

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Aeroespacial

**Estimação de atitude do satélite CBERS 4
utilizando filtro H_{∞} Estendido e dados
simulados.**

Autor: José Rômulo Cordeiro Neto
Orientador: Prof.Dr.William Reis Silva

Brasília, DF
2021



José Rômulo Cordeiro Neto

**Estimação de atitude do satélite CBERS 4 utilizando
filtro H_∞ Estendido e dados simulados.**

Monografia submetida ao curso de graduação em (Engenharia Aeroespacial) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Aeroespacial).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof.Dr.William Reis Silva

Brasília, DF

2021

José Rômulo Cordeiro Neto

Estimação de atitude do satélite CBERS 4 utilizando filtro H_∞ Estendido e dados simulados.

Monografia submetida ao curso de graduação em (Engenharia Aeroespacial) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Aeroespacial).

Trabalho aprovado. Brasília, DF, 10 de novembro de 2021:

Prof.Dr.William Reis Silva
Orientador

Prof.Dr.Ronne Toledo
Convidado 1

Prof.Dr.Domenico Simone
Convidado 2

Brasília, DF
2021

*Este trabalho é dedicado a meu pai, minha mãe e minha irmã que me apoiaram durante
minha formação.*

Agradecimentos

Agradeço aos meus pais e minha irmã que buscaram me apoiar ao longo dessa jornada. A todos os meus colegas de curso que compartilharam comigo momentos de superação e resiliência presentes nas adversidades do curso. O meu orientador, William Reis Silva, que buscou fazer o melhor referente a orientação da minha pesquisa. Obrigado por fazerem parte deste momento em que me encontro o qual deixará recordação, saudade e nostalgia daqui alguns anos.

*“As lições aprendidas na dor jamais serão esquecidas, pois só a dor gera a compreensão”
(Paulo Lima Mello)*

Resumo

O seguinte trabalho tem como objetivo o estudo de análise referente ao método de atitude do satélite CBERS 4, utilizando a modelagem matemática do filtro H_∞ Estendido que busca linearizar o sistema de estimação de atitude utilizando como ferramenta matemática a expansão em série de Taylor de primeira ordem. Aspectos teóricos que descrevem o sistema de referência do satélite localizado em seu plano orbital, em relação a terra (referencial inercial), foram utilizados com finalidade de estimação de grandezas físicas e cinemáticas localizadas no sistema de referência do corpo. As particularidades definidas no modelo de sensores foram estudadas em função de três tipos distintos de sensores Giroscópio, Sensor Solar Digital (DSS) e Infravermelho Terrestre (IRES) os quais proporcionam o movimento do satélite em relação ao seu centro de massa, em função do deslocamento rotacional e translacional na órbita em que o mesmo está inserido. A simulação computacional presente no trabalho utilizou o software Matlab e dados simulados com o objetivo de propagação dos vetores posição, velocidade e os parâmetros orbitais descritos, em função dos ângulos de Euler, para uma margem de tempo escolhida. Os modelos de sensores utilizados na simulação foram o IRES (Infravermelho Terrestre) e DSS (Sensor Solar Digital). As estimativas dos vetores de estados foram apresentadas e a análise da precisão do filtro H_∞ Estendido foi verificada, em comparação com o filtro de Kalman Estendido (FKE), e constatou a aplicabilidade da ferramenta em estudos de atitude.

Abstract

The following work, has as purpose the analysis study regarding the altitude method of CBERS 4 satellite using the mathematical modeling of extended H_∞ filter that seeks to linearize the altitude estimation system using the first order of Taylor series as a mathematical tool. The theoretical aspects that describe the satellite reference system located in its orbital plane in relation to earth (inertial reference) were used for the purpose of estimating physical and kinematic quantities located in the body's reference system. The defined particularities in the sensors models were studied in function of three distinct types of sensors the Gyroscope, Digital Solar Sensor (DSS), and Infra-red Earth Sensor (IRES) that provides the movement of the satellite in relation to its center of mass in function of the rotational and translational shifts in the orbit in which it's inserted. The computational simulation present in this study utilized the Matlab software and simulated data in order to propagate the position and velocity vectors and the orbital parameters described according to Euler's angles to a chosen time margin. The sensors models used in the simulation were IRES and DSS. The estimation of the state vectors were presented and the precision analysis of the extended H_∞ filter, was verified in comparison with extended Kalman's filter (EKF) and proved the applicability of the tool in altitude studies.

Lista de ilustrações

Figura 1 – Eixo de coordenadas tridimensional centralizado no centro da terra (plano equatorial) (FEHSE, 2003).	33
Figura 2 – Representação dos ângulos de Euler representado no eixo de coordenadas do satélite indicando sua rotação referente ao centroide. (SILVA, 2016)	34
Figura 3 – Plano orbital do satélite em relação ao centroide da terra (FEHSE, 2003)	35
Figura 4 – Sistema orbital local descrito no plano orbital. (SILVA, 2016)	36
Figura 5 – Estimação dos vetores de estado utilizando o simulador Propat e sua interação com diversos tipos de filtros. (SILVA, 2016).	53
Figura 6 – Propagação dos ângulos de Euler e os Bias definidos pelo modelo matemático.	56
Figura 7 – Propagação dos parâmetros orbitais, velocidade angular, vetor posição, vetor velocidade e variação do IRES e DSS definido pelo modelo matemático.	57
Figura 8 – Estimação dos ângulos de Euler e os Bias utilizando filtro H_∞ Estendido e o filtro de Kalman.	59
Figura 9 – Estimação dos resíduos para cada sensor utilizando filtro H_∞ Estendido e o filtro de Kalman.	60
Figura 10 – Erros para os ângulos de Euler e o Bias utilizando o filtro de Kalman e o filtro H_∞ Estendido.	61
Figura 11 – Função densidade de probabilidade ilustrativa com média μ e desvio padrão σ .) (BUSSAB, 2002)	64

Lista de tabelas

Tabela 1 – Parâmetros orbitas descritos em relação ao plano equatorial terrestre. .	35
Tabela 2 – Condições iniciais para propagação de atitude.	54
Tabela 3 – parâmetros orbitas iniciais.	55
Tabela 4 – Parâmetros escolhidos para filtragem.	58
Tabela 5 – Média e Desvio padrão dos resíduos para cada sensor.	60
Tabela 6 – Média e Desvio padrão dos erros para cada filtro.	62
Tabela 7 – Tempo de execução das rotinas.	62

Lista de abreviaturas e siglas

DSS.	Sensor solar digital.
IRES	Sensor infravermelho terrestre.
FKE	Filtro de Kalman estendido.
FKU	Filtro de Kalman Uscentend.
CBERS	Satélite sino brasileiro.
PMR	Parâmetros modificados de Rodrigues.

Lista de símbolos

ψ	Ângulo de guinada.
θ	Ângulo de rolamento.
ϕ	Ângulo de arfagem.
x_0	Componente x do vetor posição do referencial não inercial.
y_0	Componente y do vetor posição do referencial não inercial.
z_0	Componente z do vetor posição do referencial não inercial.
x	Componente x do vetor posição do referencial girante.
y	Componente y do vetor posição do referencial girante.
z	Componente z do vetor posição do referencial girante.
O_{op}	Centroide da terra .
x_{op}	Eixo de coordenadas do plano orbital na direção do nó ascendente.
y_{op}	Eixo perpendicular ao produto vetorial dos versores k e j.
z_{op}	Eixo perpendicular ao plano orbital.
ω	Ascensão reta do nó ascendente.
β	Rotação angular em relação ao nó ascendente.
\mathfrak{R}	Matriz de mudança de coordenadas do referencial inercial para o local.
φ	Latitude em relação ao eixo de coordenada terrestre.
ω_0	Velocidade angular da taxa de transferência do referencial inercial para o orbital.
$\hat{\omega}_x$	Componente x dos bias de giro do vetor velocidade angular.
$\hat{\omega}_y$	Componente y dos bias de giro do vetor velocidade angular.
$\hat{\omega}_z$	Componente z dos bias de giro do vetor velocidade angular.
$\Delta\theta_i$	Variação da velocidade angular em relação aos eixos x, y e z do modelo de sensoriamento do giroscópio.

ω_i	Vetor velocidade angular do modelo de sensoriamento do giroscópio.
β_i	Bias do vetor velocidade angular do modelo de sensoriamento do giroscópio.
Δt	Variação do tempo do modelo de sensoriamento do giroscópio.
\vec{g}	Vetor de saída do modelo de sensoriamento do giroscópio .
$\vec{\eta}$	Ruído gaussiano branco referente ao modelo de sensoriamento do giroscópio .
Q_i	diagonal principal da matriz de covariância da função .
$\delta(t)$	delta de kroneger.
$\nu_{\phi H}$	Ruído branco gaussiano em relação ao ângulo de arfagem do sensor infravermelho terrestre .
$\nu_{\theta H}$	Ruído branco gaussiano em relação ao ângulo de rolamento do sensor infravermelho terrestre .
S_x^*	Componente x do vetor unitário solar .
S_y^*	Componente y do vetor unitário solar .
S_z^*	Componente z do vetor unitário solar .
w_k	Ruído presente na estimação do vetor de estado.
$\nu_{\alpha\psi}$	Ruído Gaussiano branco de guinada referente ao (DSS) .
$\nu_{\alpha\theta}$	Ruído Gaussiano branco de rolamento referente ao (DSS).
x_{k+1}	Vetor de estado posterior.
y_k	Função de estado.
x_k	Vetor de estado.
L_k	Matriz rank completo.
z_k	Vetor de estado real .
\hat{z}_k	Vetor de estado real estimado.
x_0	Condição inicial.
\hat{x}_0	estimativa da condição inicial.

γ	coeficiente de performance.
S_k	Matriz de ponderação do estado real e estado estimado.
K_k	Ganho do sistema.
P_{k+1}	Matriz de incerteza do sistema.
Φ_{k-1}	Matriz de transição.
\tilde{Q}_k	Matriz de ruído de processo.
\tilde{R}_k	Matriz de ruído de medida.
J_1	Função custo

Sumário

1	INTRODUÇÃO	25
1.1	Objetivos gerais	26
1.1.1	Objetivos específicos	26
1.2	Estrutura do trabalho	26
1.2.1	Parte 1 - Revisão bibliográfica	26
1.2.2	Parte 2 - Simulação computacional	26
1.2.3	Conteúdo	26
2	REVISÃO BIBLIOGRÁFICA	29
3	SISTEMA DE REFERÊNCIA	33
3.1	Ângulos de Euler	34
3.2	Eixo de referência orbital	34
3.2.1	Eixo de referência orbital local	35
3.2.2	sistema de referência de controle orbital e atitude.	36
3.2.3	Estudo do movimento de atitude do satélite utilizando ângulos de Euler	37
4	SISTEMA DE SENSORES	39
4.1	Análise de funcionamento do modelo de sensores de atitude	39
4.1.1	sistema de sensor por meio de Giroscópio	39
4.1.2	Sistema de sensor Infravermelho Terrestre (IRES)	40
4.1.3	Sistema de sensor solar digital (DSS)	41
5	MODELO MATEMÁTICO DE ESTIMAÇÃO	43
5.1	O filtro de Kalman Estendido	43
5.1.1	Solução para o filtro de Kalman Estendido	43
5.2	filtro H_{∞} Estendido	44
5.2.1	Solução do filtro H_{∞} Estendido	47
5.3	Relação entre o filtro H_{∞} estendido e o filtro de Kalman	49
6	SIMULAÇÃO COMPUTACIONAL	51
6.1	Modelo computacional para análise do filtro H_{∞} Estendido utilizando dados simulados.	51
6.1.1	Parâmetros iniciais de estimação de atitude utilizando filtro H_{∞} Estendido.	51
6.2	Estimação de atitude utilizando filtro de Kalman Estendido.	51
6.2.1	Condições iniciais para filtragem utilizando o Filtro de Kalman Estendido.	51
6.2.2	Modelos de sensores.	52

6.2.3	Cálculo das matrizes Jacobianas do filtro de Kalman Estendido.	52
6.3	Estimação de atitude utilizando filtro H_∞ Estendido.	52
6.3.1	Condições iniciais	52
6.3.2	Atualização do filtro H_∞ Estendido	52
6.3.3	Predição do Filtro H_∞ Estendido	53
6.4	Estimação de atitude utilizando dados simulados	53
6.4.1	Condições iniciais para análise de atitude	54
6.4.2	Ângulos de Euler	54
6.4.3	Tensor de inércia	54
6.4.4	Parâmetros orbitais	55
6.4.5	Integrador numérico ODE 45	55
6.4.6	Resultados do modelo matemático	56
6.5	Resultados Utilizando o filtro de Kalman e o filtro H_∞ Estendido . .	58
7	CONCLUSÃO	63
7.1	Considerações finais	63
7.1.1	Observação dos resultados para o FKE e o Filtro H_∞ estendido.	63
7.1.2	Função densidade de probabilidade e desvio padrão dos resultados.	64
7.1.3	Extensão do filtro H_∞ estendido.	64
8	TRABALHOS FUTUROS	65
	REFERÊNCIAS	67
	APÊNDICE A – ALGORITMO DEMO PROPAT REAL STATE.M PROPAGAÇÃO DO MODELO MATEMÁTICO.	69
	APÊNDICE B – ALGORITMO DEMO PROPAT EKF.M DESCREVE O FILTRO DE KALMAN ESTENDIDO.	81
	APÊNDICE C – ALGORITMO DEMO PROPAT EHINF.M DES- CREVE O FILTRO H_∞ ESTENDIDO.	97

1 Introdução

Para melhor esboçar os aspectos de controle e estimação de atitude de satélites a análise teórica dos controles dinâmicos e de sensores foram formuladas utilizando a modelagem matemática e teórica dos ângulos de Euler ψ (guinada), θ (Rolamento) e ϕ (arfagem). Todos os parâmetros foram formulados utilizando a escolha do sistema de referência das matrizes de rotação de cada ângulo. Em função desses fatores, é possível estimar a posição do satélite em função das condições iniciais de um vetor posição em um dado referencial inercial. Outras grandezas físicas como velocidade, aceleração e torques aplicados a estrutura do satélite por meio do sistema dinâmico também podem ser estimadas em estudos de controle de atitude.

Para que uma determinada grandeza seja estimada é preciso que um sistema de sensor adequado possa ser aplicado. Os diferentes tipos de filtros buscam estipular essa análise de dados para que o sistema de sensores possa descrever o movimento do satélite em relação ao seu eixo de referência. Existe uma série de filtros, cada um com suas especificações, e no caso do trabalho apresentado o filtro escolhido foi o filtro H_∞ Estendido.

A formulação prática de estudo do problema para trabalhos posteriores utilizará como material de estudo os dados de atitude do satélite CBERS 4 (Satélite de Pesquisa Sino Brasileiro) que atualmente está em órbita.

O sistema descrito para estimativa de movimento do satélite, em relação a atitude, é descrito utilizando modelos não lineares. Entretanto, existem métodos de análises numéricas que buscam otimizar o processo de estimação, por meio de modelagens matemáticas como Ricatti ou expansão em série de Taylor, dada um condição inicial para avaliação dos parâmetros especificados.

O filtro H_∞ Estendido tem como característica maior precisão em relação a outros filtros estudados como o de Kalman Unscented (FKU) e o Kalman Estendido (FKE), ainda existe o filtro de kalman Sigma-Ponto que possui maior precisão se comparado com o filtro de Kalman Estendido, caso as condições iniciais escolhidas estipulem valores imprecisos (GARCIA, 2011).

Todos os filtros não possuem precisão máxima, os dados esboçados pelos sistemas apresentam erros que no caso das medições precisam ser especificados. Os Bias de giro (erros) são apresentados descritos de acordo com a movimentação do sistema. Em consequência disso, o filtro presente no sistema do satélite tem função importante na missão espacial, pois busca preservar a carga útil do sistema.

1.1 Objetivos gerais

- Aplicar a metodologia do filtro H_∞ Estendido para estimação de dados simulados de atitude do satélite CBERS 4.

1.1.1 Objetivos específicos

- Revisão bibliográfica de aspectos teóricos do filtro H_∞ Estendido como ferramenta de estimação de atitude.
- Especificar os gráficos referente as variações dos ângulos de Euler em relação ao tempo e os Bias de giros do satélite CBERS 4, utilizando o software Matlab.

1.2 Estrutura do trabalho

1.2.1 Parte 1 - Revisão bibliográfica

Este módulo tem como objetivo principal descrever de maneira teórica como a metodologia dos filtros pode ser utilizada em diversas funções, e demonstrar como a metodologia do filtro H_∞ Estendido pode ser aplicada em estimativas de atitude de satélites em órbitas em torno da terra.

1.2.2 Parte 2 - Simulação computacional

Após a descrição teórica dos aspectos do filtro H_∞ estendido o software Matlab será utilizado para apresentar os gráficos dos parâmetros de atitude, variação dos ângulos de Euler em relação ao tempo e Bias de giro, Parâmetros orbitais, Ascensão Reta do Nódulo Ascendente, Argumento do Perigeu, Semieixo Maior, Excentricidade, Anomalia Média e Inclinação Orbital. Os vetores velocidade e posição serão propagados em função do deslocamento do satélite na órbita adotando sequência de rotação 3 - 2 - 1 utilizando dados simulados. A análise teórica será aplicada para o estudo da precisão do filtro H_∞ Estendido, utilizando simulações práticas no Matlab, para o estudo de estimação e metodologia de representação de atitude.

1.2.3 Conteúdo

Capítulo 1: Descreve de maneira introdutória os conceitos abordados na temática do estudo, fornecendo uma análise generalizada da ferramenta.

Capítulo 2: Aborda as pesquisas utilizadas para argumentação do trabalho, disponibilizando um breve resumo dos documentos científicos, especificando os autores.

Capítulo 3: Elucida o sistema de representação de atitude escolhido, representado pelos ângulos de Euler, detalha a modelagem matemática da transferência dos sistemas de referência orbital. Estipula a análise cinemática do movimento de satélites inseridos em órbitas.

Capítulo 4: Detalha alguns dos tipos de sensores utilizados em métodos de estimação de atitude. Os sensores abordados são os modelos de Giroscópio, Infravermelho Terrestre (IRES) e o Sensor Solar Digital (DSS).

Capítulo 5: Apresenta a modelagem matemática do método de estimação de atitude representado pelo filtro H_∞ Estendido e descreve uma breve comparação com o filtro de Kalman.

Capítulo 6: Aplica a análise computacional utilizando a metodologia do filtro H_∞ Estendido e o filtro de Kalman Estendido (FKE), produzindo os parâmetros de comparação dos vetores de estado estimados pelos dois sistemas.

Capítulo 7: Apresenta a conclusão explicativa dos resultados da simulação computacional em formalização com aspectos teóricos utilizados ao longo do trabalho.

Capítulo 8: Descreve a continuação do trabalho para períodos posteriores buscando aplicar a metodologia do filtro H_∞ estendido utilizando dados reais do satélite CBERS 4, formaliza um novo sistema de representação de atitude e outros modelos de sensores.

Capítulo 9: Apresenta o referencial bibliográfico utilizado na pesquisa.

O último ponto do trabalho descreve os apêndices que são representados pelas rotinas descritas no software Matlab para a execução da simulação computacional.

2 Revisão Bibliográfica

O seguinte capítulo descreve as bibliografia utilizada para a formalização do trabalho, os conceitos explicitados em análises teóricas são definidos como direcionadores do objeto de pesquisa, o conteúdo é abordado em relação a data de publicação mais antiga até a pesquisa mais recente. Os documentos científicos buscam definir base argumentativa para validar aspectos de estimação de atitude utilizando o filtro H_∞ Estendido metodologia dos ângulos de Euler e estimação de atitude.

Em [Michelson \(1964\)](#) o estudo do movimento de satélites em relação ao seu centro de massa é definido, no entanto, define as diferenças e restrições quando o movimento do satélite utiliza a metodologia para os ângulos de Euler, apresenta suas particularidades em função de singularidades e valores de ângulos pequenos que não podem ser estimados utilizando a modelagem. Além disso, compara a metodologia com a análise em quatérnions que não apresenta os problemas descritos.

[Wertz \(1978\)](#) disponibiliza os modelos de sensores e suas especificações para modelos de estimação de atitude. O detalhamento dos sensores é descrito no capítulo 7 o qual descreve as modelagens matemáticas do Sensor Solar Digital, Giroscópio, Sensor de Horizonte, Magnetómetro e Sensor Estrela. Os modelos apresentados são amplamente utilizados em sistemas de representação de atitude.

[Chobotov \(1991\)](#) determina a modelagem matemática para o movimento de corpos rígidos considerando o movimento do corpo em função de um referencial inercial. A análise cinemática e dinâmica é descrita para corpos localizados em órbitas, conceitos da mecânica espacial e mecânica newtoniana são utilizados para descrever os sistemas.

Em [DISCEPOLI \(1992\)](#) o registro do lançamento do satélite Intelsat 1 é transcrito além dos subsistemas presentes no projeto, utilizando o veículo lançador Ariane 441 como transportador. O detalhamento dos aspectos estruturais dos sistemas do satélite são definidos como a carga útil e suas interações com a plataforma de dados presentes na superfície terrestre.

[SHEN X.; DENG \(1997\)](#) estabelece a metodologia quadrática linear de jogos para a sistemática do filtro H_∞ Estendido com finalidade de produzir uma modelagem discreta para a ferramenta. A consequência é evidenciada pela estimativa do erro referente a uma condição inicial proposta cujo resultado é diminuído utilizando a formulação de Ricatti. A avaliação da precisão é verificada com a comparação do filtro H_∞ Estendido e o Filtro de Kalman.

[Bussab \(2002\)](#) descreve a apresentação de ferramentas básicas no campo da esta-

tística e probabilidade as quais podem ser aplicadas em objetos de estudo que buscam trabalhar com a estimação de variáveis aleatórias.

Fehse (2003) estipula a análise informacional para engenheiros aeroespaciais do funcionamento do fenômeno de Rendezvous e Docking. Os conceitos são viabilizados para aplicação direta em missões espaciais as quais os conhecimentos sobre o processo são necessários.

Simon (2006) apresenta de forma específica a modelagem matemática de diversos filtros, a especificação teórica do filtro H_∞ Estendido é apresentada na part III. Além disso, a comparação com o filtro de Kalman é descrita no mesmo capítulo relacionando suas diferenças e limitações.

Garcia (2011) explicita a comparação de estimação de atitude de satélites aplicando o filtro de Kalman Sigma Ponto e o Filtro de Kalman Estendido, as verificações são determinadas utilizando parâmetros reais de tempo. A verificação dos resultados é estudada em função dos satélites CBERS 2 e CBERS 2B referente a análise estocástica.

Em HU J. S.; YANG (2011) o estudo comparativo da estimação de estados aplicando o mecanismo do filtro H_∞ Estendido de segunda ordem e o filtro de Kalman Estendido de segunda ordem é fornecido. A conclusão dos resultados apresenta redução da matriz quadrática erro, a variação de tempo é definida como discreta e não linear.

Em Garcia e Zanardi (2012) o algoritmo de estimação em tempo real da estimação de atitude presente no payload do satélite CBERS 2 é descrito, o modelo utilizado é o FKU (filtro de Kalman Unscented) o qual não necessita de linearização das funções para a estimativa dos vetores de estados. A comparação dos dados reais do sistema foram analisadas pelos resultados do filtro de Kalman Estendido (FKE) e o FKU.

Silva (2012) busca analisar a estabilidade de rotação de satélites simétricos em órbitas circulares expostos a torques gravitacionais. O método utilizado busca reduzir a Hamiltoniana até a 4ª ordem aplicando transformações canônicas, satélites de pequeno e médio porte viabilizaram a aplicação da modelagem empregando dados com similaridade real. Os resultados definiram pontos de equilíbrio de estabilidade dos satélites e regiões estáveis em torno dos pontos estudados.

Em Matos (2014) a abordagem do estudo de estimação de atitude de satélites aplicando a metodologia do Filtro de Kalman Sigma Ponto e o modelo de representação de atitude dos parâmetros modificados de Rodrigues (PMR) é utilizado. A matriz de covariância no método convencional de utilização e quatérnions apresentam singularidade, em consequência disso, o PMR foi utilizado para eliminar o problema na análise. Os sensores utilizados foram o Giroscópio, Sensor Solar e Sensores de Terra introduzidos no satélite CBERS 2. Os resultados foram comparados com outras ferramentas de representação de atitude correlacionadas em função da precisão dos vetores de estado definidos pelos

ângulos de Euler.

[Silva \(2016\)](#) define a aplicação da metodologia do Filtro H_∞ Estendido e o Filtro de Partículas Regularizados com finalidade de estimação de atitudes e os Bias do giros. Os dados reais do satélite CBERS 2 foram utilizados de acordo com parâmetros orbitais do satélite. A cinemática foi definida pelo modelo dos ângulos de Euler, os sensores utilizados foram dois Sensores Digitais Solares (DSS) e dois Infravermelhos Terrestres (IREs). A dinâmica de aplicação busca viabilizar a extensão dos modelos de filtro para uma estimativa não linear.

[Ricarte \(2016\)](#) descreve um estudo sobre os Giroscópios de fibra óptica de acordo com o funcionamento e suas limitações para uma gama dinâmica de $60 \frac{^\circ}{s}$ e incerteza de $1 \frac{^\circ}{s}$. A proposta de um modelo interferométrico é definida na seção de resultados o trabalho pode ser estudado para aspectos mais específicos descritos pela área de movimento de micro-satélites.

3 Sistema de Referência

Para melhor descrever o movimento de um corpo de maneira dinâmica e cinemática é preciso descrever um sistema de coordenadas em que grandezas físicas como aceleração e velocidade do corpo rígido podem ser descritas (CHOBOTOV, 1991). Em função de um sistema referencial os conceitos físicos de estudo de um corpo podem ser explicitados por um eixo de coordenadas de origem O_i e um triedro de versores ortogonais (SILVA, 2016). Para corpos que se encontram em um dado referencial inercial (repouso ou movimento retilíneo uniforme), as leis físicas básicas descritas por Newton podem descrever o sistema com facilidade e as grandezas físicas relacionadas a ele, já para corpos que não se encontram em dado sistema de referência alguns conceitos físicos precisam ser aplicados para melhor descrever o sistema. A figura 1 retrata o sistema de coordenadas tridimensional situado na terra.

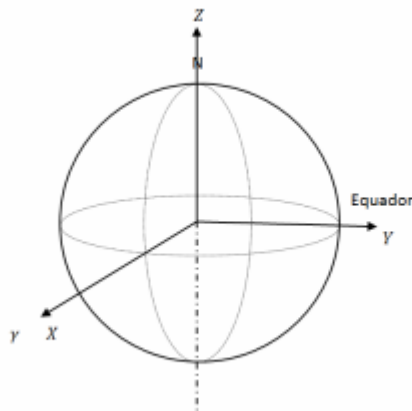


Figura 1 – Eixo de coordenadas tridimensional centralizado no centro da terra (plano equatorial) (FEHSE, 2003).

3.1 Ângulos de Euler

Para que o sistema de atitude de um satélite seja analisado, ou seja, o deslocamento do equipamento em relação ao seu centroide possa ser descrito, é necessário que o satélite apresente ângulos que possam limitar esse movimento. Os ângulos que tem a capacidade de descrever esse sistema são os ângulos de Euler que limitam a rotação livre dos corpos rígidos (MICHELSON, 1964). Para descrever o movimento do satélite, em relação ao seu eixo de coordenadas e o tempo, é preciso que o mesmo esteja interligado a um sistema de referência inercial ou quase inercial, para o caso descrito o sistema escolhido é a terra (SILVA, 2016).

O sistema de coordenadas inercial (x_0, y_0, z_0) e o não inercial (x, y, z) do referencial girante (não inercial) possuem movimento em função dos ângulos de Euler (ψ, θ, ϕ), a sequência de rotação adotada é a 3-2-1 (FUMING H.; KUGA, 1999). Em que ψ é o ângulo de guinada, θ o ângulo de rolamento e ϕ ângulo de arfagem. A figura 2 abaixo retrata o sistema de coordenadas do satélite em função dos ângulos de Euler.

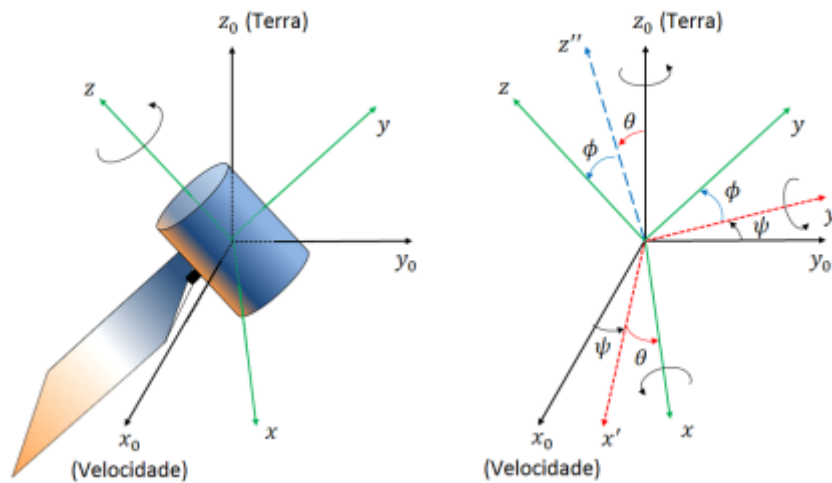


Figura 2 – Representação dos ângulos de Euler representado no eixo de coordenadas do satélite indicando sua rotação referente ao centroide. (SILVA, 2016)

3.2 Eixo de referência orbital

De acordo com as especificações do sistema terrestre a terra é apresentada como uma esfera, ou seja, centro de massa e centro geométrico são coincidentes (SILVA, 2016). A figura 3 abaixo representa a posição fictícia da órbita descrita pelo satélite em relação ao centro da terra.

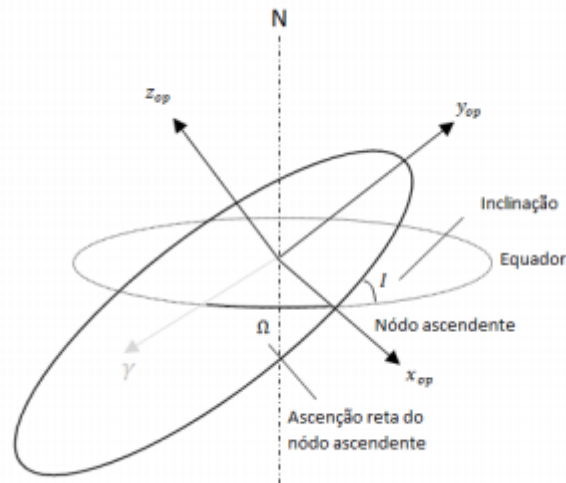


Figura 3 – Plano orbital do satélite em relação ao centroide da terra (FEHSE, 2003)

A tabela abaixo descreve os parâmetros orbitais descritos na figura em relação ao plano equatorial terrestre:

Tabela 1 – Parâmetros orbitas descritos em relação ao plano equatorial terrestre.

Parâmetros plano orbital	
O_{op}	centróide da terra
x_{op}	Eixo de coordenadas do plano orbital na direção nódo ascendende
y_{op}	eixo perpendicular ao produto vetorial dos versores k e j
z_{op}	eixo perpendicular ao plano orbital

Para descrever a passagem do sistema de referência do plano equatorial terrestre para o plano orbital utiliza-se a matriz de rotação que descreve a rotação em relação a ω Ascensão Reta do Nódulo Ascendente, em relação ao eixo z e uma inclinação (I) no eixo x_{op} , a matriz de rotação para descrever a mudança de coordenadas é descrita abaixo.

$$\begin{bmatrix} x_{op} \\ y_{op} \\ z_{op} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos I & \sin I \\ 0 & -\sin I & \cos I \end{bmatrix} \begin{bmatrix} \cos \Omega & \sin \Omega & 0 \\ -\sin \Omega & \cos \Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.1)$$

3.2.1 Eixo de referência orbital local

Após a mudança do eixo de coordenadas terrestre para o plano orbital é de principal importância que o movimento do satélite seja descrito, utilizando-se um sistema de coordenada local que descreve seu movimento em relação a pontos específicos presentes na órbita em que o mesmo se encontra (SILVA, 2016). A figura 4 abaixo descreve o sistema em relação ao plano orbital.

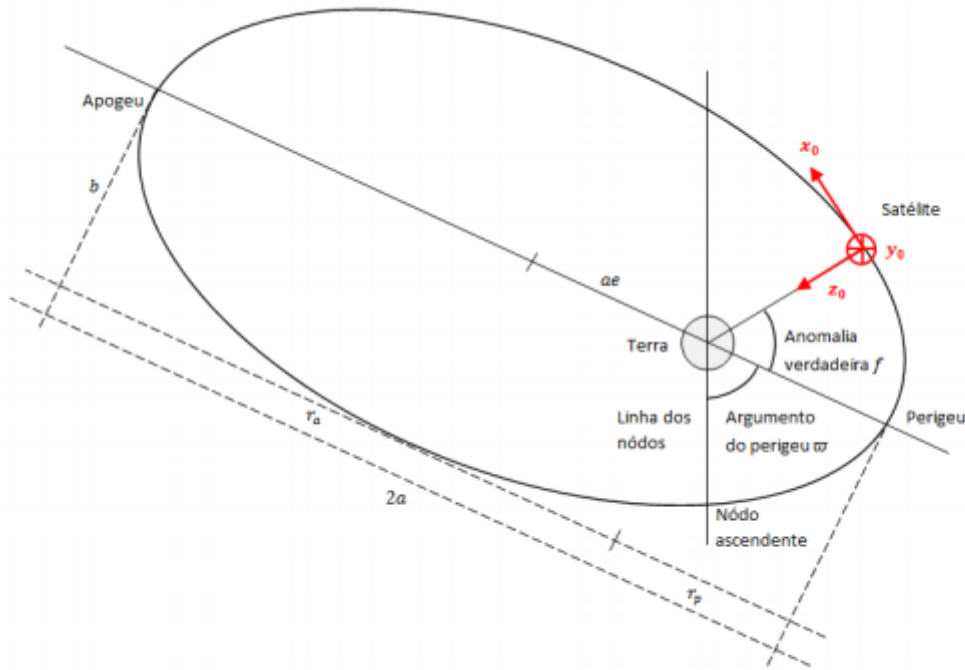


Figura 4 – Sistema orbital local descrito no plano orbital. (SILVA, 2016)

Para descrever de maneira específica a mudança de coordenadas do sistema orbital para o sistema orbital local do satélite em órbita é necessário que as coordenadas sejam descritas pela a matriz de rotação abaixo.

$$\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \cos\left(\beta + \frac{\pi}{2}\right) & \sin\left(\beta + \frac{\pi}{2}\right) & 0 \\ -\sin\left(\beta + \frac{\pi}{2}\right) & \cos\left(\beta + \frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{op} \\ y_{op} \\ z_{op} \end{bmatrix} \quad (3.2)$$

Em que o valor de β descrever a rotação em relação ao nodo ascendente e uma rotação de $\frac{\pi}{2}$ em relação ao eixos x e z_{op} .

3.2.2 sistema de referência de controle orbital e atitude.

O subsistema de controle orbital e atitude descreve as características de vôo e deslocamento do satélite em relação ao seu eixo de rotação em função da missão espacial. (DISCEPOLL, 1992)

Para que ocorra o controle de atitude (deslocamento em relação ao centroide) o sistema é controlado em relação a posição da órbita de transferência por um sistema de tração que gera torques em relação ao eixo de coordenadas tridimensional situado no centro de massa do satélite, sensores ópticos (Terra e Sol) e inerciais (Giroscópio)

podem ser utilizados (DISCEPOLL, 1992). O sistema de atitude descreve o movimento translacional e rotacional do satélite em relação a um sistema de coordenada fixo referente a um sistema inercial que no caso é a terra em determinado instante de tempo (SILVA, 2012). Utilizando os parâmetros rotacionais presentes no sistema de rotação 3-2-1 descritos pelos ângulos de Euler, em relação ao movimento do satélite, a matriz abaixo descreve a passagem do sistema de referência inercial (Terra) para o local definido na órbita.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \quad (3.3)$$

$$\mathfrak{R} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \sin \psi \cos \phi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (3.4)$$

Após a transferência do sistema de referência inercial para o orbital a análise das grandezas físicas e cinemáticas podem ser descritas para o movimento do satélite em relação ao plano orbital.

3.2.3 Estudo do movimento de atitude do satélite utilizando ângulos de Euler

O equacionamento descrito é desenvolvido em (WERTZ, 1978) e citado em (FUMING H.; KUGA, 1999) aplicando o desenvolvimento da matriz \mathfrak{R} o vetor velocidade angular para as componentes x, y e z pode ser encontrado, a matriz abaixo retrata o sistema:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \hat{\omega}_x \\ \hat{\omega}_y \\ \hat{\omega}_z \end{bmatrix} - \mathfrak{R} \left\{ \Omega_e \begin{bmatrix} \cos \varphi \\ 0 \\ \sin \varphi \end{bmatrix} + \omega_0 \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \right\} \quad (3.5)$$

O valor de Ω_e representa a velocidade angular da terra que será negligenciada, pois o corpo não encontra-se em solo terrestre, φ é a latitude em relação ao eixo de coordenada terrestre e ω_0 é a velocidade angular da taxa de transferência do sistema de coordenadas terrestre os valores de $(\hat{\omega}_x, \hat{\omega}_y, \hat{\omega}_z)$ são os Bias de giro referente ao vetor velocidade angular.

Para descrever a variação dos ângulos de Euler em relação ao sistema descrito a matriz abaixo é utilizada.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \left\{ \begin{bmatrix} \hat{\omega}_x \\ \hat{\omega}_y \\ \hat{\omega}_z \end{bmatrix} - \mathfrak{R} \begin{bmatrix} 0 \\ \omega_0 \\ 0 \end{bmatrix} \right\} \quad (3.6)$$

O vetor de estado x pode ser descrito pelos os ângulos de Euler e os Bias de giro:

$$x = \begin{bmatrix} \phi & \theta & \psi & \varepsilon_x & \varepsilon_y & \varepsilon_z \end{bmatrix}^T \quad (3.7)$$

E por fim, desenvolvendo a algebra matricial do sistema da taxa de variação dos ângulos de Euler em função do tempo e suas Bias de giro, o seguinte sistema descrito pelo espaço de estados fornecido abaixo é encontrado:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\varepsilon}_x \\ \dot{\varepsilon}_y \\ \dot{\varepsilon}_z \end{bmatrix} = \begin{bmatrix} \hat{\omega}_x + \hat{\omega}_y \sin \phi \tan \theta + \hat{\omega}_z \cos \phi \tan \theta + \omega_0 (\cos \phi \sin \psi + \sin \theta \sin \psi \tan \theta) \\ \hat{\omega}_y \cos \phi - \hat{\omega}_z \sin \phi + \omega_0 \cos \psi \\ \left(\frac{\hat{\omega}_y \sin \phi + \hat{\omega}_z \cos \phi + \omega_0 \sin \theta \sin \psi}{\cos \theta} \right) \\ 0 \\ 0 \\ 0 \end{bmatrix} + w \quad (3.8)$$

4 Sistema de sensores

4.1 Análise de funcionamento do modelo de sensores de atitude

De acordo com os tópicos descritos na teoria dos sistemas de referência, e suas características no modelo orbital, é necessário no projeto de construção do satélite a escolha do modelo de sensores específicos para melhor descrever seu movimento em relação a um dado referencial inercial que no caso específico será a Terra. Os sensores abordados podem ser divididos em modelos de Giroscópio, Infravermelho Terrestre (IRES) e Sensor Solar Digital (DSS) a descrição de cada tipo de sensor é descrita abaixo de acordo com suas especificações.

4.1.1 sistema de sensor por meio de Giroscópio

O sistema de sensores descrito por meio de Giroscópio tem como característica a facilidade de estimação do deslocamento e velocidade angular que o satélite descreve ao longo do tempo (GARCIA; ZANARDI, 2012). O Giroscópio tem funcionalidade em função de um torque de controle que busca ajustar sua orientação, o equipamento é composto por uma roda que gira em alta velocidade e responde com sensibilidade as mudanças do seu eixo de rotação que é o mesmo do satélite. Os erros produzidos de acordo com o movimento são designados como Bias de giros e serão adicionados ao vetor de estado estimado pelo satélite (MATOS, 2014). Além do modelo mecânico existe o modelo óptico que são insensíveis a aceleração e vibrações mecânicas e possuem tempo de vida longo e custo de produção reduzido (RICARTE, 2016). As equações estudadas no seguinte capítulo são definidas para o modelo mecânico.

Em função desses fatores, os integradores de velocidade podem ser utilizados para fornecer as velocidades angulares de guinada, arfagem e rolamento. O modelo matemático dos integradores é descrito abaixo(WERTZ, 1978).

$$\Delta\theta_i = \int_0^{\Delta t} (\omega_i + \beta_i) dt \quad (4.1)$$

$\Delta\theta_i$ = Variação da velocidade angular em função dos eixos $i = (x, y, z)$.

ω_i = Vetor velocidade angular.

β_i = Bias relacionadas a velocidade angular.

Δt = Variação do tempo

O vetor velocidade angular é descrito como :

$$\vec{\omega} = \left(\frac{\Delta\Theta}{\Delta t} \right) - \vec{\beta} - \vec{\eta}_1 = \vec{g} - \vec{\varepsilon} - \eta_{att} \quad (4.2)$$

Em que o valor de \vec{g} representa o vetor de saída do Giroscópio e $\vec{\eta}$ o ruído gaussiano branco referente ao acúmulo de efeitos remanescentes não modelados, para que esses efeitos possam ser descritos de maneira precisa, é necessário definir a função esperança descrita abaixo.

$$E[x] = x \sum p(x) \quad (4.3)$$

Para o deslocamento do Giroscópio a função esperança é definida como:

$$E[\eta_i(t)] = 0 \quad (4.4)$$

$$E[\eta_i(t)\eta_i(T + \Delta t)] = Q_i\delta(t) \quad (4.5)$$

Q_i = Diagonal principal da matriz de covariância da função.

$\delta(t)$ = Delta de kroneger

Após a variação do deslocamento angular em função do tempo os erros de medida (Bias) tem necessidade de ser especificados pelos os ruídos brancos gerados. As fórmulas abaixo descrevem a descrição dos Bias para o modelo do Giroscópio:

$$\dot{\varepsilon}_i = \omega_{\varepsilon i}(t) \quad (4.6)$$

$$E[\omega_{\varepsilon i}(t)] = 0 \quad (4.7)$$

$$E[\omega_{\varepsilon i}(t)\omega_{\varepsilon i}(t + \Delta t)^T] = Q_{\varepsilon i}\delta(t) \quad (4.8)$$

4.1.2 Sistema de sensor Infravermelho Terrestre (IRES)

A aplicação do sensor infravermelho terrestre tem funcionalidade em relação a sua localização e estimação no eixo longitudinal (rolamento) e transversal (arfagem) com certo nível de erro. O sensor infravermelho detecta o calor presente na atmosfera terrestre e possui sistema de referência que interliga o centro da terra com o eixo de simetria do satélite, as equações do sensores terrestres são esboçadas abaixo (FUMING H.; KUGA, 1999).

$$\phi_H = \phi + \nu_{\phi H} \quad (4.9)$$

$$\theta_H = \theta + \nu_{\theta H} \quad (4.10)$$

Em que os valores de $\nu_{\phi H}$ e $\nu_{\theta H}$ correspondem as parcelas referentes aos ruídos brancos gaussianos do sensor em relação a parâmetros de desalinhamento (erros) as formulações abaixo especificam esses desalinhamentos.

$$E[\nu_{\theta H}(t)] = 0 \quad (4.11)$$

$$E[\nu_{\phi H}(t)] = 0 \quad (4.12)$$

$$E[\nu_{\theta i}(t)\nu_{\theta i}(t + \Delta t)^T] = U_{\nu_{\theta i}(t)}\delta(t) \quad (4.13)$$

$$E[\nu_{\phi i}(t)\nu_{\phi i}(t + \Delta t)^T] = U_{\nu_{\phi i}(t)}\delta(t) \quad (4.14)$$

$U_{\nu_{\phi}(t)}$ = Diagonal da matriz de covariância da função erro.

4.1.3 Sistema de sensor solar digital (DSS)

O sensor solar digital tem como funcionalidade principal o posicionamento do satélite em relação a posição em que o sol foi detectado de acordo com a luminosidade exposta em relação ao eixo de simetria do satélite. O sensor digital solar (DSS) não tem medida definida em relação ao ângulo de guinada ψ o sensor não fornece medidas diretas e precisas, as medidas de arfagem e guinada são acopladas (FUMING H.; KUGA, 1999). As equações são esboçadas abaixo:

$$\alpha_{\psi} = \arctg\left(\frac{-S_y^*}{S_x^*\cos(60) + S_z^*\cos(150)}\right) + \nu_{\alpha\psi} \quad (4.15)$$

Para definir a posição em que o o vetor solar é detectado pelo o sensor as especificações abaixo devem ser estabelecidas:

$$|S_x^*\cos(60) + S_z^*\cos(150)| \geq 60^\circ \quad (4.16)$$

$$\left|\alpha_{\theta} = 24^\circ + \text{actg}\left(\frac{S_x^*}{S_z^*}\right)\right| < 60^\circ \quad (4.17)$$

em que S_x^* , S_y^* e S_z^* são as componentes do vetor unitário solar, cada componente pode ser encontrada utilizando as formulações abaixo:

$$\begin{aligned} S_x^* &= S_{0x} + \psi S_{0y} - \theta S_{0z} \\ S_y^* &= S_{0y} - \psi S_{0x} + \phi S_{0z} \\ S_z^* &= S_{0z} - \phi S_{0y} - \theta S_{0z} \end{aligned} \quad (4.18)$$

os valores de ϕ , θ e ψ são os ângulos de Euler para a atitude em que o satélite encontra-se em determinado momento.

Os erros referentes as medidas dos sensores devido ao deslocamento são semelhantes aos presentes no infravermelho terrestre (IRES), tem origem gaussiana. O equacionamento é esboçado abaixo em relação a matriz de covariância e a função esperança.

$$\begin{aligned} E[v_{\alpha_\psi}(t)] &= 0 \\ E[v_{\alpha_\theta}(t)] &= 0 \end{aligned} \quad (4.19)$$

$$\begin{aligned} E[v_{\alpha_\psi}(t)v_{\alpha_\psi}^T(t + \Delta t)] &= R_{v_{\alpha_\psi}} \delta(t) \\ E[v_{\alpha_\theta}(t)v_{\alpha_\theta}^T(t + \Delta t)] &= R_{v_{\alpha_\theta}} \delta(t) \end{aligned} \quad (4.20)$$

Os erros não são relacionados como esboça a formulação abaixo:

$$E[v_{\alpha_\psi}(t)v_{\alpha_\theta}^T(t + \Delta t)] = E[v_{\alpha_\theta}(t)v_{\alpha_\psi}^T(t + \Delta t)] = 0 \quad (4.21)$$

O vetor y devido as estimativas dos sensores é introduzido em espaço de estados em conjunto com os erros de desalinhamentos e montagem do sensor.

$$y = \begin{bmatrix} \arctan\left(\frac{-(S_{0y} - \psi S_{0x} + \phi S_{0z})}{(S_{0x} + \psi S_{0y} - \theta S_{0z}) \cos 60^\circ + (S_{0z} - \phi S_{0y} - \theta S_{0z}) \cos 150^\circ}\right) \\ 24^\circ + \arctan\left(\frac{S_{0x} + \psi S_{0y} - \theta S_{0z}}{S_{0z} - \phi S_{0y} - \theta S_{0z}}\right) \\ \phi \\ \theta \end{bmatrix} + v \quad (4.22)$$

em que $v = [v_{\alpha_\psi} \ v_{\alpha_\theta} \ v_{\phi_H} \ v_{\theta_H}]^T$ é o erro estimado.

5 Modelo matemático de estimação

A seção descrita tem como objetivo o detalhamento da modelagem matemática que caracteriza o método de estimação da atitude do satélite por meio da aplicação do filtro H_∞ Estendido que tem como finalidade a linearização do problema, utilizando expansão em série de Taylor de primeira ordem. Em consequência disso, busca estimar a atitude e os Bias de giro do satélite. O capítulo também apresenta a modelagem matemática do filtro de Kalman Estendido (FKE) para avaliação comparativa dos sistemas.

5.1 O filtro de Kalman Estendido

O filtro de Kalman Estendido apresenta um modelo de sistema não linear e modelo de medida discreto definidos pela variação do vetor de estado em relação ao tempo $\dot{x}(t)$ e a função de estado y_k . As equações abaixo esboçam o sistema.

$$\begin{aligned}\dot{x}(t) &= f(x(t)) + G(t)w(t) \\ y_k &= h(x_k) + v_k\end{aligned}\tag{5.1}$$

Os valores de $G(t)$ e $f(x(t))$, são respectivamente a matriz de ordem $(n \times n)$ variante com parâmetros contínuos no tempo e a função vetorial em função de $x(t)$ com dimensionamento n . Os termos v_k e $w(t)$ determinam os ruídos de medida e processo do sistema. O valor de $h(x_k)$ é o vetor de funções não lineares e não diferenciáveis em função de $x(t)$, as seguintes hipóteses são definidas para o processo de estimação.

$$\begin{aligned}E[w_k w_k^T] &= \tilde{Q}_k \\ E[v_k v_k^T] &= \tilde{R}_k \\ E[w_k v_k^T] &= 0\end{aligned}\tag{5.2}$$

Os valores de \tilde{Q}_k e \tilde{R}_k são as matrizes de incerteza de ruído de processo e de medida da atuação do filtro. A terceira formulação esboça a relação de independência dos ruídos que são definidos como branco não correlacionados e com média zero. (SIMON, 2006), (HU J. S.; YANG, 2011).

5.1.1 Solução para o filtro de Kalman Estendido

A análise da solução é definida em duas fases propagação e atualização as quais são definidas abaixo. A metodologia do processo é formulada no livro (SIMON, 2006).

Propagação do Filtro de Kalman Estendido

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, \mu_k) \quad (5.3)$$

$$\tilde{P}_k^- = \Phi_{k-1} \tilde{P}_{k-1}^+ \Phi_{k-1}^T + \tilde{Q}_k \quad (5.4)$$

Φ_{k-1} é definida como matriz de transição. A estimação da matriz é determinada pelas condições iniciais $\Phi_{k-1} = I$ e $\tilde{F}_k = \left. \frac{\partial f}{\partial x_k} \right|_{\hat{x}_k^-}$ sua definição é dada por:

$$\Phi_k = \tilde{F}_k \Phi_{k-1} \quad (5.5)$$

A matriz de ruído de processo é fornecida pela formulação:

$$\Gamma_k Q_k \Gamma_k^T = \int_{k-1}^k \Phi_t G(t) Q(t) G(t)^T \Phi_t^T dt \quad (5.6)$$

A matriz Γ_k tem ordem $(n \times n)$ e valores discretos no tempo.

Atualização do Filtro de Kalman Estendido

As equações abaixo descrevem o modelo de correção.

$$\hat{x}_k^+ = \hat{x}_k^- + \tilde{K}_k (y_k - h(\hat{x}_k^-)) \quad (5.7)$$

$$\tilde{K}_k = \tilde{P}_k^- \tilde{H}_k^T (\tilde{H}_k \tilde{P}_k^- \tilde{H}_k^T + \tilde{R}_k)^{-1} \quad (5.8)$$

$$\tilde{P}_k^+ = (I - \tilde{K}_k \tilde{H}_k) \tilde{P}_k^- \quad (5.9)$$

as grandezas \tilde{K}_k e \tilde{P}_k^+ , são definidas como o ganho do filtro e a matriz de covariância. O vetor de estado \hat{x}_k^+ é uma estimação mais precisa após o processo de atualização.

Após o processo de manipulação matemática descrito em (SIMON, 2006) e no apêndice C de (SILVA, 2016). Os valores de \tilde{K}_k e \tilde{P}_k^+ são descritos abaixo:

$$\tilde{K}_k = \tilde{P}_k^- [I + \tilde{H}_k^T \tilde{R}_k^{-1} \tilde{H}_k \tilde{P}_k^-]^{-1} \tilde{H}_k^T \tilde{R}_k^{-1} \quad (5.10)$$

$$\tilde{P}_{k+1}^- = \tilde{F}_k \tilde{P}_k^- [I + \tilde{H}_k^T \tilde{R}_k^{-1} \tilde{H}_k \tilde{P}_k^-]^{-1} \tilde{F}_k^T + \tilde{Q}_k \quad (5.11)$$

5.2 filtro H_∞ Estendido

A solução presente abaixo encontra-se disponível em (SIMON, 2006) e (SHEN X.; DENG, 1997). O Padrão não linear do sistema discreto no tempo é descrito abaixo:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k \\ y_k &= h(x_k) + v_k \end{aligned} \quad (5.12)$$

As parcelas w_k e v_k são definidas como os ruídos presentes na estimação que podem ter natureza aleatória com estatística desconhecida o índice k representa o passo discreto no tempo, as funções $f(x_k, u_k)$ e $h(x_k)$ tem característica não linear e dependem do parâmetro de controle x_k e uma entrada de controle u_k .

O padrão descrito pelas funções $f(x_k, u_k)$ e $h(x_k)$ podem ser estimados utilizando expansão em série de Taylor, pois apresentam natureza não linear, o processo é descrito abaixo:

$$f(x_k, u_k) = f(\hat{x}_k, u_k) + \left. \frac{\partial f}{\partial x_k} \right|_{\hat{x}_k} (x_k - \hat{x}_k) \quad (5.13)$$

$$h(x_k) = h(\hat{x}_k) + \left. \frac{\partial h}{\partial x_k} \right|_{\hat{x}_k} (x_k - \hat{x}_k) \quad (5.14)$$

Utilizando definição abaixo o objetivo é encontrar a combinação linear em função de x_k para a estimação de z_k a formulação é esboçada abaixo:

$$z_k = L_k x_k \quad (5.15)$$

A matriz L_k é definida como matriz rank completo, para a estimação do filtro de Kalman, L_k é igual a matriz identidade (I) a estimativa de z_k é definida como \hat{z}_k e a estimativa de x_0 é \hat{x}_0 . Os valores de z_k são estimados de acordo com as medições incluído tempo (N-1), e com isso, a função custo (J) pode ser estimada. A formulação abaixo define (J).

$$J_1 = \frac{\sum_{k=0}^{N-1} \|z_k - \hat{z}_k\|_{S_k}^2}{\|x_0 - \hat{x}_0\|_{P_0^{-1}}^2 + \sum_{k=0}^{N-1} (\|w_k\|_{Q_k^{-1}}^2 + \|v_k\|_{R_k^{-1}}^2)} \quad (5.16)$$

O critério de resolução busca a diminuição da função custo, para que isso ocorra, é necessário encontrar o valor de \hat{z}_k que buscam minimizar o numerador descrito pela função em função do fator $(z_k - \hat{z}_k)$. Entretanto, a consideração da natureza como adversa pode maximizar a função custo (J) em relação aos parâmetros w_k , v_k e x_0 . Portanto, a localização dessas variáveis são localizadas no denominador (SIMON, 2006).

As matrizes S_k , Q_k^{-1} , R_k^{-1} e P_0^{-1} são simétricas e positivas, para que a melhor estimativa de \hat{z}_k seja definida é preciso estimar um coeficiente de performance em que a função custo pode utilizar como margem. O coeficiente de performance é limitado de acordo com as definições e limitações do projeto. O ponto da estimativa de \hat{z}_k que minimiza a

função custo é limitado pelo coeficiente de performance de acordo com a relação descrita abaixo:

$$J_1 < \frac{1}{\gamma} \quad (5.17)$$

Em que γ é o coeficiente de performance definido no projeto, desenvolvendo a formulação acima em relação a restrição da função custo (J1) temos:

$$J = -\frac{1}{\gamma} \|x_0 - \hat{x}_0\|_{P_0^{-1}}^2 + \sum_{k=0}^{N-1} \left[\|z_k - \hat{z}_k\|_{S_k}^2 - \frac{1}{\gamma} \left(\|w_k\|_{Q_k^{-1}}^2 + \|v_k\|_{R_k^{-1}}^2 \right) \right] < 1 \quad (5.18)$$

Relacionando os parâmetros que otimizam a função para pontos de máximos e mínimos condicionados pela função custo a seguinte expressão é encontrada:

$$J^* = \min_{\hat{z}_k} \max_{w_k, v_k, x_0} J \quad (5.19)$$

o ruído v_k pode ser escrito em função do vetor de estado y_k de acordo com a equação:

$$y_k = h(x_k) + v_k \quad (5.20)$$

$$v_k = y_k - h(x_k) \quad (5.21)$$

Substituindo v_k no denominador da função custo (J1) é obtido:

$$\|v_k\|_{R_k^{-1}}^2 = \|y_k - h(x_k)\|_{R_k^{-1}}^2 \quad (5.22)$$

Utilizando a equação $z_k = L_k x_k$ transcrevendo $\hat{z}_k = L_k \hat{x}_k$ e substituindo ambas no numerador temos a seguinte relação:

$$\begin{aligned} \|z_k - \hat{z}_k\|_{S_k}^2 &= (z_k - \hat{z}_k)^T S_k (z_k - \hat{z}_k) \\ &= (x_k - \hat{x}_k)^T L_k^T S_k L_k (x_k - \hat{x}_k) \\ &= \|x_k - \hat{x}_k\|_{\bar{S}_k}^2 \end{aligned} \quad (5.23)$$

Substituindo as seguintes relações em (3.7) temos:

$$J = -\frac{1}{\gamma} \|x_0 - \hat{x}_0\|_{P_0^{-1}}^2 + \sum_{k=0}^{N-1} \left[\|x_k - \hat{x}_k\|_{\bar{S}_k}^2 - \frac{1}{\gamma} \left(\|w_k\|_{Q_k^{-1}}^2 + \|y_k - h(x_k)\|_{R_k^{-1}}^2 \right) \right] \quad (5.24)$$

A simplificação da função custo pode ser definida com a expressão descrita abaixo:

$$\Psi(x_0) = -\frac{1}{\gamma} \|x_0 - \hat{x}_0\|_{P_0^{-1}}^2 \quad (5.25)$$

$$L_k = \|x_k - \hat{x}_k\|_{\bar{S}_k}^2 - \frac{1}{\gamma} \left(\|w_k\|_{Q_k^{-1}}^2 + \|y_k - h(x_k)\|_{R_k^{-1}}^2 \right) \quad (5.26)$$

$$J = \Psi(x_0) + \sum_{k=0}^{N-1} L_k \quad (5.27)$$

A otimização da função, utilizando a metodologia de multiplicadores de Langrange, para pontos de máximo e mínimo condicionados por um restrição é definida abaixo de acordo com a mudança de variáveis:

$$J^* = \min_{\hat{x}_k} \max_{w_k, y_k, x_0} J \quad (5.28)$$

Os pontos estacionários são encontrados em relação as variáveis x_0 , w_k e \hat{x}_k , y_k .

5.2.1 Solução do filtro H_∞ Estendido

Após a descrição da formulação dos pontos críticos ser analisada de acordo com a modelagem desenvolvida acima e utilizando a ferramenta em expansão em série de Taylor para os vetores de estado x_k e y_k , é necessário encontrar o ponto estacionário da função custo (J), primeiramente, em relação as variáveis x_0 e w_k . As equações abaixo são utilizadas como ferramenta para desempenhar o processo:

$$x_0 = \hat{x}_0 + P_0 \lambda_0 \quad (5.29)$$

$$w_k = Q_k \lambda_{k+1} \quad (5.30)$$

$$\lambda_N = 0 \quad (5.31)$$

$$\lambda_k = G_k^{-1} \left[F_k^T \lambda_{k+1} + \gamma \bar{S}_k (\mu_k - \hat{x}_k) + H_k^T R_k^{-1} (\tilde{y}_k - H_k (\mu_k - \hat{x}_k)) \right] \quad (5.32)$$

$$P_{k+1} = F_k P_k G_k^{-1} F_k^T + Q_k \quad (5.33)$$

$$\mu_0 = \hat{x}_0 \quad (5.34)$$

$$\mu_{k+1} = (\hat{x}_k, \mu_k) + F_k (\mu_k - \hat{x}_k) + F_k P_k G_k^{-1} \left[\gamma \bar{S}_k (\mu_k - \hat{x}_k) + H_k^T R_k^{-1} (\tilde{y}_k - H_k (\mu_k - \hat{x}_k)) \right] \quad (5.35)$$

Em que λ_N é o multiplicador de Lagrange e as outras variáveis das equações são expostas abaixo:

$$F_k = \left. \frac{\partial f}{\partial x_k} \right|_{\hat{x}_k} \quad (5.36)$$

$$H_k = \left. \frac{\partial h}{\partial x_k} \right|_{\hat{x}_k} \quad (5.37)$$

$$\tilde{y}_k = y_k - h(\hat{x}_k) \quad (5.38)$$

$$G_k = I - \gamma \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \quad (5.39)$$

O valor de \tilde{y}_k é o resíduo definido pelo modelo de sensores do satélite. Posteriormente, os pontos estacionários da função (J) em relação a y_0 e \hat{x}_0 são dados como:

$$\hat{x}_k = \mu_k \quad (5.40)$$

$$y_k = h(\hat{x}_k) \quad (5.41)$$

Empregando as formulações acima encontra-se a solução do filtro que é esboçada em espaço de estados abaixo:

$$\bar{S}_k = L_k^T S_k L_k \quad (5.42)$$

$$K_k = P_k \left[I - \gamma \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} H_k^T R_k^{-1} \quad (5.43)$$

$$\hat{x}_{k+1} = f(\hat{x}_k, \mu_k) + F_k K_k (y_k - h(\hat{x}_k)) \quad (5.44)$$

$$P_{k+1} = F_k P_k \left[I - \gamma \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} F_k^T + Q_k \quad (5.45)$$

As matriz S_k e dada como uma matriz que pondera a diferença em módulo do estado real e estimado, K_k é o ganho, x_{k+1} é o vetor de estado calculado posteriormente e P_{k+1} é uma matriz de ponderação. O valor de γ precisa satisfazer a função custo (J1), para que isso aconteça, o termo descrito abaixo deve ser positivo a fim de preservar a existência do metodo.

$$P_k^{-1} - \gamma \bar{S}_k + H_k^T R_k^{-1} H_k > 0 \quad (5.46)$$

5.3 Relação entre o filtro H_∞ estendido e o filtro de Kalman

Para o filtro H_∞ Estendido as matrizes P_o , Q_k e R_k são definidas como particularidades do design do projeto, estimadas pela definição dos ruídos w_k e v_k e também pela diferença entre o estado real e estimado da condição inicial definida como $(x_0 - \hat{x}_0)$. No entanto, para o filtro de Kalman os ruídos w_k , v_k e a diferença $(x_0 - \hat{x}_0)$ tem média zero e as matrizes definidas acima são suas respectivas covariâncias. A escolha das matrizes L_k e S_k como matrizes indentidade (I) em todas as formulações do filtro H_∞ Estendido, a fim de ponderar todos os possíveis erros de estado, e a utilização do coeficiente de performace θ da função custo como zero. Alteram as formulações do filtro H_∞ Estendido para as formulações do filtro de Kalman que de acordo com a restrição faz o desempenho tender ao infinito. O filtro de Kalman e o H_∞ tem diferença notável em relação ao estimador, para o filtro de Kalman o estimador é o mesmo independente da combinação linear adotada, então a matriz L_k não interfere na combinação linear $x_k L_k$ (SIMON, 2006).

As particularidades dos filtros são esboçadas pelas as equações abaixo:

$$P_{k+1} = F_k P_k \left[I - \gamma \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} F_k^T + Q_k \quad (5.47)$$

$$K_k = P_k \left[I - \gamma \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} H_k^T R_k^{-1} \quad (5.48)$$

As formulações do filtro de Kalman não possuem o termo $\gamma \bar{S}_k P_k$ resultando em uma matriz P_{k+1} de covariância maior e um ganho K_k maior, em função disso, a matriz Q_k também aumenta. É possível notar que os parâmetros x_0 , w_k e v_k buscam maximizar a função custo já que a natureza é considerada adversa. O filtro H_∞ Estendido é mais robusto de acordo com os equacionamentos do design.

6 Simulação computacional

6.1 Modelo computacional para análise do filtro H_∞ Estendido utilizando dados simulados.

6.1.1 Parâmetros iniciais de estimação de atitude utilizando filtro H_∞ Estendido.

- Demo propat Real State.
- Demo propat EKF.
- Demo propat EHinf.

Os seguintes códigos foram produzidos em sequência em função das condições iniciais de tempo, parâmetros orbitais, velocidade angular e posição. Buscando a comparação da aplicação do filtro de Kalman Estendido e o filtro H_∞ infinito Estendido e firmou a comprovação da metodologia mais precisa em estudos de atitude. A definição descrita pela rotina Demo Propat Real State, forneceu os dados simulados propagados para verificação de estimação de atitude pelos filtros.

6.2 Estimação de atitude utilizando filtro de Kalman Estendido.

6.2.1 Condições iniciais para filtragem utilizando o Filtro de Kalman Estendido.

Em relação aos parâmetros de estimação de propriedades descritas pelo Demo Propat EKF, utilizando análise estatística, a estimação das condições iniciais do filtro definiram as matrizes de ponderação relacionadas no sistema do filtro cujas foram definidas de acordo com o grau de incerteza das medidas e avaliações com grandezas estatísticas de covariância e desvio padrão que para variáveis aleatórias avaliam a dispersão gerada, em função do método de propagação. Os resultados buscaram aproximar o modelo estimado pelo filtro do modelo matemático gerado.

A definição dos parâmetros iniciais foi definida no software Matlab, o valor das matrizes de incertezas de ruídos do modelo matemático e do sensor foram definidas para viabilizar o método, as matrizes de incerteza de estados do modelo matemático e do sensor também foram definidas. Buscando valores mais precisos para a estimação dos vetores de

estado definidos pelo filtro, a análise da eficiência do sistema foi avaliada na comparação de resultados expostos pelos modelos de filtragem.

6.2.2 Modelos de sensores.

Os modelos de sensores utilizados para as simulações no software Matlab foram o Sensor Solar Digital (DSS) e o Infravermelho Terrestre (IRES). Para o primeiro, foram definidos os versores orbitais que indentificam o posicionamento em que o satélite se encontra na órbita em que está inserido, em função de sua posição, o valor do vetor posição unitário no sistema orbital foi calculado. Além disso, foi calculado também o vetor posição do sol no sistema geocêntrico em função dos parâmetros de lançamentos localizados na referencial terrestre.

6.2.3 Cálculo das matrizes Jacobianas do filtro de Kalman Estendido.

As Matrizes Jacobianas foram calculadas para o filtro de Kalman Estendido, já que a funções de estado $F(X)$ e medição $H(X)$ são descritas como não lineares. Utilizou-se as derivadas em relação aos vetores de estado para estimação dessas matrizes com precisão.

6.3 Estimação de atitude utilizando filtro H_∞ Estendido.

6.3.1 Condições iniciais

Para a estimação de atitude utilizando o filtro H_∞ Estendido os valores das condições iniciais foram definidas para o início das simulações de acordo com o vetor de estados iniciais x_0 que possui como suas componentes, os valores dos ângulos de Euler descritos por θ (rolamento), ϕ (arfagem) e ψ (guinada) e os, Bias, erros definidos para a variação de cada ângulo no sistema de coordenadas. Após a definição dos valores iniciais a representação dessas grandezas na órbita foram estimadas.

6.3.2 Atualização do filtro H_∞ Estendido

A atualização pode ser definida em função da modelagem matemática descrita e o modelo de sensor escolhido. Para o caso em questão, os sensores IRES e DSS foram escolhidos como sensores presentes no filtro, indiretamente, os vetores de estado nos quais as estimações do filtro são avaliadas necessitam das funções de estado do sensor e do modelo matemático para que a finalidade do filtro torne-se possível, ou seja, vetores de estado mais precisos sejam fornecidos. A melhor estimativa desses vetores está intimamente ligada com a diferença entre a funções de estado determinadas pelo sensor e o modelo matemático fornecido, este equacionamento é chamado de resíduo. Outro parâmetro necessário para que o processo de atualização ocorra é o ganho do filtro o qual é estimado de acordo com

a ponderação entre as matrizes de incerteza devido o modelo de sensoriamento e o modelo matemático. O ganho regula a estimativa inicial do sistema, quanto maior o seu valor, o palpite inicial do filtro tenderá a utilizar a estimativa do sensor, entretanto, se o valor do ganho é pequeno a estimativa inicial utilizada será descrita pelo modelo matemático. O processo busca aprimorar as estimativas dos vetores de estado a cada passo de tempo definido pelo filtro encontrando um modelo mais preciso e eficiente em função de suas interações.

6.3.3 Predição do Filtro H_∞ Estendido

O processo de predição do filtro é definido para melhorar o palpite inicial descrito no processo de atualização no qual o novo vetor de estado fornece uma medida mais precisa em relação ao valor utilizado para inicializar o processo. A cada passo de tempo, o vetor de estado descrito no processo de atualização é utilizado como chute inicial do modelo analítico visando uma melhor aproximação e diminuição do erro do vetor de estado real e o vetor de estado estimado. A figura abaixo retrata a estimação dos vetores de estado utilizando o simulador Propat para diversos filtros.

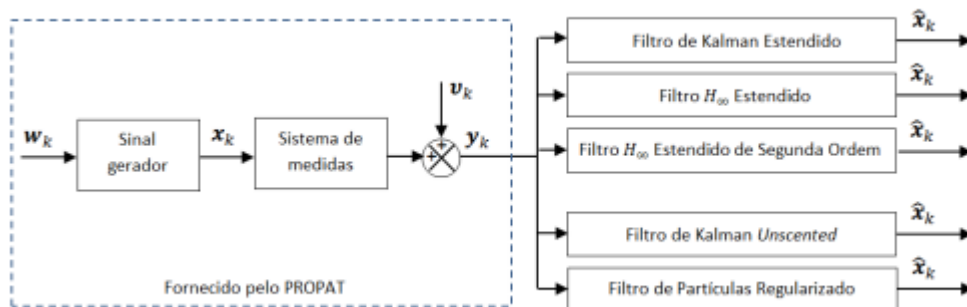


Figura 5 – Estimação dos vetores de estado utilizando o simulador Propat e sua interação com diversos tipos de filtros. (SILVA, 2016).

6.4 Estimação de atitude utilizando dados simulados

Em função da análise teórica estipulada em tópicos anteriores o software Matlab foi utilizado com a finalidade de simulação do método de estimação de atitude, utilizando a ferramenta matemática descrita pelos ângulos de Euler, com o objetivo de especificar o movimento do satélite no referencial do seu centro de massa e localização orbital. O modelo descrito pelo código tem como aplicação a descrição de dados iniciais relacionados com o movimento rotacional e translacional que o satélite descreve na órbita, de acordo com esses valores, é possível estimar os ângulos θ (arfagem), ϕ (rolamento) e ψ (guinada) e os Bias como também os parâmetros orbitais de acordo com uma margem de tempo definida.

Os modelos de sensoriamento escolhidos para a aplicação da simulação foram o Sensor Solar Digital (DSS) e o Infravermelho Terrestre (IRES).

6.4.1 Condições iniciais para análise de atitude

6.4.2 Ângulos de Euler

Os ângulos de Euler foram definidos para a simulação computacional utilizando a sequência de rotação tradicional do tipo 3-2-1 em que a rotação inicial ocorre primeiramente em relação ao eixo z e posteriormente em relação aos eixos y e x. A matriz de rotação faz referência direta da rotação do centro de massa do satélite com o sistema de referência inercial que é a terra.

Para definir o processo de estimação da atitude descrita pelo satélite as seguintes condições iniciais são definidas na tabela abaixo:

Tabela 2 – Condições iniciais para propagação de atitude.

Condições iniciais			
Velocidade angular (rad/s)	ω_x	ω_y	ω_z
	0.000002	0.000003	0.000005
Ângulos de Euler (graus)	θ	ϕ	ψ
	3°	5°	2°
Bias (graus/hora)	x	y	z
	0.1	0.1	0.1
Vetor de estado real (m)	x	y	z
	0	0	0

6.4.3 Tensor de inércia

Após a escolha dos parâmetros iniciais aplicou-se os cálculos necessários para o momento de inércia do satélite em relação ao seu centro de massa e o sistema de referência tridimensional. O momento de inércia é uma grandeza física que relaciona a facilidade ou dificuldade que o corpo possui para rotacionar em torno do seu eixo de rotação. A formulação abaixo esboça o tensor de inércia:

$$\sigma = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (6.1)$$

Em que σ representa o tensor de inércia para um eixo de coordenadas tridimensional presente no centro de massa do corpo. Utilizando as formulações descritas pela teoria de geometria de massas o tensor de inércia para o satélite em específico é descrito abaixo:

$$\sigma = \begin{bmatrix} 2.3220 & -0.0035 & 0.0150 \\ -0.0035 & 1.0880 & -0.0040 \\ 0.0150 & -0.0040 & 2.0470 \end{bmatrix} \quad (6.2)$$

6.4.4 Parâmetros orbitais

Para definição do movimento do satélite em relação a órbita em que o mesmo está inserido é necessário a definição de parâmetros que buscam definir tamanho, formato e orientação da órbita (WERTZ, 1978). Os elementos orbitais são descritos como seis, a tabela abaixo indentifica suas especificações para uma órbita Kleperiana definida na rotina Demo Propat descrita no Matlab.

Tabela 3 – parâmetros orbitais iniciais.

Parâmetros orbitais	
semieixo maior (a)	7149000 (m)
Excentricidade (e)	0.01
inclinação orbital (I)	1.7104 (rad)
ascensão reta do nóculo ascendente (Ω)	0
Argumento do perigeu (ω)	0
Anomalia média (M_0)	0

É possível notar de acordo com a tabela acima que a Ascensão Reta do Nóculo Ascendente, Anomalia Média e o Argumento do Perigeu têm valores zero, ou seja, são desconsideradas para a simulação. Depois da escolha dos parâmetros orbitais a propagação foi efetuada correspondente ao passo de tempo escolhido.

6.4.5 Integrador numérico ODE 45

A função ODE 45 definida pelo software Matlab tem como finalidade a resolução de equações diferenciais ordinárias utilizando o método de Runge Kutta de quarta ordem para determinado tempo e condições iniciais. Para o problema em questão, a simulação foi estipulada para uma margem de tempo de 0 até 600 segundos. As condições iniciais descritas acima foram utilizadas para propagação dos dados referentes aos ângulos de Euler, Bias, parâmetros orbitais, vetores de estado real, vetor posição e vetor velocidade descritos pelo o satélite.

6.4.6 Resultados do modelo matemático

Os resultados da rotina Demo Propat foram propagados no software Matlab os gráficos são esboçados abaixo:

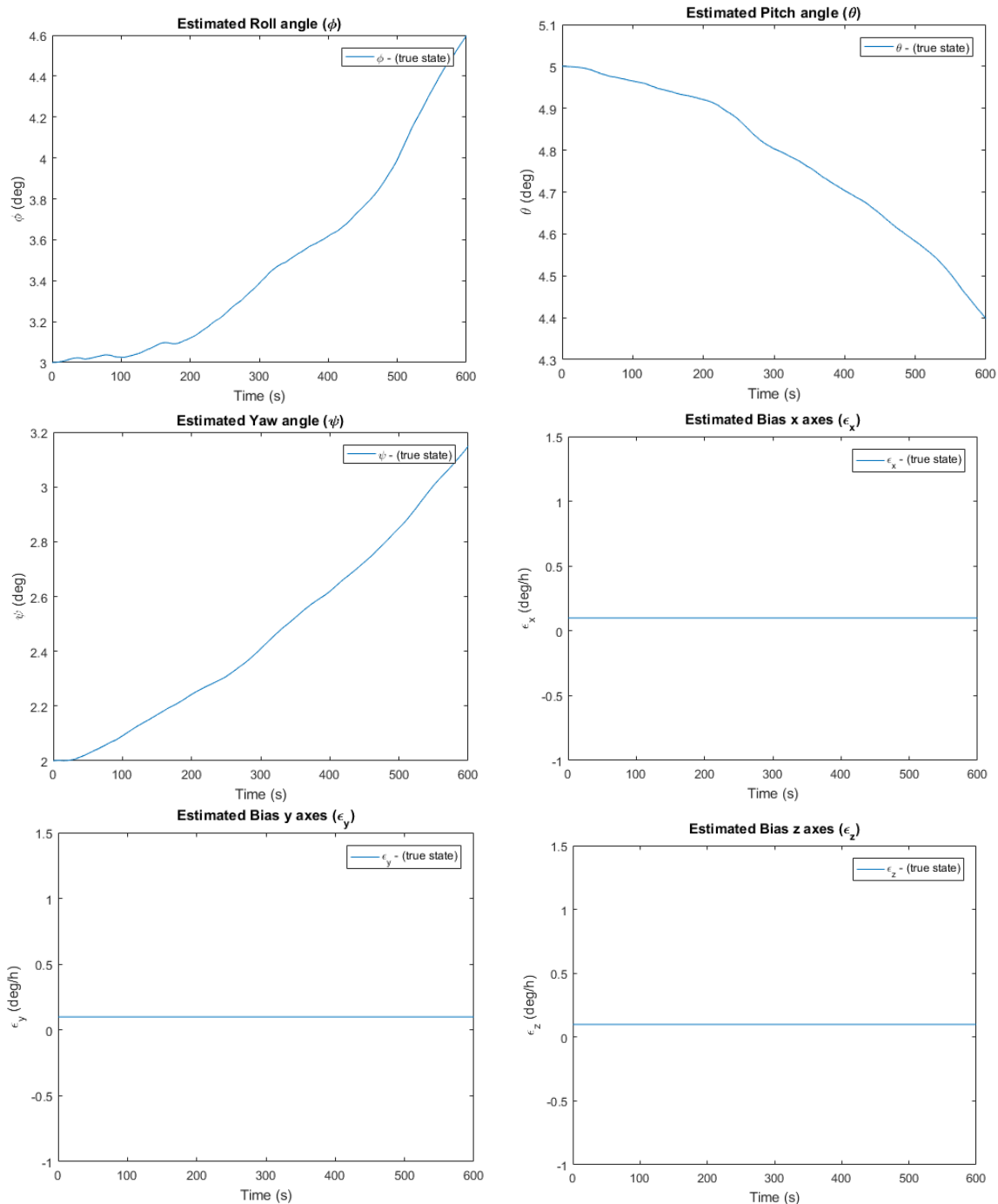


Figura 6 – Propagação dos ângulos de Euler e os Bias definidos pelo modelo matemático.

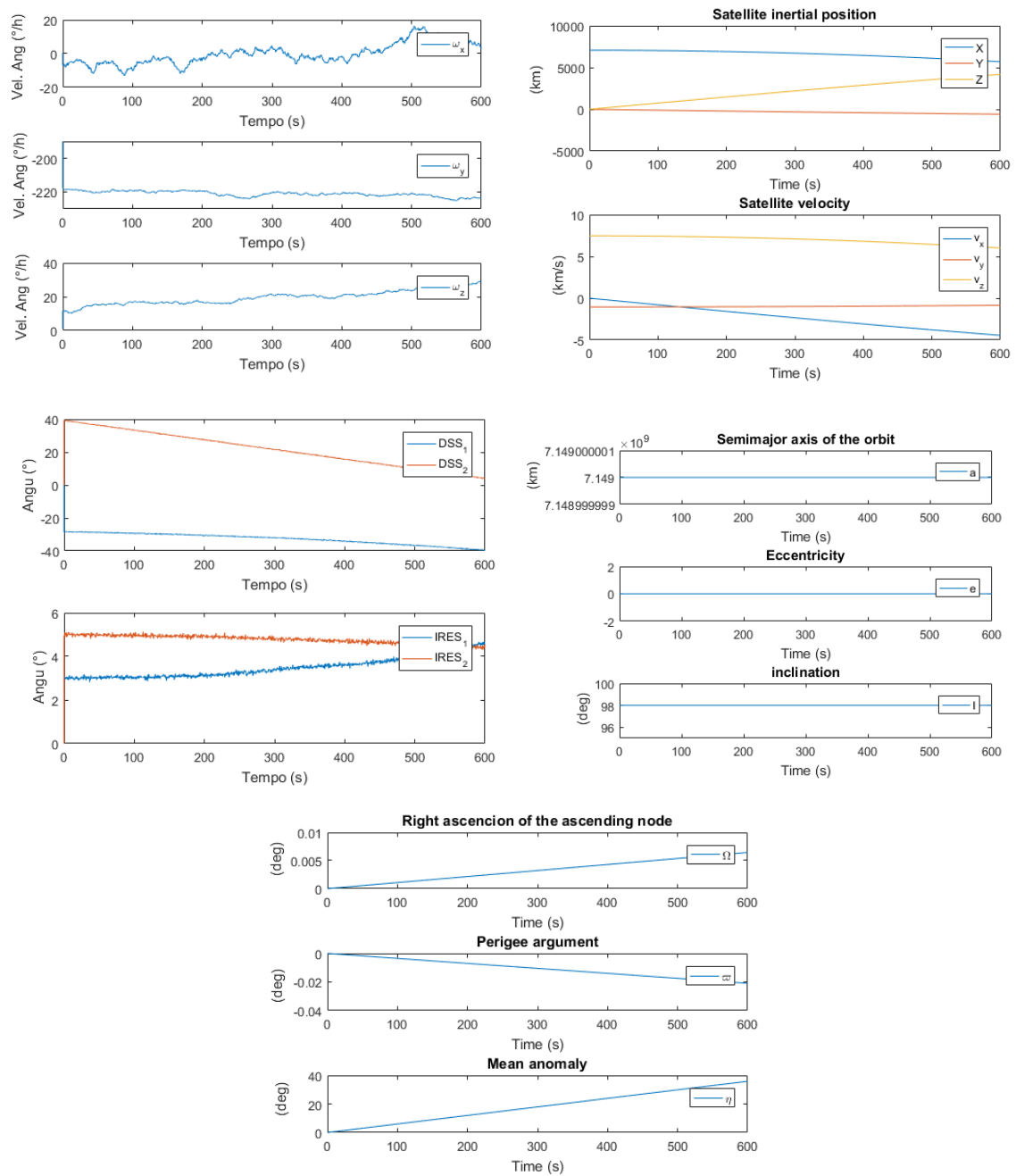


Figura 7 – Propagação dos parâmetros orbitais, velocidade angular, vetor posição, vetor velocidade e variação do IRES e DSS definido pelo modelo matemático.

6.5 Resultados Utilizando o filtro de Kalman e o filtro H_∞ Estendido

Após a propagação dos vetores de estados reais relacionados com o modelo matemático do filtro a estimativa gerada pelos filtros foram definidas utilizando as rotinas Demo Propat EKF e Demo Propat EHinf. As condições iniciais descritas pelo filtro para a estimação de estados são esboçadas na tabela abaixo:

Tabela 4 – Parâmetros escolhidos para filtragem.

Condições iniciais para os filtros		
Matrizes diagonais		
Q	6x6	$[0.1268 ; 0.1127 ; 0.0986 ; 0 ; 0 ; 0] * 10^{-5}$
P	6x6	$[0.0011; 0.0009 ; 0.0010 ; 0 ; 0 ; 0]$
R	4x4	$[0.1097 ; 0.1097 ; 0 .0011 ; 0.0011] * 10^{-3}$
γ	Escalar	1/4
S	6x6	Identidade
L	6x6	Identidade

Em que as matrizes Q e R são as matrizes de incertezas dos ruídos do modelo matemático e do sensor aplicado e P, S são as matrizes de incerteza do modelo matemático e do sensor. A definição da matriz L e o valor de gama são parâmetros utilizados para o filtro H_∞ estendido em que γ é o fator de performance e L a matriz Rank completo. Algumas propriedades do filtro podem ser alteradas em relação ao filtro de Kalman, aumentando sua precisão e robustez, as matrizes Q_k e P_{k+1}^{-1} definidas como matriz de incerteza do ruído do processo e matriz inversa de incerteza do modelo aumentam o ganho K_k . No entanto, para o filtro H_∞ estendido a alteração de P_{k+1}^{-1} é definida pelo fator $\gamma \bar{S}_k P_k$ em P_{k+1} e K_k . (SILVA, 2016) e (HU J. S.; YANG, 2011)

Após a escolha dos parâmetros dos filtros os vetores de estado e os gráficos foram definidos para o filtro de Kalman e o filtro H_∞ Estendido a figura 8 esboça o processo:

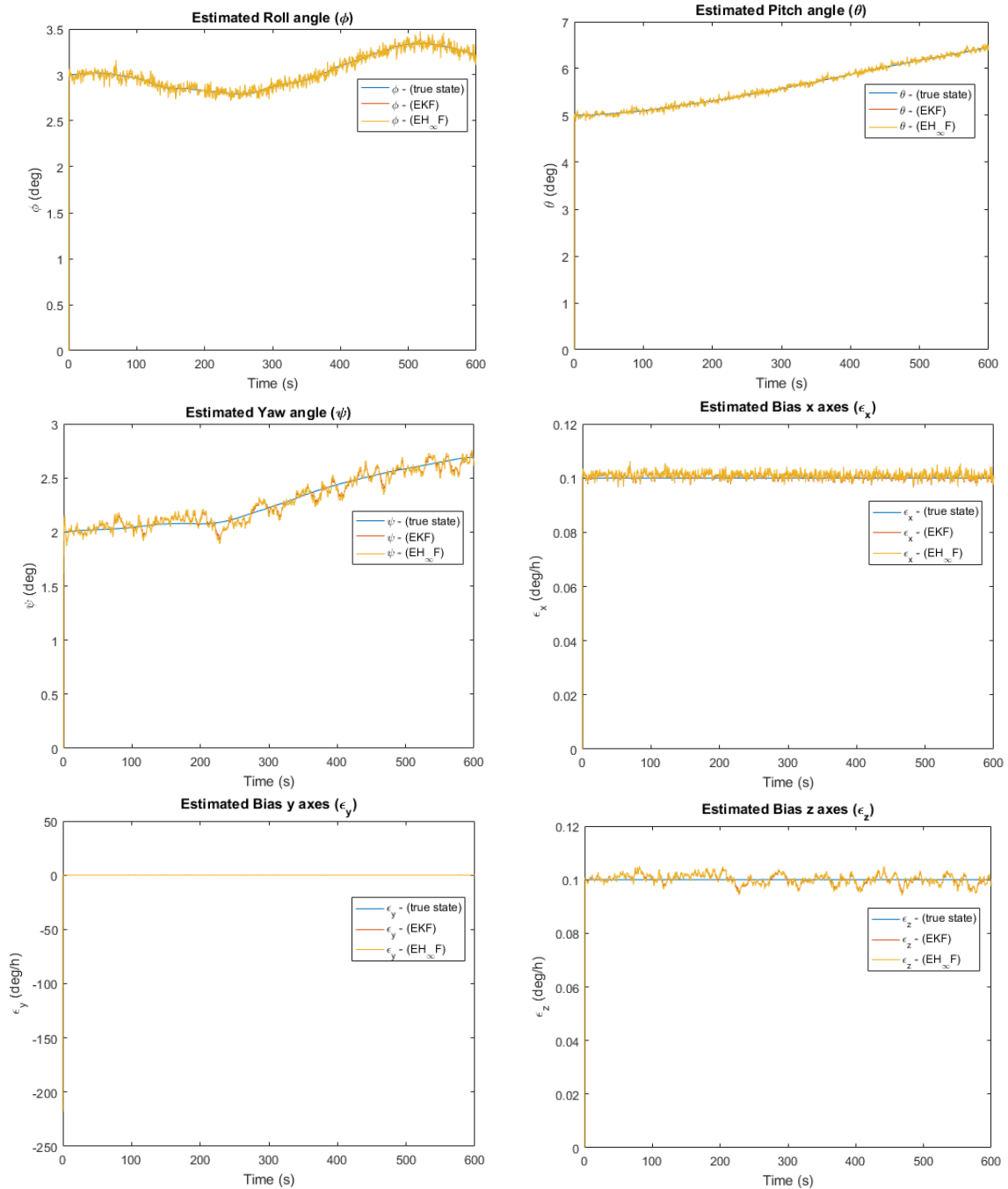


Figura 8 – Estimação dos ângulos de Euler e os Bias utilizando filtro H_∞ Estendido e o filtro de Kalman.

A convergência dos filtros é avaliada em relação aos resíduos relacionados as funções de estado definidas pelo sensor e o modelo matemático Demo Propat Real State. Os resíduos definem a precisão do modelo de sensor utilizado para estimação dos estados. As formulações abaixo retratam a análise do desvio padrão das funções de estado e sua média.

$$\sigma = \sqrt{\sum_{k=1}^n \frac{(\tilde{y}_k - \bar{y}_k)^2}{n}} \quad (6.3)$$

$$\bar{y}_k = \sum_{k=1}^n \frac{\tilde{y}_k}{n} \quad (6.4)$$

As variáveis das equações indicam n como o número de interações \bar{y}_k como a média dos resíduos e \tilde{y}_k o valor dos resíduos. Os gráficos abaixo definem os resíduos para o IRES e o DSS.

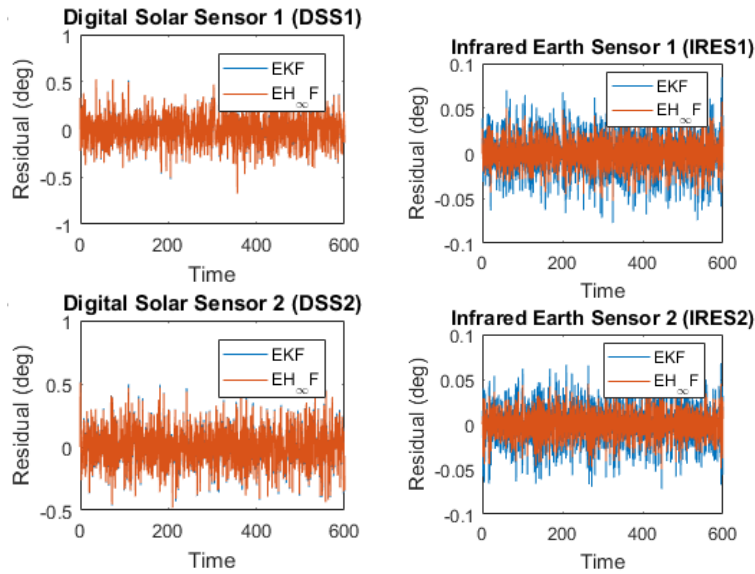


Figura 9 – Estimação dos resíduos para cada sensor utilizando filtro H_∞ Estendido e o filtro de Kalman.

Para melhor visualização a tabela de médias e desvios padrões dos sensores é definida abaixo:

Tabela 5 – Média e Desvio padrão dos resíduos para cada sensor.

Sensores	FKE	Filtro H_∞ estendido
$DSS_1(^{\circ})$	$-1,5*10^{-3} \pm 0.1651$	$1,5*10^{-3} \pm 0.1664$
$DSS_2(^{\circ})$	$-0.6*10^{-3} \pm 0.1486$	$0.7*10^{-3} \pm 0.1467$
$IRES_1(^{\circ})$	0 ± 0.0249	0 ± 0.0171
$IRES_2(^{\circ})$	$0.1*10^{-3} \pm 0.0244$	0 ± 0.0167

Os valores apresentados descrevem os sensores em relação a sua capacidade de convergência é possível notar que os valores descritos apresentam valores semelhantes para o DSS_1 e o $IRES_1$ para ambos os filtros. No entanto, alguns casos apresentaram variação pequena em seus resultados é o caso do $IRES_2$ e o DSS_2 . Os melhores valores são definidos pelos sensores $IRES_1$ e $IRES_2$ que manifestaram convergência absoluta. O primeiro para os dois filtros e o segundo para o Filtro H_∞ Estendido.

A análise dos parâmetros de precisão dos filtros é avaliada pelo erro referente aos vetores de estado real e estimado. A finalidade do filtro é fornecer o melhor estimador

de estados possível, em consequência disso, o filtro que apresenta o menor erro pode ser avaliado como mais preciso. Abaixo a formulação do erro é definida:

$$\tilde{x} = x_k - \hat{x}_k \quad (6.5)$$

Además, é necessário definir o valor do desvio padrão e a média dos vetores de estado as formulações são descritas abaixo:

$$\sigma = \sqrt{\frac{\sum_{k=1}^n (\tilde{x}_k - \bar{x}_k)^2}{n}} \quad (6.6)$$

$$\bar{x}_k = \sum_{k=1}^n \frac{\tilde{x}_k}{n} \quad (6.7)$$

Os valores de \tilde{x}_k e \bar{x}_k são respectivamente os erros dos vetores de estado e a média n representa o numero de interações.

Os gráficos dos erros dos vetores de estado são esboçados a seguir:

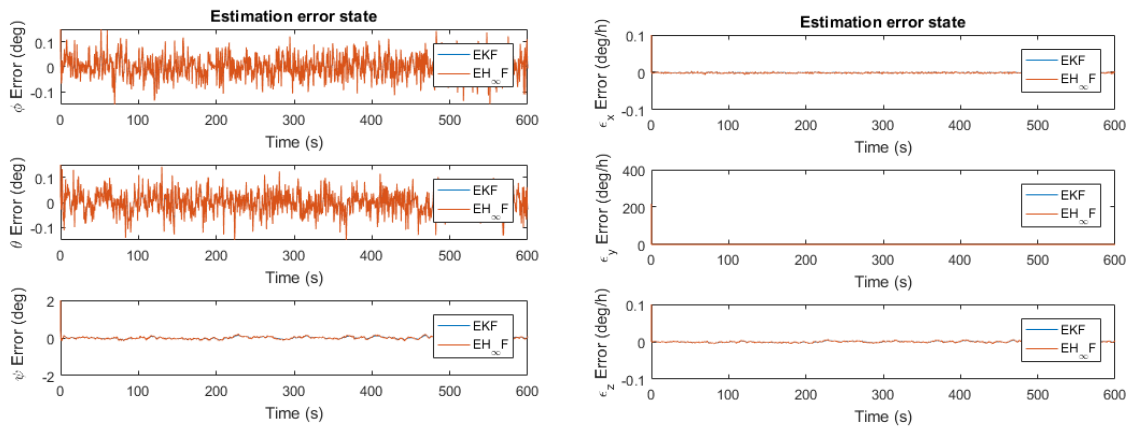


Figura 10 – Erros para os ângulos de Euler e o Bias utilizando o filtro de Kalman e o filtro H_∞ Estendido.

A tabela das médias e desvios padrões dos erros é esboçada abaixo:

Tabela 6 – Média e Desvio padrão dos erros para cada filtro.

Vetores de estado	FKE	Filtro H_∞ estendido
ϕ (<i>Rolamento</i>)	$-0.7*10^{-3} \pm 0.0391$	$-0.7*10^{-3} \pm 0.0449$
θ (<i>Arfagem</i>)	$-1.6*10^{-3} \pm 0.0385$	$-1.6*10^{-3} \pm 0.0441$
ψ (<i>Guinada</i>)	$4.3*10^{-3} \pm 0.0501$	$4.1*10^{-3} \pm 0.0600$
B_x (<i>Biax</i>)	$0.6*10^{-3} \pm 0.0013$	$0,5*10^{-3} \pm 0.0013$
B_y (<i>Biax</i>)	$-1.8*10^{-3} \pm 0.0013$	$-0,7*10^{-3} \pm 0.0013$
B_z (<i>Biaz</i>)	$-0.3*10^{-3} \pm 0.0016$	$-0,3*10^{-3} \pm 0.0018$

Os valores esboçados pela tabela representam a comparação entre os valores de erros dos vetores de estado estimados pelo filtro de Kalman e o filtro H_∞ estendido. É possível notar que os valores são idênticos ou apresentam pouca variação em seus resultados, o ϕ (rolamento) e θ (arfagem), apresentaram o mesmo valor para os dois filtros. Já o ψ (guinada) apresentou uma pequena variação descrevendo um menor erro no filtro H_∞ estendido. Os Bias apresentaram pequenas alterações nos resultados de B_x e B_y para ambos, os menores valores de erro foram estipulados pelo filtro H_∞ Estendido. O valor de B_z apresentou valores iguais para as duas ferramentas.

O tempo de processamento para as rotinas descritas no software Matlab referente aos filtros é exposto na tabela abaixo:

Tabela 7 – Tempo de execução das rotinas.

Tempo de processamento	
FKE	Filtro H_∞ estendido
0.5879s	2.5987s

A tabela apresenta maior tempo de processamento para o filtro H_∞ estendido pois utiliza parâmetros definidos na rotina do FKE (Demo propat EKF) que foram utilizados na simulação. Os códigos foram feitos em sequência adicionando mais informação na rotina do filtro H_∞ estendido (Demo propat EHinf) e possibilitando a comparação entre os resultados presentes nos dois filtros.

7 Conclusão

7.1 Considerações finais

Em relação a análise descrita no referencial bibliográfico, o método do filtro H_∞ Estendido possui modelagem matemática mais robusta em comparação com o filtro de Kalman, pois a natureza dos ruídos referentes a instalação e erros do sensor não necessitam de indentificação em relação a sua natureza para que o método funcione diminuindo as restrições de parâmetros definidos no início do projeto. É possível notar em função das formulações expostas que os parâmetros de ruídos e a condição inicial definida no sistema já são consideradas como grandezas que aumentam a complexidade do projeto em que a missão espacial está inserida. Portanto, a localização das grandezas no denominador presente na função custo define um sistema mais eficiente, pois busca otimizar o custo do projeto definido pela missão espacial, disponibilizando valores mais eficientes mesmo com piores condições de ruídos e grandezas iniciais. Entretanto, as simulações executadas no software Matlab não validaram muitas diferenças entre a estimação do filtro H_∞ Estendido e o filtro de Kalman Estendido (FKE) aproximando de maneira notável a estimação dos erros dos vetores de estado e conseqüentemente a precisão definida pelos modelos de filtragem. Para a utilização em missões espaciais reais é necessário definir a viabilidade do projeto e o custo específico para cada sistema, como a função de estimação é a mesma para o referente trabalho, o sistema menos complexo e com menos gasto seria a melhor escolha.

7.1.1 Observação dos resultados para o FKE e o Filtro H_∞ estendido.

Alguns parâmetros são especificados de acordo com as limitações encontradas no modelo de filtragem. O filtro de Kalman estendido (FKE) tem como finalidade aproximar o sistema real do sistema do modelo analítico não linear, assim como o filtro H_∞ estendido, entretanto, algumas propriedades como as matrizes de ruídos de medida e processo \tilde{R}_k e \tilde{Q}_k podem influenciar na determinação da precisão do sistema. Os valores das matrizes não foram alterados para o teste de ambos os filtros. Portanto, os parâmetros estocásticos dos modelos forneceram resultados próximos para os vetores de estado. O fator γ quando atribuído valores próximos de zero no filtro H_∞ Estendido aproxima o valor do ganho K_k e a matriz de incerteza do modelo P_{k+1} para resultados semelhantes ao Filtro de Kalman Estendido (FKE), a matriz P_{k+1} é a mesma para os dois dispositivos, direcionando a estimativa dos filtros para os mesmos valores. Para o caso das matrizes S_k e L_k são restritas para o FKE limitando sua forma apenas ao valor da matriz identidade, no entanto, para o filtro H_∞ estendido a única exigência é que a matriz seja simétrica e

positiva, ou seja, o valor dela é igual a sua transposta, como as matrizes na simulação são definidas como indentidade o processo aproxima mais uma vez as duas metodologias.

7.1.2 Função densidade de probabilidade e desvio padrão dos resultados.

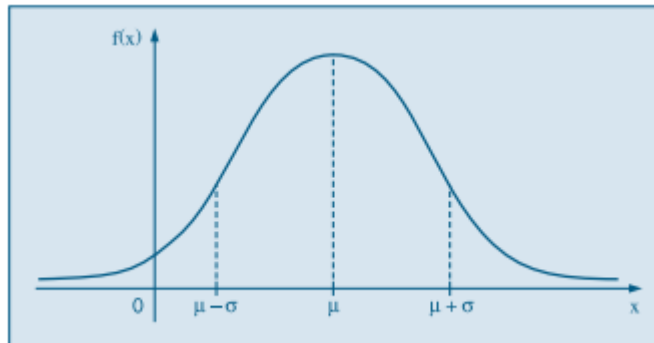


Figura 11 – Função densidade de probabilidade ilustrativa com média μ e desvio padrão σ .) (BUSSAB, 2002)

Para uma determinada variável aleatória é possível determinar a probabilidade de um evento ocorrer, em função da curva densidade de probabilidade, um sistema padrão apresenta valores $\mu = 0$ e $\sigma = 1$ (BUSSAB, 2002). O sistema definido pelos métodos de estimação apresentam valores relativamente próximos ou iguais para os referentes valores de média, fornecendo mesmo valor de centralização das áreas da função densidade de probabilidade ou áreas quase centralizadas para o FKE e o filtro H_{∞} Estendido. Os desvios padrões produzidos estão definidos em relação a dimensão no eixo das abscissas, todos os valores apresentam estimativas perto de zero estreitando a área em relação a horizontal aproximando os resultados da média. Portanto, é possível avaliar que o espaçamento entre os pontos é reduzido para ambos os filtros estipulando um modelo preciso comparado aos dados propagados pela rotina Demo Propat Real State. Os Gráficos presentes nos resultados avaliam a precisão fundamentada pelo equacionamento dos sistemas.

7.1.3 Extensão do filtro H_{∞} estendido.

Algumas fontes de pesquisas determinam métodos mais precisos para ambas ferramentas os documentos científicos (SILVA, 2016) e (HU J. S.; YANG, 2011) disponibilizam testes com o filtro H_{∞} Estendido de segunda ordem e o FKE de segunda ordem. As ferramentas não descartam ordens superiores na linearização de termos das funções presentes na modelagem matemática em função da expansão em série de Taylor, possibilitando melhores aproximações dos resultados dos vetores de estado estimado, aumentando a robustez da ferramenta. Para o trabalho fornecido a alteração das propriedades dos filtros foi o que disponibilizou os resultados expostos já que não utilizou os termos de ordem superior.

8 Trabalhos futuros

Para trabalhos posteriores os dados reais do satélite CBERS 4 serão utilizados para estimação de atitude utilizando a aplicação da metodologia do filtro H_∞ Estendido exposta no seguinte trabalho. A aplicação de ferramentas mais precisas e robustas como o Filtro H_∞ estendido de segunda ordem será aplicada e estudada, para avaliar a diferença comparativa de estimativa dos vetores de estado estimados pelo filtro H_∞ estendido. Além disso, a escolha de outro sistema de representação de atitude será escolhido. A modelagem matemática em função de quatérnions formaliza uma ferramenta mais eficiente, pois retira os pontos de singularidade definidos para a metodologia dos ângulos de Euler. A aplicação de sistemas distintos de modelos de sensores definidos pelo trabalho, também será utilizada, a fim de validar a eficiência e precisão dentro da aplicação dos filtros. Alguns tipos de modelos avaliados para a mesma temática são o Sensor Estrela e o Magnetômetro cujas aplicações também são determinadas em estudos de atitude.

Referências

- BUSSAB, e. M. Estatística básica. In: . [S.l.: s.n.], 2002. p. 62. Citado 3 vezes nas páginas 13, 29 e 64.
- CHOBOTOV, V. A. Spacecraft attitude dynamics and control. In: . [S.l.: s.n.], 1991. p. 29. Citado 2 vezes nas páginas 29 e 33.
- DISCEPOLL, F. A. Italsat f1 satellite, the system and its performance. p. 32, 1992. Citado 3 vezes nas páginas 29, 36 e 37.
- FEHSE, W. Automated rendezvous and docking of spacecraft. In: . [S.l.: s.n.], 2003. p. 29. Citado 4 vezes nas páginas 13, 30, 33 e 35.
- FUMING H.; KUGA, H. K. Cbers simulator mathematical models. cbtt project, cbtt/2000/mm/001, são josé dos campos. 1999. Citado 4 vezes nas páginas 34, 37, 40 e 41.
- GARCIA, H. K. K. R. V.; ZANARDI, M. C. F. P. S. Unscented kalman filter applied to the spacecraft attitude estimation with euler angles. p. 35, 2012. Citado 2 vezes nas páginas 30 e 39.
- GARCIA, R. V. Filtro não linear de kalman sigma - ponto com algoritmo unscented aplicado a estimativa dinâmica da atitude de satélites artificiais. p. 25, 2011. Citado 2 vezes nas páginas 25 e 30.
- HU J. S.; YANG, C. H. Second-order extended h filter for nonlinear discrete-time systems using quadratic error matrix approximation. iee transactions on signal processing,. p. 52, 2011. Citado 4 vezes nas páginas 30, 43, 58 e 64.
- MATOS, N. de F. O. Estimação de atitude de satélites artificiais com quatérnions. p. 35, 2014. Citado 2 vezes nas páginas 30 e 39.
- MICHELSON. Eulerangles for libration analysis. p. 30, 1964. Citado 2 vezes nas páginas 29 e 34.
- RICARTE, J. C. P. Giroscópio de fibra Óptica. In: . [S.l.: s.n.], 2016. p. 39. Citado 2 vezes nas páginas 31 e 39.
- SHEN X.; DENG, L. Gametheory approach to discrete h_{∞} filter design. In: . [S.l.: s.n.], 1997. p. 39. Citado 2 vezes nas páginas 29 e 44.
- SILVA. Estudo da estabilidade do movimento rotacional de satélites artificiais com variáveis canônicas.. p. 33, 2012. Citado 2 vezes nas páginas 30 e 37.
- SILVA, W. R. da. Filtro h_{∞} estendido de segunda ordem e filtro de particulas regularizado com roughening aplicados na estimação de atitudes de satélites artificiais. p. 29, 2016. Citado 10 vezes nas páginas 13, 31, 33, 34, 35, 36, 44, 53, 58 e 64.
- SIMON, D. Optimal state estimation: Kalman, h_{∞} , and nonlinear approaches. In: . [S.l.: s.n.], 2006. p. 39. Citado 5 vezes nas páginas 30, 43, 44, 45 e 49.

WERTZ, J. R. Spacecraft attitude determination and control. In: . [S.l.: s.n.], 1978. p. 33. Citado 4 vezes nas páginas [29](#), [37](#), [39](#) e [55](#).

APÊNDICE A – Algoritmo Demo Propat Real State.m propagação do modelo matemático.

A rotina descrita foi desenvolvida no software matlab para análise computacional.

```

demo_propat_real_state
%
% Objetivo: Simular a atitude e órbita de satélites artificiais
%
% Comentários:
% ...
%
%Sensores:
%...DSS_psi==>sensor solar digital [rad]
%...DSS_teta==> sensor solar digital [rad]
%...IRES_fi==> sensor de horizonte [deg]
%...IRES_teta==> sensor de horizonte [deg]
% Inputs: x0, P0,
%
% Outputs: "True States"          >>>>>>>          x_real;
%          "Angular Velocity"     >>>>>>>          omeg;
%          "Observated State"     >>>>>>>          z_real;
%          "Orbital Parameters"   >>>>>>>          keorb;
%          "Orbital Position and Velocity" >>>>>> orbit;
%
%.....
% Ultima Atualização: 19/05/2014
% Revisado por: William R. S. ; Hélio K. K., Roberta,
%.....
%
% contador
contad = 1;

```

```

% Orbit keplerian elements:
kepel = [7149000, 0.01, 1.7104, 0, 0, 0]; % see function delkep

% Orbit state vector:
stat = kepel_statvec(kepel);

% Attitude elements in Euler angles of a 3-2-1 (z-y-x) rotation
eulzyx = [3.0, 5.0, 2.0]'*pi/180; % converted from degrees to radians
%eulzyx = [-0.48, -0.48, -1.3]'*pi/180;

% Attitude in quaternions
quat = ezyxquat(eulzyx); % converted from Euler angles

% Angular velocity vector in body frame:
w_ang = [0.000002, 0.000003, 0.000005]'; % in radians/sec

% Initial bias
%BIAS = [0.76, 4.83, 2.68]'*(pi/180)/3600;; % in radians/sec
BIAS = [0.1, 0.1, 0.1]'*(pi/180)/3600; % converted from degrees/hours to radians/sec

% Real measurements
z_real = [0, 0, 0, 0]';

% Compute the variations in keplerian elements due to the Earth oblateness
delk = delkep(kepel);

% Ephemerides date in Modified Julian date:
year = 2006;
mjd = djm(22, 4, year); % Format (day, month, year)
mjdo = djm(1, 1, year); % modified julian date of 1/1/year
mjd1 = djm(1, 1, year+1); % modified julian date of 1/1/(year+1)
year_frac = year + (mjd - mjdo)/(mjd1 - mjdo); % year and fraction

% Ephemerides time:
dfra = time_to_dayf (13, 46, 25); % UTC time in (hour, minute, sec)

% Propagation time in seconds:
tstart = 0; % initial time (sec)

```

```

tstep = 0.5;    % step time (sec)
tend = 600;    % end time (10 minutes)

% Inertia matrix of axis-symmetric rigid body:
alfa = 0;
Trotpai = [cos(alfa) 0 sin(alfa); 0 1 0; -sin(alfa) 0 cos(alfa)];
Tpai = [1236 -0.5 0; -0.5 31 -16; 0 -16 1267];
iner = [1086 -3 15; -3 1057 12; 15 12 780] + Trotpai'*Tpai*Trotpai;    % in kg*m*

% Inverse inertia matrix:
invin = inv(iner);

% Initial control torque:
contq = [0 0 0]';

% Magnetic moment torque flag and moment:
flag_mag = 0;    % 1=compute magnetic moment / 0=discard magnetic moment
mag_mon = [0; 0; 0.1];    % in A.m

% ODE solver precision:
options = odeset('abstol', 1e-4, 'reltol', 1e-4);

% Initial vectors
time = tstart;    % to store time
euler = eulzyx*180/pi;    % Euler angles
omeg = w_ang;    % Angular velocity
BIas = BIAS*(180/pi)*3600;    % gyros bias
orbit = stat';    % Orbit elements (state vector)
keorb = kepel';    % Orbit elements (keplerian)

% Attitude and orbit propagation
for t = tstart:tstep:tend

    % Velocidade angular orbital [rad/s]
    rr = sqrt(orbit(1,contad)^2 + orbit(2,contad)^2 + orbit(3,contad)^2);    %vet
    w0 = sqrt(3.986005e14/rr^3);    % movimento médio orbital (n^2=mi/r^3)

    % Orbit propagation

```

```

kep2 = kepel + delk*t;

% To convert from keplerian elements to state vector (if needed)
stat = kepel_statvec(kep2);

% Perturbation torques:
ambt = [0 0 0]';

% External torques (perturbation + control)
ext_torq = ambt + contq;

% Initial attitude vector:
att_vec = [quat; w_ang]';           % Transposed

% ODE Solver parameters
tspan = [t, t+tstep/2, t+tstep];

% Numeric integration (ODE45)
if flag_mag == 0
    [T, Y] = ode45('rigbodyA', tspan, att_vec, options, ext_torq, iner, invin);
else
    % To convert from inertial state vector to terrestrial vector
    geoc = inertial_to_terrestrial(gst(mjd, dfra+t), stat);

    % Earth's magnetic field
    sphe = rectangular_to_spherical(geoc);
    alt = sphe(3)/1000;
    elong = sphe(1);
    colat = pi/2 - sphe(2);
    earth_field = 1.e-9*igrf_field(year_frac, alt, colat, elong);

    [T, Y] = ODE45('rigbodyA', tspan, att_vec, options, ext_torq, iner, invin, ...
        mag_mom, earth_field);
end

% covariances matrix for random states
Q1 = [0.000000008, 0, 0, 0; 0, 0.000000007, 0, 0; 0, 0, 0.000000009, 0; 0, 0, 0, 0.000000008];
Q2 = [0.0000000008, 0, 0; 0, 0.00007, 0; 0, 0, 0.0000000009];

```

```
% Output visualization
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% True States
```

```
figure
```

```
plot(time, x_real(1,:));
```

```
xlabel('Time (s)')
```

```
ylabel('\phi (deg)')
```

```
title('Estimated Roll angle (\phi)')
```

```
legend('\phi - (true state)')
```

```
figure
```

```
plot(time, x_real(2,:));
```

```
xlabel('Time (s)')
```

```
ylabel('\theta (deg)')
```

```
title('Estimated Pitch angle (\theta)')
```

```
legend('\theta - (true state)')
```

```
figure
```

```
plot(time, x_real(3,:));
```

```
xlabel('Time (s)')
```

```
ylabel('\psi (deg)')
```

```
title('Estimated Yaw angle (\psi)')
```

```
legend('\psi - (true state)')
```

```
figure
```

```
plot(time, x_real(4,:));
```

```
xlabel('Time (s)')
```

```
ylabel('\epsilon_{x} (deg/h)')
```

```
title('Estimated Bias x axes (\epsilon_{x})')
```

```
legend('\epsilon_{x} - (true state)')
```

```
figure
```

```
plot(time, x_real(5,:));
```

```
xlabel('Time (s)')
```

```
ylabel('\epsilon_{y} (deg/h)')
```

```
title('Estimated Bias y axes (\epsilon_{y})')
```

```
legend('\epsilon_{y} - (true state)')
```

```

figure
plot(time, x_real(6,:));
xlabel('Time (s)')
ylabel('\epsilon_{z} (deg/h)')
title('Estimated Bias z axes (\epsilon_{z})')
legend('\epsilon_{z} - (true state)')

```

% Angular Velocity

```

figure
%subplot(3, 1, 1);
subplot(3, 1, 1,'FontSize',14);
plot(time, omeg(1,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
xlabel('Tempo (s)')
ylabel('Vel. Angular (/h)')
legend('\omega_{x}')

```

```

%subplot(3, 1, 2);
subplot(3, 1, 2,'FontSize',14);
plot(time, omeg(2,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
xlabel('Tempo (s)')
ylabel('Vel. Angular (/h)')
legend('\omega_{y}')
ylim([-230,-190]);

```

```

%subplot(3, 1, 3);
subplot(3, 1, 3,'FontSize',14);
plot(time, omeg(3,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
xlabel('Tempo (s)')

```

```
ylabel('Vel. Angular (/h)')
legend('\omega_{z}')

% Observated State

figure
%subplot(2, 1, 1);
subplot(2, 1, 1,'FontSize',14);
plot(time, z_real(1,:)*180/pi, time, z_real(2,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Angle (deg)')
%title('Measurements Digital Sun Sensor')
xlabel('Tempo (s)')
ylabel('Ângulo (°)')
legend('DSS_{1}', 'DSS_{2}')

%subplot(2, 1, 2);
subplot(2, 1, 2,'FontSize',14);
plot(time, z_real(3,:)*180/pi, time, z_real(4,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Angle (deg)')
%title('Measurements Digital Sun Sensor')
xlabel('Tempo (s)')
ylabel('Ângulo (°)')
legend('IRES_{1}', 'IRES_{2}')

% Orbital Parameters

figure
subplot(3, 1, 1);
plot(time, keorb(1,:)*1000);
xlabel('Time (s)')
ylabel('semimajor axis (km)')
title('Semimajor axis of the orbit')
legend('a')

subplot(3, 1, 2);
plot(time, keorb(2,:));
xlabel('Time (s)')
```

```
ylabel('eccentricity')
title('Eccentricity')
legend('e')

subplot(3, 1, 3);
plot(time, keorb(3,:)*180/pi);
xlabel('Time (s)')
ylabel('inclination (deg)')
title('inclination')
legend('I')

figure
subplot(3, 1, 1);
plot(time, keorb(4,:)*180/pi);
xlabel('Time (s)')
ylabel('Ascencion node (deg)')
title('Right ascencion of the ascending node')
legend('\Omega')

subplot(3, 1, 2);
plot(time, keorb(5,:)*180/pi);
xlabel('Time (s)')
ylabel('Perigee argument (deg)')
title('Perigee argument')
legend('\varpi')

subplot(3, 1, 3);
plot(time, keorb(6,:)*180/pi);
xlabel('Time (s)')
ylabel('Mean anomaly (deg)')
title('Mean anomaly')
legend('\eta')

% Orbital Position and Velocity

figure
subplot(2, 1, 1);
plot(time, orbit(1:3,:)/1000);
xlabel('Time (s)')
```



```
ylabel('Position (km)')  
title('Satellite inertial position')  
legend('X', 'Y', 'Z')
```

```
subplot(2, 1, 2);  
plot(time, orbit(4:6,:)/1000);  
xlabel('Time (s)')  
ylabel('Velocity (km/s)')  
title('Satellite velocity')  
legend('v_{x}', 'v_{y}', 'v_{z}');
```


APÊNDICE B – Algoritmo Demo propat EKF.m descreve o Filtro de Kalman Estendido.

```

% demo_propat_EKF
%
% Objetivo: Estimar a atitude de satélites artificiais utilizando EKF.
%
% Comentários:
% vetor de estado inicial==> x0=[fi teta psi bp bq br]'
%           (fi,teta,psi)=angulos de euler [rad]
%           (bp,bq,br)=bias do giroscópio [rad/s]
% x_prop,x_est==> estado propagado e atualizado para instante k
%
%
%Sensores:
%...DSS_psi==>sensor solar digital [rad]
%...DSS_teta==> sensor solar digital [rad]
%...IRES_fi==> sensor de horizonte [deg]
%...IRES_teta==> sensor de horizonte [deg]
% Inputs: x0, P0,
%
% Outputs: x_est, P_est
%
%.....
% Ultima Atualização: 19/05/2014
% Revisado por: William R. S. ; Hélio K. K.
%.....
%

tic

% contador
contador = 2;

```

```

% Inicialização das Condições Iniciais para a filtragem de dados:
condicoes_iniciais;

%Inicializa com as condições iniciais utilizada na filtragem de
%dados
x_est = x0;
P_est = P0;

% Propagation time in seconds:
tstart = 0;      % initial time (sec)
tstep = 0.5;    % step time (sec)
tend = 600;     % end time (10 minutes)

% Initial vectors
time = tstart;      % to store time
sigma1 = SigAttX;   % ver condições iniciais
sigma2 = SigAtty;
sigma3 = SigAttz;
sigma4 = SigBiasx;
sigma5 = SigBiasy;
sigma6 = SigBiasz;
xest = [0;0;0;0;0;0];
res_prop = [0;0;0;0];
res_atual = [0;0;0;0];

% Attitude and orbit propagation
for t = tstart:tstep:tend

    % AQUI COMEÇA!!!!!!!

    Wgyro = omeg(:,contador);% - x_est(4:6);    % medida dos giros com bias

    % Velocidade angular orbital [rad/s]
    rr = sqrt(orbit(1,contador)^2 + orbit(2,contador)^2 + orbit(3,contador)^2); %ve
    w0 = sqrt(3.986005e14/rr^3); % movimento médio orbital (n^2=mi/r^3)

```

```

%% Filtro de Kalman Estendido

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Propagação o estado (Runge Kutta 4):
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%y = zeros(1,3);
y = zeros(1,6);
%ne = 3;
ne = 6;
%West = Wgyro + x_est(4:6);
West = Wgyro;% + x_est(4:6);

y = rk4_nova(w0,ne,tstep,West,x_est(1:6));
%x_prop = [y';x_est(4:6)];
x_prop = y';

%.....

% Aqui é realizado o manuseio matemático par encontrar a Jacobiana
% de h(x), ou seja H

% Calculo dos versores orbitais (Vp, Vq, Vw)
[Vp, Vq, Vw] = orbPQW(orbit(:,contador));

% Calculo do vetor posição do sol no sistema inercial geocentrico:
S = SUN_50(mjd, dfra);

% Calculo do vetor solar unitario no sistema orbital (xo,yo,zo)
S0 = SUN_BF(S, Vp, Vq, Vw);

% Sensores (DSS & IRES):
[DSS,HPSIdss,HTETdss,HDSS] = CBERS_DSS(S0,x_prop(1:3));
[IRES,HFIires,HTETires,HIRES] = CBERS_IRES(x_prop(1),x_prop(2),0,0);

% Jacobiana H
H = [HDSS;HIRES];

% vetor de observação (4x1)

```

```

z = [DSS IRES]';% + sqrt(Rx)*randn(4,1);
%medida(:, :, t) = z;

%.....

% Calculo das matrizes jacobianas de f(x), ou seja F;
F = JACOB_F(w0,West,x_prop(1:3));

% Matriz de transição (fi):
fi = expm(tstep*F);

% Matriz de covariancia propagada:
P_prop = fi*P_est*fi' + Q;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Atualização do filtro:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Cálculo do Resíduo do Filro de Kalman Estendido
res = z_real(:, contador) - z(:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Linhas de comando do Filtro de Kalman Estendido
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

K = P_prop*H'*inv(H*P_prop*H' + Rx);

P_est = (eye(6) - K*H)*P_prop;
x_est = x_prop + K*(res);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%.....Resíduos Pós Atualização.....

```

```

[DSSa,HPSIdssa,HTETdssa,H_DSSa] = CBERS_DSS(S0,x_est(1:3));
[IRESa,HFIiresa,HTETiresa,H_IRESa] = CBERS_IRES(x_est(1),x_est(2),0,0);

z_atual = [DSSa IRESa]';

% Cálculo do Resíduo do Filro de Kalman Estendido
res_a = z_real(:,contador) - z_atual(:);

% Desvio-Padrão da Covariância

Sigma1 = sqrt(P_est(1,1));           % [rad]
Sigma2 = sqrt(P_est(2,2));           % [rad]
Sigma3 = sqrt(P_est(3,3));           % [rad]
Sigma4 = sqrt(P_est(4,4));           % [rad/s]
Sigma5 = sqrt(P_est(5,5));           % [rad/s]
Sigma6 = sqrt(P_est(6,6));           % [rad/s]

%.....

%% Store data to be plotted
time = cat(2, time, t);

sigma1 = cat(2, sigma1, Sigma1);
sigma2 = cat(2, sigma2, Sigma2);
sigma3 = cat(2, sigma3, Sigma3);
sigma4 = cat(2, sigma4, Sigma4);
sigma5 = cat(2, sigma5, Sigma5);
sigma6 = cat(2, sigma6, Sigma6);

%x_est(5) = x_est(5) - w0;
xest = cat(2,xest,x_est);
res_prop = cat(2,res_prop,res);
res_atual = cat(2,res_atual,res_a);

```

```

    contador = contador + 1;
end

tempo_proc = toc;

xest(5,:) = xest(5,:) - w0;
x_estg = xest;           % [deg; deg; deg; deg/s; deg/s; deg/s]
x_estg(1,:) = xest(1,)*deg;
x_estg(2,:) = xest(2,)*deg;
x_estg(3,:) = xest(3,)*deg;
x_estg(4,:) = xest(4,)*deg*3600;
x_estg(5,:) = xest(5,)*deg*3600;
x_estg(6,:) = xest(6,)*deg*3600;
inov = res_prop*deg;    % resíduo pós-propagação (inovação)
residuo_atual = res_atual*deg; % resíduo pós atualização

% Mean of Residual
MeanResidual = mean(residuo_atual,2);

% Standard Deviation of Residual
StdResidual = std(residuo_atual,1,2);

close all

% Estimation Error
Error = x_real - x_estg;
>Error(4,:) = abs(Error(4,:));
>Error(5,:) = abs(Error(5,:));
>Error(6,:) = abs(Error(6,:));

% Mean of Estimation Error
MeanError = mean(Error(:,2:1202),2);

% Standard Deviation of Estimation Error
StdError = std(Error(:,2:1202),1,2);

```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output visualization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Estimated States

figure
plot(time, x_real(1,:),time,x_estg(1,:));
xlabel('Time (s)')
ylabel('\phi (deg)')
title('Estimated Roll angle (\phi)')
legend('\phi - (true state)', '\phi - (EKF)')

figure
plot(time, x_real(2,:),time,x_estg(2,:));
xlabel('Time (s)')
ylabel('\theta (deg)')
title('Estimated Pitch angle (\theta)')
legend('\theta - (true state)', '\theta - (EKF)')

figure
plot(time, x_real(3,:),time,x_estg(3,:));
xlabel('Time (s)')
ylabel('Euler angles (3-2-1) (deg)')
title('Estimated Yaw angle (\psi)')
legend('\psi - (true state)', '\psi - (EKF)')

figure
plot(time, x_real(4,:),time,x_estg(4,:));
xlabel('Time (s)')
ylabel('\epsilon_{x} (deg/h)')
title('Estimated Bias x axes (\epsilon_{x})')
legend('\epsilon_{x} - (true state)', '\epsilon_{x} - (EKF)')

figure
plot(time, x_real(5,:),time,x_estg(5,:));
```

```

xlabel('Time (s)')
ylabel('\epsilon_{y} (deg/h)')
title('Estimated Bias y axes (\epsilon_{y})')
legend('\epsilon_{y} - (true state)', '\epsilon_{y} - (EKF)')

figure
plot(time, x_real(6,:),time,x_estg(6,:));
xlabel('Time (s)')
ylabel('\epsilon_{z} (deg/h)')
title('Estimated Bias z axes (\epsilon_{z})')
legend('\epsilon_{z} - (true state)', '\epsilon_{z} - (EKF)')

% Angular Velocity

%figure
%subplot(3, 1, 1);
%plot(time, omeg(1,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
%legend('\omega_{x}')

%subplot(3, 1, 2);
%plot(time, omeg(2,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
%legend('\omega_{y}')

%subplot(3, 1, 3);
%plot(time, omeg(3,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
%legend('\omega_{z}')

% Observed State

%figure

```

```
%subplot(2, 1, 1);
%plot(time, z_real(1,:)*180/pi, time, z_real(2,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Angle (deg)')
%title('Measurements Digital Sun Sensor')
%legend('DSS_{1}', 'DSS_{2}')
```

```
%subplot(2, 1, 2);
%plot(time, z_real(3,:)*180/pi, time, z_real(4,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Angle (deg)')
%title('Measurements Digital Sun Sensor')
%legend('IRES_{1}', 'IRES_{2}')
```

```
% Orbital Parameters
```

```
%figure
%subplot(3, 1, 1);
%plot(time, keorb(1,:)*1000);
%xlabel('Time (s)')
%ylabel('semimajor axis (km)')
%title('Semimajor axis of the orbit')
%legend('a')
```

```
%subplot(3, 1, 2);
%plot(time, keorb(2,:));
%xlabel('Time (s)')
%ylabel('eccentricity')
%title('Eccentricity')
%legend('e')
```

```
%subplot(3, 1, 3);
%plot(time, keorb(3,:)*180/pi);
%xlabel('Time (s)')
%ylabel('inclination (deg)')
%title('inclination')
%legend('I')
```

```
%figure
```

```

%subplot(3, 1, 1);
%plot(time, keorb(4,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Ascencion node (deg)')
%title('Right ascencion of the ascending node')
%legend('\Omega')

%subplot(3, 1, 2);
%plot(time, keorb(5,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Perigee argument (deg)')
%title('Perigee argument')
%legend('\varpi')

%subplot(3, 1, 3);
%plot(time, keorb(6,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Mean anomaly (deg)')
%title('Mean anomaly')
%legend('\eta')

% Orbital Position and Velocity

%figure
%subplot(2, 1, 1);
%plot(time, orbit(1:3,:)/1000);
%xlabel('Time (s)')
%ylabel('Position (km)')
%title('Satellite inertial position ')
%legend('X', 'Y', 'Z')

%subplot(2, 1, 2);
%plot(time, orbit(4:6,:)/1000);
%xlabel('Time (s)')
%ylabel('Velocity (km/s)')
%title('Satellite velocity')
%legend('v_{x}', 'v_{y}', 'v_{z}')

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%Plot  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
% Plota os resíduos das medidas  
  
figure;  
%suptitle('Teste (DSS1)')  
subplot(2,2,1);  
plot(time, inov(1,:), '--', 'Color', 'blue', 'LineWidth', 1);  
title('Digital Solar Sensor 1 (DSS1)');  
ylabel('Inovation (deg)');  
xlabel('Time');  
legend('EKF');  
  
subplot(2,2,2);  
plot(time, residuo_atual(1,:), '--', 'Color', 'red', 'LineWidth', 1);  
title('Digital Solar Sensor 1 (DSS1)');  
ylabel('Residual (deg)');  
xlabel('Time');  
legend('EKF');  
  
subplot(2,2,3);  
plot(time, inov(2,:), '--', 'Color', 'blue', 'LineWidth', 1);  
title('Digital Solar Sensor 2 (DSS2)');  
ylabel('Inovation (deg)');  
xlabel('Time');  
legend('EKF');  
  
subplot(2,2,4);  
plot(time, residuo_atual(2,:), '--', 'Color', 'red', 'LineWidth', 1);  
title('Digital Solar Sensor 2 (DSS2)');  
ylabel('Residual (deg)');  
xlabel('Time');  
legend('EKF');  
  
figure;  
subplot(2,2,1);  
plot(time, inov(3,:), '--', 'Color', 'blue', 'LineWidth', 1);
```

```

title('Infrared Earth Sensor 1 (IRES1)');
ylabel('Inovation (deg)');
xlabel('Time');
legend('EKF');

subplot(2,2,2);
plot(time, residuo_atual(3,:), '-','Color','red','LineWidth',1);
title('Infrared Earth Sensor 1 (IRES1)');
ylabel('Residual (deg)');
xlabel('Time');
legend('EKF');

subplot(2,2,3);
plot(time, inov(4,:), '-','Color','blue','LineWidth',1);
title('Infrared Earth Sensor 2 (IRES2)');
ylabel('Inovation (deg)');
xlabel('Time');
legend('EKF');

subplot(2,2,4);
plot(time, residuo_atual(4,:), '-','Color','red','LineWidth',1);
title('Infrared Earth Sensor 2 (IRES2)');
ylabel('Residual (deg)');
xlabel('Time');
legend('EKF');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plota as covariâncias

%covariância da dos ângulos de Euler
figure
subplot(3,1,1)
plot(time,sigma1*deg, '-','Color','blue');
title('Covariance Attitude');
ylabel('\sigma_{\phi} (deg)');
xlabel('Time (h)');

```



```
% Estimation Error

figure
subplot(3,1,1)
plot(time,Error(1,:));
title('Estimation error state');
ylabel('\phi Error (deg)');
xlabel('Time (s)');
legend('EKF')
ylim([-0.15,0.15])

subplot(3,1,2)
plot(time,Error(2,:));
ylabel('\theta Error (deg)');
xlabel('Time (s)');
legend('EKF')
ylim([-0.15,0.15])

subplot(3,1,3)
plot(time,Error(3,:));
ylabel('\psi Error (deg)');
xlabel('Time (s)');
legend('EKF')

figure
subplot(3,1,1)
plot(time,Error(4,:));
title('Estimation error state');
ylabel('\epsilon_{x} Error (deg/h)');
xlabel('Time (s)');
legend('EKF')

subplot(3,1,2)
plot(time,Error(5,:));
ylabel('\epsilon_{y} Error (deg/h)');
xlabel('Time (s)');
legend('EKF')

subplot(3,1,3)
```



```
plot(time,Error(6,:));  
ylabel('\epsilon_{z} Error (deg/h)');  
xlabel('Time (s)');  
legend('EKF')
```


APÊNDICE C – Algoritmo Demo propat EHinf.m descreve o Filtro H_∞ Estendido.

```

% demo_propat_EHinfF
%
% Objetivo: Estimar a atitude de satélites artificiais utilizando EHinfF.
%
% Comentários:
% vetor de estado inicial==> x0=[fi teta psi bp bq br]'
%           (fi,teta,psi)=angulos de euler [rad]
%           (bp,bq,br)=bias do giroscópio [rad/s]
% x_prop,x_est==> estado propagado e atualizado para instante k
%
%
%Sensores:
%...DSS_psi==>sensor solar digital [rad]
%...DSS_teta==> sensor solar digital [rad]
%...IRES_fi==> sensor de horizonte [deg]
%...IRES_teta==> sensor de horizonte [deg]
% Inputs: x0, P0,
%
% Outputs: x_est, P_est
%
%.....
% Ultima Atualização: 10/02/2014
% Revisado por: William R. S. ; Hélio K. K.
%.....
%

tic

% contador
contador = 2;

% Inicialização das Condições Iniciais para a filragem de dados:
condicoes_iniciais_H;

```

```

%Inicializa com as condições iniciais utilizada na filtragem de
%dados
x_estH = x0;
P_estH = P0;

% Propagation time in seconds:
tstart = 0;      % initial time (sec)
tstep = 0.5;    % step time (sec)
tend = 600;     % end time (10 minutes)

% Initial vectors
time = tstart;      % to store time
sigma1H = SigAttX;  % ver condições iniciais
sigma2H = SigAtty;
sigma3H = SigAttz;
sigma4H = SigBiasx;
sigma5H = SigBiasy;
sigma6H = SigBiasz;
xestH = [0;0;0;0;0;0];
res_propH = [0;0;0;0];
res_atualH = [0;0;0;0];

% Attitude and orbit propagation
for t = tstart:tstep:tend

    % AQUI COMEÇA!!!!!!!

    Wgyro = omeg(:,contador);% - x_estH(4:6);    % medida dos giros com bias

    % Velocidade angular orbital [rad/s]
    rr = sqrt(orbit(1,contador)^2 + orbit(2,contador)^2 + orbit(3,contador)^2); %ve
    w0 = sqrt(3.986005e14/rr^3); % movimento médio orbital (n^2=mi/r^3)

    %% Filtro de Kalman Estendido

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Propagação o estado (Runge Kutta 4):
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

yH = zeros(1,6);

ne = 6;
%West = Wgyro + x_est(4:6);
WestH = Wgyro;% + x_estH(4:6);

yH = rk4_nova(w0,ne,tstep,WestH,x_estH(1:6));
%x_prop = [y';x_est(4:6)];
x_propH = yH';

%.....

% Aqui é realizado o manuseio matemático par encontrar a Jacobiana
% de h(x), ou seja H

% Calculo dos versores orbitais (Vp, Vq, Vw)
[Vp, Vq, Vw] = orbPQW(orbit(:,contador));

% Calculo do vetor posição do sol no sistema inercial geocentrico:
S = SUN_50(mjd, dfra);

% Calculo do vetor solar unitario no sistema orbital (xo,yo,zo)
S0 = SUN_BF(S, Vp, Vq, Vw);

% Sensores (DSS & IRES):
[DSSH,HPSIdssH,HTETdssH,HDSSH] = CBERS_DSS(S0,x_propH(1:3));
[IRESH,HFIiresH,HTETiresH,HIRESH] = CBERS_IRES(x_propH(1),x_propH(2),0,0);

% Jacobiana H
Hh = [HDSSH;HIRESH];

% vetor de observação (4x1)
zH = [DSSH IRESH]';% + sqrt(Rx)*randn(4,1);
%medida(:, :, t) = z;

```

```

%.....

% Calculo das matrizes jacobianas de f(x), ou seja F;
FH = JACOB_F(w0,WestH,x_propH(1:3));

% Matriz de transição (fi):
fiH = expm(tstep*FH);

% Matriz de covariancia propagada:
P_propH = fiH*P_estH*fiH' + Q;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Cálculo do Resíduo do Filro H Infinito Estendido
resH = z_real(:,contador) - zH(:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Linhas de comando do Filtro H Infinito Estendido
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

temp = eye(6) - gamma*Sbar*P_propH + Hh'*inv(Rx)*Hh*P_propH;

if cond(temp) > 1e10
    disp('ill conditioned H-infinity estimation problem')
    return;
end

G = inv(temp);

KH = P_propH*G*Hh'*inv(Rx);

x_estH = x_propH + fiH*KH*(resH);
P_estH = fiH*P_propH*G*fiH' + Q;

```

```

Gtest = eig(inv(P_propH) - gamma*Sbar + Hh'*inv(Rx)*Hh);

if min(real(Gtest)) <= 0
    disp('H-infinity performance bound out of range')
    return;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%.....Resíduos Pós Atualização.....

[DSSaH,HPSIdssaH,HTETdssaH,H_DSSaH] = CBERS_DSS(S0,x_estH(1:3));
[IRESaH,HFIiresaH,HTETiresaH,H_IRESaH] = CBERS_IRES(x_estH(1),x_estH(2),0,0);

z_atualH = [DSSaH IRESaH]';

% Cálculo do Resíduo pós do Filro H Infinito Estendido
res_aH = z_real(:,contador) - z_atualH(:);

% Desvio-Padrão da Covariância

Sigma1H = sqrt(P_estH(1,1));           % [rad]
Sigma2H = sqrt(P_estH(2,2));           % [rad]
Sigma3H = sqrt(P_estH(3,3));           % [rad]
Sigma4H = sqrt(P_estH(4,4));           % [rad/s]
Sigma5H = sqrt(P_estH(5,5));           % [rad/s]
Sigma6H = sqrt(P_estH(6,6));           % [rad/s]

%.....

```

```

%% Store data to be plotted
time = cat(2, time, t);

sigma1H = cat(2, sigma1H, Sigma1H);
sigma2H = cat(2, sigma2H, Sigma2H);
sigma3H = cat(2, sigma3H, Sigma3H);
sigma4H = cat(2, sigma4H, Sigma4H);
sigma5H = cat(2, sigma5H, Sigma5H);
sigma6H = cat(2, sigma6H, Sigma6H);

%x_estH(5) = x_estH(5) - w0;
xestH = cat(2,xestH,x_estH);
res_propH = cat(2,res_propH,resH);
res_atualH = cat(2,res_atualH,res_aH);

    contador = contador + 1;
end

tempo_proc = toc;

xestH(5,:) = xestH(5,:) - w0;
x_estgH = xestH;                % convert to [deg; deg; deg; deg/s; deg/s; deg/s]
x_estgH(1,:) = xestH(1,)*deg;
x_estgH(2,:) = xestH(2,)*deg;
x_estgH(3,:) = xestH(3,)*deg;
x_estgH(4,:) = xestH(4,)*deg*3600;
x_estgH(5,:) = xestH(5,)*deg*3600;
x_estgH(6,:) = xestH(6,)*deg*3600;
inovH = res_propH*deg;          % resíduo pós-propagação (inovação)
residuo_atualH = res_atualH*deg; % resíduo pós atualização

% Mean of ResidualH
MeanResidualH = mean(residuo_atualH,2);

% Standard Deviation of ResidualH
StdResidualH = std(residuo_atualH,1,2);

```

```
close all
```

```
% Estimation ErrorH
```

```
ErrorH = x_real - x_estgH;
```

```
%ErrorH(4,:) = abs(ErrorH(4,:));
```

```
%ErrorH(5,:) = abs(ErrorH(5,:));
```

```
%ErrorH(6,:) = abs(ErrorH(6,:));
```

```
% Mean of Estimation ErrorH
```

```
MeanErrorH = mean(ErrorH(:,2:1202),2);
```

```
% Standard Deviation of Estimation ErrorH
```

```
StdErrorH = std(ErrorH(:,2:1202),1,2);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Output visualization
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Estimated States
```

```
figure
```

```
plot(time, x_real(1,:),time,x_estg(1,:),time,x_estgH(1,:));
```

```
xlabel('Time (s)')
```

```
ylabel('\phi (deg)')
```

```
title('Estimated Roll angle (\phi)')
```

```
legend('\phi - (true state)', '\phi - (EKF)', '\phi - (EH_{\infty}F)')
```

```
figure
```

```
plot(time, x_real(2,:),time,x_estg(2,:),time,x_estgH(2,:));
```

```
xlabel('Time (s)')
```

```
ylabel('\theta (deg)')
```

```
title('Estimated Pitch angle (\theta)')
```

```
legend('\theta - (true state)', '\theta - (EKF)', '\theta - (EH_{\infty}F)')
```

```
figure
```

```
plot(time, x_real(3,:),time,x_estg(3,:),time,x_estgH(3,:));
```

```
xlabel('Time (s)')
```

```

ylabel('Euler angles (3-2-1) (deg)')
title('Estimated Yaw angle (\psi)')
legend('\psi - (true state)', '\psi - (EKF)', '\psi - (EH_{\infty}F)')

figure
plot(time, x_real(4,:),time,x_estg(4,:),time,x_estgH(4,:));
xlabel('Time (s)')
ylabel('\epsilon_{x} (deg/h)')
title('Estimated Bias x axes (\epsilon_{x})')
legend('\epsilon_{x} - (true state)', '\epsilon_{x} - (EKF)', '\epsilon_{x} - (EH_{\infty}F)')

figure
plot(time, x_real(5,:),time,x_estg(5,:),time,x_estgH(5,:));
xlabel('Time (s)')
ylabel('\epsilon_{y} (deg/h)')
title('Estimated Bias y axes (\epsilon_{y})')
legend('\epsilon_{y} - (true state)', '\epsilon_{y} - (EKF)', '\epsilon_{y} - (EH_{\infty}F)')

figure
plot(time, x_real(6,:),time,x_estg(6,:),time,x_estgH(6,:));
xlabel('Time (s)')
ylabel('\epsilon_{z} (deg/h)')
title('Estimated Bias z axes (\epsilon_{z})')
legend('\epsilon_{z} - (true state)', '\epsilon_{z} - (EKF)', '\epsilon_{z} - (EH_{\infty}F)')

% Angular Velocity

%figure
%subplot(3, 1, 1);
%plot(time, omeg(1,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
%legend('\omega_{x}')

%subplot(3, 1, 2);
%plot(time, omeg(2,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')

```

```
%title('Attitude angular velocity')
%legend('\omega_{y}')

%subplot(3, 1, 3);
%plot(time, omeg(3,:)*(180/pi)*3600);
%xlabel('Time (s)')
%ylabel('Angular velocity (deg/h)')
%title('Attitude angular velocity')
%legend('\omega_{z}')

% Observated State

%figure
%subplot(2, 1, 1);
%plot(time, z_real(1,:)*180/pi, time, z_real(2,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Angle (deg)')
%title('Measurements Digital Sun Sensor')
%legend('DSS_{1}', 'DSS_{2}')
```

```
%subplot(2, 1, 2);
%plot(time, z_real(3,:)*180/pi, time, z_real(4,:)*180/pi);
%xlabel('Time (s)')
%ylabel('Angle (deg)')
%title('Measurements Digital Sun Sensor')
%legend('IRES_{1}', 'IRES_{2}')
```

```
% Orbital Parameters

%figure
%subplot(3, 1, 1);
%plot(time, keorb(1,:)*1000);
%xlabel('Time (s)')
%ylabel('semimajor axis (km)')
%title('Semimajor axis of the orbit')
%legend('a')
```

```
%subplot(3, 1, 2);
%plot(time, keorb(2,:));
```

```

xlabel('Time (s)')
ylabel('eccentricity')
title('Eccentricity')
legend('e')

subplot(3, 1, 3);
plot(time, keorb(3,:)*180/pi);
xlabel('Time (s)')
ylabel('inclination (deg)')
title('inclination')
legend('I')

figure
subplot(3, 1, 1);
plot(time, keorb(4,:)*180/pi);
xlabel('Time (s)')
ylabel('Ascension node (deg)')
title('Right ascension of the ascending node')
legend('\Omega')

subplot(3, 1, 2);
plot(time, keorb(5,:)*180/pi);
xlabel('Time (s)')
ylabel('Perigee argument (deg)')
title('Perigee argument')
legend('\varpi')

subplot(3, 1, 3);
plot(time, keorb(6,:)*180/pi);
xlabel('Time (s)')
ylabel('Mean anomaly (deg)')
title('Mean anomaly')
legend('\eta')

% Orbital Position and Velocity

figure
subplot(2, 1, 1);
plot(time, orbit(1:3,:)/1000);

```

```

%xlabel('Time (s)')
%ylabel('Position (km)')
%title('Satellite inertial position ')
%legend('X', 'Y', 'Z')

%subplot(2, 1, 2);
%plot(time, orbit(4:6,:)/1000);
%xlabel('Time (s)')
%ylabel('Velocity (km/s)')
%title('Satellite velocity')
%legend('v_{x}', 'v_{y}', 'v_{z}')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plota os resíduos das medidas

figure;
subplot(2,2,1);
plot(time, inov(1,:),time, inovH(1,:));
title('Digital Solar Sensor 1 (DSS1)');
ylabel('Inovation (deg)');
xlabel('Time');
legend('EKF', 'EH_{\infty}F');

subplot(2,2,2);
plot(time, residuo_atual(1,:),time, residuo_atualH(1,:));
title('Digital Solar Sensor 1 (DSS1)');
ylabel('Residual (deg)');
xlabel('Time');
legend('EKF', 'EH_{\infty}F');

subplot(2,2,3);
plot(time, inov(2,:),time, inovH(2,:));
title('Digital Solar Sensor 2 (DSS2)');
ylabel('Inovation (deg)');
xlabel('Time');

```



```

%Plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plota as covariâncias

%covariância da dos ângulos de Euler
figure
subplot(3,1,1)
plot(time,sigma1*deg,'-',time,sigma1H*deg,'-');
title('Covariance Attitude');
ylabel('\sigma_{\phi} (deg)');
xlabel('Time (h)');
legend('\sigma_{\phi} - EKF', '\sigma_{\phi} - EH_{\infty}F')

subplot(3,1,2)
plot(time,sigma2*deg,'-',time,sigma2H*deg,'-');
ylabel('\sigma_{\theta} (deg)');
xlabel('Time (h)');
legend('\sigma_{\theta} - EKF', '\sigma_{\theta} - EH_{\infty}F')

subplot(3,1,3)
plot(time,sigma3*deg,'-',time,sigma3H*deg,'-');
ylabel('\sigma_{\psi} (deg)');
xlabel('Time (h)');
legend('\sigma_{\psi} - EKF', '\sigma_{\psi} - EH_{\infty}F')

%Covariância dos Bias dos gyros
figure
subplot(3,1,1)
plot(time,sigma4*deg*3600,'-',time,sigma4H*deg*3600,'-');
title('Covariance Gyros Bias');
ylabel('\sigma_{\epsilon_{x}} (deg/h)');
xlabel('Time (h)');
legend('\sigma_{\epsilon_{x}} - EKF', '\sigma_{\epsilon_{x}} - EH_{\infty}F')

subplot(3,1,2)
plot(time,sigma5*deg*3600,'-',time,sigma5H*deg*3600,'-');
ylabel('\sigma_{\epsilon_{y}} (deg/h)');
xlabel('Time (h)');

```

```

legend('\sigma_{\epsilon_y} - EKF', '\sigma_{\epsilon_y} - EH_{\infty}F')

subplot(3,1,3)
plot(time,sigma6*deg*3600,'-',time,sigma6H*deg*3600,'-');
ylabel('\sigma_{\epsilon_z} (deg/h)');
xlabel('Time (h)');
legend('\sigma_{\epsilon_z} - EKF', '\sigma_{\epsilon_z} - EH_{\infty}F')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Estimation Error

figure
subplot(3,1,1)
plot(time>Error(1,:),time>ErrorH(1,:));
title('Estimation error state');
ylabel('\phi Error (deg)');
xlabel('Time (s)');
legend('EKF', 'EH_{\infty}F')
ylim([-0.15,0.15])

subplot(3,1,2)
plot(time>Error(2,:),time>ErrorH(2,:));
ylabel('\theta Error (deg)');
xlabel('Time (s)');
legend('EKF', 'EH_{\infty}F')
ylim([-0.15,0.15])

subplot(3,1,3)
plot(time>Error(3,:),time>ErrorH(3,:));
ylabel('\psi Error (deg)');
xlabel('Time (s)');
legend('EKF', 'EH_{\infty}F')

figure
subplot(3,1,1)
plot(time>Error(4,:),time>ErrorH(4,:));

```

```
title('Estimation error state');
ylabel('\epsilon_{x} Error (deg/h)');
xlabel('Time (s)');
legend('EKF', 'EH_{\infty}F')

subplot(3,1,2)
plot(time,Error(5,:),time,ErrorH(5,:));
ylabel('\epsilon_{y} Error (deg/h)');
xlabel('Time (s)');
legend('EKF', 'EH_{\infty}F')

subplot(3,1,3)
plot(time,Error(6,:),time,ErrorH(6,:));
ylabel('\epsilon_{z} Error (deg/h)');
xlabel('Time (s)');
legend('EKF', 'EH_{\infty}F')
```