabilidade dos resultados de alinhamento, será um novo alinhamento automático que pode ser aplicado em algumas situações de emergência.

Para isso, melhorar o alinhamento grosseiro é primordial para atingir o alinhamento otimizado. Por este motivo, este trabalho se concentra nessa etapa do alinhamento completo.

Como uma alternativa aos estudos anteriores, este trabalho propõe o uso do algoritmo REQUEST para citar a matriz de atitude inicial e realizar o auto-alinhamento.

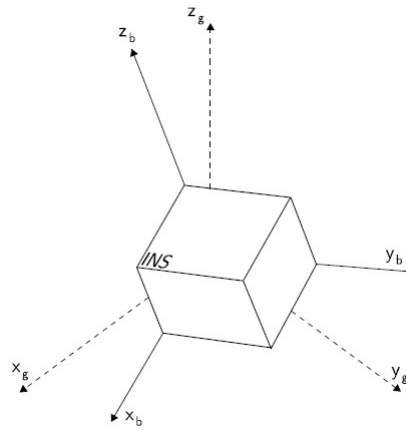
Como alternativa a estudos anteriores, este trabalho propõe o uso do algoritmo REQUEST (BAR-ITZHACK, 1996) para encontrar a matriz de atitude inicial.

Portanto, neste trabalho um método de alinhamento grosseiro baseado no algoritmo REQUEST com uma rede neural artificial e com a modelagem de um estimador de bias é proposto para alcançar o alinhamento grosseiro com mais precisão e acurácia, respectivamente.

O alinhamento é um processo pelo qual a orientação dos eixos da navegação da plataforma (Corpo) é determinado em relação ao eixo de referência Global (Local).

Segundo Savage (2007b) o problema do Alinhamento de sistemas de navegação inercial é determinar a matriz de atitude inicial entre as coordenadas da Estrutura do corpo e as coordenadas da estrutura Local.

Logo, a integração das medidas dos sensores inerciais deve ser realizada somente após o alinhamento inicial, isto é, após os valores iniciais da velocidade,

Figura 1.1: Estrutura de referência global (Local) e do corpo.

Fonte: (WOODMAN, 2007)

posição e orientação serem determinados.

Caso isso não seja realizado, ao integrar as medidas inerciais, o erro dos valores iniciais causarão uma diferença muito grande nos valores calculados durante a navegação.

O processo de encontrar a atitude inicial e a posição inicial é denominado inicialização. Logo, defini-se Auto-Alinhamento a inicialização antes da navegação sem nenhum auxílio externo (SAVAGE, 2007b).

No entanto, existem muitas complicações nesse processo, entre elas os ruídos aleatórios dos sensores inerciais, que fazem o alinhamento ser bastante demorado e complexo (TITTERTON; WESTON, 2004b), isto ocorre devido a necessidade de utilizar filtros não lineares e a dedução de modelos matemáticos para descrever esses filtros.

Por isso, o problema de realizar o Auto-Alinhamento de forma simples, rápida e precisa é investigado nesse trabalho.

Ao longo desse trabalho vamos nos referir a estrutura do sistema de navegação como a estrutura do corpo e à sistema de referência como a estrutura local ou global (WOODMAN, 2007), como mostrado na figura 1.1.

1.1 **Objetivo Geral**

Propor novas soluções e perspectivas para solucionar o problema de auto-alinhamento.

1.2 Objetivo Especifico

Os objetivos específicos são

- Avaliar os métodos de auto-alinhamento.
- Investigar o uso do algoritmo REQUEST para essa finalidade.
- Investigar projetos de estimadores e *Machine Learnings* para melhorar a performance do algoritmo REQUEST.
- Analisar a performance do algoritmo REQUEST usando os estimadores e *Machine Learnings* propostos e comparar em relação ao tradicional.

1.3 Justificativas

Como já dito, os SNIs são amplamente utilizados para a navegação de aeronaves, mísseis guiados, veículos espaciais, navios, submarinos e veículos terrestres (TITTERTON; WESTON, 2004b).

Em muitas aplicações é essencial alcançar um alinhamento preciso dentro de um período muito curto de tempo. Isto é particularmente verdadeiro em muitas aplicações militares, nas quais um tempo de resposta rápido ou instantâneo é um requisito primordial (TITTERTON; WESTON, 2004b). Assim, esta se torna a motivação desse trabalho.

1.4 Metodologias

Os métodos utilizados nesse trabalho são baseados em melhorias do algoritmo REQUEST proposto por Bar-Itzhack (1996) utilizando redes neurais com o algoritmo de treinamento *extreme learning machine* indicado por Huang, Zhu e Siew (2006) e um estimador de *bias*.

O software MATLAB foi empregado para realizar tais simulações. Além disso, a dados práticos foram coletados no Instituto de Aeronáutica e Espaço (IAE) com um SNIS construído na própria instituição.

1.5 Organização da dissertação

No capítulo 2, realiza-se uma breve revisão de literatura. No capítulo 2.8 os principais e mais atuais trabalhos relacionados são apresentados. A descrição do auto-alinhamento tradicional é descrito no capítulo 3. O alinhamento grosseiro usando o algoritmo REQUEST, REQUEST com estimador de bias e REQUEST com redes neurais são apresentados nos capítulos 4, 6 e 5, respectivamente. Análise dos resultados é apresentada no capítulo 7 e as conclusões no capítulo 8

2 Revisão Bibliográfica

Uma revisão de bibliográfica do tema é apresentada neste capítulo, a fim de contextualizar o leitor. Assim, detalhes sobre o alinhamento de maneira global são definidos.

2.1 Sistemas de Navegação Inercial

A navegação, resume-se em encontrar o caminho de um lugar para o outro. Uma forma bastante simples de navegação é seguir direções ou instruções, entretanto esse método depende de características conhecidas ou objetos fixos (TITTERTON; WESTON, 2004b).

O Sistema de navegação Inercial (SNI) é o sistema que calcula a posição, a velocidade, e a atitude de um veículo usando informações obtidas a partir de sensores inerciais (Acelerômetros e Girômetros), isso pode ser verificado a partir da figura 2.1.

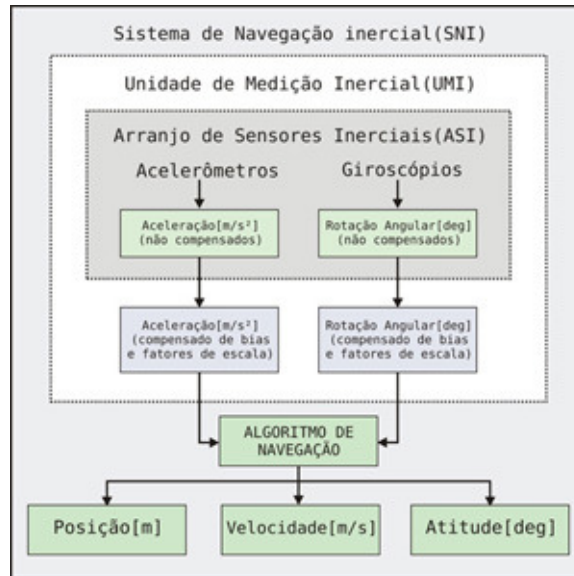
O SNI localiza o norte magnético pela rotação da Terra a partir dos sensores inerciais. Assim, possibilita-se encontrar os movimentos angulares tridimensionais de uma plataforma inercial, como por exemplo um veículo lançador de satélites .

Os sinais originados desses sensores são corrompidos por ruídos instrumentais (ruídos aleatórios), por isso são necessários algoritmos específicos para minimizar esse problema, caso contrário a trajetória reconstruída através desses sinais poderá ter desvios significativos.

Tais sistemas são amplamente utilizados para a navegação de aeronaves, mísseis guiados, veículos espaciais, navios, submarinos e veículos terrestres (TITTERTON; WESTON, 2004b)

O SNI deve ter vantagens particulares como autonomia e alta precisão em curto espaço de tempo. (Xu F; Fang J.C , 2008).

Dessa forma, como mostrado na figura 2.1 o sistema de navegação inercial é

Figura 2.1: Diagrama de blocos da estrutura de uma plataforma inercial

Fonte: (VELGER, 200)

constituído por dois sistemas:

- Uma Unidade de medida inercial (UMI) onde os sensores inerciais (Acelerômetros e giroscópios) são fixados;
- Um sistema microprocessado com o algoritmo de navegação.

Com isso, a partir das medidas inerciais fornecidas pela UMI, o sistema microprocessado calcula através do algoritmo de navegação a posição, a velocidade e a atitude.

2.1.1 Unidade de Medida Inercial

A UMI é um sub-conjunto do SNI completo. Essa estrutura contém os sensores inerciais em conjunto com a eletrônica de suporte necessária para operar os sensores, além do microprocessador para compensar os desvios nos sinais de medidas gerados pelos sensores (WOODMAN, 2007).

O acoplamento dos sensores inerciais na UMI é realizado de forma que sua configuração espacial caracterizem uma tríade ortogonal, uma para acelerômetro e outra para girômetros, a modelagem matemática dessas medidas serão mostradas detalhadamente no capítulo 2.7.1.

Portanto, a posição e a orientação do objeto pode ser calculada com essas medidas.

A UMI é caracterizada em duas categorias: Sistema *Strapdown* (cap 2.1.3) e Sistema de plataforma Estável (cap 2.1.2). A diferença entre as duas categorias é a referência que os sensores inerciais utilizam. Esse trabalho se concentra no sistema *Strapdown*, assim toda a descrição da UMI realizada até aqui é referente a essa categoria.

Alem disso, A figura 2.6 indica a necessidade de corrigir a gravidade, já que há variações na intensidade da gravidade da Terra. Isto ocorre devido a distribuição não uniforme da massa da Terra e da aceleração centrífuga, que varia em função da latitude da Terra(SANTANA, 2011).

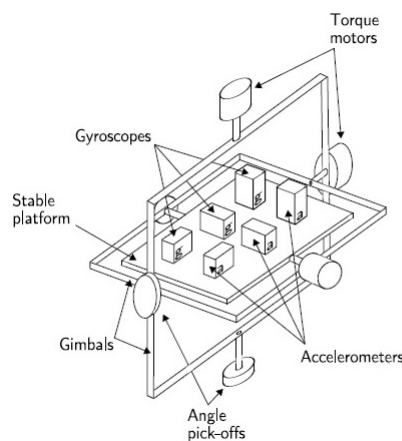
Segundo Santana (2011) a gravidade pode ser calculada pela equação 2.1:

$$\begin{bmatrix} \frac{-\Omega(R_0+h)}{2} \sin 2L \\ 0 \\ g - \frac{\Omega(R_0+h)}{2} (1 + \cos 2L) \end{bmatrix} \quad (2.1)$$

2.1.2 Sistema de plataformas Estáveis

Nesses sistemas, os sensores inerciais são montados sobre uma plataforma estável e são mecanicamente isolados a partir do movimento de rotação do veículo.

Figura 2.2: Sistema de plataformas Estáveis.



Fonte: (LAWRENCE, 2012)

Em outras palavras, a plataforma é mantida em alinhamento com a estrutura local. Isto é conseguido através da montagem da plataforma com Gimbals (Janelas), que permitem a liberdade da plataforma em todos os seus três eixos cartesiano (WOODMAN, 2007), conforme mostrado na figura 2.2.

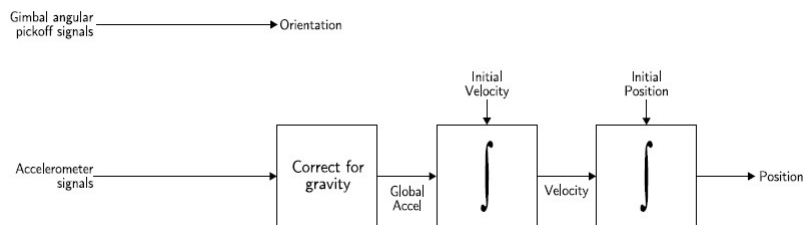
A plataforma com giroscópios realiza a detecção das rotações da plataforma.

Estes sinais são enviados de volta para motores de torque que rodam os balancins a fim de anular essas rotações, mantendo, portanto, a plataforma alinhada com o quadro local.

Para o cálculo da posição do dispositivo, os sinais provenientes dos acelerômetros são duplamente integrados (WOODMAN, 2007). Nota-se que é necessário subtrair a aceleração devido a gravidade em relação à vertical do canal antes de executar a integração.

O algoritmo de navegação inercial pelo sistema de plataforma estável é mostrado na figura 2.3. Sistemas de plataformas tem uso comum, especialmente para aquelas aplicações que requerem estimativa muito precisa dos dados de navegação, como navios e submarinos.

Figura 2.3: Algoritmo do SNI: Sistema de Plataforma Estável



Fonte: (TITTERTON; WESTON, 2004b)

Uma falha pode ocorrer no sistema de plataforma estável se os *Gimbals* da plataforma ficarem alinhados, isso ocorre devido a um conjunto de manobras do veículo, assim o veículo perde o eixo de referência. A figura 2.4 ilustra essa falha claramente.

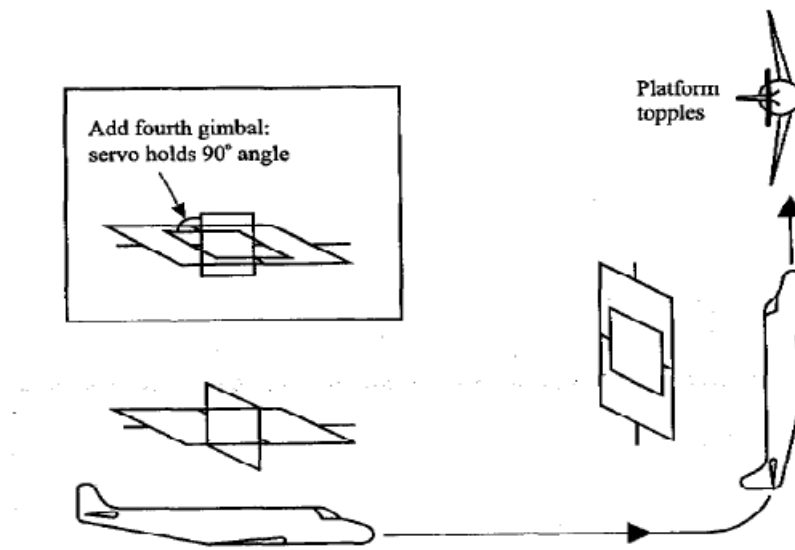
2.1.3 Sistema *Strapdown*

No sistema *Strapdown*, os sensores inerciais são presos diretamente ao veículo e, conseqüentemente, não são isolados a partir do seu movimento angular (Figura 2.5).

Esse sistema é frequentemente definido como um sistema analítico. Os sinais produzidos pelos sensores inerciais são resolvidos matematicamente em um sistema microprocessado.

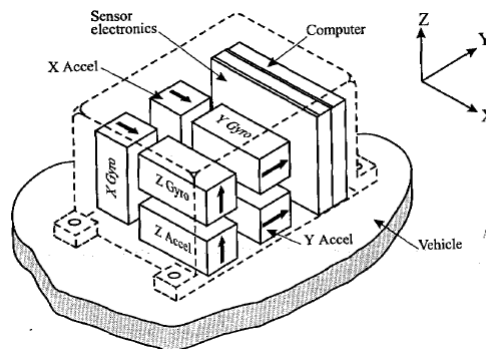
O algoritmo de navegação inercial pelo sistema de plataforma *Strapdown* é mostrado na figura 2.6

Figura 2.4: *Gimbal Lock*



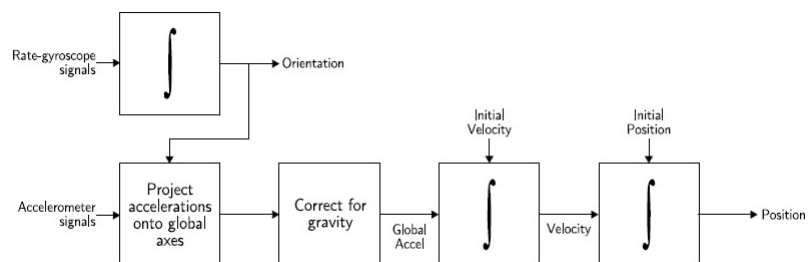
Fonte: (TITTERTON; WESTON, 2004b)

Figura 2.5: Sistema de plataforma *Strapdown*



Fonte: (LAWRENCE, 2012)

Figura 2.6: O algoritmo de navegação inercial pelo sistema *Strapdown*



Fonte: (TITTERTON; WESTON, 2004b)

As vantagens desta abordagem é a redução no custo, tamanho e complexidade mecânica. Além disso, possui confiabilidade maior em comparação com os sistemas de plataforma estável, já que não ocorre o *Gimbal Lock*¹.

As principais desvantagens é um aumento substancial na complexidade computacional e a necessidade de utilizar sensores mais precisos (TITTERTON; WESTON, 2004b).

2.1.4 Ruídos

A exatidão de um SNI depende, fundamentalmente, da precisão e confiança dos seus principais componentes.

Entretanto, as unidades de medição inercial (UMI) de baixo desempenho possuem sensores inerciais com baixa sensibilidade para medidas da velocidade da rotação terrestre.

Com isso, uma diferença de milímetros inicialmente, evolui um erro de quilômetros com o tempo.

As fontes potenciais de erros mais significativas são:

- Erros causados pelo movimento de rotação diário da Terra;
- Atrito nos sistemas giroscópicos;
- Desalinhamento da plataforma estável, resultando que componentes verticais do campo gravitacional da Terra sejam falsamente interpretados como componentes horizontais;
- Imperfeições na construção dos giroscópios e acelerômetros.

Devido ao erro combinado causado por estes fatores, todos os Sistemas de Navegação Inercial apresentam algum grau de erro cumulativo que aumenta com o tempo de operação. Assim, a posição fornecida pelo sistema é periodicamente comparada com posições obtidas por outros meios e, ainda, o SNI deve ser atualizado e calibrado em determinados intervalos de tempo (MARINHA, 2000).

A figura 2.1, mostra a necessidade de compensar os bias (ϵ_{ba}) das medidas inerciais, já que variações na temperatura gera um erro nos *bias* das medidas dos acelerômetros. Isso é indesejado, visto que um erro de *bias* cria um erro de posição

¹Nome dado a falha do sistema de plataforma estável, que ocorre quando, os Gimbals da plataforma ficam alinhados, devido a um conjunto de manobras do veículo

que cresce de forma quadraticamente com o tempo, pois ao integrar duplamente o erro da aceleração (*bias*) a equação da posição se torna, $\frac{S(t) = \epsilon_{ba}(\theta)t^2}{2}$. Portanto, esse erro pode ser corrigido através de um sensor interno da UMI para medi-lo ou através de modelagens matemáticas para estima-lo.

No caso da UMI possuir sensor interno de temperatura, então será possível aplicar correções ao sinal de saída dos acelerômetros no sentido de compensar a variação do *bias* causada pela variação de temperatura

2.2 *Dead Reckoning e Estrelas fixas*

Nessa seção será apresentada os princípios de navegação por estrelas fixas e *Dead Reckoning*², pois a essência desses métodos é utilizada na navegação moderna.

Uma técnica antiga e bem estabelecida é avistar estrelas fixas como referencia (TITTERTON; WESTON, 2004b). No passado, a navegação por esse método era dificultada, pois não havia formas de medir o tempo e de observar referências precisamente.

Assim, após o advento do cronômetro marítimo, o princípio *Dead Reckoning* pode ser utilizado para navegação, já que a posição atual ($s[n]$) é a soma entre a última posição ($s[n-1]$) e o espaço percorrido (Δs) até a posição atual, visto que esse espaço é dado pela velocidade média (\bar{v}) vezes a variação do tempo ($t_n - t_{n-1}$), medida com precisão, entre a última posição e a posição atual. Assim, a equação 2.2 a seguir demonstra essa ideia.

$$s[n] = s[n-1] + (t_n - t_{n-1})\bar{v} \quad (2.2)$$

Esse princípio pode ser implementado usando sensores inerciais (SIs) para detectar o movimento de rotação e de translação em relação a um referencial local. Isto é conhecido como navegação (TITTERTON; WESTON, 2004b). Por isso, esse principio é enfatizado ao longo do trabalho.

2.3 Sistemas de Coordenadas

A navegação possui como dificuldade a existência sistemas de coordenadas múltiplos. Por exemplo, os sensores inerciais medem o movimento em relação a um referen-

²Método de cálculo da posição do veículo, especialmente no mar, estimando-se a direção e distância percorrida em vez de usar pontos de referência, observações astronômicas, ou métodos de navegação eletrônicos.

cial inercial, porem o movimento da Terra não pode ser desprezada, por isso o sistema não pode ser considerado inercial como será mostrado na seção 2.3.1.

Logo, a relação entre os diversos sistemas de coordenadas devem ser modelados e avaliados.

2.3.1 Triedro de Referencia Inercial

A atitude é sempre relacionada a um sistema de referência arbitrário, isto é, um sistema de coordenadas definido pelo engenheiro em relação ao qual a atitude é referida ou medida.

O Triedro é definido como inercial quando está em repouso ou apenas em movimento de translação uniforme.

Para que o calculo da atitude não seja afetado pelo sistema de referencia é desejável que ele seja inercial.

Portanto, o sistema de referencia inercial é usado para denotar um sistema de referencia que não tem aceleração resultante, alem disso ele é desejado na navegação devido as equações de navegação que se tornam mais simples.

2.3.2 Triedro do Corpo

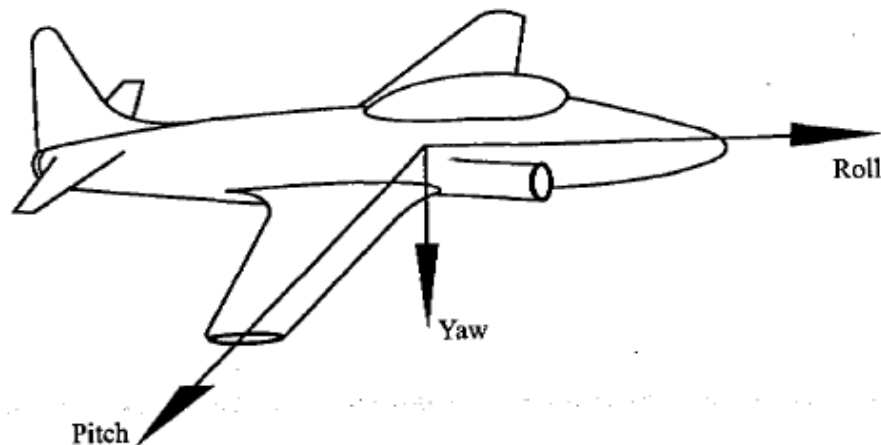
O movimento angular eixos XYZ do sistema de coordenadas de um veículo (SCV) são definidos como Roll, Pitch e Yaw (RPY) respectivamente. O sistema de coordenadas do veiculo da figura 2.7 é caracterizado como *North* (eixo x), *East* (eixo y) e *Down* (eixo z) conhecido como sistema NED. Esse sistema é o mais utilizado e representa a superfície terrestre como um plano cartesiano xy.

Logo, o sistema de coordenadas do veiculo deve ser completamente alinhado com o sistema de coordenadas cartesiano da UMI, visto que os sensores inerciais não serão colocados exatamente nos eixos do sistema de coordenadas do veiculo. Entretanto, esse alinhamento pode ser feito através de referencias mecânicas onde a UMI deve ser posicionado.

2.3.3 Triedro de Local e Global

O sistema de coordenadas local é normalmente anexado a algum ponto de interesse e também pode utilizar o sistema NED.

Como mostrado pela figura 2.8 a disposição dos eixos do SCL define que a

Figura 2.7: Conjunto de eixos de um veículo

Fonte: (TITTERTON; WESTON, 2004b)

origem está no centro de massa do veículo e os eixos são dispostos como o sistema NED determina.

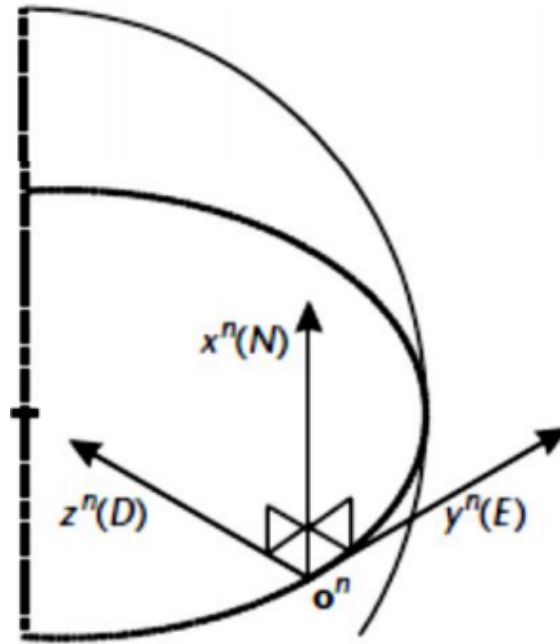
2.4 Alinhamento de Sensores Inerciais

Essencialmente, os métodos de alinhamento se resumem em determinar a orientação do conjunto de eixos ortogonais, definidos pelos eixos de entrada do acelerômetro (Corpo), em relação aos eixos de referência designado (Local), minimizando os efeitos dos ruídos.

Como já mencionado, erros aleatórios no sistema limitam a precisão do SNI alinhado em ambos os métodos. Isso inclui os efeitos dos erros dos sensores inerciais, latência de dados causada por atrasos de transmissão, quantização do sinal, efeitos de vibração e outros movimentos indesejáveis ou não quantificáveis. Por isso, várias técnicas tem sido desenvolvidas para superar os efeitos desses erros aleatórios e sistemáticos.

Existem dois tipos fundamentais de alinhamento: Auto-Alinhamento e o Alinhamento de transferência em relação a um mestre. Logo, esse trabalho se limita em discutir o Auto-Alinhamento.

Figura 2.8: Eixos de referencia Local



Fonte: (GROVES, 2008)

2.5 Análise de Erro: Modelo Quasi-Estacionário

Como visto, há a necessidade de quantificar o erro para aplicar na modelagem do sistema e assim estimar o erro de alinhamento pelo FK. Aqui será considerado um sistema Quasi-Estacionário, ou seja, ele não está em voo e nem em repouso, porém possui pequenas vibrações.

Com isso, esse modelo será aplicado em um exemplo de Auto-Alinhamento com FKI. A diferença entre o FK normal e o indireto é que, na forma da modelagem, como já dito, o indireto estima o erro da variável de estado (Erro de estado) e o normal estima a variável de estados (CHRITENSEN; FOGH, 2008).

Assim, o modelo do sistema é descrito a seguir:

$$\dot{x} = A(t)x(t) + G_p(t)\eta_p(t) \quad (2.3)$$

Tal que, x é o vetor de erro de estado, A é o matriz dinâmica de erro de estado, $G_p(t)$ é o vetor de ruído do processo e $\tau_p(t)$ é o ruído dinâmico. Entretanto, o vetor de erro de estado x é corrompido com ruído, assim o valor medido é na verdade dado pela seguinte equação, tal que z é o valor medido e x o valor real.

$$z(t) = H(t)x(t) + G_m(t)\eta_m(t) \quad (2.4)$$

No FK Indireto é permitido ajustar os estados do sistema através de um processo chamado de "Redefinição de Controle". O erros de estado são corrigidos com a seguinte equação:

$$x(+c) = x(+e) + u_{nc} \quad (2.5)$$

Visto que u_{nc} é o vetor de controle usado para controlar o erro que se acumula nos estados, $x(+c)$ é a designação do estado de erro corrigido após os estados terem sido redefinidos e $x(+e)$ é a designação do estado de erro estimado antes do vetor de controle de redefinição ser aplicado.

A equação de medida 2.4 é uma linearização geral da equação de observação não linear z_{obs} que é definida como mostrado a seguir:

$$z_{obs} = \Delta r_H^n - \Delta r_{refH}^n \quad (2.6)$$

Onde, Δr_{refH}^n é posição de referência para a posição de divergência que é aproximadamente igual a zero devido ao sistema Quasi-Estacionário.

Assim, é necessário desenvolver o modelo do erro do sistema para aplicar no modelo do FK Indireto descrito.

O objetivo do modelo de erro é descrever a propagação dos erros na equações de navegação. Os erros são definidos como o ângulo entre as DCMs reais e as DCMs calculadas e a diferença entre a velocidade real e a velocidade calculada. Os erros a serem definidos são os ângulos de erro associados a DCM calculada pelo alinhamento grosso (γ), os ângulos de erro associados ao DCM de rotação entre o sistema de coordenadas do corpo em relação a Terra (ϵ) e, por último, o erro da velocidade (δv).

As equações simplificadas de navegação que descrevem o movimento de um veículo em relação à terra em um sistema de navegação inercial *strapdown* são dadas a seguir:

$$\dot{C}_b^n = C_b^m \Omega_{ib}^b - (\Omega_{ie}^b + \Omega_{en}^b) C_b^n \quad (2.7)$$

$$\omega_{ie}^n = C_b^m \omega_{ie}^e \quad (2.8)$$

$$\omega_{en}^n = \frac{1}{R}(u_{zn}^n \times v^n) + \rho_R u_{zn}^n \quad (2.9)$$

$$\dot{v}^n = C_b^m a^n + g^n - (\omega_{en}^n + 2\omega_{ie}^n \times v^n) \quad (2.10)$$

$$\dot{C}_e^n = C_e^m \Omega_{en}^n \quad (2.11)$$

$$\dot{h} = v^n u_{zn}^n \quad (2.12)$$

As variáveis das equações de 2.7 até 3.24 consideram as variáveis ideais, entretanto devido aos erros instrumentais dos sensores, deve-se indicar através da notação $\hat{\cdot}$ que as variáveis utilizadas são estimativas dos valores reais.

$$\dot{\hat{C}}_b^n = \hat{C}_b^n \tilde{\Omega}_{ib}^b - (\hat{\Omega}_{ie}^b + \hat{\Omega}_{en}^b) \hat{C}_b^n \quad (2.13)$$

$$\hat{\omega}_{ie}^n = \hat{C}_b^n \hat{\omega}_{ie}^e \quad (2.14)$$

$$\hat{\omega}_{en}^n = \frac{1}{\hat{R}}(u_{zn}^n \times \hat{v}^n) + \hat{\rho}_R u_{zn}^n \quad (2.15)$$

$$\dot{\hat{v}}^n = \hat{C}_b^n \tilde{a}^n + \hat{g}^n - (\hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n \times \hat{v}^n) \quad (2.16)$$

$$\dot{\hat{C}}_e^n = \hat{C}_e^n \hat{\Omega}_{en}^n \quad (2.17)$$

$$\dot{\hat{h}} = \hat{v}^n u_{zn}^n \quad (2.18)$$

Assim, pode-se definir o erro com a notação δ . Para o alinhamento a variável de interesse é a matriz de DCM, assim o erro dela é descrito como:

$$\delta C_b^n = \hat{C}_b^n - C_b^n \quad (2.19)$$

$$\hat{C}_b^n = (I - \Gamma^n) C_b^n \Rightarrow \delta C_b^n = -\Gamma^n C_b^n \quad (2.20)$$

Γ é uma matriz simétrica com os erros γ associados a matriz DCM encontrada no alinhamento grosseiro.

Dessa forma, os ângulos de Euler da matriz DCM diretamente corrigida é obtida da seguinte forma:

$$\phi_k = \arctan \frac{C_{32k}}{C_{33k}} \quad (2.21)$$

$$\theta_k = \arctan \frac{-C_{31k}}{\sqrt{1 - C_{31k}^2}} \quad (2.22)$$

$$\Psi_k = \arctan \frac{C_{21k}}{C_{11k}} \quad (2.23)$$

A análise é executada ao calcular os valores médios verdadeiros, os quais, constituem a melhor estimativa para os valores verdadeiros das grandezas medidas. (VUOLO, 1992) para se reduzir o erro causado pelos ruídos aleatórios dos girômetros na estimativa do alinhamento em azimute, (TITTERTON; WESTON, 2004b) mostra que as leituras dos sensores inerciais devem ser acumuladas durante um intervalo de tempo T e depois convertidas nesse valor médio, para então serem utilizadas nos cálculos. Este tempo T , chamado de tempo de alinhamento em azimute é calculado como:

$$\mathbf{T} = \left(\frac{\mathbf{r}\omega_y 180}{\delta\gamma\Omega \cos \mathbf{L}\pi} \right)^2 \quad (2.24)$$

Onde $\delta\gamma$ é o erro esperado para o alinhamento em azimute e $\mathbf{r}\omega_y$ é a integração do ruído branco (Random Walk Angular) do giroscópio apontado para a direção leste, podendo ser estimado através do desvio padrão de Allan (HAN; KNIGHT, 2009).

2.5.1 Ângulos de Euler

A FGD é aplicada na informação da DCM diretamente, assim é necessário transformar a DCM em rotações entres os eixos x , y e z do sistema de navegação do corpo e do sistema de navegação de referência. Essa representação é melhor para determinar a atitude ou identificar rotações no SNI. A rotação nos eixos Roll, Pitch e Yaw são definidas como rolagem ϕ , elevação θ e arfagem ψ .

Segundo o Teorema sobre sequências de rotações enunciado por Leonard Euler dois sistemas de coordenadas ortogonais e independentes podem ser associados por até três sequências de rotações sobre os eixos de coordenadas desde que não haja duas rotações consecutivas (GRANZIERA, 2006). Assim, conclui-se que a matriz DCM pode ser representada por uma sequência dos Ângulos de Euler.

O sistema de coordenadas de referência e o sistema de coordenadas de um

corpo estão relacionados através dos ângulos de Euler pela matriz DCM (Matriz de atitude ou Matriz de Rotação).

$$\hat{C}_B^N = \mathbf{R} = \mathbf{R}_\phi^x \mathbf{R}_\theta^y \mathbf{R}_\psi^z \quad (2.25)$$

onde,

$$\mathbf{R}_\phi^x = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{vmatrix} \quad (2.26)$$

$$\mathbf{R}_\theta^y = \begin{vmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{vmatrix} \quad (2.27)$$

$$\mathbf{R}_\psi^z = \begin{vmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (2.28)$$

$$\hat{C}_B^N = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{vmatrix} \begin{vmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{vmatrix} \begin{vmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (2.29)$$

$$\hat{C}_B^N = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{vmatrix} \begin{vmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\sin \psi & \cos \psi & 0 \\ \sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta \end{vmatrix} \quad (2.30)$$

$$\hat{C}_B^N = \begin{vmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\ \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{vmatrix} \quad (2.31)$$

Nota-se que há um limite para os ângulos de Euler

$$\psi = [-\pi, \pi] \theta = \left[\frac{-\pi}{2}, \frac{\pi}{2} \right] \phi = [-\pi, \pi] \quad (2.32)$$

Segundo Oliveira e Leite Filho (2013), através da ferramenta piograma (Pio, 1966), pode-se exprimir os ângulos de Euler em função da DCM:

$$\psi = \arcsin(c_{12}) \quad (2.33)$$

$$\theta = \arctan \left(\frac{-c_{13}}{c_{11}} \right) \quad (2.34)$$

$$\phi = \arctan \left(\frac{-c_{32}}{c_{22}} \right) \quad (2.35)$$

Assim,

$$\psi = \arcsin \left(\frac{f_y}{g} \right) \quad (2.36)$$

$$\theta = \arctan \left(\frac{-f_z}{f_x} \right) \quad (2.37)$$

$$\phi = \arctan \left(\frac{f_x^2 \omega_y - f_x f_y \omega_x - f_y f_z \omega_z + f_z^2 \omega_y}{g(f_z \omega_x - f_x \omega_z)} \right) \quad (2.38)$$

Com isso, na FKD os ângulos de Euler descritos aqui são encontrados e filtrados diretamente. Logo, ao aplicar esse modelo no Filtro de Kalman estendido, verifica-se que os erros dos ângulos de Euler converge para zero.

2.5.2 Quatérnions

Quatérnions são vetores utilizados para calcular rotações em mecânica aeroespacial, computação gráfica, processamento de visão, e outras aplicações. São formados por quatro elementos, tal que três são números imaginários e um define a amplitude de rotação. Os quatérnions são comumente denotado por:

$$\mathbf{q} = w + x.i + y.j + z.k \quad (2.39)$$

onde:

$$i^2 = j^2 = k^2 = i * j * k = -1 \quad (2.40)$$

Esse formato possui uma eficiência computacional maior que a matriz de rotação comum, por isso é preferencialmente utilizado em aplicações que a matriz de transformação é utilizada.

2.5.3 Desvio Padrão de Allan

O desvio padrão de Allan (ad) é uma técnica de análise no domínio do tempo desenvolvida para ser aplicada a qualquer sinal sob o qual se deseja determinar ruídos de processos ocultos (OLIVEIRA; Leite Filho, 2013). O ad pode ser determinado dividindo-se a sequência original de dados em n partições de comprimento temporal τ ($n = 9$), e extraído-se, em seguida, o valor médio de cada partição, isto é, $\mathbf{x}_1(\tau)$, $\mathbf{x}_2(\tau)$, $\mathbf{x}_3(\tau)$, ..., $\mathbf{x}_n(\tau)$. A partir dos valores médios de cada partição, o desvio padrão de Allan é calculado como:

$$\mathbf{ad}(\tau) = \sqrt{\frac{\sum(\mathbf{x}_n(\tau) - \mathbf{x}_{n-1}(\tau))^2}{2(n-1)}} \quad (2.41)$$

Através do gráfico do desvio padrão com todos os eixos na escala logarítmica, diversos tipos de processos aleatórios produzem curvas com diferentes gradientes de inclinação (HAN; KNIGHT, 2009).

Sabendo o tipo de processo aleatório, é possível obter seu valor numérico diretamente por análise gráfica. Para o caso específico dos girômetros, o valor do Randow Walk angular é obtido pela interseção da curva do desvio padrão de Allan com a linha perpendicular ao eixo das ordenadas no tempo de partição $\tau = 1$. Uma vez estimado o valor do Randow Walk angular do giroscópio, pode-se utilizar a equação 2.41, para estimar o tempo necessário para um erro convergir para $\delta\gamma$.

2.6 Alinhamento

O SNI é um sistema *Dead-Recnoking* que fornece a posição, velocidade e a informação da atitude, sendo assim ele deve ser inicializado antes da navegação. Essa inicialização é uma parte crucial na operação do sistema, já que nela é definida, *a priori*, a velocidade inicial e a posição inicial, para assim determinar a atitude do SNI. Logo, a parte mais difícil desse procedimento é a determinação da atitude, conhecida nesse caso como **Alinhamento** (BAR-ITZHACK, 1990).

Como mencionado anteriormente um SNI simplesmente integra a aceleração medida e a velocidade de rotação para determinar a posição do chassi. Mas, a posição é sempre relativa a posição de partida e, por este motivo, o SNI precisa saber tanto a posição no corpo como a posição antes da navegação.

A posição antes da navegação é conhecida, entretanto a atitude tem de ser de-

terminada, ou seja, deve-se fazer o alinhamento. A ideia por trás do alinhamento é usar as medições dos acelerômetros e girômetros para determinar a orientação da estrutura do corpo com respeito a uma referência (Coordenadas de navegação).

Por isso, o objetivo do alinhamento é determinar os ângulos ou a matriz de rotação entre um sistema de coordenadas de referência local e o sistema de coordenadas do corpo. No sistema *Strapdown*, a informação da Attitude³ pode ser armazenada como uma matriz de cossenos DCM (Direct Cossine Matrix) ou conjunto de quatérnions.

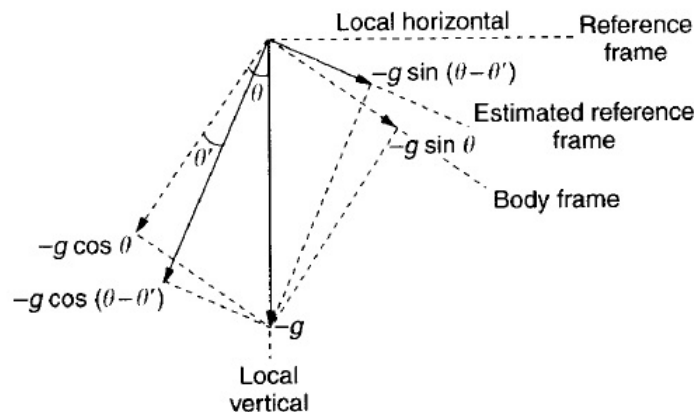
2.6.1 Alinhamento Plataforma Fixa

Aqui um exemplo prático de alinhamento em uma plataforma fixa em um único plano é descrito para facilitar a compreensão.

Os acelerômetros fornecem medidas da aceleração verdadeira no sistema de coordenadas do corpo, $-g \sin(\theta)$ e $-g \cos(\theta)$. O valor θ , ângulo entre o corpo e a referência, pode ser encontrado com o seguinte procedimento: Estima-se um valor θ' para o ângulo que o corpo faz com a referência. Assim, dada que a equação da gravidade decomposta no eixo x seja $g_x = -g \sin(\theta - \theta')$, quando o ângulo $\theta' = \theta$, a gravidade em x é $g_x = 0$.

Portanto, θ' pode ser ajustado até que $\theta' = \theta$. Essa ideia pode ser avaliada na figura 2.9.

Figura 2.9: Alinhamento do vetor gravidade é um único plano.



Fonte: (TITTERTON; WESTON, 2004b)

No segundo passo para executar o alinhamento, é necessário determinar a

³Orientação de um veículo ou outro objeto em relação ao horizonte, direção do movimento e etc.

posição ou orientação azimutal ⁴ do corpo em relação ao plano horizontal, isto é, determinar a direção em relação ao norte verdadeiro. Isso pode ser feito ao conhecer os verdadeiros componentes da velocidade angular da Terra no sistema de coordenadas local (TITTERTON; WESTON, 2004b).

A equação da velocidade angular é dada por $\omega_E = \Omega \cos(\psi - \psi')$, quando o ângulo $\psi' = \psi$, a velocidade na direção x é $\omega_E = 0$.

Com isso, assumindo que os girômetros possuem precisão suficiente para detectar a velocidade angular da Terra, de forma análoga ao esquema anterior, a informação armazenada da atitude é agora ajustada até que o componente da velocidade angular, medida na direção leste, se reduz a zero (TITTERTON; WESTON, 2004b).

2.7 Auto-Alinhamento

O Auto-Alinhamento é um método de alinhamento que utiliza a velocidade de erro nas coordenadas norte e leste (V_N e V_E) para estimar o erro de desalinhamento para ajustar as coordenadas do SNI através de uma malha fechada. (BARITZHACK, 1990).

Normalmente há duas etapas no processo de Auto-Alinhamento: O Auto-Alinhamento fino e o Auto-Alinhamento Grosseiro (SAVAGE, 2007b). O Alinhamento grosseiro é um alinhamento analítico que sofre com a falta de precisão devido aos erros de medição e de outras fontes de ruído. Dessa forma, um alinhamento mais preciso (Alinhamento fino) pode ser executado com um filtro. (BRITTING, 1971) .

O Auto-Alinhamento grosseiro é feito através da medição dos vetores gravidade e velocidade angular da Terra para encontrar valores aproximados da atitude. Portanto, o Auto-Alinhamento fino é feito após o grosseiro para corrigir esses valores.

Este capítulo irá explicar o processo de Auto-Alinhamento básico. A primeira seção irá descrever o alinhamento grosseiro. A segunda seção descreve a procedimento de alinhamento fino utilizando um filtro de Kalman direto.

⁴Medida angular entre o norte geográfico até a projeção de um alvo com o horizonte

2.7.1 Alinhamento Grosseiro: Abordagem Tradicional

Define-se aqui os eixos de navegação x_N , y_N e z_N apontando, respectivamente, para as direções vertical local, leste e norte (configuração UEN - Up, East, North) e o veículo é suposto estar no hemisfério sul (OLIVEIRA; Leite Filho, 2013). O Auto-Alinhamento grosseiro tradicional, como já dito, é feito através das medições de três vetores:

- Vetores gravidade \vec{g} ;
- Velocidade angular da Terra $\vec{\omega}_E$;
- Produto Vetorial $\vec{\omega}_E \times \vec{g}$.

Assim, encontra-se esses vetores facilmente ao decompor essas grandezas no triedro x,y,z no sistema de coordenadas local (Inercial), ou seja, no triedro de navegação (Índice N). Ao considerar que L , g e Ω são, respectivamente, o valor da latitude, magnitude da gravidade e magnitude da velocidade angular da Terra, temos que:

$$\vec{g}^N = \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix} \quad (2.42)$$

$$\vec{\omega}_E^N = \begin{bmatrix} \Omega \sin(L) \\ 0 \\ \Omega \cos(L) \end{bmatrix} \quad (2.43)$$

$$\vec{\omega}_E^N \times \vec{g}^N = \begin{bmatrix} 0 \\ \vec{g}\Omega \cos(L) \\ 0 \end{bmatrix} \quad (2.44)$$

No sistema de coordenadas do corpo, ou seja, no triedro do corpo (Índice B), ao considerar que f_i e ω_i são, respectivamente, o valor da força específica medida pelos acelerômetros no eixo i e a velocidade angular medida pelo girômetro no eixo i , temos que:

$$\vec{g}^B = \begin{bmatrix} -f_x \\ -f_y \\ -f_z \end{bmatrix} \quad (2.45)$$

$$\vec{\omega}_E^B = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.46)$$

$$\vec{\omega}_E^B \times \vec{g}^B = \begin{bmatrix} f_z \omega_y - f_y \omega_z \\ f_x \omega_z - f_z \omega_x \\ f_y \omega_x - f_x \omega_y \end{bmatrix} \quad (2.47)$$

Os dois sistemas de coordenadas r^B e r^N , relacionam-se através de uma matriz de transformação C_B^N (DCM) da seguinte forma, dado que:

$$r^B = \begin{bmatrix} \vec{g}^B \\ \omega_E^B \\ \omega_E^B \times \vec{g}^B \end{bmatrix} \quad (2.48)$$

$$r^N = \begin{bmatrix} \vec{g}^N \\ \omega_E^N \\ \omega_E^N \times \vec{g}^N \end{bmatrix} \quad (2.49)$$

então:

$$r^N = C_B^N r^B \quad (2.50)$$

Assim, a matriz de alinhamento é definida desde que a sua matriz inversa exista, ou seja, se nem uma das linhas for combinação linear das outras. Isso é garantido, já que os vetores \vec{g} e $\vec{\omega}_E$ são não colineares, a não ser nos polos onde o alinhamento analítico torna-se impossível, como será mostrado a seguir. (OLIVEIRA; Leite Filho, 2013).

Desde que $C_B^N = (C_N^B)^T = (C_N^B)^{-1}$, então é possível escrever que:

$$C_B^N = (r_n^T)^{-1} r_b^T \quad (2.51)$$

Dessa forma, encontra-se que:

$$C_B^N = \begin{bmatrix} c_{11} c_{12} c_{13} \\ c_{21} c_{22} c_{23} \\ c_{31} c_{32} c_{33} \end{bmatrix} \quad (2.52)$$

Tal que:

$$c_{11} = \frac{f_x}{g} \quad (2.53)$$

$$c_{12} = \frac{f_y}{g} \quad (2.54)$$

$$c_{13} = \frac{f_z}{g} \quad (2.55)$$

$$c_{21} = \frac{f_z \omega_y - f_y \omega_z}{g \Omega \cos L} \quad (2.56)$$

$$c_{22} = \frac{f_x \omega_z - f_z \omega_x}{g \Omega \cos L} \quad (2.57)$$

$$c_{23} = \frac{f_y \omega_x - f_x \omega_y}{g \Omega \cos L} \quad (2.58)$$

$$c_{31} = \frac{g \omega_x - f_x \Omega \sin L}{g \Omega \cos L} \quad (2.59)$$

$$c_{32} = \frac{g \omega_y - f_y \Omega \sin L}{g \Omega \cos L} \quad (2.60)$$

$$c_{33} = \frac{g \omega_z - f_z \Omega \sin L}{g \Omega \cos L} \quad (2.61)$$

Com isso, verifica-se pelas equações de 3.11 até 3.19 que essa DCM é definida unicamente pelas medidas dos sensores inerciais em uma latitude diferente de 90° . Isso significa que o procedimento desse alinhamento não é possível em um navegador inercial posicionado nos polos norte e sul.

A análise de erro pode ser encontrada em (OLIVEIRA; Leite Filho, 2013).

2.7.2 Alinhamento Grosseiro: Abordagem Alternativa

Uma abordagem alternativa é analisada por (JIANG, 1998) e (OLIVEIRA; Leite Filho, 2013), semelhante a metodologia tradicional apresentada anteriormente, também se baseia na medição de três vetores não colineares resultantes da combinação dos vetores gravidade e velocidade angular da terra.

Esta metodologia, todavia, não utiliza a tríade $\vec{g}, \vec{\omega}_E^N$ e $\vec{g}^N \times \vec{\omega}_E^N$, mas sim a tríade $\vec{g}, \vec{g}^N \times \vec{\omega}_E^N$ e $(\vec{g}^N \times \vec{\omega}_E^N) \times \vec{g}^N$. (OLIVEIRA; Leite Filho, 2013) compararam os resultados das duas abordagens e a única diferença entre as DCM obtida pela metodologia tradicional e a alternativa recaem sobre a terceira linha de cada uma delas.

Logo, encontra-se que:

$$C_B^N = \begin{bmatrix} c_{11} c_{12} c_{13} \\ c_{21} c_{22} c_{23} \\ c_{31} c_{32} c_{33} \end{bmatrix} \quad (2.62)$$

Tal que:

$$c_{11} = c'_{11} \quad (2.63)$$

$$c_{12} = c'_{12} \quad (2.64)$$

$$c_{13} = c_{13} \quad (2.65)$$

$$c_{21} = c'_{21} \quad (2.66)$$

$$c_{22} = c'_{22} \quad (2.67)$$

$$c_{23} = c'_{23} \quad (2.68)$$

$$c'_{31} = \frac{f_z^2 \omega_x - f_x f_z \omega_z - f_x f_y \omega_y + f_y^2 \omega_x}{g^2 \Omega \cos L} \quad (2.69)$$

$$c'_{32} = \frac{f_x^2 \omega_y - f_x f_y \omega_x - f_y f_z \omega_z + f_z^2 \omega_y}{g^2 \Omega \cos L} \quad (2.70)$$

$$c'_{33} = \frac{f_y^2 \omega_z - f_y f_z \omega_y - f_x f_z \omega_x + f_x^2 \omega_z}{g^2 \Omega \cos L} \quad (2.71)$$

Segundo Oliveira e Leite Filho (2013) a DCM calculada pela metodologia

tradicional apresenta erros de ortogonalidade. Entretanto, estes erros não estão presentes na DCM calculada pela metodologia alternativa, além disso (OLIVEIRA; Leite Filho, 2013) constatou que ao tentar ortogonalizar a DCM pela metodologia tradicional os erros de ortogonalidade eram reduzidos, entretanto os erros de alinhamento eram aumentados. Com isso, a metodologia alternativa, mostrou-se melhor.

2.7.3 Alinhamento Fino: Abordagem FKD

O alinhamento fino, resume-se em determinar ou estimar a atitude fornecida pelo alinhamento grosseiro. Atualmente, isso pode ser feito de diversas formas, uma delas é por "Filtragem Direta". Entretanto, a abordagem mais comum é determinar ou estimar os **erros** de alinhamento Ψ e utiliza-los para corrigir a atitude fornecida pelo alinhamento grosseiro (BAR-ITZHACK, 1990), sendo essa uma ideia semelhante a descrita na seção 2.6.1. Em outras palavras, usa-se a estimativa do erro para corrigir recursivamente o sistema através de uma "Filtragem Indireta".

Em ambos os casos o Filtro de Kalman é normalmente usado. O primeiro caso é mais simples e conhecido como Filtragem de Kalman Direta (FKD). O segundo é mais complexo, devido ao trabalho extra em derivar um modelo de erro e é conhecido como Filtragem de Kalman Indireta (FKI) (CHRITENSEN; FOGH, 2008).

Segundo Chritensen e Fogh (2008), o motivo para a FKI ser normalmente aplicada pode ser atribuído a elevação da complexidade computacional é aumentada devido a dois fatores:

- A FKD exige que o processador faça cálculos de Kalman antes do próximo ciclo do microprocessador.
- A FKD em um sistema não linear deve ser realizada com um filtro não linear.

Como mencionado anteriormente, o uso de filtragem Kalman indireta requer o modelo de erro no sistema. Este modelo de erro pode ser aproximado para ser linear. Por estas razões, um filtro de Kalman linear normal pode ser usado com um ciclo de Kalman muito menor definindo exigências menores para o processador. Por isso, como o modelo do SNI é não linear e tem uma dinâmica elevada, a FKI é geralmente escolhida.

Ao usar a FKI, é necessário executar o algoritmo somente quando novos dados estão disponíveis, sendo esse outro motivo para a FKI ter uma aplicação maior, já que o custo computacional diminui.

2.7.4 Alinhamento Fino por Filtragem Indireta

O alinhamento fino é a parte mais difícil do processo de Auto-Alinhamento (BAR-ITZHACK, 1990), por isso esse capítulo será dedicado exclusivamente para descrever detalhadamente a forma clássica de executar essa etapa: A Filtragem Indireta.

Inicialmente para facilitar o entendimento será descrito o alinhamento em repouso (*Alignment at Rest* **AAR**), já que ele é mais simples que o alinhamento em voo (*Alignment at Flight*) **AAF**.

A seguir será descrito a abordagem clássica, na qual os ângulos de desalinhamento são determinados diretamente e a sua derivação na qual os ângulos são estimados pelo Filtro de Kalman (FK). Em seguida, será descrito o Alinhamento Quasi-Estacionário que será utilizado para simular um exemplo.

2.7.5 Alinhamento Fino: Abordagem Clássica

A gravidade da Terra constitui um sinal forte em relação aos ruídos gerados pelos acelerômetros nessa componente. Por outro lado, a velocidade angular da Terra tem uma magnitude muito pequena em relação aos ruídos gerados pelos girômetros na mesma componente.

Dessa forma, para extrair o sinal dos girômetros é necessário acumular a saída dos girômetros durante algum tempo. Usualmente, nota-se que durante o processo de alinhamento, ao rodar o SNI em relação a coordenada de referência, o alinhamento dura mais quanto maior o erro gerado pelos ruídos (BAR-ITZHACK, 1990).

Assim, esse é um problema dinâmico, cuja a natureza é descrita pelas equações diferenciais. Ao colocar na representação de variáveis de estado:

$$\dot{x} = z_t x_t + \beta_t + \eta_t \quad (2.72)$$

Ou na forma de equações de diferenças

$$x_{n+1} = z_n x_n + \beta_n + \eta_n \quad (2.73)$$

Assim, essas equações diferenciais que descrevem adequadamente o comportamento não amortecido do erro do SNI durante o alinhamento, é reescrita da seguinte forma.

$$\frac{d}{dx} \begin{bmatrix} V_N \\ V_E \\ V_D \\ \Psi_N \\ \Psi_E \\ \Psi_D \end{bmatrix} = \begin{bmatrix} 0 & 2W_D & 0 & 0 & g & 0 \\ -2W_D & 0 & 2W_N & -g & 0 & 0 \\ 0 & 2W_N & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_D & 0 \\ 0 & 0 & 0 & -W_D & 0 & W_N \\ 0 & 0 & 0 & 0 & W_N & 0 \end{bmatrix} \begin{bmatrix} V_N \\ V_E \\ V_D \\ \Psi_N \\ \Psi_E \\ \Psi_D \end{bmatrix} + \begin{bmatrix} B_N \\ B_E \\ B_D \\ D_N \\ D_E \\ D_D \end{bmatrix} + \begin{bmatrix} \eta_{BN} \\ \eta_{BE} \\ \eta_{BD} \\ \eta_{DN} \\ \eta_{DE} \\ \eta_{DD} \end{bmatrix} \quad (2.74)$$

O equacionamento pode ser deduzido em função de diversos padrão de coordenadas. Como exemplo, o equacionamento anterior foi desenvolvido utilizando as coordenadas *North*, *East* e *Down* que são representadas pelos Os índices N, E e D, respectivamente. A variável ϵ é o erro dos sensores, η é o ruído dos sensores e g a gravidade da Terra. Tal que, assumi-se que os ruído é gaussiano.

A variável de estado \vec{x} contem as componentes a seguir:

- V_N Componente *North* da velocidade do erro;
- V_E Componente *East* da velocidade do erro;
- V_D Componente *Down* da velocidade do erro;
- Ψ_N Componente *North* do desalinhamento;
- Ψ_E Componente *East* do desalinhamento;
- Ψ_D Componente *Down* do desalinhamento.

além disso,

- B_N Bias do acelerômetro *North*;
- B_E Bias do acelerômetro *East*;
- B_D Bias do acelerômetro *Down*;
- D_N Constante de Drift do girômetros *North*;

- D_E Constante de Drift do girômetros *East*;
- D_D Constante de Drift do girômetros *Down*.

- η_{BN} Ruído branco do acelerômetro *North*;
- η_{BE} Ruído branco do acelerômetro *East*;
- η_{BD} Ruído branco do acelerômetro *Down*;
- η_{DN} Ruído branco do girômetros *North*;
- η_{DE} Ruído branco do girômetros *East*;
- η_{DD} Ruído branco do girômetros *Down*.

A partir da equação 2.74 obtemos as seguintes equações:

$$\dot{v}_N = g\psi_E + B_N + \eta_{BN} \quad (2.75)$$

$$\dot{v}_E = -g\psi_N + B_E + \eta_{BE} \quad (2.76)$$

$$\dot{\psi}_N = \omega_D\psi_E + D_N + \eta_{DN} \quad (2.77)$$

$$\dot{\psi}_E = -\omega_D\psi_N + \omega_N\psi_D + D_E + \eta_{DE} \quad (2.78)$$

$$\dot{\psi}_D = -\omega_N\psi_E + D_D + \eta_{DD} \quad (2.79)$$

Os sinais utilizados para indicar os ângulos de desalinhamento são as velocidades de erros V_N e V_E . Essas variáveis são obtidos pela leitura de componentes N e E dos sensores de navegação inercial então, para que o SNI seja adequado essas medidas devem ser idealmente iguais a zero. Com isso, qualquer valor diferente de zero nas leituras é considerado erro.

Por inspeção da equação 2.75 e da equação 2.76 é evidente que no caso ideal, onde os ruídos gaussianos η são ignorados, verifica-se que para $V_E = 0$ e $V_N = 0$. Isso implica que $\Psi_E = 0$ e $\Psi_N = 0$ (BAR-ITZHACK, 1990).

Logo, fica claro que o valor Ψ_E e Ψ_N sã indicados pelos valores V_N e V_E . Assim, os ângulos de rotação adequados do SNI sobre o *North* e *East* ocorrem quando $V_N = 0$ e $V_E = 0$. Esse procedimento é denominado **Leveling**.

A partir da equação 2.77 e da equação 2.79, verifica-se que ao zerar Ψ_N o Ψ_D não zera, assim Ψ_D ficará mudando, embora ele possa ser temporariamente eliminado como um resultado de nivelamento ou *Leveling*.

Portanto, para efetuar o alinhamento completo é necessário rotacionar o sistema de coordenadas do corpo, sobre os três eixos de coordenadas até que $V_N = 0$ e $V_E = 0$. Assim, durante um intervalo de tempo, o SNI utiliza V_N e V_E gerados e uma malha com realimentação para ajustar os três eixos o sistema de coordenada do SNI até zerar V_N e V_E (BAR-ITZHACK, 1990).

No caso ideal onde o erro dos acelerômetros e girômetros são zero, assim como V_N e V_E são zero, pela equação 2.75 a equação 2.79, os erros de desalinhamento são zero. Entretanto, uma vez que os acelerômetros e giros não são ideais, os erros do desalinhamento alcançam um valor diferente de zero. Assim é evidente que o limite inferior do desalinhamento é:

$$\psi_N = \frac{-B_E}{g} \quad (2.80)$$

$$\psi_E = \frac{B_N}{g}$$

$$\psi_D = \frac{-D_E}{\omega_N} \quad (2.82)$$

Logo, esse método usando V_N e V_E para ajustar as coordenadas do SNI através de uma malha fechada é conhecido como **Auto-Alinhamento** (BAR-ITZHACK, 1990).

2.7.6 Alinhamento Fino: Abordagem FKI

Nessa abordagem, ao invés de determinar os ângulos de desalinhamento diretamente, um FK irá estimar essas variáveis através das medidas de V_N e V_E .

Com a introdução do FK nos meados dos anos sessenta, percebeu-se que isso poderia ser usado com sucesso para alinhamento do SNI (BAR-ITZHACK, 1990).

Na abordagem com FK, o alinhamento fino é dividido em duas partes: **Es-**

timação e o **Ajuste** (BAR-ITZHACK, 1990).

Na primeira parte estima-se o desalinhamento dos ângulos baseado nas medidas V_N e V_E e na segunda é efetuado o ajuste das coordenadas do SNI para eliminar os ângulos de desalinhamento. Normalmente, as coordenadas do SNI são ajustadas continuamente com o objetivo de manter elas o mais próximo possível da atitude em relação a orientação desejada durante o processo de estimação.

Para um processo de alinhamento em um curto espaço de tempo, pode-se considerar $\dot{\epsilon} = 0$ (Ruído gaussiano), assim o modelo de erro utilizado é:

$$\dot{x} = Ax + \eta \quad (2.83)$$

Logo, a matriz dinâmica A e a variável x conterá ambos os sistema dinâmicos, ou seja o erro ϵ , as velocidades dos erros \vec{V} e os desalinhamentos $\vec{\Psi}$. Assim é possível reescrever a equação como:

$$\frac{d}{dx} \begin{bmatrix} V_N \\ V_E \\ V_D \\ \Psi_N \\ \Psi_E \\ \Psi_D \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 2W_D & 0 & 0 & g & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2W_D & 0 & 2W_N & -g & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2W_N & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_D & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -W_D & 0 & W_N & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & W_N & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_N \\ V_E \\ V_D \\ \Psi_N \\ \Psi_E \\ \Psi_D \\ B_N \\ B_E \\ B_D \\ D_N \\ D_E \\ D_D \end{bmatrix} + \begin{bmatrix} \eta_{BN} \\ \eta_{BE} \\ \eta_{BD} \\ \eta_{DN} \\ \eta_{DE} \\ \eta_{DD} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.84)$$

Portanto, o FK pode ser utilizado para estimar os ângulos de desalinhamento para fazer o **Auto-Alinhamento**, mas também pode ser utilizado para estimar os outros parâmetros, como velocidade de erro, os erros de *Bias* do acelerômetro, e os erros de *Drift* dos girômetro. A estimação dessas fontes de erros dos acelerômetro e dos girômetros e a sua remoção é conhecida como **Auto-Calibração**.

O FK não aceita modelos dinâmicos com forças motrizes com ruído com média diferente de zero. Essa condição é uma qualidade disfarçada de defeito, visto que ela nos força a incluir as fontes de erros no modelo para estima-los e removê-los.

O ângulo de desalinhamento mínimo descrito na abordagem clássica também existe na abordagem com FK. Considerando a equação 2.75 e que estamos lidando com o ângulos mínimos, nós vamos assumir que todos os desalinhamentos exceto Ψ_E , assim como D_E são zero. O FK processa V_N e obtém uma estimativa de Ψ_E . Dessa forma:

$$g\Psi_E \neq -B_N \quad (2.85)$$

Agora V_N não será constantemente zero e FK estimará continuamente Ψ_E . Mas, quando a inequação 2.85 virar uma equação, V_N será uma integral de ruídos gaussianos. Nesse caso, Ψ_E seria dado pela equação 2.76 novamente.

A partir das equação 2.75 até a equação 2.79, percebe-se que os ângulos de desalinhamento que têm um efeito imediato e visível no desenvolvimento de V_N e V_E são Ψ_E e Ψ_N , respectivamente, pois eles são multiplicados por g que é um sinal forte. Por isso, a melhor parte desse desalinhamento é estimada imediatamente e eliminadas através do ajuste.

A influência de Ψ_D e V_N é sentida somente em um estágio posterior e depois da integração. Nota-se que, assim como a gravidade amplifica os desalinhamentos *East* e *North* o desalinhamento *Down* é atenuado pela velocidade angular da Terra. Com isso, o processo descrito acima para estimar Ψ_D através de V_N é conhecido como ***Gyrocompassing***. O *Gyrocompassing* é um fator limitante da velocidade na qual o alinhamento em repouso pode ser executado.

2.8 Estado da Arte

Nesta seção, um resumo das principais técnicas discutidas para solucionar o problema de auto-alinhamento é realizado.

Liu et al. (2015) indicou um algoritmo de alinhamento grosseiro usando vetores de movimento aparente gravitacional em três momentos diferentes e operação vetorial. Este método mostrou tolerar o ruído aleatório do acelerômetro.

Em uma das etapas, essa abordagem precisa de um design tradicional de filtro kalman. Dessa forma, o algoritmo mostrou uma melhora significativa no alinhamento completo ao melhorar o alinhamento grosseiro.

Um alinhamento analítico chamado método seletivo foi proposto por Tan et al. (2015). Tradicionalmente, o alinhamento analítico grosso usa seis saídas da unidade de medição inercial (IMU) para calcular a atitude. A principal contribuição do método seletivo é que ele usa três unidades de medida.

Um método de auto-alinhamento para SINS baseado no q-método foi estudado por Xu et al. (2017b). Essa abordagem é interessante, visto que o algoritmo REQUEST é inspirado no QUEST que é derivado do q-método. Um método melhorado baseado na integração de movimento aparente gravitacional para formar velocidade aparente é projetado. Com isso, Nesta abordagem foi mostrado que a taxa de convergência se torna mais rápida.

Recentemente, um alinhamento grosso usando REQUEST foi proposto por Zhu, Zhang e Xu (2018) e por Campos et al. (2017). O artigos mostram que o algoritmo REQUEST proposto tem algumas vantagens em relação ao algoritmo tradicional de alinhamento grosseiro.

Zhu, Zhang e Xu (2018) pesquisou o algoritmo REQUEST paralelamente ao trabalho apresentado nessa dissertação e confirmou as vantagens encontradas nessa pesquisa.

Finalmente, neste trabalho, uma melhoria do algoritmo REQUEST usando redes neurais artificiais é proposta para alcançar uma melhor taxa de convergência.

3 Auto-Alinhamento Tradicional

Neste capítulo, o auto-alinhamento tradicional usando o método TRIAD para determinar a atitude da nave espacial a partir de duas medidas de vetores é investigada. A fim de melhorar a estabilidade do auto-alinhamento tradicional, um algoritmo REQUEST baseado em método melhorado é proposto na seção 4.

Além disso, a metodologia experimental também é descrita a seguinte na seção 3.1.

3.1 Metodologia Experimental

No alinhamento fino tradicional, o alinhamento grosseiro é fundamental para alcançar um alto desempenho. Se a estimativa de alinhamento grosseiro for próxima do valor real, a taxa de convergência do alinhamento fino aumentará e o tempo total de alinhamento pode ser reduzido (SHEN et al., 2016; LÜ; XIE; CHEN, 2009; XU et al., 2017a).

Portanto, neste trabalho, um método de alinhamento grosso baseado no algoritmo REQUEST com rede neural é proposto para alcançar um alinhamento grosseiro mais rápido e preciso e, conseqüentemente, diminuir o tempo total de alinhamento. Além disso, um estimador de *bias* é proposto para melhorar a acurácia do algoritmo REQUEST.

3.2 Dados Reais

O experimento foi realizado no Instituto de Aeronáutica e Espaço (IAE). O sistema de navegação inercial *strapdown* SNIS utilizado nos testes foi construído pelo IAE e inclui três giroscópios de alta precisão. A precisão do giroscópio é 0,01 [°/h], e a precisão do acelerômetro é de 100 μ g.

Para a comparação prática de métodos, dados de um sistema de navegação

inercial comercial acoplado a um veículo de lançamento por satélite foram empregados. No total, 56000 amostras com uma taxa de amostragem de 100 Hz.

Os dados foram obtidos de um veículo no hemisfério sul, e o sistema de coordenadas considerado abaixo é o *Up, East e Down*.

3.3 Simulation Data

Apenas o *bias* e o ruído na UMI foram considerados para gerar as medidas inerciais simuladas. O sistema de navegação inercial foi considerado estático.

Inspirado no experimento realizado por Xu et al. (2017a), os valores dos vários parâmetros para gerar as medidas inerciais foram escolhidos da seguinte forma: $L = 32^\circ$, $g = 9.8m/s^2$, $\omega_{ie} = 7.2921158 \times 10^{-5}rad/s$. Os vetores são definidos como:

- $[roll, pitch, yaw] = [10^\circ, 10^\circ, 10^\circ]^T$
- $[\nabla_x, \nabla_y, \nabla_z] = [10^{-4}, 10^{-4}, 10^{-4}]^T \times g$
- $[\epsilon_x, \epsilon_y, \epsilon_z] = [10^{-2}, 10^{-2}, 10^{-2}]^T [^\circ/h]$
- $[\sigma_{f_z}, \sigma_{f_y}, \sigma_{f_x}] = [5 \times 10^{-5}, 5 \times 10^{-5}, 5 \times 10^{-5}] \times g$
- $[\sigma_{\omega_z}, \sigma_{\omega_y}, \sigma_{\omega_x}] = [5 \times 10^{-5}, 5 \times 10^{-5}, 5 \times 10^{-5}] [^\circ/h]$.

Onde ∇_i indica o *Bias* do acelerômetro eixo-i, ϵ_i é o *drift* do giroscópio axis-i, σ_{f_i} é o desvio padrão do ruído branco para o acelerômetro no eixo i σ_{ω_i} é o desvio padrão do ruído branco para o giroscópio do eixo i.

A taxa de amostragem foi de 100 Hz, tal que a força específica e a rotação nos eixos x y e z foram denotadas como f_x, w_x, f_y, w_y, f_z e w_z , respectivamente.

Em seguida, assim como foi assumido por (XU et al., 2017a) os valores de $[roll, pitch, yaw]$ são definidos como $[0^\circ, 0^\circ, 0^\circ]$, visto que ocorrem comumente na prática e os valores dos outros parâmetros ficaram inalterados.

3.4 Alinhamento Grosseiro

O método TRIAD, as medidas dos dois vetores são a gravidade local e a taxa de rotação da Terra. Esses vetores são dados pelas equações 3.1 e 3.2, respectivamente. Onde g, Ω, L é a gravidade local, rotação da Terra e latitude local.

$$\mathbf{g}^N = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T \quad (3.1)$$

$$\boldsymbol{\omega}_E^N = \begin{bmatrix} \Omega_N & 0 & \Omega_D \end{bmatrix}^T \quad (3.2)$$

É importante notar que $\Omega_N = \Omega \cos(L)$ e $\Omega_D = -\Omega \sin(L)$.

Onde f_i e ω_i indicam os elementos do eixo i da força específica e da taxa angular na estrutura do corpo. Assim, o vetor de aceleração e a taxa de rotação do corpo são obtidos diretamente pela Equações 3.3 e 3.4.

$$\mathbf{g}^B = \begin{bmatrix} -f_x & -f_y & -f_z \end{bmatrix} \quad (3.3)$$

$$\boldsymbol{\omega}_E^B = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix} \quad (3.4)$$

Assim, a equação 3.5 pode ser obtida diretamente.

$$\boldsymbol{\omega}_E^B \times \mathbf{g}^B = \begin{bmatrix} f_z \omega_y - f_y \omega_z \\ f_x \omega_z - f_z \omega_x \\ f_y \omega_x - f_x \omega_y \end{bmatrix} \quad (3.5)$$

Os sistemas de coordenadas de navegação e do corpo, \mathbf{r}^B e \mathbf{r}^N , são fornecidos por Equações 3.6 e 3.7, respectivamente. A relação entre ambos é dada pela matriz de transformação C_B^N pela Equação 3.8.

$$\mathbf{r}^B = \begin{bmatrix} \mathbf{a}^B \\ \boldsymbol{\omega}_E^B \\ \boldsymbol{\omega}_E^B \times \mathbf{g}^B \end{bmatrix} \quad (3.6)$$

$$\mathbf{r}^N = \begin{bmatrix} \mathbf{g}^N \\ \boldsymbol{\omega}_E^N \\ \boldsymbol{\omega}_E^N \times \mathbf{g}^N \end{bmatrix} \quad (3.7)$$

$$\mathbf{r}^N = C_B^N \mathbf{r}^B \quad (3.8)$$

Portanto, a matriz de transformação $C_B^N = (C_N^B)^T = (C^B)^{-1}$ pode ser definida pela Equação 3.9.

$$C_B^N = (\mathbf{r}_N^T)^{-1} \mathbf{r}_B^T \quad (3.9)$$

$$\mathbf{C}_B^N = \begin{bmatrix} c_{11} c_{12} c_{13} \\ c_{21} c_{22} c_{23} \\ c_{31} c_{32} c_{33} \end{bmatrix} \quad (3.10)$$

Os coeficientes c_{ni} na matriz \mathbf{C}_B^N foram definidos pela Equações 3.11 para 3.19.

$$c_{11} = \frac{f_x}{g} \quad (3.11)$$

$$c_{12} = \frac{f_y}{g} \quad (3.12)$$

$$c_{13} = \frac{f_z}{g} \quad (3.13)$$

$$c_{21} = \frac{f_z \omega_y - f_y \omega_z}{g \Omega \cos L} \quad (3.14)$$

$$c_{22} = \frac{f_x \omega_z - f_z \omega_x}{g \Omega \cos L} \quad (3.15)$$

$$c_{23} = \frac{f_y \omega_x - f_x \omega_y}{g \Omega \cos L} \quad (3.16)$$

$$c_{31} = \frac{g \omega_x - f_x \Omega \sin L}{g \Omega \cos L} \quad (3.17)$$

$$c_{32} = \frac{g \omega_y - f_y \Omega \sin L}{g \Omega \cos L} \quad (3.18)$$

$$c_{33} = \frac{g \omega_z - f_z \Omega \sin L}{g \Omega \cos L} \quad (3.19)$$

As equações 3.11 para 3.19 são definidas por medições inerciais. Os ângulos de Euler podem ser expressos pela função DCM (Equation 3.10). Por método de piograma (PIO, 1966), as equações 3.20 para 3.22 são estabelecidas.

$$\psi = \arcsin(c_{12}) \quad (3.20)$$

$$\theta = \arctan\left(\frac{-c_{13}}{c_{11}}\right) \quad (3.21)$$

$$\phi = \arctan\left(\frac{-c_{32}}{c_{22}}\right) \quad (3.22)$$

As medidas do acelerômetro e do giroscópio são calculadas em média cerca de um ou dois minutos durante o alinhamento grosseiro para estimar os ângulos de euler (SILSON, 2011).

3.5 Alinhamento Fino

O alinhamento fino realiza a estimativa de um pequeno desalinhamento entre o quadro verdadeiro e o calculado, este é utilizado para ajustar o alinhamento calculado anteriormente.

O modelo de erro do SNIS foi derivado usando o método da perturbação descrito por tit.

$$\dot{x}(t) = Ax(t) + Gx(t) \quad (3.23)$$

$$z(t) = Hx(t) + v(t) \quad (3.24)$$

O modelo geral dado pelas Equações 3.23 e 3.24 pode ser representado pelas Equações 3.25 e 3.26, tal que $\Delta t = t_k - t_{k-1}$. Onde $F_{k,k-1} = \sum_{i=0}^{\infty} \frac{1}{i!} A(t_k) \Delta t$. w_k e v_k são modelados como sequências de ruído branco descorrelacionado.

Dessa forma, as seguintes esperanças devem ser definidas como $E[w_k] = 0$, $E[w_k w_l^T] = Q_k$, $E[v_k] = 0$, $E[v_k v_l^T] = R_k$. Tal que, R_k é a matriz de covariância de ruído das medidas e Δt é o período de filtragem.

$$x[k] = F[k-1]x[k-1] + \Delta t[k-1]w[k-1] \quad (3.25)$$

$$z[k] = H[k]x[k] + v[k], \quad (3.26)$$

A matriz de covariância do ruído do processo Q_k é calculada pela Equação 3.27 assumindo que $M_1 = Q_{\Delta t}$.

$$Q_k \cong \sum_{i=0}^{\infty} M_j \frac{\Delta t^j}{j!}, M_{j+1} = AM_j + (AM_j)^T \quad (3.27)$$

A equação 3.28 determina $\hat{X}_{k,k-1}$ que representa o estado propagado no instante t_k baseado nas medidas do instante t_{k-1} .

$$\hat{X}_{k,k-1} = F_{k,k-1} \hat{X}_{k-1} \quad (3.28)$$

Com isso, a covariância de erro de $\hat{X}_{k,k-1}$ é fornecida pela equação 3.29.

$$P_{k,k-1} = F_{k,k-1} P_{k-1} F_{k,k-1}^T + \Delta t_{k,k-1} Q_{k-1} \Delta t_{k,k-1}^T \quad (3.29)$$

O calculo do ganho de Kalman é definido pela Equação 3.30.

$$K_k = P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + R_k)^{-1} \quad (3.30)$$

Finalmente, o estado estimado atualizado e a covariância de erro atualizada são dadas pelas Equações 3.31 e 3.32.

$$\hat{x}_k = \hat{x}_{k,k-1} + k_k (z_k - H_k x_{k,k-1}) \quad (3.31)$$

$$P_k = (I - K_k H_k) P_{k,k-1} (I - K_k H_k)^T + K_k R_k H_k^T \quad (3.32)$$

3.6 Resultados

Os Resultados do algoritmo são apresentados nessa seção, verifica-se que a acurácia e a precisão estão de acordo com os resultados apresentados na literatura.

3.6.1 Dados reais

O erro no alinhamento grosseiro utilizando o algoritmo TRIAD é da ordem esperada. Com isso, a ordem do erro fica em torno de $0,01^\circ$ sendo essa a magnitude esperada. Visto que os ângulos reais são igual a zero, o gráfico de acurácia pode ser suprimido dessa análise, já que ele é igual ao da figura 3.1

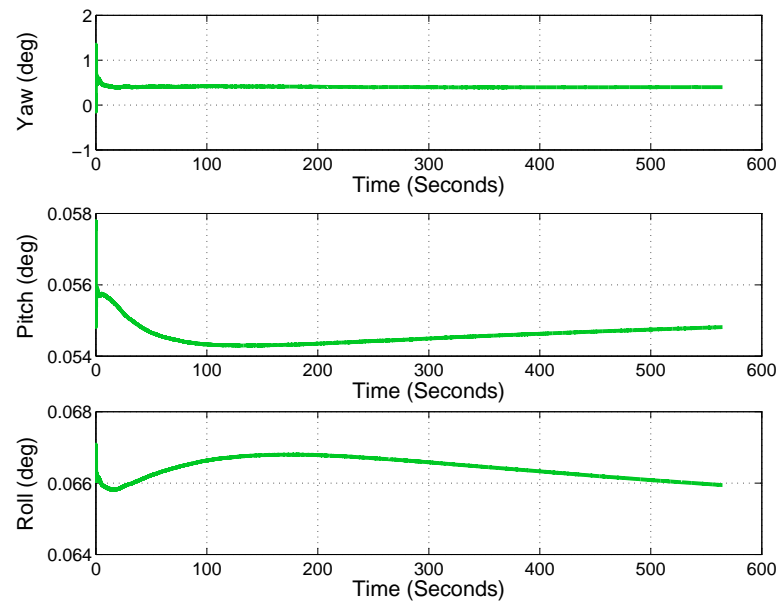


Figura 3.1: Ângulos TRIAD com dados reais.

3.6.2 Dados simulados

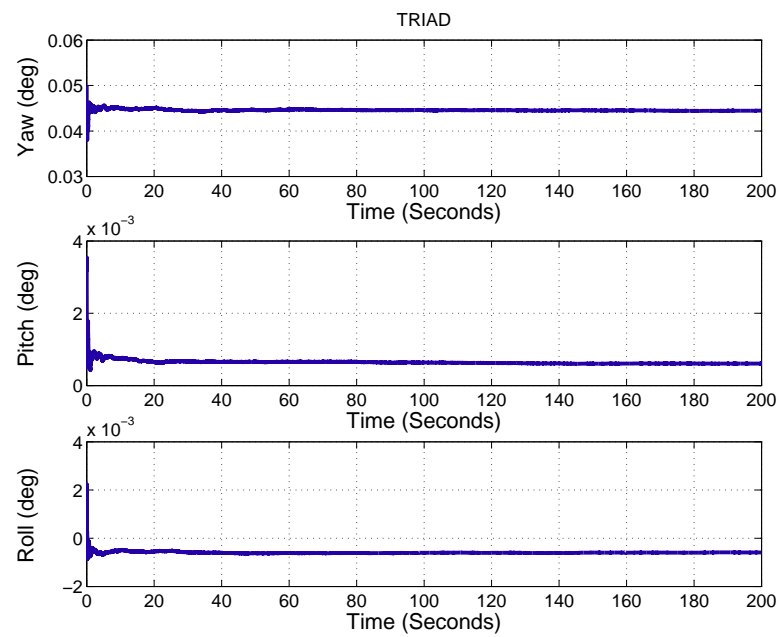


Figura 3.2: Ângulos de alinhamento grosseiro TRIAD.

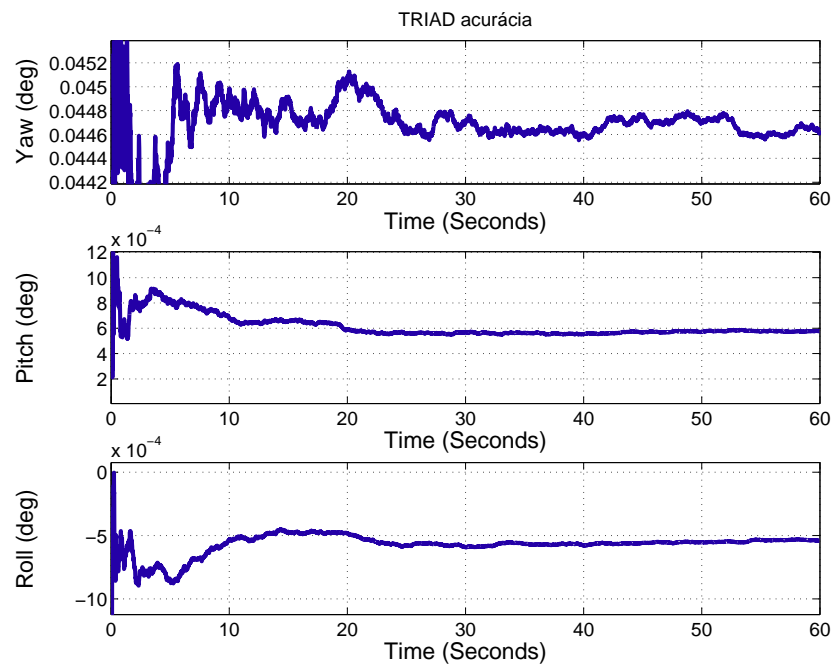


Figura 3.3: Acurácia alinhamento grosseiro TRIAD .

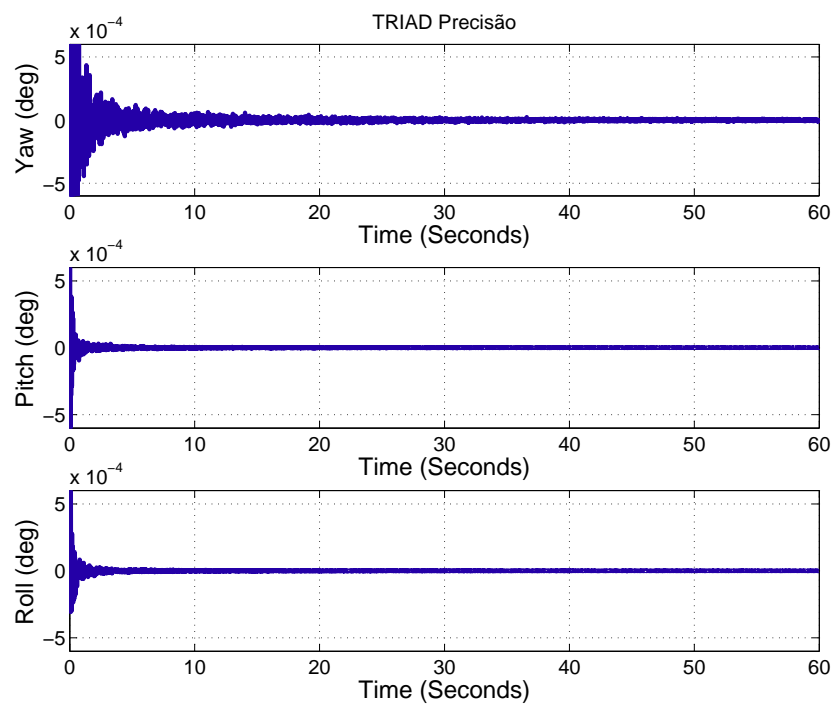


Figura 3.4: Precisão alinhamento grosseiro TRIAD.

A acurácia dos dados simulados estão de acordo com os reais, verifica-se pelas Figuras 3.3 e 3.2 que o resultado desses dados foram semelhantes ao da figura 3.1. Isto pode indicar que o modelo utilizado para simular as medidas dos sensores inerciais está adequado.

A precisão pode ser avaliada observando a derivada dos ângulos de alinhamento, assim a convergência do ângulo e a precisão do algoritmo pode ser avaliada pela Figura 3.4.

4 Alinhamento Grosseiro usando REQUEST

Neste capítulo, o auto-alinhamento usando o método REQUEST é investigado, a fim de estimar atitude recursivamente.

O objetivo do algoritmo REQUEST (BAR-ITZHACK, 1996) é encontrar o quatérnion \mathbf{q}^* que minimiza a Equação 4.1. Assim, esse trabalho propõe obter a matriz de transformação C_B^N em uma única etapa.

$$J(C_B^N(\mathbf{q})) = \frac{1}{2} \sum_{i=1}^k a_i |\mathbf{r}_i^B - C_B^N(\mathbf{q})\mathbf{r}_i^N|^2, \quad (4.1)$$

Onde a_i é um atributo de normalização, \mathbf{r}_i^B é a medida da coordenada do corpo e \mathbf{r}_i^N é a medida da coordenada de referencia no instante t_i , tal que $i = 1, 2, 3 \dots k$.

$$g(\mathbf{q}) = 1 - J(\mathbf{q}) \quad (4.2)$$

Para a definição $g(\mathbf{q})$ pode ser representada pela Equation 4.3.

$$g(\mathbf{q}) = \mathbf{q}^T K \mathbf{q} \quad (4.3)$$

K é o autovalor de λ e λ é o autovetor de \mathbf{q}^* , temos as seguintes Equações 4.4 para 4.5.

$$K \mathbf{q}^* = \lambda \mathbf{q}^* \quad (4.4)$$

$$g(\mathbf{q}^*) = \lambda \quad (4.5)$$

Devemos encontrar K para determinar o multiplicador Lagrange λ e quatérnion \mathbf{q}^* que minimizam a função de Wabah.

$$m_k = \sum_{i=1}^k a_i \quad (4.6)$$

$$\sigma_k = \frac{1}{m_k} \sum_{i=1}^k a_i b_i^T r_i \quad (4.7)$$

$$B_k = \frac{1}{m_k} \sum_{i=1}^k a_i b_i r_i^T \quad (4.8)$$

$$S_k = B_k + B_k^T \quad (4.9)$$

$$z_k = \frac{1}{m_k} \sum_{i=1}^k a_i (b_i \times r_i) \quad (4.10)$$

Por definição, K_k , γ^* e \mathbf{q}^* são calculados pelas equações 4.11, 4.12 e 4.13.

$$K_k = \begin{bmatrix} S_k - \sigma \mathbf{I} & z_k \\ z_k^T & \sigma_k \end{bmatrix} \quad (4.11)$$

$$\gamma^* = [(\lambda + \sigma)\mathbf{I} + S]^{-1} \mathbf{z} \quad (4.12)$$

$$\mathbf{q}^* = \frac{1}{\sqrt{1 + \gamma^*}} \begin{bmatrix} \gamma^* \\ 1 \end{bmatrix} \quad (4.13)$$

A extração apropriada do autovetor de \mathbf{q}^* foi feita pelo algoritmo QUEST (SHUSTER, 1989).

No entanto, QUEST calcula a atitude para uma única janela de tempo, enquanto a REQUEST calcula o ponto de atitude de ponto de forma recursiva.

Analogamente, podemos afirmar que δK_{k+1} é dado computando as equações 4.14 para 4.23.

$$\delta K_{k+1} = \begin{bmatrix} \delta S_{k+1} - \delta \sigma \mathbf{I} & \delta z_{k+1} \\ z_{k+1}^T & \delta \sigma_{k+1} \end{bmatrix} \quad (4.14)$$

$$m_{k+1} = \sum_{i=1}^{k+1} a_i \quad (4.15)$$

$$\sigma_{k+1} = \frac{1}{m_{k+1}} \sum_{i=1}^{k+1} a_i b_i^T r_i \quad (4.16)$$

$$\sigma_{k+1} = \frac{1}{m_{k+1}} \sum_{i=1}^k a_i b_i^T r_i + \frac{1}{m_{k+1}} a_{k+1} b_{k+1}^T r_{k+1} \quad (4.17)$$

$$B_{k+1} = \frac{1}{m_{k+1}} \sum_{i=1}^{k+1} a_i b_i r_i^T \quad (4.18)$$

$$B_{k+1} = \frac{1}{m_{k+1}} \sum_{i=1}^k a_i b_i r_i^T + \frac{1}{m_{k+1}} a_{k+1} b_{k+1} r_{k+1}^T \quad (4.19)$$

$$S_{k+1} = B_{k+1} + B_{k+1}^T \quad (4.20)$$

$$S_{k+1} = B_k + B_k^T + \delta B_{k+1} + \delta B_{k+1}^T \quad (4.21)$$

$$z_{k+1} = \frac{1}{m_{k+1}} \sum_{i=1}^{k+1} a_i (b_i \times r_i) \quad (4.22)$$

$$z_{k+1} = \frac{\sum_{i=1}^k a_i (b_i \times r_i)}{m_{k+1}} + \frac{a_{k+1} (b_{k+1} \times r_{k+1})}{m_{k+1}} \quad (4.23)$$

As seguintes frases em 4.24 para 4.27 são necessárias para perceber a computação de K_{k+1} .

$$\delta \sigma_{k+1} = a_{k+1} b_{k+1}^T r_{k+1} \quad (4.24)$$

$$\delta B_{k+1} = a_{k+1} b_{k+1} r_{k+1}^T \quad (4.25)$$

$$\delta S_{k+1} = \delta B_{k+1} + \delta B_{k+1}^T \quad (4.26)$$

$$\delta z_{k+1} = a_{k+1} (b_{k+1} \times r_{k+1}) \quad (4.27)$$

Assim, essas equações são reduzidas nas Equações 4.28 para z1.

$$\sigma_{k+1} = \frac{1}{m_{k+1}} (m_k \sigma_k + \delta \sigma_{k+1}) \quad (4.28)$$

$$S_{k+1} = \frac{1}{m_{k+1}} (m_k S_k + \delta S_{k+1}) \quad (4.29)$$

$$z_{k+1} = \frac{1}{m_{k+1}}(m_k z_k + \delta z_{k+1}) \quad (4.30)$$

Ao substituir a equação 4.28 e 4.30 na equação 4.11, temos a Equação 4.31.

$$K_{k+1} = \frac{1}{m_{k+1}}(m_k K_k + \delta K_{k+1}) \quad (4.31)$$

A equação 4.31 dá o autovalor λ de K_{k+1} , então a Equação 4.12 e 4.13 fornecem \mathbf{q}_{k+1} imediatamente.

Portanto, \mathbf{q}_{k+1}^* depende de \mathbf{q}_k^* . Podemos com o autovalor e o autovetor de \mathbf{q}_k^* encontrar \mathbf{q}_{k+1}^* .

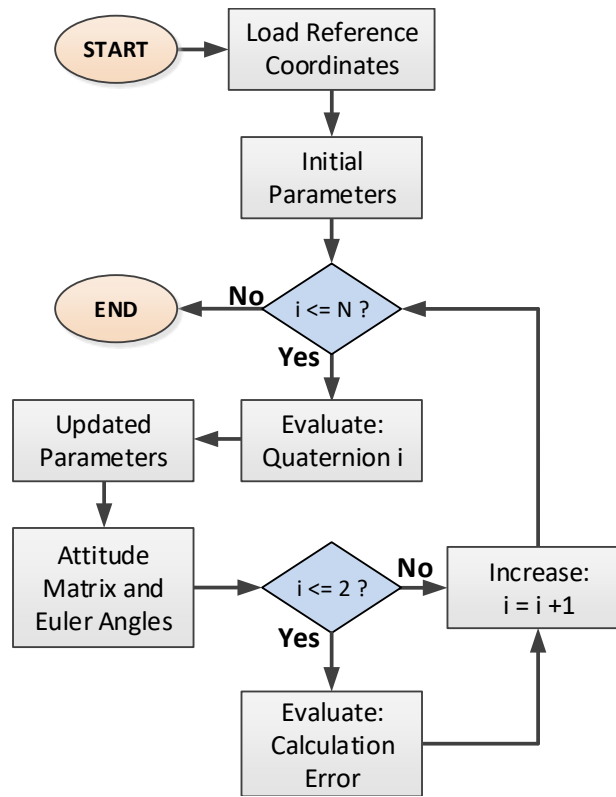


Figura 4.1: Diagrama de Bloco REQUEST

A figura 4.1 mostra a previsão dos quatérnions do algoritmo em forma de diagrama de blocos. Inicialmente, verifica-se que as coordenadas de referência devem ser definidas. Em seguida, os parâmetros iniciais são carregados. A partir desses parâmetros iniciais o quatérnion \mathbf{q}_k^* é computado. Por fim, novos parâmetros são calculados para determinar o quatérnion atualizado \mathbf{q}_{k+1}^* .

4.1 Resultados

Os resultados do algoritmo REQUEST puro é apresentado nessa seção, em seguida breve comparações em relação ao algoritmo tradicional serão realizadas.

4.1.1 Dados reais

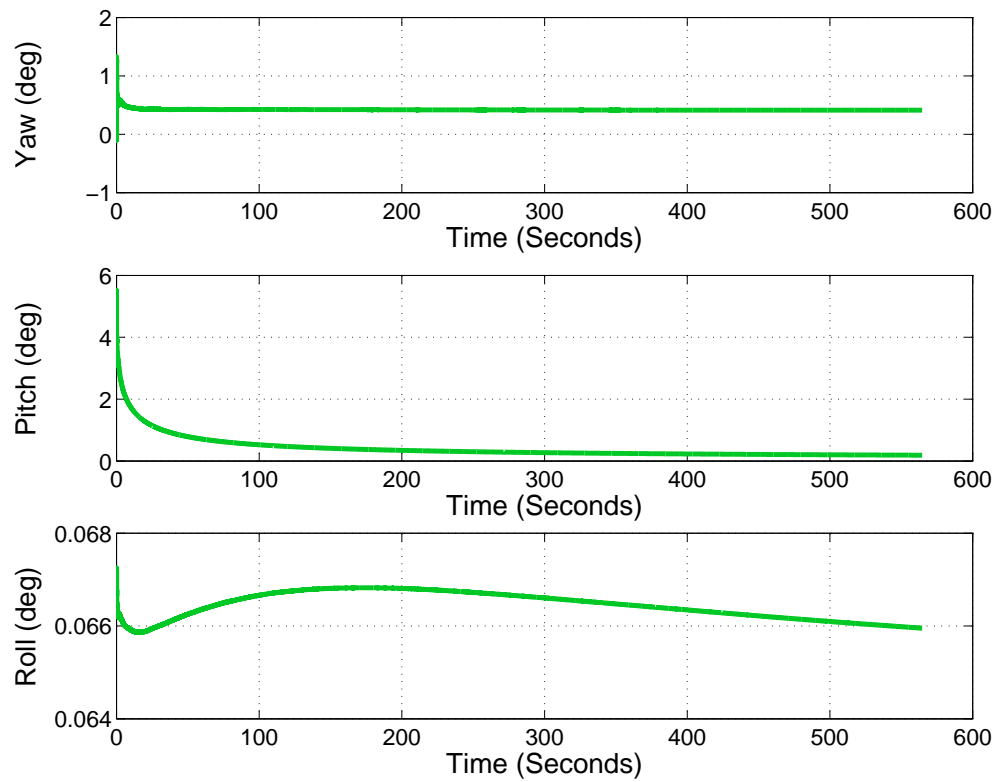


Figura 4.2: Ângulos REQUEST com dados reais.

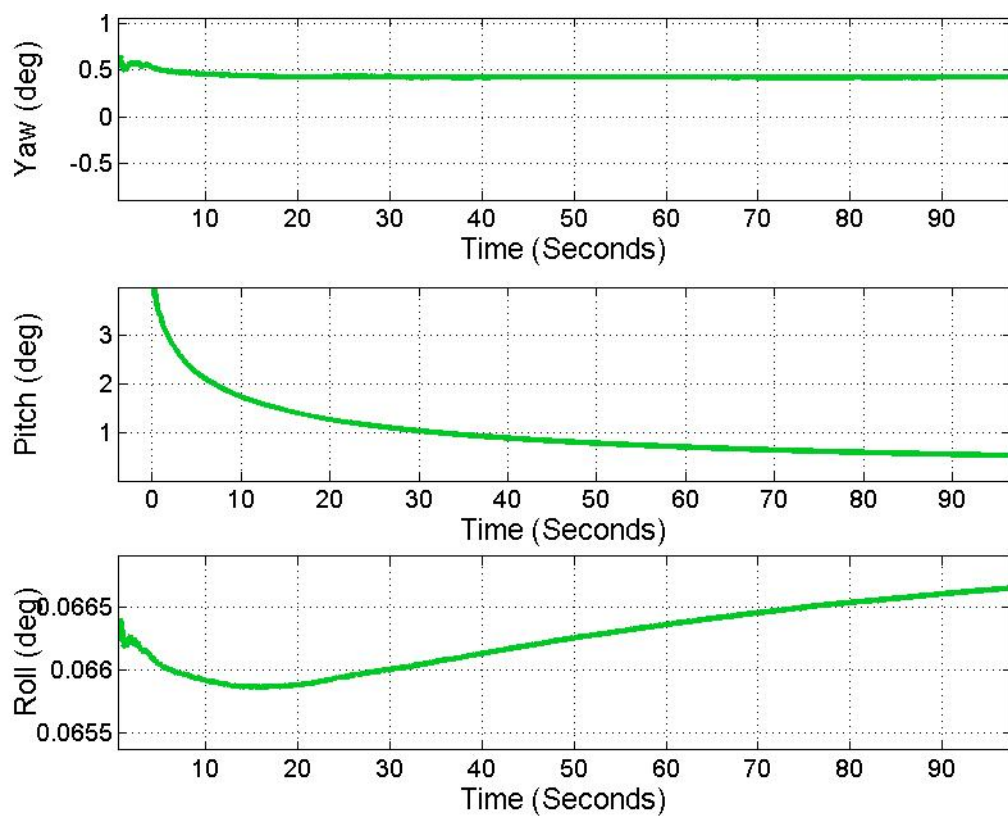


Figura 4.3: Ângulos REQUEST com dados reais em regime transitório.

A Figura 4.2 indica que os resultados são idênticos, deve-se levar em consideração que esses dados estão calibrados.

Além disso, a inicialização do algoritmo REQUEST pode gerar um ponto de singularidade. Como resultado disso, a primeira estimativa pode iniciar de valores elevados.

A figura 4.3 mostra o comportamento inicial da resposta fornecida pelo algoritmo.

4.1.2 Dados simulados

Como visto na figura 4.4, o resultado do algoritmo REQUEST simulado é semelhante ao real. Dessa forma, o resultado é praticamente idêntico ao resultado do algoritmo TRIAD.

Esse resultado é coerente, visto que o REQUEST possui as mesmas características que o algoritmo QUEST em relação a acurácia, assim como discutido em (MARKLEY, 1998) e (SHUSTER, 2009).

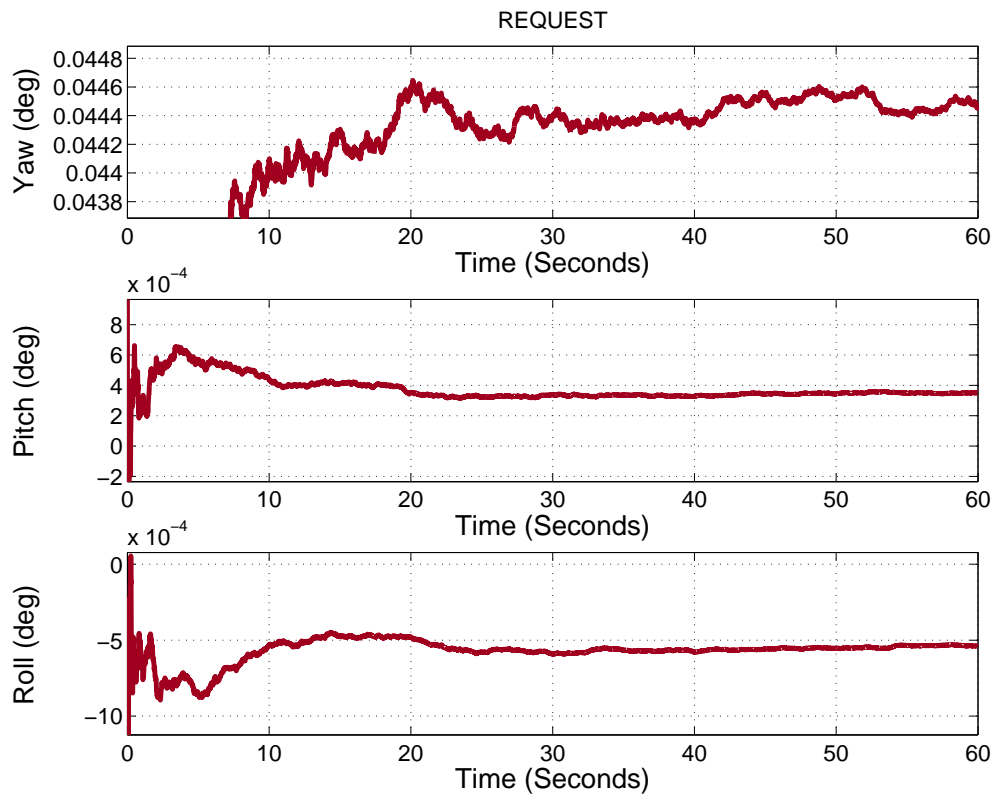


Figura 4.4: Ângulos de alinhamento grosseiro REQUEST.

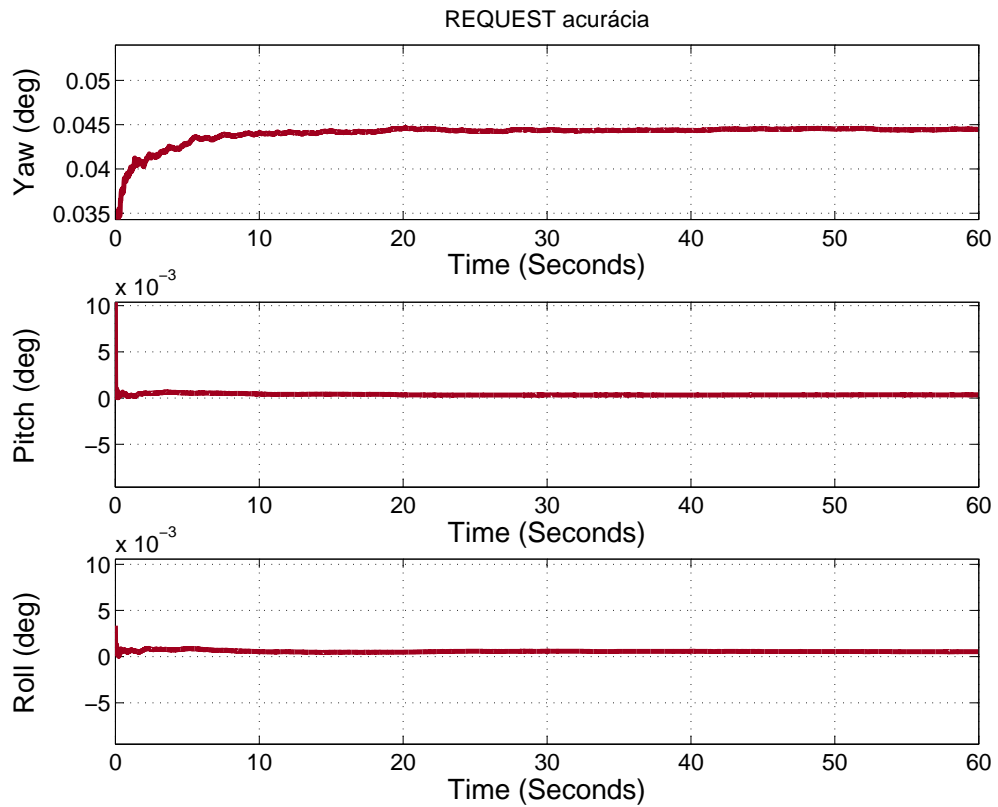


Figura 4.5: Acurácia alinhamento grosseiro REQUEST .

Além disso, a Figura 4.6 mostra que o algoritmo REQUEST possui precisão idêntica ao TRIAD

Apesar disso, o algoritmo REQUEST possui algumas vantagens sobre o QUEST, assim como o QUEST possui algumas vantagens sobre o TRIAD.

No algoritmo REQUEST possui claramente a vantagem de ser recursivo, enquanto o QUEST e o TRIAD não. Além disso, assim como demonstrado por Markley (1998), o QUEST e o REQUEST possui uma complexidade computacional menor.

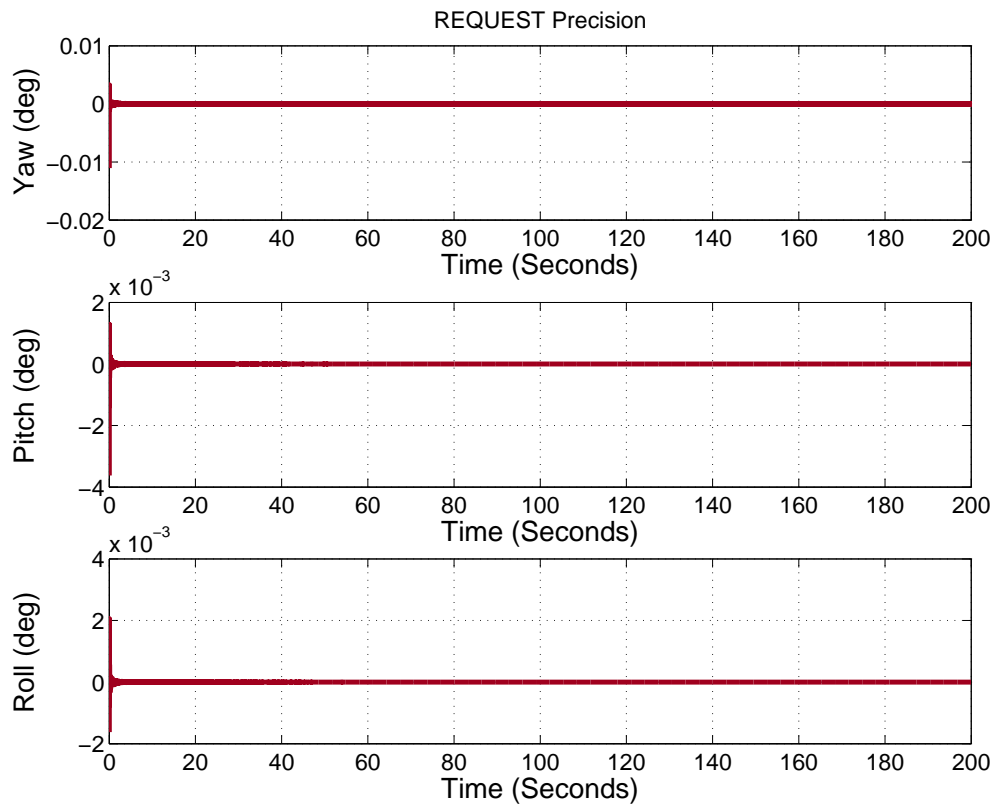


Figura 4.6: Precisão alinhamento grosseiro REQUEST .

Dessa forma, o algoritmo REQUEST pode ser considerado uma melhor prática para determinação de atitude para realizar o alinhamento grosseiro.

5 REQUEST com estimador de *bias*

O impacto do *bias* na acurácia dos algoritmos propostos é alto. Dessa forma, o fato do algoritmo REQUEST não estimar o *bias* pode ser investigado para melhorar a sua performance.

5.1 Estimador

Nesse trabalho, as Equações 5.1 e 5.2 foram utilizadas para de modelar o estimador de *bias*. Tal que, \mathbf{B}_g e \mathbf{B}_a são os vetores com os *bias* do girômetro e do aceleômetro em cada eixo, respectivamente.

$$C_N^B \omega_{ie} \cong \omega_E^N - \mathbf{B}_g \quad (5.1)$$

$$C_N^B \omega_{ie} \cong \omega_E^N - \mathbf{B}_a \quad (5.2)$$

A matriz de rotação estimada no instante k é utilizada para transformar a rotação da Terra do eixo de local para o eixo do corpo. Idealmente, os valores dos sensores inerciais sem o *bias* deve ser igual ao resultado dessa transformação.

Por isso, Assume-se que a Equação 5.3 é verdadeira.

$$0 \cong C_N^B \omega_{ie} - (\omega_E^N - \mathbf{B}) \quad (5.3)$$

A fim de diminuir a resolução necessária do sistema computacional, a raiz quadrada pode ser adicionada ao sistema. Resultando na Equação 5.4.

$$0 \cong \sqrt{C_N^B \omega_{ie} - (\omega_E^N - \mathbf{B})} \quad (5.4)$$

Dessa forma, o *bias* em cada eixo pode ser determinado ao minimizar a equação 5.5. Diversos algoritmos para encontrar o mínimo de uma função multivariável podem

ser utilizados. Nesse trabalho, o algoritmo simplex em (LAGARIAS et al., 1998) foi utilizado para realizar essa tarefa.

$$E = \sqrt{C_N^B \omega_{ie} - (\omega_E^N - B)} \quad (5.5)$$

5.2 Inicialização do Algoritmo

Busca por força bruta, isto é a partir de um conjunto de *bias* arbitrários diversos resultados para a Equação 5.5 são gerados. Com esses resultados, o valor mínimo é encontrado. A partir disso, o conjunto do *bias* que gerou a esse valor mínimo é utilizado para inicializar o algoritmo *simplex*. A Figura 5.1 e 5.2 foram

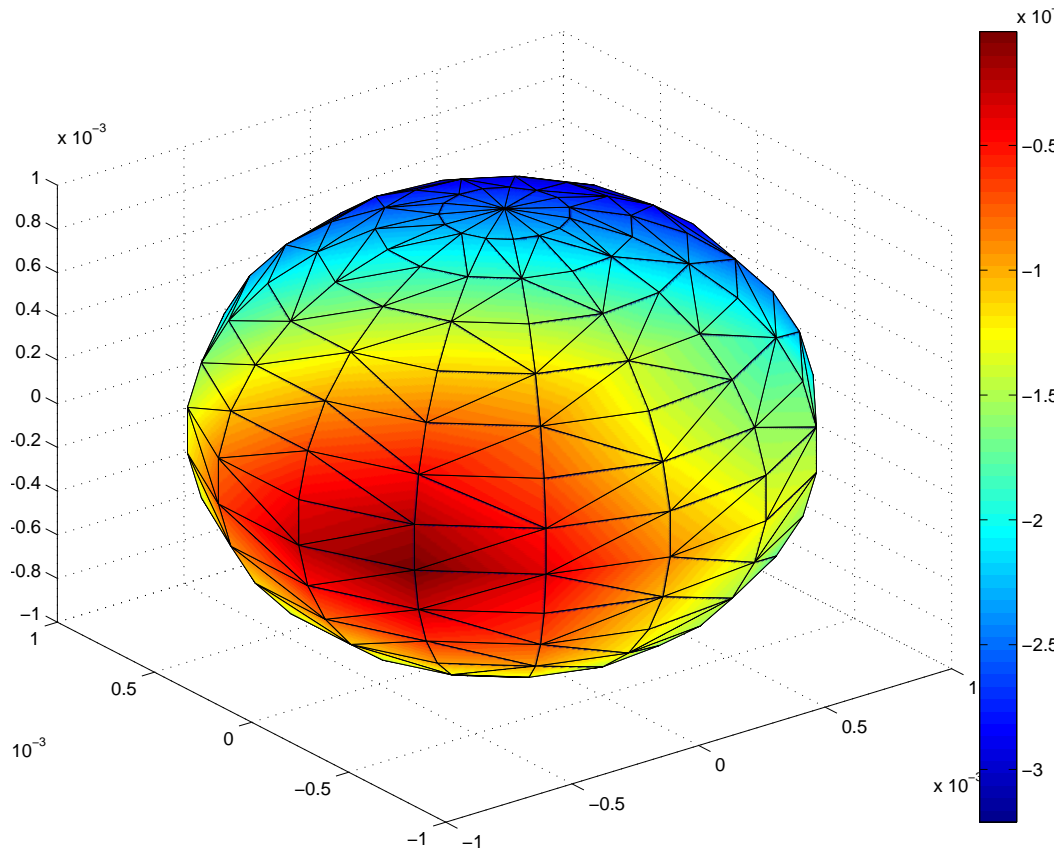


Figura 5.1: Função $E(b_x, b_y, b_z)$ em 1 segundo.

produzidas pelo algoritmo de força bruta. Cada eixo representa a magnitude do *bias* correspondente e a cor indica o valor da função, tal que o vermelho claro mostra o resultado igual a 0. Pode ser observado que não há mínimos locais, portanto isso pode indicar robustez à metodologia.

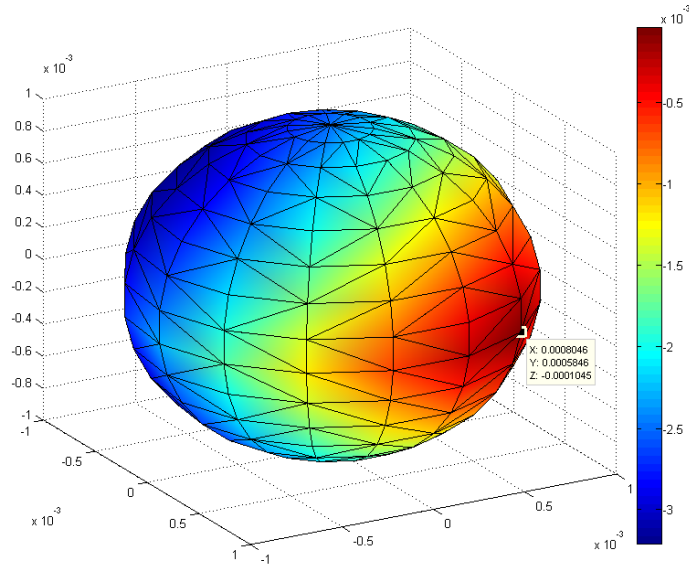


Figura 5.2: Função $E(b_x, b_y, b_z)$ em 1 segundo.

5.3 Resultados

Os resultados do algoritmo REQUEST com estimador são apresentados nesta seção. Com isso, deve-se considerar a significativa melhora na acurácia da estimativa.

5.3.1 Dados simulados

As figuras 5.3, 5.4 indicam que a acurácia melhorou a performance do algoritmo.

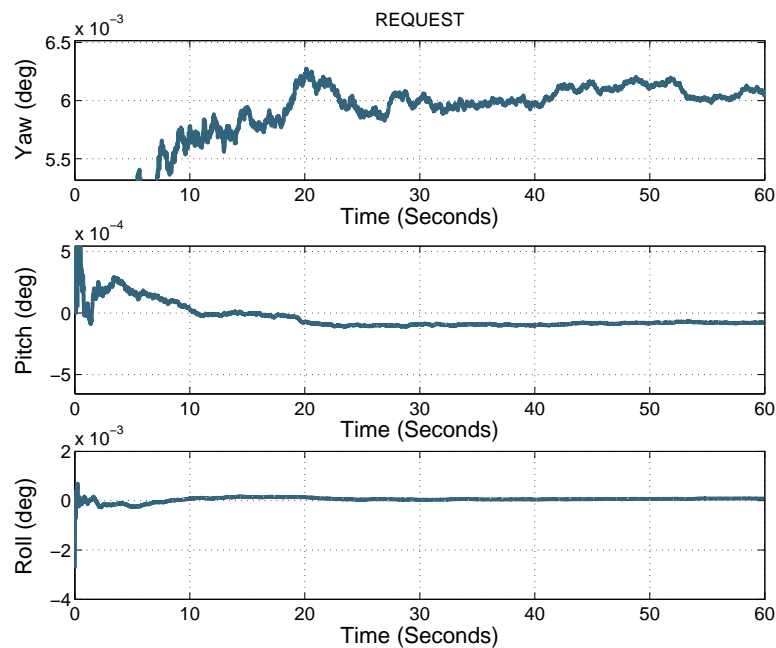


Figura 5.3: Ângulos de alinhamento grosseiro REQUEST usando estimador usando dados simulados.

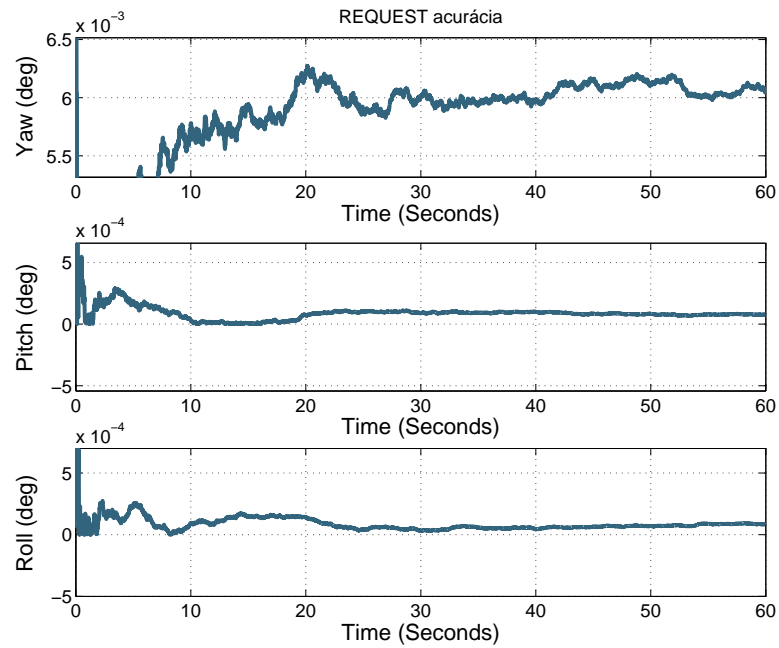


Figura 5.4: Acurácia alinhamento grosseiro REQUEST usando estimador usando dados simulados.

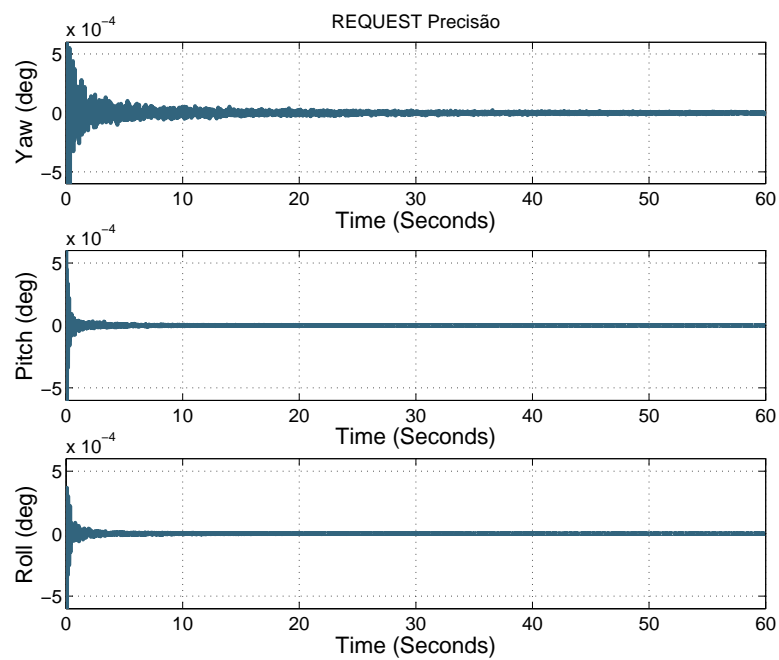


Figura 5.5: Precisão alinhamento grosseiro REQUEST usando estimador usando dados simulados.

A figura 5.5 mostra que a convergência não sofreu impacto, entretanto o método proposto na seção 6 pode ser utilizado para melhorar esse quesito.

Com isso, a acurácia do alinhamento grosseiro utilizando o algoritmo REQUEST com estimador de *bias* pode tornar o Auto-Alinhamento completo muito mais rápido em relação ao que utiliza o método tradicional.

6 Método proposto *Neural REQUEST*

Neste capítulo, o auto-alinhamento usando REQUEST com redes neurais *feed-forward* é proposto, para diminuir o tempo de alinhamento.

Uma rede neural é uma máquina com a capacidade de se adaptar à medida que ocorrem os eventos em torno dela (máquina adaptável). ANN tem uma habilidade natural para armazenar conhecimento experimental, aprender e tomar novas decisões que estão disponíveis para uso (HAYKIN, 1999).

Em outras palavras, uma rede neural é um sistema que armazena o comportamento anterior do sistema alvo, a fim de reproduzir seu comportamento em circunstâncias futuras gerais.

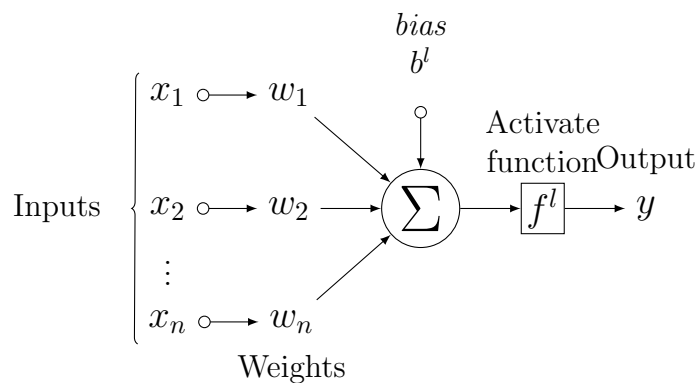


Figura 6.1: Neuron Model.

6.1 Esquema geral

Os principais recursos de rede neural feedforward são números de camada oculta, números de neurônios, função de ativação. Cada recurso é detalhado na seção 6.2, 6.3 e 6.4.

A função de ativação sigmoid χ é tradicionalmente considerada. O recurso diferenciável desta função torna popular, devido aos tradicionais algoritmos de aprendizagem baseados em gradientes clássicos que funcionam apenas

para funções de ativação diferenciáveis. Portanto, esta função é definida pela Equação 6.1.

$$\chi(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}} \quad (6.1)$$

Com uma função de ativação descrita, uma rede neural pode ser apresentada. Uma descrição matemática de uma rede neural clássica feed-forward é definida pela Equação 6.2. Onde $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{in}]$ é a matriz de pesos sinápticos conectando o i th nó escondido e os nós de saída, $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{ik}]$ é a matriz de pesos sinápticos conectando o i th nó escondido e os nós de entrada e b_i é o *bias* do nó oculto.

$$\tilde{y}(\mathbf{x}_k) = \mathbf{v}_i \chi(\mathbf{w}_i \mathbf{x}_k + b_i) \quad (6.2)$$

Portanto, a saída da rede neural \tilde{y}_k com N amostras de entrada x_k pode ser aproximada da saída de destino y_k , de modo que haja v_i , w_i e b_i tal que $\sum_{k=1}^N \tilde{y}_k - y_k = 0$. Para encontrar esses coeficientes pode ser usado um algoritmo de aprendizado baseado em gradiente.

Neste trabalho, o resultado alvo é considerado um quatérnion q_{k+p} , tal que $y_k = q_{k+p}$ com base no vetor definido pela equação 6.3 como entrada.

$$\mathbf{x}_k = [q_k, \Omega_k, L, g, f_k^x, f_k^y, f_k^z, \omega_k^x, \omega_k^y, \omega_k^z]^T \quad (6.3)$$

A matriz de coeficientes \mathbf{v}_i , \mathbf{w}_i e \mathbf{b}_i fornece uma memória que foi obtida a partir de um padrão de amostra. Neste artigo, esta condição é investigada para melhorar o desempenho do algoritmo REQUEST.

O diagrama de bloco na Figura 6.2 mostra a diferença entre o algoritmo REQUEST e NREQUEST. Como pode ser visto, o método usado para NREQUEST tem uma rede neural entre o quatérnion k e quatérnion $k + 1$. Portanto, uma RNA de experiências anteriores nas mesmas condições ambientais é adicionada por RNA.

O NN considerado neste trabalho tem dez neurônios, uma função de ativação sigmoide, uma camada oculta e o algoritmo de aprendizagem de máquina extremo (HUANG; ZHU; SIEW, 2004, 2006).

Essas características são os principais fatores que influenciam a rede neural de desempenho. Portanto, a motivação para escolher cada atributos é detalhada nas seções 6.2, 6.3 e 6.4.

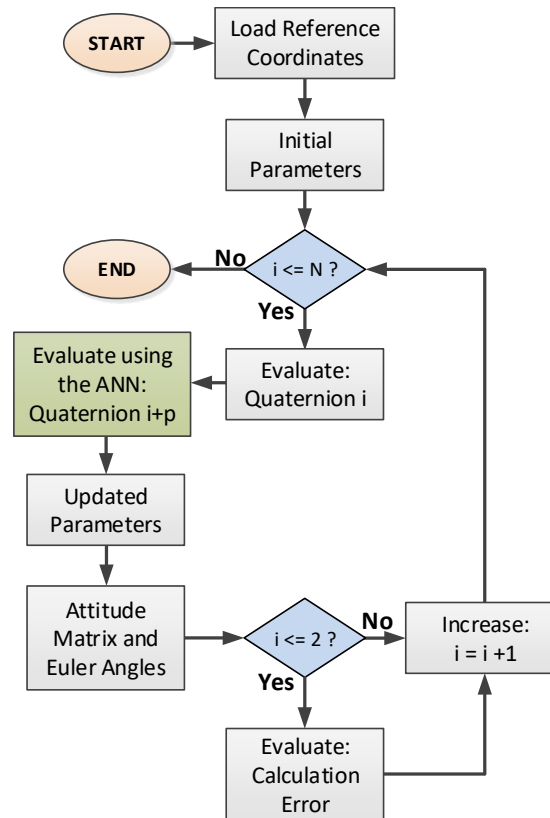


Figura 6.2: Diagrama de blocos do NREQUEST

6.2 Algoritmo de aprendizagem

As redes neurais foram utilizadas para realizar mapeamentos não-lineares do sistema a posteriori da resposta do algoritmo REQUEST, no entanto, os algoritmos de aprendizado tradicionais são extremamente lentos. Este problema torna algumas aplicações impraticáveis.

A fim de utilizar um algoritmo de aprendizado rápido, este trabalho aplica uma aprendizagem algoritmo para redes neuronais feedforward de camada unidimensional chamado máquina de aprendizagem extrema ou *Extreme learning machine* (ELM), que foi provado com rigor por Huang, Zhu e Siew (2006).

O ELM tem várias características e benefícios significativos diferentes dos tradicionais algoritmos de aprendizagem para redes neurais *feedforward*, como o algoritmo de *backpropagation* proposto por Rumelhart et al. (1988).

A velocidade de aprendizagem do ELM é extremamente rápida, tem melhor desempenho de generalização do que a aprendizagem com base em gradientes, como *backpropagation*, e tende a alcançar as soluções diretas sem problemas triviais como mínimos locais, taxa de aprendizado incorreta e superação excessiva.

Ao contrário dos tradicionais algoritmos de aprendizagem baseados em gradientes clássicos, o algoritmo de aprendizagem ELM poderia ser usado com funções de ativação não diferenciáveis (HUANG; ZHU; SIEW, 2006).

6.2.1 Algoritmo geral baseado em gradiente

Normalmente, a fim de treinar uma rede neural *feedforward* com uma única camada, pode-se desejar para encontrar w_i , v_i e b_i , de modo que a função de custo na Equação 6.4, de modo que N seja um número de amostra e \tilde{N} é o número de neurônios da camada oculta.

$$E = \sum_{j=1}^N \left(\sum_{i=1}^{\tilde{N}} v_i \chi(w_i x_k + b_i) - y_k \right)^2 \quad (6.4)$$

No procedimento de minimização usando algoritmos baseados em gradiente, o vetor \mathbf{W} , que é o conjunto de pesos (w_i , v_i) e o *bias* (b_i), é ajustado iterativamente da seguinte forma:

$$W_k = W_{k-1} - \eta \cdot E(W)x \quad (6.5)$$

Onde η é uma taxa de aprendizado.

Dessa forma, alguns problemas nos algoritmos tradicionais podem ser levantados, entre eles estão:

- Quando a taxa de aprendizado η é muito pequena, o algoritmo de aprendizagem converge muito devagar. No entanto, quando η é muito grande, o algoritmo torna-se instável e diverge;
- A presença de mínimos locais (HAYKIN, 1999), o que indica um falso aprendizado.
- A rede neural pode ser superdimensionada usando algoritmos BP e obtendo uma pior performance de generalização. Assim, a validação e métodos de parada adequados são necessários no procedimento de minimização da função de custo.
- O tempo de treinamento elevado nas maioria das aplicações.

Em (HUANG; ZHU; SIEW, 2006), o ELM foi proposto com sucesso para solucionar o diminuir estes problemas. Por isso, este algoritmo foi escolhido para a aplicação proposta neste trabalho.

6.2.2 Extreme learning machine

O ELM para redes neuronais feedforward de camada unidimensional escolhe aleatoriamente nós ocultos e determina analiticamente os pesos de saída.

Dado um conjunto de treinamento em Equation 6.6 e uma função de ativação $\chi(x)$, e o número de nó oculto \tilde{N} , portanto, um algoritmo de aprendizagem de máquina de aprendizagem extrema pode ser resumido da seguinte maneira: 1) Randomicamente atribuir pesos de entrada \mathbf{w}_i e *bias* b_i , $i = [1, \dots, \tilde{N}]$, 2) Calcular a matriz de peso da saída da camada oculta e 3) Calcular o peso de saída $\mathbf{v}.I$.

$$\iota = (x_i, y_i) | x_i \in R^n, \quad y_i \in R^m, \quad i = [1, \dots, N] \quad (6.6)$$

6.3 Número de neurônios

Vários pesquisadores tentaram e propuseram muitas metodologias para corrigir o número de neurônios escondidos. A pesquisa foi feita para encontrar o número de neurônios escondidos na rede neural. A seguinte declaração deve ser considerada.

Como indicado por Huang, Zhu e Siew (2006), se o número \tilde{N} de nós ocultos for igual ao número N de amostras de treinamento distintas, $\tilde{N} = N$, as redes *feedforward* podem aproximar essas amostras de treinamento como erro zero.

Se a função de ativação g for infinitamente diferenciável, Huang, Zhu e Siew (2004) provou que o número necessário de nós ocultos é $\tilde{N} \leq N$. Em outras palavras, as redes *feedforward* podem exigir menos de N nós ocultos se o erro de aprendizagem for permitido.

Sheela e Deepa (2013) fizeram uma extensa revisão sobre métodos para corrigir o número de neurônios escondidos em redes neurais e descreveram vários métodos de forma cronológica.

Shibata e Ikeda (2009) investigaram o efeito da estabilidade de aprendizagem e do neurônio escondido na rede neural. Os resultados da simulação mostram que o peso da conexão de saída oculta torna-se pequeno, pois o número de neurônios ocultos N_h torna-se grande. Isso é implementado em problemas de mapeamento de números aleatórios. A fórmula para nós ocultos é apresentada na Equação 6.7 onde N_{in} é o neurônio de entrada e N_{out} é o neurônio de saída.

$$N_h = \sqrt{N_{in}N_{out}} \quad (6.7)$$

6.4 Função de ativação

Hornik (1991) provou que, se a função de ativação for contínua, limitada e não constante, os mapeamentos contínuos podem ser aproximados em função das redes neurais sobre conjuntos de entrada compactos camada oculta.

Tais funções de ativação incluem as funções sigmoidais, bem como a base radial, seno, cosseno, exponencial e muitas outras funções não-regulares, como mostrado por Huang e Babri (1998).

Conforme rigorosamente provado por Huang, Zhu e Siew (2006), ao contrário das teorias de aproximação de função tradicionais que requerem ajustar pesos de entrada e polaridades de camada oculta, os pesos de entrada e as polaridades da camada oculta podem ser atribuídos aleatoriamente se a função de ativação for infinitamente diferenciável.

A função de ativação usada em nossos algoritmos propostos é uma função sigmoideal simples chamada função logística

6.5 Resultados

Os resultados do algoritmo são apresentados nessa seção, para isso é apresentado gráficos da precisão, acurácia e ângulo estimado ao utilizar essa técnica.

6.5.1 Dados Simulados

As Figuras 6.3 e 6.4 mostram que a acurácia não é modificada ao adicionar a rede neural, visto que o objetivo dela no sistema é melhorar a precisão.

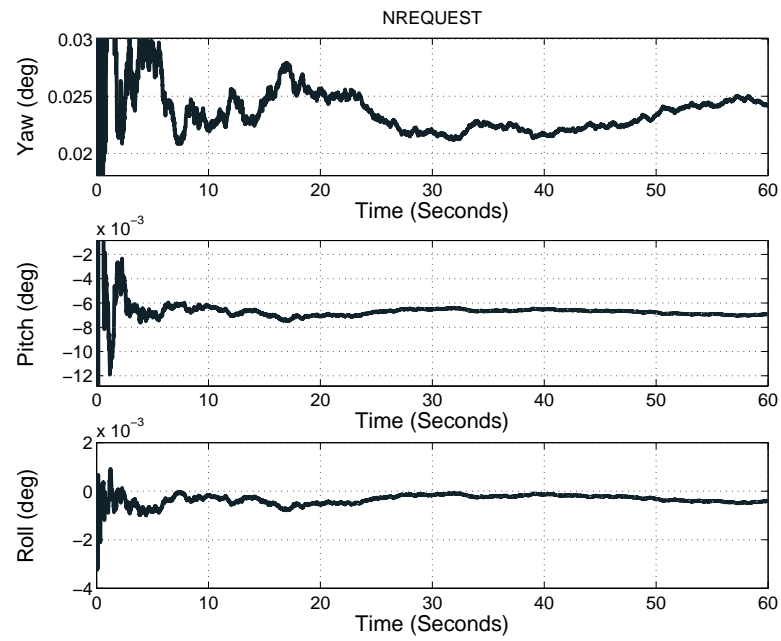


Figura 6.3: Ângulos de alinhamento grosseiro NREQUEST.

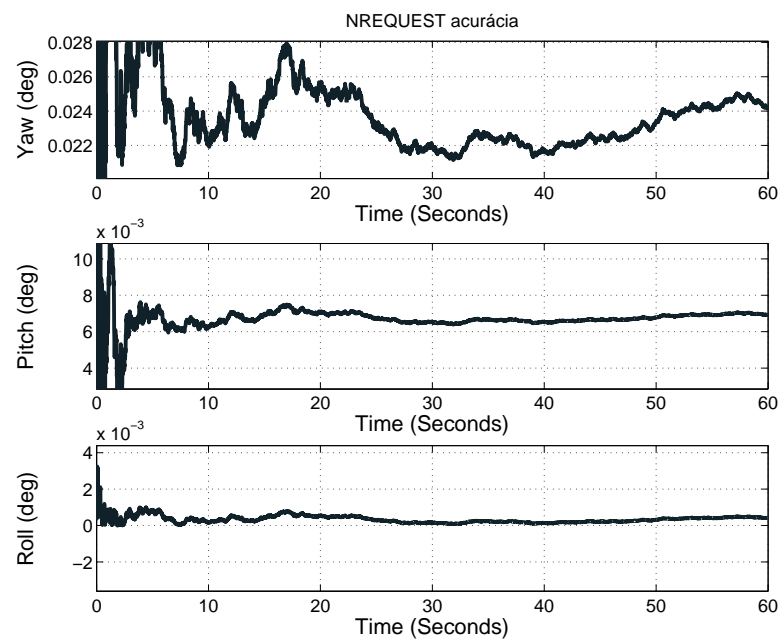


Figura 6.4: Acurácia alinhamento grosseiro NREQUEST .

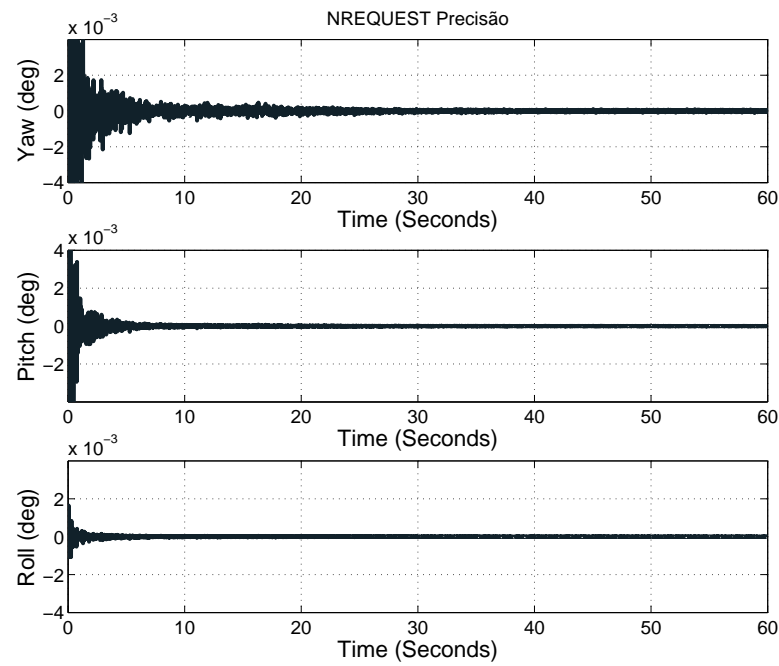


Figura 6.5: Precisão alinhamento grosseiro NREQUEST .

A Figura 6.5 mostra que a taxa de convergência do ângulo de *yaw* é melhorada. Na seção 7, a acurácia e a taxa de convergência são discutidas utilizando valores numéricos, a fim de comparar melhores os métodos apresentados.

6.6 Validação da rede neural

Apesar da validação fazer parte do treinamento da rede o qual não pode ser concluído sem a validação adequada, a revalidação é considerada uma boa prática para confirmar o resultado.

Nessa seção, a validação da rede neural foi realizada ao treinar a rede neural com amostras de ângulos distintos próximo de zero. Com isso, onze amostras foram geradas, tal que 7 são para treino e uma para testar a rede neural. Esse tipo de validação é conhecida como *leave one out* ou validação cruzada.

Como resultado disso, a Tabela 6.1 foi criada para mostrar que a rede neural manteve o equilíbrio e estimou adequadamente os resultados. As linhas da Tabelas (Samples) são o conjunto de ângulos para teste e as colunas da tabela são os erros dos ângulos resultantes.

Tabela 6.1: Validação

Erro RMS [s]	Erro de Yaw	Erro de Pitch	Erro de Roll
Amostra 1	0.0321	0.0034	0.0011
Amostra 2	0.0132	0.0011	0.0018
Amostra 3	0.0134	0.0011	0.0011
Amostra 4	0.0198	0.0027	0.0039
Amostra 5	0.0421	0.0062	0.0016
Amostra 6	0.0324	0.0014	0.0066
Amostra 7	0.0512	0.0013	0.0041
Amostra 8	0.0144	0.0012	0.0020

Dessa forma, como é possível verificar, o resultado parece aceitável mostrando que o treinamento foi correto, embora tenha algumas variações, o erro é coerente.

7 Análise dos resultados

Em resumo, as evidências relevantes e originais desse trabalho são: A modelagem de um estimador de bias que pôde ser minimizado com o algoritmo simplex e o uso do algoritmo *extreme learning machine* para treinar uma rede neural, tal que o objetivo de tais técnicas é, respectivamente, melhorar a acurácia e a precisão do alinhamento.

Considerando que o estado da arte do tema, como a proposta em (ZHU; ZHANG; XU, 2018), indica o algoritmo REQUEST como uma técnica adequada para realizar o alinhamento grosseiro. As otimizações e revisões do REQUEST, propostas neste trabalho, devem ser julgadas como alternativas importantes para as próximas pesquisas.

A fim de estimar o *bias* e melhorar a acurácia do algoritmo REQUEST, nesse trabalho foi definida a Equação 5.5.

Os valores dessa função podem ser plotados através de um algoritmo de busca por força bruta, com isso é possível há um valor mínimo para um conjunto de *bias* específico, os quais são considerados como *bias* estimados.

É imprescindível notar que um conjunto de *bias* bem estimado melhora a acurácia do algoritmo REQUEST. Então, o estimador proposto melhora a performance do auto-alinhamento completo utilizando o algoritmo REQUEST comparado com a solução tradicional.

Somente um conjunto de dados reais estava disponível para realizar os testes. Entretanto, os resultados obtidos com os dados simulados não deixa dúvida que a metodologia é adequada.

A fim de mostrar a confiabilidade da metodologia proposta, 15 ângulos espaçados de cada eixos foram gerados equi-espaçados e combinados, com isso sequência de ângulos *pitch*, *roll* e *yaw* foram geradas. Por fim, a média (RMS) do erro de cada ângulo foi calculada. A comparação de cada algoritmo é listada na tabela 7.1.

Tabela 7.1: Erro médio quadrático de estimação

	Yaw	Pitch	Roll
TRIAD	0.0448	0.0006	0.0006
REQUEST	0.0441	0.0004	0.0006
NREQUEST	0.0233	0.0068	0.0003
EREQUEST	0.0061	0.0001	0.0001

Como pode ser observado, quando o algoritmo REQUEST com estimador é utilizado há uma redução na acurácia de 10 a 7 vezes em relação ao método tradicional. Portanto, a acurácia é altamente melhorada devido a supressão aproximada do *bias*.

A figura 7.1 indica a superioridade do algoritmo EREQUEST contra o algoritmo TRIAD em relação a acurácia.

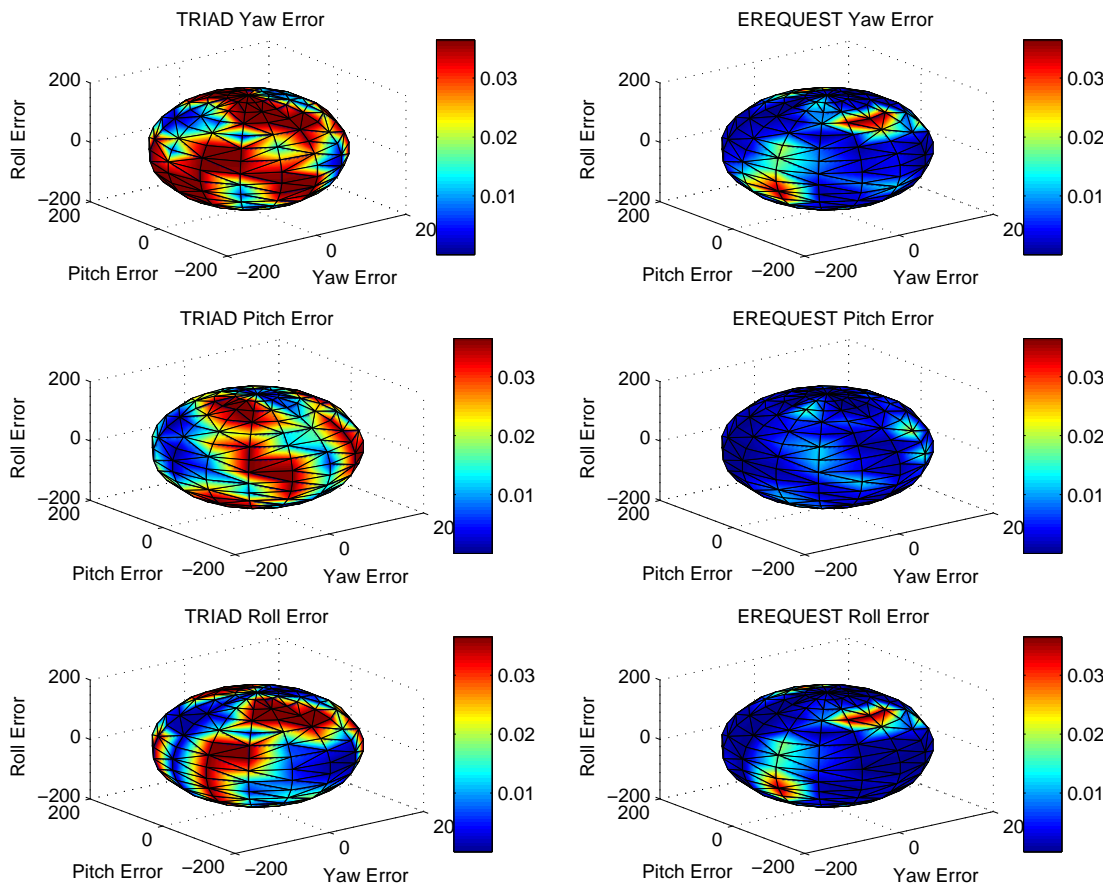


Figura 7.1: Erro em quatro dimensões.

Cada eixo representa um ângulo específico das 256 sequências geradas através das combinação dos 15 ângulos e a cor representa o erro do Yaw, Pitch ou Roll conforme indicado.

Dessa forma, perceba que os ângulos estimados pelo TRIAD ficam com a tonalidade mais avermelhada do que o EREQUEST, o que indica um erro maior na maioria dos

ângulos.

Em relação ao uso da rede neural, a acurácia não possui melhora. Entretanto, o tempo de convergência diminui ao utilizar essa metodologia.

O fato da rede neural mapear qualquer sistema linear, ajuda o sistema estimar ângulos a frente, visto que a rede neural foi treinada para estimar o quatérnion uma iteração à frente.

Diante desse cenário, verifica-se uma bela oportunidade de utilizar a rede neural para diminuir o tempo de convergência do algoritmo EREQUEST.

Com isso, um algoritmo com convergência e acurácia melhorada foi proposto.

8 Conclusões

Nesse trabalho, um novo método de alinhamento grosseiro baseado no algoritmo REQUEST, estimador de *bias* e redes neurais foi demonstrado. Essa técnica utiliza as medidas de acelerômetro e girômetro para realizar a estimativa.

O estimador foi aplicado para melhorar a acurácia do algoritmo REQUEST. Ao combinar o estimador e a aprendizagem de máquina com o algoritmo REQUEST, um novo algoritmo com acurácia e exatidão adequada pode ser projetado.

Lembrando que, o alinhamento utiliza o ângulo calculado pelo alinhamento grosseiro como ponto inicial. Dessa forma, é obvio e trivial que quanto mais próximo o valor inicial é do valor final, o tempo para o alinhamento fino chegar no ponto desejado será menor. Tal resultado é comum e bastante conhecido para todos os estimadores. A prova desta ideia pode ser vista em Bar-Itzhack (1990).

Com isso, o tempo de auto-alinhamento completo irá diminuir. Isto ocorre devido a melhora da acurácia, visto que, se necessário, o alinhamento fino irá demorar menos tempo para corrigir o erro nesse cenário. Além disso, a melhora da precisão finalizará o alinhamento grosseiro antes.

Este trabalho apresentou o uso do algoritmo REQUEST para realizar o Auto-Alinhamento de SNIS. Adicionalmente, um algoritmo recursivo utilizando RNA e estimador de *bias* foi derivado do REQUEST.

Os testes de simulação indicaram que a acurácia e a taxa de convergência do alinhamento melhoraram significativamente. Os algoritmos propostos neste trabalho podem ser úteis em muitas aplicações que requerem o alinhamento em condição quasi-estacionária.

8.1 Sugestões para Trabalho Futuros

Numa próxima etapa, deve-se testar o algoritmo REQUEST com estimador e rede neural, simultaneamente. Além disso, o REQUEST pode ser aplicado variando o tempo usando a noção de Memória de Desvanecimento, na medida em que o peso de contribuições de antigas medições seja reduzido com o tempo como sugerido por Bar-Itzhack (1996).

Outra ideia, seria utilizar médias longas e autocalibrar o sistema. Testes em mesa de rotação são essenciais também para validar os modelos propostos.

Como seguimento deste trabalho, o algoritmo com RNA pode ser testado para realizar estimativas de outros parâmetros do REQUEST. Outras RNAs podem ser testadas, por exemplo, redes de função base radial, redes neurais caóticas, redes Bayesianas, sistemas Neurofuzzy e Redes recorrentes.

No caso da aplicação de redes recorrentes, outros modelos de auto-alinhamento devem ser considerados, por exemplo o auto-alinhamento em movimento. Dessa maneira, uma comparação justa poderia ser realizada para aproveitar a característica dinâmica das redes recorrentes.

Referências

- BAR-ITZHACK, I. Y. *Advances in Aeronautical Systems*. Elsevier, 1990. 369–396 p. (Control and Dynamic Systems, v. 38). ISSN 00905267. ISBN 9780120127382. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780120127382500130>>.
- BAR-ITZHACK, I. Y. Request - a recursive quest algorithm for sequential attitude determination. *Journal of Guidance, Control, and Dynamics*, Vol. 19, n. No. 5, p. pp. 1034–103, 1996.
- BRITTING, K. *Inertial navigation systems analysis*. John Wiley & Sons Canada, Limited, 1971. ISBN 9780471104858. Disponível em: <<https://books.google.com.br/books?id=Mn9TAAAAMAAJ>>.
- CAMPOS, R.; BATISTA, D. S.; JR, F. G.; TOSIN, M. C.; MELO, L. F. de. Auto-alinhamento de sistemas de navegação inercial usando algoritmo request com redes neurais artificiais. *Simpósio Brasileiro de automação Inteligente*, 2017.
- CHRITENSEN, R.; FOGH, N. *Inertial Navigation System*. Tese (Masterthesis) — AALBORG UNIVERSITY, 2008.
- GAO, W.; BEN, Y.; ZHANG, X.; LI, Q.; YU, F. Rapid fine strapdown ins alignment method under marine mooring condition. *IEEE Tran. Aero. Elec. Syst.*, v. 47, n. 2887?2896., 2011.
- GRANZIERA, F. *Simulação e Implementação de um Determinador de Atitude em Tempo Real Utilizando Sensores Microfabricados*. Dissertação (Mestrado) — Departamento de Engenharia Elétrica da Universidade Estadual de Londrina (UEL), 2006.
- GROVES, P. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008. (GNSS technology and applications series). ISBN 9781580532556. Disponível em: <<https://books.google.com.br/books?id=WssZAQAIAAJ>>.
- HAN, S.; KNIGHT, N. Using allan variance to determine the calibration model of inertial sensors for gps / ins integration. 2009. Disponível em: <<http://www.sage.unsw.edu.au/snap/publications/hansetal2009a.pdf>>.
- HAYKIN, S. *Redes Neurais: Princípios e Práticas*. [S.l.]: ARTMED S.A, 1999.
- HORNIK, K. Approximation capabilities of multilayer feedforward networks. *Neural networks*, Elsevier, v. 4, n. 2, p. 251–257, 1991.
- HUANG, G.-B.; BABRI, H. A. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks*, IEEE, v. 9, n. 1, p. 224–229, 1998.
- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: a new learning scheme of feedforward neural networks. In: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. [S.l.: s.n.], 2004. v. 2, p. 985–990 vol.2. ISSN 1098-7576.
- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing*, v. 70, n. 1, p.

- 489 – 501, 2006. ISSN 0925-2312. Neural Networks. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231206000385>>.
- JEKELI, C. *Inertial Navigation Systems with Geodetic Applications*. Walter de Gruyter, 2001. (Lien au titre de la Collection). ISBN 9783110159035. Disponível em: <<https://books.google.com.br/books?id=YRaCD-JHsecC>>.
- JIANG, Y. F. Error analysis of analytic coarse alignment methods. *IEEE Transactions on Aerospace and Electronic*, 1998.
- LAGARIAS, J. C.; REEDS, J. A.; WRIGHT, M. H.; WRIGHT, P. E. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, SIAM, v. 9, n. 1, p. 112–147, 1998.
- LAWRENCE, A. *Modern Inertial Technology: Navigation, Guidance, and Control*. Springer New York, 2012. (Mechanical Engineering Series). ISBN 9781461217343. Disponível em: <<https://books.google.com.br/books?id=2oHlBwAAQBAJ>>.
- LI, H.; PAN, Q.; WANG, X.; JIANG, X.; DENG, L. Kalman filter design for initial precision alignment of a strapdown inertial navigation system on a rocking base. *The Journal of Navigation*, Cambridge University Press, v. 68, n. 1, p. 184–195, 2015.
- LIU, X.; XU, X.; LIU, Y.; WANG, L. A method for sins alignment with large initial misalignment angles based on kalman filter with parameters resetting. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, v. 2014, 2014.
- LIU, Y.; XU, X.; LIU, X.; YAO, Y.; WU, L.; SUN, J. A self-alignment algorithm for sins based on gravitational apparent motion and sensor data denoising. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 15, n. 5, p. 9827–9853, 2015.
- LÜ, S.; XIE, L.; CHEN, J. New techniques for initial alignment of strapdown inertial navigation system. *Journal of the Franklin Institute*, Elsevier, v. 346, n. 10, p. 1021–1037, 2009.
- MARINHA. Tutorial de navegação. *Online*, 2000.
- MARKLEY, F. L. Attitude determination using two vector measurements. 1998.
- OLIVEIRA, F.; Leite Filho, W. Análise do tempo de alinhamento em sistemas de navegação inercial solidários. *Simpósio Brasileiro de Automação Inteligente (SBAI)*, v. 1, p. 1–6, 2013.
- OLIVEIRA, F.; Leite Filho, W. Sobre o procedimento de ortogonalização da DCM no Auto-Alinhamento Grosseiro de sistemas de Navegação Inercial Solidários. In: *Simpósio Brasileiro de Automação Inteligente (SBAI)*. São José dos Campos, SP, Brasil: [s.n.], 2013. Disponível em: <www.sbai2013.ufc.br/pdfs/4272.pdf>.
- PIO, R. Euler angle transformations. *IEEE Transactions on Automatic Control*, v. 11, n. 4, p. 707–715, Oct 1966. ISSN 0018-9286.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. et al. Learning representations by back-propagating errors. *Cognitive modeling*, v. 5, n. 3, p. 1, 1988.
- SANTANA, D. D. S. *Navegação Terrestre Usando Unidade De Medição Inercial de Baixo Desempenho e Fusão Sensorial com Filtro de Kalman Adaptativo Suavizado*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo (USP), 2011.
- SAVAGE, P. G. *Strapdown Analytics*. [S.l.]: Strapdown Associates, 2007.
- SAVAGE, P. G. *Strapdown Analytics*. [S.l.]: Strapdown Associates, 2007.
- SHEELA, K. G.; DEEPA, S. N. Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, v. 2013, 2013.

- SHEN, C.; CAO, H.; LI, J.; TANG, J.; ZHANG, X.; SHI, Y.; YANG, W.; LIU, J. Hybrid de-noising approach for fiber optic gyroscopes combining improved empirical mode decomposition and forward linear prediction algorithms. *Review of Scientific Instruments*, AIP Publishing, v. 87, n. 3, p. 033305, 2016.
- SHIBATA, K.; IKEDA, Y. Effect of number of hidden neurons on learning in large-scale layered neural networks. In: IEEE. *ICCAS-SICE, 2009*. [S.l.], 2009. p. 5008–5013.
- SHUSTER, M. D. A simple kalman filter and smoother for spacecraft attitude. *The Journal of the Astronautical Sciences*, 1989.
- SHUSTER, M. D. Filter quest or request. *Journal of guidance, control, and dynamics*, v. 32, n. 2, p. 643, 2009.
- SHUSTER, M. D.; OH, S. D. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, v. 4, n. 1, p. 70–77, jan. 1981. ISSN 0731-5090.
- SILSON, P. Coarse alignment of a ship's strapdown inertial attitude reference system using velocity loci. *IEEE Trans. Instrum. Meas.*, v. 60., n. 19301941., 2011.
- TAN, C.; ZHU, X.; SU, Y.; WANG, Y.; WU, Z.; GU, D. A new analytic alignment method for a sins. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 15, n. 11, p. 27930–27953, 2015.
- TITTERTON, D.; WESTON, J. *Strapdown Inertial Navigation Technology, 2nd Edition*. Institution of Engineering and Technology, 2004. (IEE radar, sonar, navigation, and avionics series). ISBN 9780863413582. Disponível em: <<http://books.google.com.br/books?id=WwrCrn54n5cC>>.
- TITTERTON, D.; WESTON, J. *Strapdown Inertial Navigation Technology, 2nd Edition*. Institution of Engineering and Technology, 2004. (IEE radar, sonar, navigation, and avionics series). ISBN 9780863413582. Disponível em: <<http://books.google.com.br/books?id=WwrCrn54n5cC>>.
- VELGER, M. *Inertial Navigation And GPS Integration*. [S.l.: s.n.], 200.
- VUOLO, J. H. Fundamentos da teoria de erros. In: LTDA, E. E. B. (Ed.). [S.l.: s.n.], 1992.
- WOODMAN, O. J. An introduction to inertial navigation. n. 696, 2007.
- XU, X.; XU, X.; ZHANG, T.; LI, Y.; WANG, Z. A coarse alignment method based on digital filters and reconstructed observation vectors. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 17, n. 4, p. 709, 2017.
- XU, X.; XU, X.; ZHANG, T.; LI, Y.; TONG, J. A kalman filter for sins self-alignment based on vector observation. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 17, n. 2, p. 264, 2017.
- ZHU, Y.; ZHANG, T.; XU, X. A coarse-alignment method based on the optimal-request algorithm. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 1, p. 239, 2018.

Apêndice A – Produção Científica

Esse capítulo indica, a lista de artigos produzidos, publicados ou submetidos.

A.1 Artigos Publicados

CAMPOS, R.; BATISTA, D. S.; JR, F. G.; TOSIN, M. C.; MELO, L. F. de. Auto-alinhamento de sistemas de navegação inercial usando algoritmo request com redes neurais artificiais. *Simpósio Brasileiro de automação Inteligente*, 2017.

A.2 Artigos Submetidos

CAMPOS, R.; BATISTA, D. S.; JR, F. G.; TOSIN, M. C.; MELO, L. F. de. A comparison of feedforward and recurrent neural networks in aircraft control system. *Sensors*, 2017.

CAMPOS, R.; BATISTA, D. S.; JR, F. G.; TOSIN, M. C.; MELO, L. F. de. A coarse alignment method based on request algorithm with extreme learning machine. *Sensors*, 2018.

CAMPOS, R.; BATISTA, D. S.; JR, F. G.; TOSIN, M. C.; MELO, L. F. de. A self-alignment method for sins based on request algorithm with bias estimator. *Sensors*, 2018.