MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2016/07.14.00.14-TDI

PROJETO E SIMULAÇÃO DO SISTEMA DE VALIDAÇÃO E TESTES DE CONTROLE DE ATITUDE COM APLICAÇÃO EM MICRO-SATÉLITES

Arnaldo Ferreira de Oliveira Junior

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelos Drs. Helio Koiti Kuga, e Valdemir Carrara, aprovada em 25 de maio de 2016.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34P/3M47EC5>

INPE
São José dos Campos
2016

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
Fax: (012) 3208-6919
E-mail: pubtc@inpe.br

COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):
Presidente:
Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação (CPG)
Membros:
Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)
Dr. André de Castro Milone - Coordenação de Ciências Espaciais e Atmosféricas (CEA)
Dra. Carina de Barros Melo - Coordenação de Laboratórios Associados (CTE)
Dr. Evandro Marconi Rocco - Coordenação de Engenharia e Tecnologia Espacial (ETE)
Dr. Hermann Johann Heinrich Kux - Coordenação de Observação da Terra (OBT)
Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CPT)
Silvia Castro Marcelino - Serviço de Informação e Documentação (SID)
BIBLIOTECA DIGITAL:
Dr. Gerald Jean Francis Banon
Clayton Martins Pereira - Serviço de Informação e Documentação (SID)
REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:
Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SID)
Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)
EDITORAÇÃO ELETRÔNICA:
Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)
André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2016/07.14.00.14-TDI

PROJETO E SIMULAÇÃO DO SISTEMA DE VALIDAÇÃO E TESTES DE CONTROLE DE ATITUDE COM APLICAÇÃO EM MICRO-SATÉLITES

Arnaldo Ferreira de Oliveira Junior

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelos Drs. Helio Koiti Kuga, e Valdemir Carrara, aprovada em 25 de maio de 2016.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34P/3M47EC5>

INPE
São José dos Campos
2016

Dados Internacionais de Catalogação na Publicação (CIP)

Oliveira Junior, Arnaldo Ferreira de.

Ol4p Projeto e simulação do sistema de validação e testes de controle de atitude com aplicação em micro-satélites / Arnaldo Ferreira de Oliveira Junior. – São José dos Campos : INPE, 2016.

xvi + 83 p. ; (sid.inpe.br/mtc-m21b/2016/07.14.00.14-TDI)

Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2016.

Orientadores : Drs. Helio Koiti Kuga, e Valdemir Carrara.

1. Controle de atitude. 2. Filtro de Kalman. 3. Simulação. I.Título.

CDU 629.7.062.2

Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.

Aluno (a): ***Arnaldo Ferreira de Oliveira Junior***

Título: "PROJETO E SIMULAÇÃO DO SISTEMA DE VALIDAÇÃO E TESTES DE CONTROLE DE ATITUDE COM APLICAÇÃO EM MICRO-SATÉLITES".

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de *Mestre* em

Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle

Dr. Helio Koiti Kuga

Presidente / Orientador(a) / INPE / SJCampos - SP

Dr. Valdemir Carrara

Orientador(a) / INPE / SJCampos - SP

Dr. Élcio Jeronimo de Oliveira

Membro da Banca / IAE/DCTA / São José dos Campos - SP

Dr. Osamu Saotome

Convidado(a) / ITA/DCTA / SJCampos - SP

Este trabalho foi aprovado por:

( ) maioria simples

(✓) unanimidade

São José dos Campos, 25 de Maio de 2016

RESUMO

Neste trabalho apresentaremos por meio de modelagem matemática dos sensores (magnetômetro, sensor solar e giroscópios MEMS) e dos atuadores (bobinas magnéticas) um simulador que represente as características reais do sistema de determinação e controle de atitude a serem desenvolvidos pelo INPE e que serão embarcados no computador de bordo do satélite NBR-2, um Cubesat 2U em desenvolvimento com lançamento previsto para 2016. De forma a extrair informações úteis dos experimentos de bordo, o satélite deverá contar com um sistema de determinação de atitude. O algoritmo de determinação de atitude deve ser determinístico, obtendo os parâmetros de atitude diretamente a partir das medidas efetuadas pelos sensores, que são contaminadas com ruído. O método TRIAD será utilizado na determinação de atitude; o processo de estimação de atitude e do viés do giroscópio será executado por um filtro de Kalman. A propagação do estado será feita pela dinâmica do corpo rígido livre de torques externos, e pela equação da cinemática em quatérnio. O sistema de controle de atitude deverá, em seu modo nominal, efetuar o apontamento de uma das faces do NBR-2 para a Terra com precisão ao redor de cinco graus. O equacionamento do filtro, as equações de propagação e os resultados de simulação do movimento do satélite em órbita serão apresentados. Estes resultados serão utilizados na codificação do software a ser embarcado nas próximas missões brasileiras baseadas em Cubesats. Para codificação do software serão apresentadas as principais características do computador de bordo NanoMind do NBR-2, bem como o sistema operacional FreeRTOS. Os resultados mostram que o algoritmo do Filtro de Kalman, mesmo sob condições iniciais imprecisas, é capaz de convergir. Estes resultados serão úteis na codificação de software dos satélites do programa NBR-2, além de outros programas a serem desenvolvidos pelo INPE baseados na plataforma Cubesat.

DESIGN AND SIMULATION OF THE VALIDATION AND TESTS SYSTEM AND ATTITUDE CONTROL WITH APPLICATION IN MICROSATELLITES

ABSTRACT

This work shows by mathematical modeling of the sensors (magnetometer, solar sensor and MEMS Gyros) and the actuators (magnetic coils), a computer simulator representing the real caracteristics of attitude determination and control system for Cubesat missions at INPE and will be embedded in onboard computer of sattelite NBR-2, a Cubesat in development with scheduled launch at 2016. In order to extract usefull information from onboard experiments, the sattelite should rely in an attitude determination system. The algorithm of attitude determination should be deterministic, obtaining the attitude parameters direct from measurements of sensors wich are contaminated with noise. The TRIAD or QUEST methods will be used for attitude determination; the estimation process of attitude and gyros bias will be executed by a Kalman filter. A rigid body dynamics without external torques, together with the kynematic equations, will be employed for the filter state propagation in terms of quaternions. The attitude control system, in nominal mode, should point one face of NBR-2 to Earth, with accuracy around five degrees. The Kalman filter equations, the state variables dynamic equations and the results from attitude and orbit simulation will be presented. This results will be used in embedded code of ADCS (Attitude Determination and Control System) for next Brazilian missions based on Cubesat. The main characteristics of NanoMind onboard computer of the NBR-2 is presented, as well as the FreeRTOS operational system. Results show that the Kalman filter algorithm, even under inaccurate initial conditions, can converge. This results will be usefull in software coding for NBR sattelite program, or others ADCS softwares to be developed by INPE based on the Cubesat plataform.

LISTA DE FIGURAS

Figura 2.1 – Exemplo de rotação 3-1-3 7
Figura 2.2 – Eixos do NCBR-2 9
Figura 2.3 – Eixos do magnetômetro 10
Figura 2.4 – Posição do sensor solar na placa fotovoltaica 11
Figura 2.5 – Momento magnético gerado em uma espira 19
Figura 2.6 – Atuador magnético eixo X e Y (ISIS 2012) 20
Figura 2.7 – Atuador magnético eixo Z (ISIS 2012) 20
Figura 3.1 – Diagrama de blocos da placa NanoMind. Fonte: ISIS(2012) 24
Figura 3.2 – Modos de operação. Fonte: ISIS(2012) 25
Figura 3.3 – Diagrama transição entre os estados. Fonte: ISIS(2012) 26
Figura 4.1 – Separação dos ambientes de simulação 28
Figura 4.2 – Diagrama de blocos do simulador 29
Figura 4.3 – Simulador desenvolvido em Simulink 30
Figure 4.4 – Saídas do sistema simulado armazenadas 31
Figura 4.5 – Bloco 1 33
Figura 4.6 – Bloco 2 34
Figura 4.7 – Bloco 3 35
Figura 4.8 – Bloco 4 36
Figura 4.9 – Foto acopladores em anti paralelo 37
Figura 4.10 – Bloco 5 37
Figura 4.11 – Bloco 6 38
Figura 4.12 – Giroscópio simulado 39
Figura 4.13 – Magnetômetro simulado 39
Figura 5 1 – Órbita prevista do NBR1 41
Figura 5 2 – Erro na estimativa da atitude do satélite (eixo X) 42
Figura 5 3 – Erro na estimativa da atitude do satélite (eixo Y) 43
Figura 5 4 – Erro na estimativa da atitude do satélite (eixo Z) 43
Figura 5 5 – Velocidade angular 44
Figura 5 6 – Erro na estimativa da velocidade angular 44
Figura 5 7 – Erro na estimativa da velocidade angular 45
Figura 5 8 – Erro na estimativa da velocidade angular em regime permanente 45
Figura 5.9 – Diagrama de blocos do sistema de determinação de atitude 46

Figura 5.10 – Diagrama de blocos do sistema de controle de atitude 47
Figura 5.11 – Erro na estimativa da atitude do satélite 47
Figura 5.12 – Erro na estimativa de atitude do satélite durante a primeira órbita 48
Figura 5.13 – Erro na estimativa da atitude do satélite em regime permanente 48
Figura 5.14 – Momento magnético 49
Figura 5.15 – Velocidade angular do satélite 50
Figura 5.16 – Velocidade angular do satélite durante primeira órbita 50
Figura 5.17 – Velocidade angular do satélite em regime permanente 51
Figura 5.18 – Erro de atitude no sistema orbital 52
Figura 5.19 – Erro de atitude em regime permanente 52
Figura 5.20 – Viés do giroscópio estimado pelo filtro de Kalman 53
Figura 5.21 – Viés do giroscópio em regime permanente 53
Figura 5.22 – Atitude inercial em ângulos de Euler 54
Figura 5.23 – Atitude inercial em ângulos de Euler em regime permanente 54
Figura 5.24 – Otimização da execução pelo Windows e Matlab 55
Figura 5.25 – Tempo de execução utilizando integrador Runge Kuta
de passo variável - ODE45 56
Figura 5.26 – Teste integrador ODE23T 56
Figura 5.27 – Teste integrador ODE23T em modo alto desempenho 57
Figura 5.28 – Simulador em Simulink 58
Figura 5.29 – Relação entre a saída do Arduino e entrada Nanomind 59
Figura 5.30 – Tempo em nível alto versus comando PWM 60
Figura A.1 – Menu 67
Figura A.2 – Support Package 67
Figura A.3 – Seleção dos Pacotes 68
Figura A.4 – Contrato 68
Figura A.5 – Licensa de terceiros 69
Figura A.6 – Confirmar a instalação 69
Figura A.7 – Download 70
Figura A.8 – Instalação Completa 70
Figura A.9 – Setup 71
Figura A.10 – Atualizar o Firmware embarcado 71
Figura A.11 – Configurar Rede 72

Figura A.12 – Selecione o Drive para instalação 72
Figura A.13 – Conecte o Raspberry 73
Figura A.14 – Menu 73
Figura A.15 – Configurações de hardware 74
Figura A.16 – Configurações de IP 74
Figura A.17 – Elevação de nível de permissão 75
Figura A.18 – Exemplo de aplicação 75
Figura A.19 – Configuração do Putty 76
Figura A.20 – Inicializando o sistema 76
Figura A.21 – Erro no Diagnóstico 77
Figura A.22 – Montando uma USB no Linux 78
Figura A.23 – Configuração do Debuger 79
Figura A.24 – Erro de configuração debuger 80
Figura A.25 – Erro de localização da pasta 81
Figura A.26 – Mudando o caminho no Matlab 82

LISTA DE ABREVIAÇÕES

COTS	*commercial off-the-shelf*, ou comerciais de prateleira
EKF	*Extended Kalman Filter*, filtro estendido de Kalman
IGRF	International Geomagnetic Reference Field, referência internacional para o campo magnético terrestre
INPE	Instituto Nacional de Pesquisas Espaciais
NBR-1	NanosatC BR1
NBR-2	NanosatC BR2
MEMS	*Micro Electro-Mechanical System* ou sistemas micro eletromecânicos
PD	Proporcional-Derivativo
PID	Proporcional-Integral-Derivativo
QUEST	*Quaternion estimator*, estimador de quatérnios
SAS	(Small Astronomy Satellite)
Seasat	Satélite de sensoriamento remoto dos oceanos – NASA/ JPL
TRIAD	Método de determinação de atitude baseado em 2 vetores conhecidos

SUMÁRIO

1. INTRODUÇÃO 1
1.1 Objetivos 2
1.2 Revisão bibliográfica 3
1.3 Organização do trabalho 5
2. FUNDAMENTAÇÃO TEÓRICA 7
2.1 Equações da cinemática 7
2.2 Sensores 9
2.3 Estimação de atitude 12
2.4 Vetor de estado 16
2.5 Controle 18
3. NANOMIND 23
3.1 Dados técnicos do nanomind 23
3.2 Máquina de estado 24
3.3 Gerenciamento de modos de operação 25
4. HARDWARE IN THE LOOP 27
5. RESULTADOS Simulação 41
5.1 Resultados para o NBR-1 41
5.2 Resultados para o NBR-2 45
5.3 Resultados EGSE 55
6. CONCLUSÕES 61
7. REFERÊNCIAS BIBLIOGRÁFICAS 63
APENDICE A 67
APENDICE B 83

1. INTRODUÇÃO

O Instituto Nacional de Pesquisas Espaciais (INPE) participa de projetos de microssatélites do tipo Cubesats, sendo que até o momento houve o lançamento bem sucedido do NanosatC BR1 (NBR-1), um satélite do tipo 1 U, ou seja, um cubo de 100 mm de aresta, que tem como função mapear o campo magnético terrestre e a precipitação de partículas energéticas. Este foi o primeiro Cubesat a ser lançado pelo Brasil.

O satélite NBR-1 conta com um sistema simplificado de estabilização de atitude, que provê apenas a redução da velocidade angular e eventual imobilização do satélite em relação ao campo magnético terrestre. Para um satélite em órbita quase polar, a direção do campo magnético gira duas vezes a cada órbita em relação a um sistema de coordenadas fixado à Terra ou a um sistema inercial, o que faz com que a velocidade angular mínima seja da ordem de 0,02 rpm, aproximadamente.

O software de estimação de atitude e estabilização da velocidade angular foi fornecido pelo próprio fabricante do satélite, sem, portanto, envolvimento da equipe brasileira. O segundo satélite deste programa, denominado de NanosatC BR2 (NBR-2), é também um satélite da classe Cubesat, mas do tipo 2 U, com dimensões de 100 mm de largura e de comprimento por 200 mm de altura.

O programa computacional que efetua a determinação e controle de atitude do satélite NBR-2 será desenvolvido e qualificado no país. Os equipamentos que integram o subsistema de controle de atitude, composto por giroscópios MEMS (do inglês *Micro Electro-Mechanical System* ou sistemas micro eletromecânicos), magnetômetro com 3 eixos sensores, sensores solares do tipo fotovoltaico, bobinas magnéticas de núcleo de ar também em 3 direções ortogonais e computador de atitude serão fornecidos pela mesma empresa que fornece todos os sistemas (exceto as cargas úteis) do NBR-1. O sistema de controle de atitude do NBR-2 deverá ser mais complexo do que seu predecessor, provendo o apontamento de uma das faces do satélite para a Terra. A principal missão deste satélite será o estudo da dinâmica da atmosfera ionizada, realizada por uma das cargas úteis compostas por um fotômetro e uma sonda de Langmuir.

Como os satélites da classe Cubesat têm custos reduzidos devido à utilização de componentes COTS (*commercial off-the-shelf*, ou comerciais de prateleira, no sentido de serem componentes com requisitos de produção comerciais ou industriais), espaço reduzido para acomodar componentes devido ao seu tamanho, e capacidade de geração de energia

limitada, será necessária a utilização de técnicas de controle ótimo com foco na redução do consumo de energia, mesmo que a manobra demore várias órbitas. O NBR-2 conta com um sensor solar em cada uma das 6 faces e um magnetômetro de três eixos, além de um giroscópio do tipo MEMs. Tais sensores são considerados de baixa precisão e suscetíveis a diversas fontes de interferência ruidosa. Porém, com técnicas adequadas de estimação e filtragem, como o filtro de Kalman, consegue-se reduzir os erros e aumentar a precisão de apontamento utilizado pelo sistema de controle de atitude.

1.1 Objetivos

O objetivo deste trabalho é propor, testar e simular a lei de controle que irá reger o SDCA (Sistema de Determinação e Controle de Atitude) nos modos de estabilização de atitude (*detumbling*) e nominal a serem desenvolvidos pelo INPE e que serão embarcados no computador de bordo do NBR-2. Para isso os seguintes tópicos serão examinados:

a) Implementar modelos de simulação dos sensores (magnetômetro, sensor solar e giroscópios MEMS) e atuadores (bobinas magnéticas) no ambiente de simulação em linguagem de Matlab.

b) Implementar um modelo de determinação de atitude baseado no método TRIAD ou QUEST, para servir de apoio ao processo de estimação de atitude ou, eventualmente, para substituí-lo caso o filtro apresente problemas de convergência.

c) Implementar um estimador de estados por filtro de Kalman para estimar a atitude.

d) Propor e implementar um método de controle de atitude no modo nominal que permita um apontamento geocêntrico com precisão ao redor de 5º.

e) Integrar os modelos acima e simular a entrada e manutenção do satélite no modo nominal, incluindo passagens pela sombra da Terra, quando as informações dos sensores solares não estarão disponíveis.

Os testes e simulações do sistema de estabilização e redução de velocidades angulares e sistema de determinação de atitude que foram utilizados na simulação de atitude do satélite NBR-1, servirão de base para geração dos códigos para o modo *detumbling* e nominal do NBR-2.

Pelo fato do controle por bobinas não permitir controle efetivo em 3 eixos, devido à interação do momento magnético das bobinas com o campo magnético terrestre, pretende-se utilizar um controle proporcional e derivativo (PD) combinado a outro que considere o sistema parcialmente controlável ao longo de uma órbita, criando zonas mortas quando o campo magnético estiver no mesmo sentido do vetor torque a ser gerado, ou ainda quando a magnitude do vetor campo magnético for pequena.

Adicionalmente, é também objetivo deste trabalho estudar e apresentar as principais características do computador de bordo NanoMind do NBR-2. Este estudo visa conciliar os principais requisitos estabelecidos pela modelagem e simulação do processo de determinação, estimação e controle de atitude realizados em Matlab com as limitações existentes (memória, velocidade de processamento, recursos de bibliotecas) do NanoMind. Se houver tempo e as condições permitirem, pretende-se converter o código desenvolvido na linguagem Matlab para C, e implementá-lo, ainda que parcialmente, no NanoMind.

1.2 Revisão bibliográfica

Kalman (1960) apresenta o artigo "A new approach to linear filtering and prediction problems" com a formulação do filtro de Kalman.

Já em Farrel (1970) a determinação de atitude é feita por um filtro de Kalman onde a confiabilidade de sensores pode ser aumentada pelo uso de algoritmos sofisticados. A orientação do veículo é obtida por meio dos dados provenientes de sensores solares e magnéticos.

De acordo com Shuster e Oh (1981) os métodos de determinação de atitude em três eixos são apresentados utilizando-se de dois ou mais vetores de observação. O TRIAD é o primeiro método apresentado, é determinístico e foi utilizado em várias missões, entre elas o SAS (*Small Astronomy Satellite*) e o Seasat. O segundo método apresentado é o QUEST (QUaternion ESTimator); tem a vantagem de ser um método de otimização onde a busca pelo valor ótimo pode ser alcançada de maneira mais rápida.

Viegas e Waldmann (2010) utilizam em seu artigo um controle proporcional derivativo, com torques gerados por magneto torqueadores e rodas de reação. Um sistema de determinação de atitude e velocidade angular, baseado em filtro de Kalman em sua versão estendida, será utilizado no controle de atitude do satélite Itasat.

Bak et al. (1996) apresenta o sistema de controle de atitude do satélite Oersted em seus vários modos. A lei de controle do modo *rate detumbling* é baseada na teoria de Lyapunov onde o controle é do tipo $\dot{B}$ ou Bdot.

A integração de um magnetômetro de estado sólido ao Labview e ao microcontrolador da Texas Instruments MSC1210 é apresentado no artigo de Ferreira et al. (2008), onde a atitude é determinada por meio do método QUEST e os testes foram realizados utilizando-se uma bobina de Helmholtz.

Na dissertação de mestrado de Santos (2008) o sistema de determinação de atitude e o estimador de estado, formado por quatérnio e velocidade angular são baseados em filtro estendido de Kalman, além de outras versões baseadas em sua versão *unscented* para o projeto do satélite Itasat.

Oliveira Junior et al. (2014) publicaram um trabalho sobre estimação de atitude por quatérnio com filtro de Kalman, e apresentaram resultados da estimação de atitude e velocidade angular para o satélite NBR-1.

Em Carrara et al. (2014) são apresentados os modos de operação do satélite Conasat, satélite em desenvolvimento pelo INPE, baseados na plataforma Cubesat, com fator de forma 8U. Naquele trabalho utilizou-se um filtro de Kalman com matriz de covariância reduzida, para evitar a singularidade causada pela redundância no quatérnio.

Quadrino (2014) apresenta o desenvolvimento de um sistema de teste com hardware in the loop para o microssatélite MicroMAS (Microsized Microwave Atmospheric Satellite), um Cubesat com fator de forma de 3U, desenvolvido pelo laboratório Lincoln e Laboratório de Sistemas Espaciais do MIT (Massachussetes Institute of Tecnology), onde é possível verificar o uso de mancais aerostáticos e bobinas de Helmhotz sendo comparados à simulação de sistemas.

Na dissertação de Walter (2014) é apresentada uma metodologia orientada por modelo, onde é utilizado um SBC (Single Board Computer – Computador de placa única) para simular Hardware em tempo real. Neste trabalho foi utilizado um Raspberry Pi para simular um sistema eletromecânico, demostrando baixo esforço computacional a um custo relativo de dez por cento de um sistema comercial.

Na dissertação de Pomyen (2015) é apresentada uma forma de embarcar códigos utilizando-se do pacote de suporte disponível do Matlab Simulink. Neste trabalho é apresentada a utilização de bibliotecas e scripts para o processamento de imagens.

Ling (2015) apresenta o uso de funções do tipo Sfunction no Matlab Simulink para ter acesso às funções não existentes nas bibliotecas padrões. Neste artigo ela descreve a forma de comunicação serial entre um Arduino e um Raspberry Pi.

1.3 Organização do trabalho

O Capítulo 2 contempla a fundamentação teórica, com os conceitos e técnicas que serão aplicados neste trabalho. Entre eles citam-se as equações da cinemática e dinâmica de atitude, o processo de determinação e controle de atitude e o filtro estendido de Kalman.

O Capítulo 3 mostra as informações relativas ao computador de bordo a ser utilizado no NBR-2. Neste capítulo são demonstradas as interfaces do sistema, a máquina de estado e o gerenciamento de modos de operação.

No Capítulo 4, são apresentados os diagramas do sistema Hardware in the loop, as conexões entre os sistemas reais e os simulados.

Os resultados parciais obtidos por simulação no Matlab para os Satélites NBR-1 e NBR-2 e os provenientes de simulação com o hardware (Satélite) na malha de controle são apresentados no Capítulo 5.

As conclusões, considerações finais e os próximos passos para a conclusão da dissertação do mestrado serão apresentados no Capítulo 6.

O Capítulo 7 apresenta as referências bibliográficas deste trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão apresentados os conceitos básicos aplicados neste trabalho: as equações da cinemática e dinâmica de atitude e o filtro de Kalman em seu modo estendido.

2.1 equações da cinemática

A orientação de um corpo no espaço é definida por um conjunto de parâmetros que permitem correlacionar, num instante de tempo qualquer, um sistema inercial em relação a um sistema de coordenadas que gira com o corpo móvel. A essa correlação damos o nome de atitude (HUGHES, 1986).

Há uma série de representações de atitude em uso, tais como ângulos de Euler, matrizes de rotação (conhecida como matriz de co-senos diretores ou matriz de atitude) e os quatérnios (CARRARA, 2012).

A representação de atitude por meio de ângulos de Euler é considerada mínima, uma vez que utiliza apenas três parâmetros, ao passo que a matriz de rotação e o quatérnio são considerados não mínimos. Representações da atitude não mínimas possuem a vantagem de serem globalmente não singulares. No entanto, com frequência surgem problemas de ambiguidade causados pelo fato de se utilizar mais parâmetros do que o necessário para representar a atitude.

A representação em ângulos de Euler é comumente utilizada por permitir a visualização da atitude com relativa facilidade. Porém, devido às singularidades que podem ocorrer quando os ângulos são múltiplos de 90 graus (denominadas de *gimbal lock*), nem sempre é utilizada.

Os ângulos de Euler indicam um conjunto de 3 rotações sucessivas realizadas sobre os eixos cartesianos de forma a girar o sistema inercial até o alinhamento com o sistema fixado ao corpo móvel. Como cada ângulo estará associado a um determinado eixo, e a combinação permite que se tenha 12 transformações distintas. A Fig. 2.1 apresenta a sequência de rotações 3-1-3, ou *Z*-*X*-*Z*, na qual se utilizaram os ângulos φ, θ e ψ .

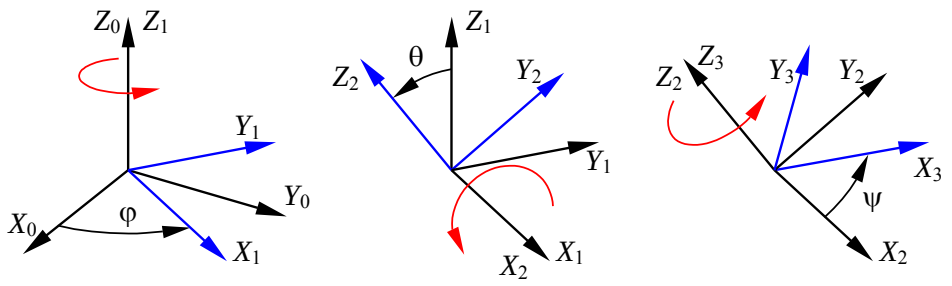

Figura 2.1 - Exemplo de rotação 3-1-3

As matrizes de rotação apresentam os componentes dos versores da base do sistema móvel em relação ao sistema inercial nas suas linhas. Na matriz, apenas três componentes são independentes, o que significa uma desvantagem em relação às outras formas de representação de atitude.

Os quatérnios são uma extensão do conceito de números complexos em três dimensões e possuem quatro componentes: um vetor de três dimensões e um escalar. O módulo do quatérnio é obrigatoriamente unitário. Eles são representados por $\mathbf{q} = q_1\,\mathbf{i} + q_2\,\mathbf{j} + q_3\,\mathbf{k} + q_4$, com **i**, **j** e **k** os versores da base, que representam números hipercomplexos que obedecem a $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i}\,\mathbf{j}\,\mathbf{k} = -1$.

Devido ao fato de que o quatérnio não tem interpretação física, é necessário o uso de métodos de conversão para, por exemplo, ângulos de Euler. Considerando a notação de Wertz (1978) do quatérnio:

$$\mathbf{Q} = (\boldsymbol{\varepsilon} \quad \eta) \tag{2.1}$$

onde $\boldsymbol{\varepsilon}$ é a parte vetorial, neste trabalho apresentado como $\boldsymbol{\varepsilon} = (q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k})$, e η a parte escalar do quatérnio, representada por $\eta = (q_4)$. Utilizando as fórmulas a seguir obtêm-se o ângulo e eixo de Euler.

$$\boldsymbol{\varepsilon} = \mathbf{a}\left(\operatorname{sen}\frac{\theta}{2}\right) \tag{2.2}$$

$$\eta = \frac{\cos\theta}{2} \tag{2.3}$$

onde $\mathbf{a}$ é eixo de Euler e θ é o ângulo de Euler. Pode-se ainda converter em matriz de rotação utilizando a seguinte relação:

$$\mathbf{C}_{ba} = (\eta^2 - \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon})\mathbf{1} + 2\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T - 2\eta\boldsymbol{\varepsilon}^{\times} \tag{2.4}$$

onde o índice representa a rotação do sistema do corpo girante, neste caso o satélite para o sistema inercial, o índice é a representação da transposição da matriz e o índice é a representação da matriz antissimétrica, de maneira explicita:

$$\mathbf{C}_{\mathrm{ba}} = \begin{bmatrix} \eta^2 + \varepsilon_1^2 - \varepsilon_2^2 - \varepsilon_3^2 & 2(\varepsilon_1\varepsilon_2 + \eta\varepsilon_3) & 2(\varepsilon_1\varepsilon_3 - \eta\varepsilon_2) \\ 2(\varepsilon_1\varepsilon_2 - \eta\varepsilon_3) & \eta^2 - \varepsilon_1^2 + \varepsilon_2^2 - \varepsilon_3^2 & 2(\varepsilon_2\varepsilon_3 + \eta\varepsilon_1) \\ 2(\varepsilon_1\varepsilon_3 + \eta\varepsilon_2) & 2(\varepsilon_2\varepsilon_3 - \eta\varepsilon_1) & \eta^2 - \varepsilon_1^2 - \varepsilon_2^2 + \varepsilon_3^2 \end{bmatrix} \tag{2.5}$$

A equação diferencial da cinemática de atitude em quatérnio pode ser escrita como:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{q} \tag{2.6}$$

onde a matriz $\boldsymbol{\Omega} = \begin{bmatrix} -\boldsymbol{\omega}^{\mathrm{x}} & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{T} & 0 \end{bmatrix}$ é formada pela matriz antissimétrica do vetor velocidade angular no sistema do corpo do satélite. Substituindo os termos obtém-se:

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}\mathbf{q} \tag{2.7}$$

2.2 Sensores

Para entender como para controlar a atitude é importante compreender o processo de sensoriamento do ambiente espacial, qualquer filtragem feita nas medidas coletadas ou atrasos nessas medições serão realimentados para a lei de controle, gerando erros. Este processo inicia-se com os sensores que o satélite usa para conhecer a sua atitude. Estes vetores podem então ser combinados para produzir um quatérnio. O NBR-2 tem sensores solares, magnetômetro de 3 eixos e um giroscópio do tipo MEMs. Todos os eixos serão referidos no sistema do satélite conforme Fig. 2.2.

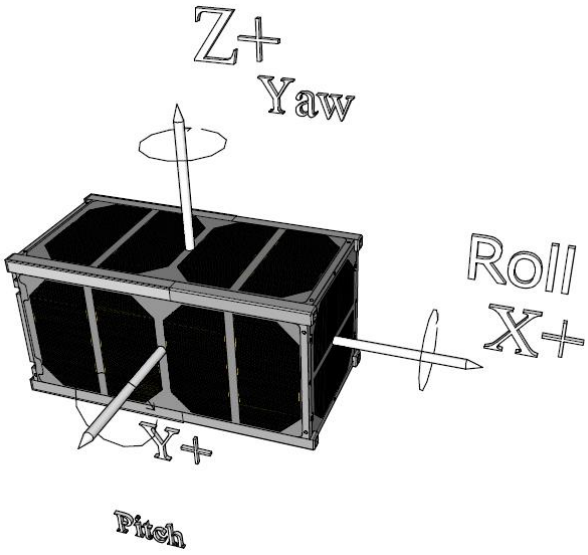

Figura 2.2 – Eixos do NCBR-2

2.2.1 Magnetometro

O NBR2 dispõe de um magnetômetro incorporado à placa de controle Nanomind, com características baseadas no sensor HMC5843 da Honeywell. É triaxial com range de ± 4 G (± 4 µT) e uma resolução de 10µG (0,001µT).

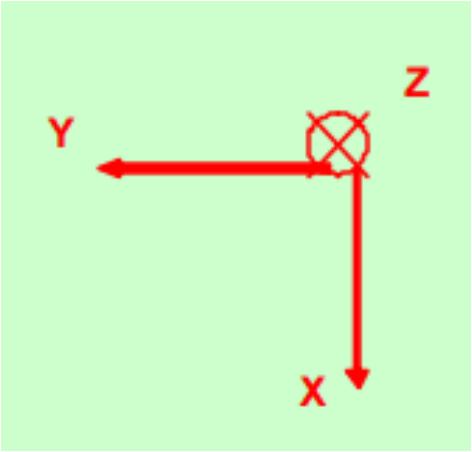

Figura 2.3 - Eixos do magnetômetro

Como o magnetômetro estará montado em uma placa a ser empilhada no satélite a direção dos eixos passam a ser: Z do magnetômetro tem a direção de X do Satélite, X do magnetômetro tem a direção de Z- do satélite e Y do magnetômetro e do satélite são coincidentes.

O magnetômetro será utilizado para medir o campo magnético terrestre, e então compará-lo ao modelo de referência internacional para campo geomagnético (IGRF – "International Geomagnetic Reference Field").

2.2.2 Sensor solar

Um dos dados necessários para estimar a atitude é o vetor direção do sol. O NBR-2 usa foto diodos sensíveis ao espectro de luz visível, fabricados pela Silonex, sob o código SLSD-61N8. Estes estão montados nos painéis solares, conforme podemos observar na Fig. 2.4.

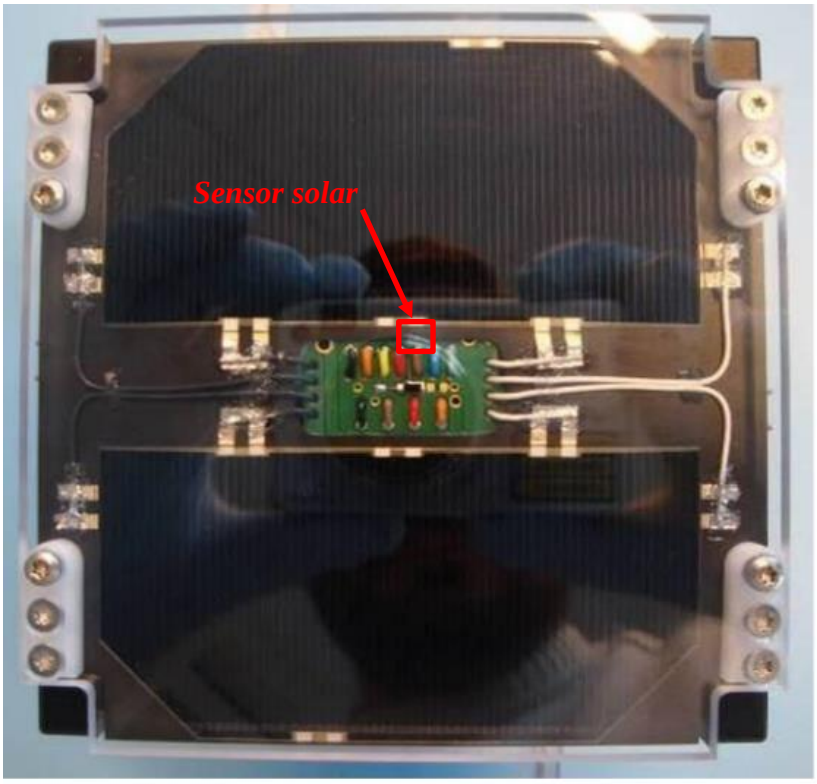

Figura 2.4 – Posição do sensor solar na placa fotovoltaica

A gama de comprimentos de onda detectáveis é 400-1100 nanómetros, com um ângulo de meio cone 60º. A tensão gerada pelos fotodiodos poderá ser interpolada para um ângulo, a partir de um gráfico presente na folha de dados [Silonex SLCD-61N8].

Os sensores são descritos como grosseiro porque seu campo de visão é de 60 graus, o que significa que a informação que pode fornecer oferece uma incerteza maior do que um sensor com um campo de visão mais estreito. Existem sensores montados em cada uma das faces (±X, ±Y e ± Z), o que é ideal devido ao sensor estar voltado ao eixo que está

observando. A estimativa da localização do sol pode ser encontrada tomando em consideração a configuração dos seis sensores em conta.

Através de um algoritmo que envolve a combinação de suas seis medições de tensão, os sensores de sol podem dar uma direção aproximada para o sol, quando o satélite não está em eclipse. Este algoritmo leva a tensão de cada sensor, dado bruto sem qualquer tratamento ou filtragem, mapeando os eixos do sistema. Ele compara as duas tensões provenientes do mesmo eixo (positivo e negativo), e mantém o maior dos dois, enquanto a tensão é acima do nível pré-estabelecido de ruído. O próximo passo é interpolar a tensão a um ângulo θ, definido como o ângulo entre o sensor e a fonte de luz, a partir de uma curva dada na folha de dados do fotodiodo com base na sua resposta. O cosseno deste ângulo é então usado como o componente do vector sol.

$$S_{x,y,z} = Cos(\theta_{x,y,z}) \tag{2.8}$$

2.2.3 Giroscópio

A velocidade angular pode ser obtida pela leitura dos giroscópios, com erros advindos de viés ou ruídos inerentes ao sensor, ou ainda calculada pelo processo de estimação de atitude, reduzindo os erros.

Considerando que no satélite NBR-2 ter-se-á um giroscópio de 3 eixos, deve-se levar em consideração suas medidas na estimativa. O modelo matemático do giroscópio, segundo Wertz (1978), é dado por:

$$\boldsymbol{\omega}_i^M = (1 + \mathbf{k}_i)\,\boldsymbol{\omega}_i + \mathbf{b}_i + \mathbf{n}_i \tag{2.9}$$

onde ^{M} significa que o valor foi medido na saída do giroscópio, $\mathbf{k}_i$ é um pequeno fator de correção de escala, que neste caso será desconsiderado para fins de simplificação, $\mathbf{b}_i$ é a deriva ou viés do giroscópio e $\mathbf{n}_i$ é um ruído branco nas medidas do giroscópio, que representa as incertezas não modeladas. Devido às simplificações, pode-se reescrever a equação como:

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}_i + \mathbf{b}_i + \mathbf{n}_i \tag{2.10}$$

2.3 Estimação de atitude

A estimação da atitude pelo filtro de Kalman pode ser realizada considerando-se as medidas efetuadas pelos sensores ou então com base na atitude calculada a partir destas mesmas medidas. O primeiro método é mais complexo e pode apresentar instabilidades na

convergência do filtro em virtude das linearizações necessárias do modelo que relacionam as medidas com os parâmetros estimados. O segundo método é mais simples, mas requer um processamento realizado a bordo para determinar a atitude, isto é, para computar a atitude com base nas medidas dos sensores. Vários métodos existem para isso, e variam em função dos tipos de sensores utilizados.

2.3.1 TRIAD

Os métodos utilizados para a determinação da atitude de um veículo espacial podem ser basicamente de dois tipos: métodos determinísticos e métodos ótimos. Nos métodos determinísticos é necessário um número mínimo de dois vetores de medida para se determinar a atitude corretamente e qualquer dado adicional será obrigatoriamente descartado. Já os métodos ótimos, de maneira diferente dos métodos determinísticos, utilizam todos os dados disponíveis para determinar a atitude que é obtida a partir da otimização de uma função de custo. O método TRIAD é o mais conhecido da primeira abordagem. A segunda abordagem foi criada para resolver o problema de Wahba e tem como principais exemplos o método Q e o QUEST. No presente trabalho será utilizado o método TRIAD.

O algoritmo TRIAD foi desenvolvido por Harold Black na década de 1960, quando ele estava trabalhando para a Marinha dos EUA para desenvolver uma solução para resolver a atitude de uma nave espacial. Embora não seja a solução ótima para a estimativa de atitude, é simples e não consome grande quantidade de recursos computacionais. O algoritmo leva em consideração dois vetores de referência não paralelos em um sistema de referência inercial, e compara-os com dois vetores medidos no sistema do satélite. Ele assume que uma medição do vetor no sistema do satélite é mais preciso do que o outro, e se baseia em que a medição para dois eixos de atitude conhecida. O usuário pode selecionar qualquer medição como a mais precisa para usar como vetor "mais confiável". O algoritmo Triad utiliza o segundo vetor de medição para deduzir informação de rotação em torno do terceiro eixo. A partir desta informação, uma matriz de cossenos diretores é desenvolvida, e a partir desse, a atitude pode ser calculada.

Como os satélites do tipo Cubesat usam geralmente sensores solares e magnéticos, que fornecem dois vetores no sistema fixado ao satélite, pode-se utilizar o método TRIAD na

determinação da atitude. Este método monta dois sistemas de coordenadas usando, respectivamente, os vetores obtidos das medidas dos sensores no sistema fixado ao corpo, e as direções destes vetores computados a partir de modelos matemáticos que fornecem a direção do Sol e do campo geomagnético no local onde se encontra o satélite. A atitude decorre da relação que envolve o produto entre os cossenos diretores dos dois sistemas de coordenadas. Salienta-se que este sistema é contaminado por ruídos provindos das medidas dos sensores. Para reduzir este erro a níveis aceitáveis na missão, será utilizado um filtro de Kalman, que será apresentado no próximo capítulo.

2.3.2 Equações de Euler

Utilizam-se as equações diferenciais do movimento de um corpo rígido nas simulações, conhecidas como equações de Euler

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\mathbf{N}_c - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}) \tag{2.11}$$

onde **I** é a matriz de inércia, considerada constante, e $\mathbf{N}_c$ resulta dos torques de perturbações e de controle aplicados ao satélite (WERTZ,1978).

Devido ao NBR- 2 ser lançado como carga secundária, não há como prever qual será a fração de órbita que o satélite passará na sombra da Terra ou qual fração poderá visualizar o Sol. Portanto, é difícil prever quantas informações estarão disponíveis para efetuar a determinação completa da atitude em uma órbita. Deve-se acrescentar ainda que, devido à precessão orbital causada pelo achatamento terrestre, é provável que a relação entre a fração iluminada e a fração na sombra altere-se ao longo do ano. Na sombra da Terra, a atitude não poderá ser determinada pelo algoritmo TRIAD devido à falta do vetor de medidas da direção do Sol. Ainda assim, a atitude poderá ser calculada, propagando-se a equação da cinemática com as medidas efetuadas pelos giroscópios.

2.3.3 Filtro de Kalman

O filtro de Kalman é um instrumento fundamental para analisar e resolver uma ampla classe de problemas de estimação de parâmetros ou de variáveis de estado (MCGEE e SCHMIDT, 1985). Apesar do formalismo matemático do filtro de Kalman, ser já muito conhecido, uma quantidade substancial de conhecimento e experiência são necessários para desenvolver um algoritmo operacional. Um modelo matemático, tanto da dinâmica do estado

como das incertezas, é inerentemente incorporada na estrutura do filtro de Kalman. Descrever um modelo matemático adequado sobre o qual o filtro irá basear-se é o ponto crucial do problema de projeto. Além disso, o projeto deve atender às especificações de desempenho, explorando alternativas para alcançar um filtro simples, adicionando ou excluindo a complexidade do modelo de maneira a atender os resultados adequados. É de fundamental importância no processo de projeto a avaliação de desempenho real de forma a validar os erros, evitando que ocorra divergência do filtro. Precisão numérica e problemas de estabilidade podem danificar substancialmente o desempenho, especialmente quando o filtro é implementado num computador de bordo (MAYBECK, 1979).

Tendo em vista que o filtro de Kalman em seu modelo mais simples foi criado para estimar parâmetros de sistemas lineares, foram desenvolvidas outras formas de estimação para sistemas não lineares, sendo uma delas a versão estendida do filtro de Kalman (EKF) que será apresentada neste trabalho.

Considere-se um sistema dinâmico não-linear cuja equação de estado é dada por:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[x(t), t] + \mathbf{G}(t)\boldsymbol{\eta}(t) \tag{2.12}$$

onde $\mathbf{x}$ é o vetor de estado de dimensão n; $\mathbf{f}$ é uma função vetorial dos elementos do estado; $\mathbf{G}$ é a matriz $(n \times r)$com elementos contínuos no tempo; $\boldsymbol{\eta}$ é um vetor de dimensão r que representa o ruído dinâmico no estado.

Segundo Kuga (2005) o modelo do estado a ser estimado é, para o caso discreto variando em instantes discretos, dado por:

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_{k+1,k}\mathbf{x}_k + \boldsymbol{\Gamma}_k\mathbf{w}_k \tag{2.13}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.14}$$

onde $\mathbf{x}_{k+1}$ é o vetor de n estados a ser estimado, $\boldsymbol{\Phi}_{k+1,k}$ é a matriz de transição de estados entre os instantes k e k+1, $\mathbf{x}_k$ é o vetor de estado atual, $\boldsymbol{\Gamma}_k$ é a matriz de adição de ruídos dinâmicos, $\mathbf{w}_k$ representa os erros de natureza aleatória ou impossíveis de serem modelados, caracterizado como ruído branco com média nula e variância $\mathbf{Q}_k$, $\mathbf{y}_k$ é o vetor de m observações coletadas, $\mathbf{H}_k$ é a matriz que relaciona as medidas ao estado e $\mathbf{v}_k$ representa os erros de modelagem ou

de natureza aleatória relacionados às medidas, caracterizado por um ruído branco com média nula e variância $\mathbf{R}_k$.

O filtro de Kalman possui as fases de propagação, também chamada de predição, e atualização, também chamada de correção (KUGA, 2005). A etapa de propagação é utilizada para propagar o estado e a covariância por meio do modelo dinâmico, e é definida por:

$$\bar{\mathbf{x}}_k = \mathbf{\Phi}_{k,k-1}\hat{\mathbf{x}}_{k-1} \tag{2.15}$$

$$\bar{\mathbf{P}}_k = \mathbf{\Phi}_{k,k-1}\hat{\mathbf{P}}_{k-1}\mathbf{\Phi}_{k,k-1}^T + \mathbf{\Gamma}_k\mathbf{Q}_k\mathbf{\Gamma}_k^T \tag{2.16}$$

onde $\bar{\mathbf{x}}_k$ representa o estado propagado para o instante k, $\mathbf{\Phi}_{k,k-1}$ é a matriz de transição do estado anterior para o atual, $\hat{\mathbf{x}}_k$ é o estado estimado no instante anterior, $\bar{\mathbf{P}}_k$ representa a matriz de covariância propagada, e $\hat{\mathbf{P}}_{k-1}$ é a matriz de covariância estimada no instante anterior. A etapa de atualização é utilizada para corrigir o estado e a covariância no instante k considerando a medida por meio do modelo de observações, computando o ganho de Kalman e a nova estimativa do estado.

$$\mathbf{K}_k = \bar{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{2.17}$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\bar{\mathbf{P}}_k \tag{2.18}$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\bar{\mathbf{x}}_k) \tag{2.19}$$

onde $\mathbf{K}_k$ é o ganho de Kalman e $\mathbf{I}$ é a matriz identidade.

Especificamente, o EKF é uma formulação que usa modelos dinâmicos não-lineares de plantas contínuas com sensores amostrados de maneira discreta. Este filtro será implementado em um computador digital. A natureza digital do microprocessador presta-se a funções discretas. Assim, a dinâmica do satélite e cinemática são aproximadas com equações não-lineares discretas e os sensores reais são sensores analógicos amostrados de maneira discreta.

2.4 Vetor de estado

O vetor de estado é então definido como composto pelo quatérnio advindo da determinação de atitude e do viés do giroscópio:

$$\mathbf{x} \equiv \begin{bmatrix} \mathbf{q} \\ \mathbf{b} \end{bmatrix} \tag{2.20}$$

cuja derivada constitui as equações da dinâmica do filtro:

$$\dot{\mathbf{x}} \equiv \begin{bmatrix} \frac{1}{2}\mathbf{\Omega q} \\ \mathbf{0} \end{bmatrix} + \mathbf{w} \tag{2.21}$$

onde $\mathbf{\Omega} = \begin{bmatrix} -\boldsymbol{\omega}^{\times} & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{T} & 0 \end{bmatrix}$, $\mathbf{w}$ é o ruído branco que representa as somatórias das incertezas ligadas ao modelo, com covariância $\mathbf{Q}$. De outra forma, pode-se escrever $\mathbf{w} = N(0, \mathbf{Q})$.

Supondo que a velocidade angular do satélite possa ser considerada como constante durante o intervalo entre duas amostras consecutivas do giroscópio, a equação da cinemática em quatérnios apresenta neste caso uma solução analítica, que pode ser escrita como:

$$\bar{\mathbf{q}}_{\mathbf{k+1}} = \mathbf{\Phi}_{\mathbf{q|k+1,k}} \hat{\mathbf{q}}_{\mathbf{k}} \tag{2.22}$$

Para propagar o estado inicial para o próximo estado deve-se conhecer a matriz de transição, é obtida de Lefferts et al. (1982):

$$\mathbf{\Phi}_{\mathbf{q|k+1,k}} = \cos\left(\frac{|\boldsymbol{\omega}|\Delta t}{2}\right)\mathbf{I}_{4\times4} + \frac{1}{|\boldsymbol{\omega}|} sen\left(\frac{|\boldsymbol{\omega}|\Delta t}{2}\right)\mathbf{\Omega}(\boldsymbol{\omega}) \tag{2.23}$$

e, como o viés é constante, a sua propagação resulta em:

$$\bar{\mathbf{b}}_{\mathbf{k+1}} = \hat{\mathbf{b}}_{\mathbf{k}} \tag{2.24}$$

e sua matriz de transição:

$$\mathbf{\Phi}_{b|\mathbf{k+1,k}} = \mathbf{I}_{3\times3} \tag{2.25}$$

A representação das equações de estado por meio da matriz de transição facilita a etapa de propagação do filtro, pois evita a utilização de métodos de integração numéricos. A matriz de transição completa dos estados é dada por:

$$\mathbf{\Phi}_{\mathbf{k+1,k}} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{q|k+1,k}} & \mathbf{I}_{4\times3} \\ \mathbf{I}_{3\times3} & \mathbf{\Phi}_{\mathbf{b|k+1,k}} \end{bmatrix} \tag{2.26}$$

O modelo de medidas é apresentado em tempo discreto, dado por:

$$\mathbf{y}_{\mathbf{k}} = \mathbf{H}_{\mathbf{k}}\mathbf{x}_{\mathbf{k}} + \mathbf{v}_{\mathbf{k}} \tag{2.27}$$

onde $\mathbf{y_k}$ é o vetor de medidas, $\mathbf{H_k}$ é a matriz de correlação entre a medida e o estado, definida neste trabalho por $\mathbf{H_k} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$ e $\mathbf{v_k}$ representa o ruído branco associado às incertezas das medidas de covariância, ou seja, $\mathbf{v_k} = N(0, \mathbf{R_k})$.

Para efetuar a estimação por filtro de Kalman, a escolha dos quatérnios para representação de atitude possui a desvantagem de apresentar singularidade na matriz de covariância, devido à dependência entre os elementos do quatérnio, que possui módulo unitário. Para contornar esse problema, nesse trabalho foi escolhida a representação por matriz de covariância reduzida, sugerida originalmente no trabalho de Lefferts et al. (1982), e reproduzidas por Garcia et al. (2011) e Carrara et al. (2014).

2.5 Controle

Controle de atitude para satélites se refere às técnicas que são empregadas para manter a atitude do satélite dentro de uma faixa de valores pré-definidos. Para o controle da atitude de um veículo, são necessários sensores para definir a atitude atual do veículo, já apresentados no capítulo 2.2, atuadores para aplicar o torque necessário para reorientar o veículo para a atitude desejada, e algoritmos para comandar os atuadores baseados em medidas dos sensores da atitude corrente e especificação da atitude desejada.

2.5.1 Atuadores

No NBR-2 serão utilizadas bobinas magnéticas, ou magnetotorqueadores, a fim de prover o controle por meio da interação com o campo magnético terrestre.

Bobinas magnéticas são amplamente utilizadas em sistemas de controle de atitude (SIDI, 1997). A sua aplicação varia de amortecimento dos distúrbios causados por torques externos (como, por exemplo, gradiente de gravidade, pressão de radiação solar, etc.), a dessaturação de dispositivos que armazenam momento angular (como rodas de reação, por exemplo) e o controle de satélites pequenos, como é o caso do Itasat e do NBR-1.

São, normalmente, empregados nas missões espaciais devido à sua construção simples - sem partes móveis, o que garante uma longa vida útil – seu baixo peso em comparação aos outros atuadores e por utilizar energia renovável.

Sua construção básica consiste em um núcleo magnético e uma bobina. Quando a bobina é energizada, é gerado um momento magnético dado por:

$$\mathbf{M} = NIA\mathbf{n} \tag{2.28}$$

onde N é o número de espiras, I é a corrente aplicada, A é a área da espira e n é um vetor normal ao plano da espira, com direção positiva pela regra da mão direita com relação ao sentido da corrente elétrica. Uma representação do momento gerado pode ser vista na Fig 2.5:

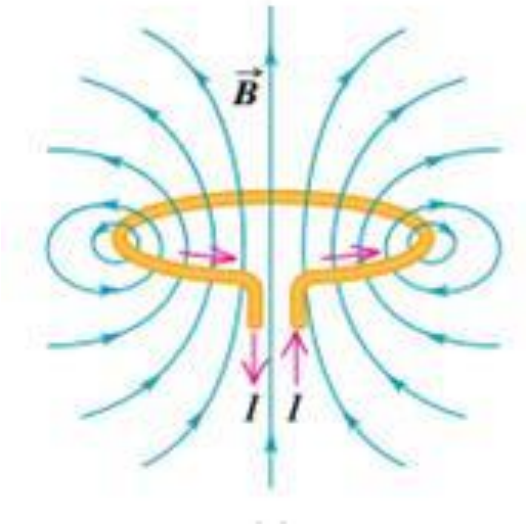

Figura 2.5 - Momento magnético gerado em uma espira (Apostila física no cotidiano USP-2005)

Um parâmetro importante durante a especificação das bobinas é o material do núcleo. Um núcleo ferromagnético possui alta permeabilidade, o que leva a um baixo consumo de energia e possui baixo peso. Entretanto, saturam rapidamente. É ainda bastante empregado em satélites bobinas com um núcleo de "ar" (WERTZ, 1978).

No caso do NBR-2 serão empregados 3 atuadores magnéticos para o controle de atitude. Duas bobinas com núcleo ferromagnético, eixos X e Y, e uma com núcleo de ar, eixo Z. Apesar do material do núcleo ser diferente, as três bobinas apresentam torque igual, quando submetidos à mesma intensidade de corrente. O acionamento das bobinas pode ser efetuado por aplicação de uma tensão constante, ou sinal modulado em frequência. Este acionamento é feito pela placa de controle, Nanomind no caso do NBR2.

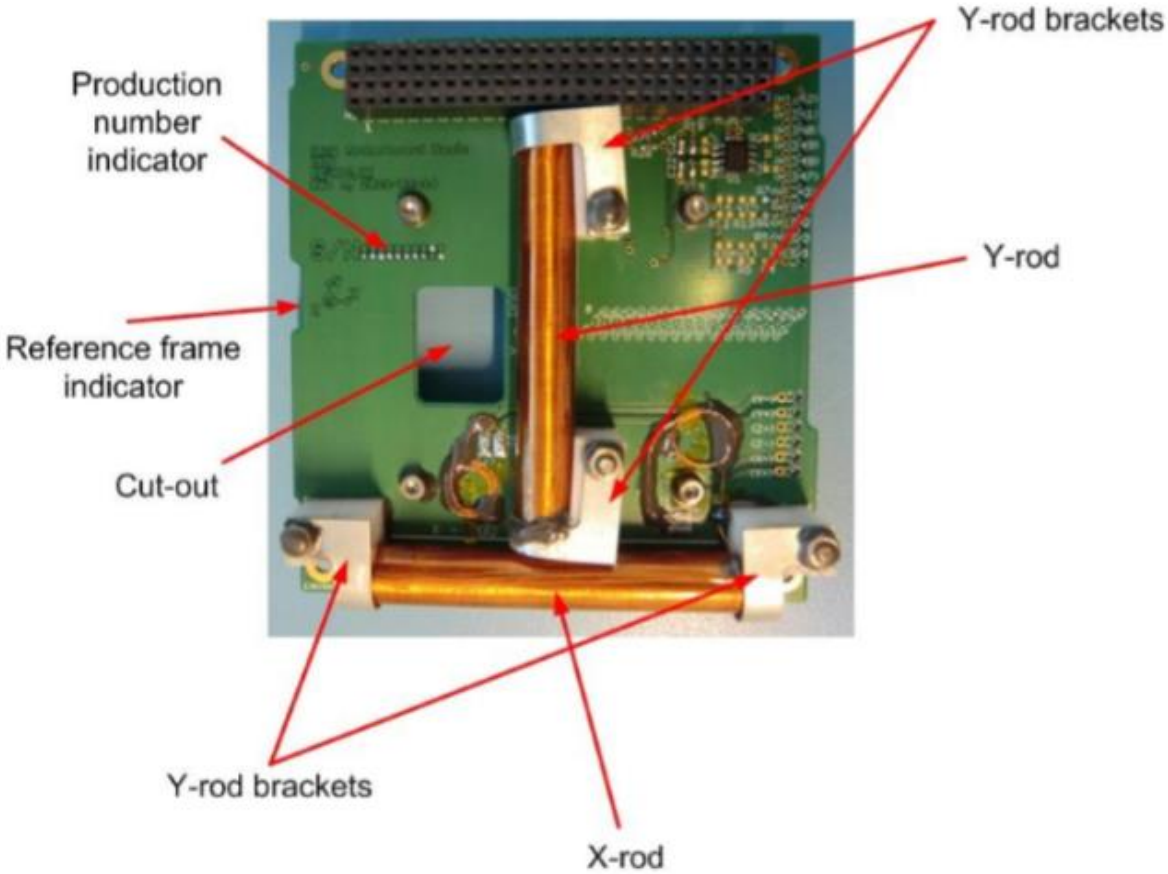

Figura 2.6 – Atuador magnético eixo X e Y (ISIS 2012)

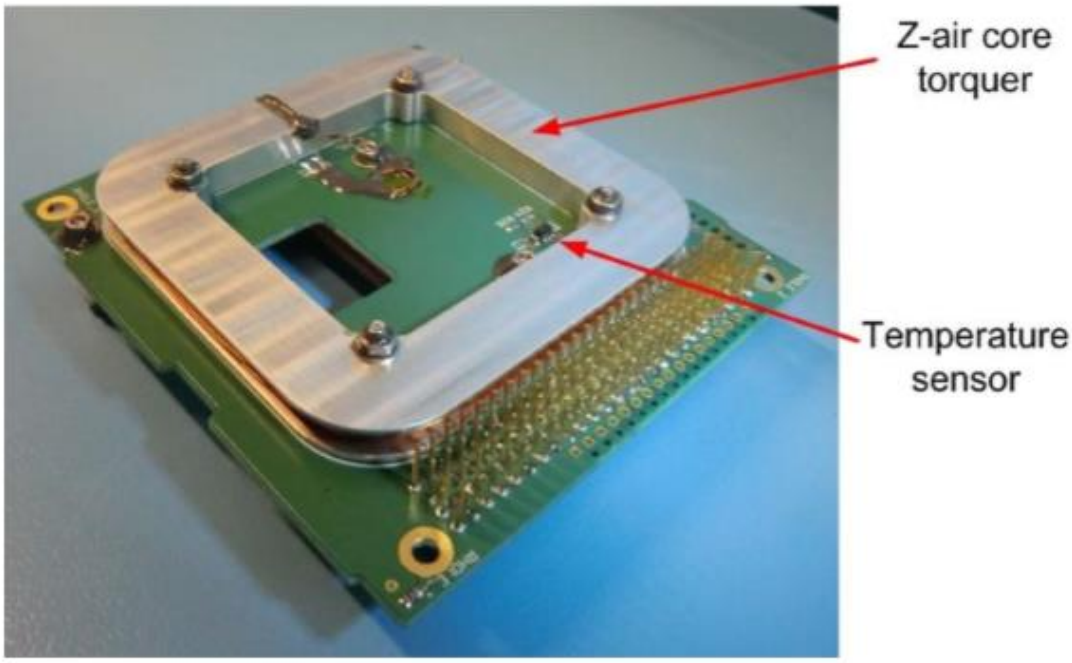

Figura 2.7 - Atuador magnético eixo Z (ISIS 2012)

O torque gerado pelas bobinas pode ser calculado como:

$$\mathbf{T}_{magnético} = \mathbf{M} \times \mathbf{B} \tag{2.29}$$

onde **M** é o momento magnético das bobinas e **B** é o campo magnético terrestre. Observando a equação anterior, nota-se que a direção do torque gerado pelas bobinas é perpendicular à direção do campo magnético terrestre, ou seja, não fornece torque em uma direção arbitrária para o sistema de controle.

Outro fator negativo é que a força do campo magnético terrestre é inversamente proporcional ao cubo da distância:

$$\mathrm{B} = \frac{\boldsymbol{m}}{\mathrm{R}^3} \tag{2.30}$$

onde m é a constante magnética terrestre, o que limita a aplicação das bobinas aos satélites com órbitas próximas à Terra. O uso das bobinas também depende das medidas do campo magnético terrestre, sendo em geral necessário o uso de magnetômetro, conforme apresentado na seção 2.2.

2.5.2 Algoritimo B-DOT

Reduzir a velocidade angular do satélite é uma das funções do sistema de controle. Observações de como o campo magnético varia em função do tempo é uma forma simples de mitigar o capotamento (tumbling), chamada de controle B-Dot. Este consiste em aplicar um torque em direção oposta a encontrada na variação do campo magnético, conforme apresentada na equação 2.31

$$\dot{\mathbf{B}}(t) = -\boldsymbol{\omega}(t) \times \mathbf{B}(t) \tag{2.31}$$

$$\mathbf{m}(t) = -k\dot{\mathbf{B}}(t) \tag{2.32}$$

$$\boldsymbol{\tau}_{\mathrm{C}}(t) = \mathbf{m}(t) \times \mathbf{B}(t) \tag{2.33}$$

onde $\dot{\mathbf{B}}(t)$ é a variação do vetor campo magnético ($\mathbf{B}(t)$), $\boldsymbol{\omega}(t)$ é a rotação angular do satélite, $\mathbf{m}(t)$ é o torque a ser aplicado pelas bobinas magnéticas para reduzir a velocidade angular, k é um ganho proporcional e $\boldsymbol{\tau}_{\mathrm{C}}(t)$ é o torque de controle

A lei de controle B-Dot é comumente usada em satélites para reduzir a velocidade em satélites que orbitam a Terra, permitindo o controle em todas as direções exceto a direção alinhada ao vetor campo magnético.

3. NANOMIND

O computador de bordo a ser utilizado no microssatélite NBR-2 é o NanoMind, modelo A712C, com processador Atmel ARM 7, produzido pela empresa Gom Aerospace. Este computador de bordo utiliza o sistema operacional FreeRTOS, junto com bibliotecas em C (drivers) para gerenciar os modos de operação bem como executar as tarefas necessárias ao funcionamento do NBR-2.

3.1 Dados técnicos do Nanomind

O computador de bordo do Cubesat NBR-2 é o NanoMind A712C fabricado pela empresa GOM Space. Como características importantes para o projeto, ele tem um processador ARM7 de 32-bits com arquitetura RISC (do inglês *Reduced Instruction Set Computer* ou Computador com um Conjunto Reduzido de Instruções) com suporte a um sistema operacional de tempo real (neste projeto são utilizados o FreeRTOS e o eCos), com velocidade de *clock* de 8 a 40 MHz, 2 Mb de memória RAM, 4 Mb de memória para o código e 4 Mb para armazenamento dos dados. Nesta placa da GOM o barramento utilizado é semelhante ao do PC104, também adotado em Cubesats, com protocolo I2C de comunicação entre as placas, o que evita a utilização de pinos do processador para comunicação e aquisição de dados. Ele conta ainda com um sensor magneto resistivo (magnetômetro) de 3 eixos, que será utilizado para determinação do campo magnético terrestre para fins de determinação de atitude e controle do satélite, 6 entradas analógicas para os sensores solares e 3 saídas PWM para o controle das bobinas, além de conexão com os giroscópios por meio de uma interface SPI. Para a alimentação da placa é utilizado uma fonte única de 3,3 V DC, fornecida no barramento do satélite. A Fig. 3.1 mostra o diagrama de blocos da placa NanoMind.

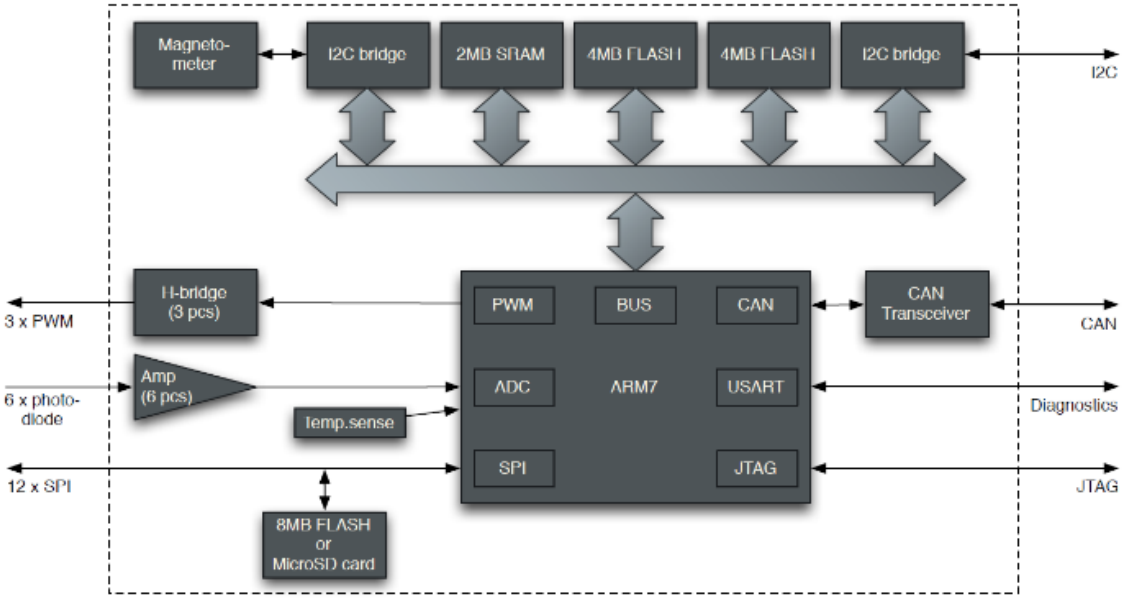

Figura 3.1 - Diagrama de blocos da placa NanoMind. Fonte: ISIS(2012)

A placa NanoMind tem dois barramentos I2C com suporte de transferência bidirecional de dados entre mestre e escravo, podendo ser utilizado como multimestre, onde a arbitragem entre transmissões simultâneas evita a perda ou corrupção dos dados na comunicação serial. A sincronização do *clock* permite equipamentos com diferentes taxas de transmissão comunicarem-se de maneira serial. A velocidade máxima de transferência é de 400 kbits/s, com um buffer de 68bytes na transmissão e outro na recepção.

O software padrão é escrito para uso com o micro *kernel* FreeRTOS. Este inclui o controle de tarefas, semáforos, tratamento da interrupção e código de início em *assembler*.

3.2 Máquina de estado

Tendo em vista que a máquina de estados implementada prevê os estados *idle*, *deployment*, *safe, nominal* e *detumbling* para o satélite NBR-2, neste trabalho iremos ater ao estado nominal com as tarefas de apontamento do eixo Nadir e determinação de atitude. Todos os estados e tarefas disponíveis na máquina de estado podem ser vistos na Fig. 3.2.

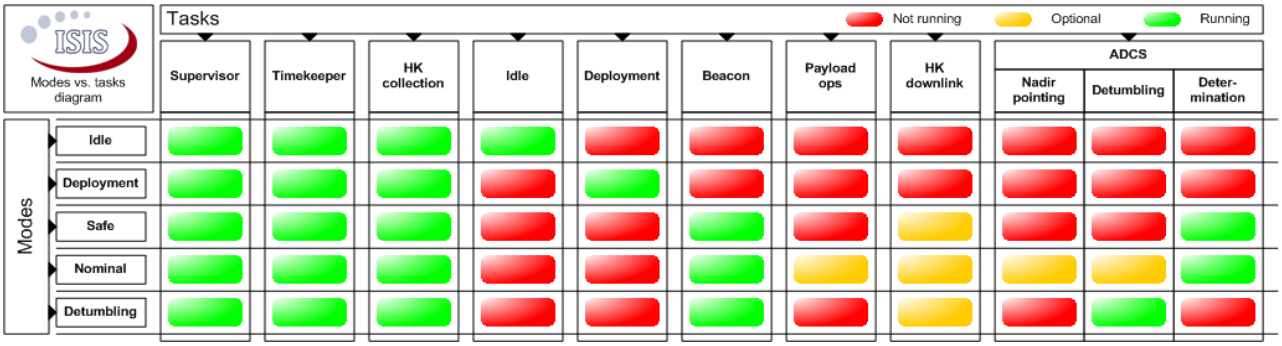

Figura 3.2 - Modos de operação. Fonte: ISIS(2012)

3.3 Gerenciamento de modos de operação

O diagrama da Fig. 3.3 demonstra como ocorrerá a transição entre os estados. Ao ligar a alimentação da placa ou após o reset pelo telecomando, o satélite irá para o estado *idle*. Após as inicializações do sistema operacional, o sistema vai para o estado *deployment*, onde ocorre a abertura das antenas durante o primeiro ciclo, mantendo a mesma estendida durante a vida útil do satélite, se a abertura for realizada com sucesso, ele passa ao estado *safe*. No estado *safe,* ele aguarda um telecomando de seleção do modo de operação. Caso seja enviado telecomando para o mesmo passar ao modo *detumbling,* ele irá executar os algoritmos de estabilização, reduzindo a velocidade angular ao patamar pré-estabelecido. Pode-se enviar o telecomando para o satélite entrar em modo nominal, onde o satélite irá executar a determinação da atitude e controle para apontamento do eixo Nadir. O satélite poderá entrar em modo *safe* caso algum gatilho de segurança seja atuado, estando em qualquer estado.

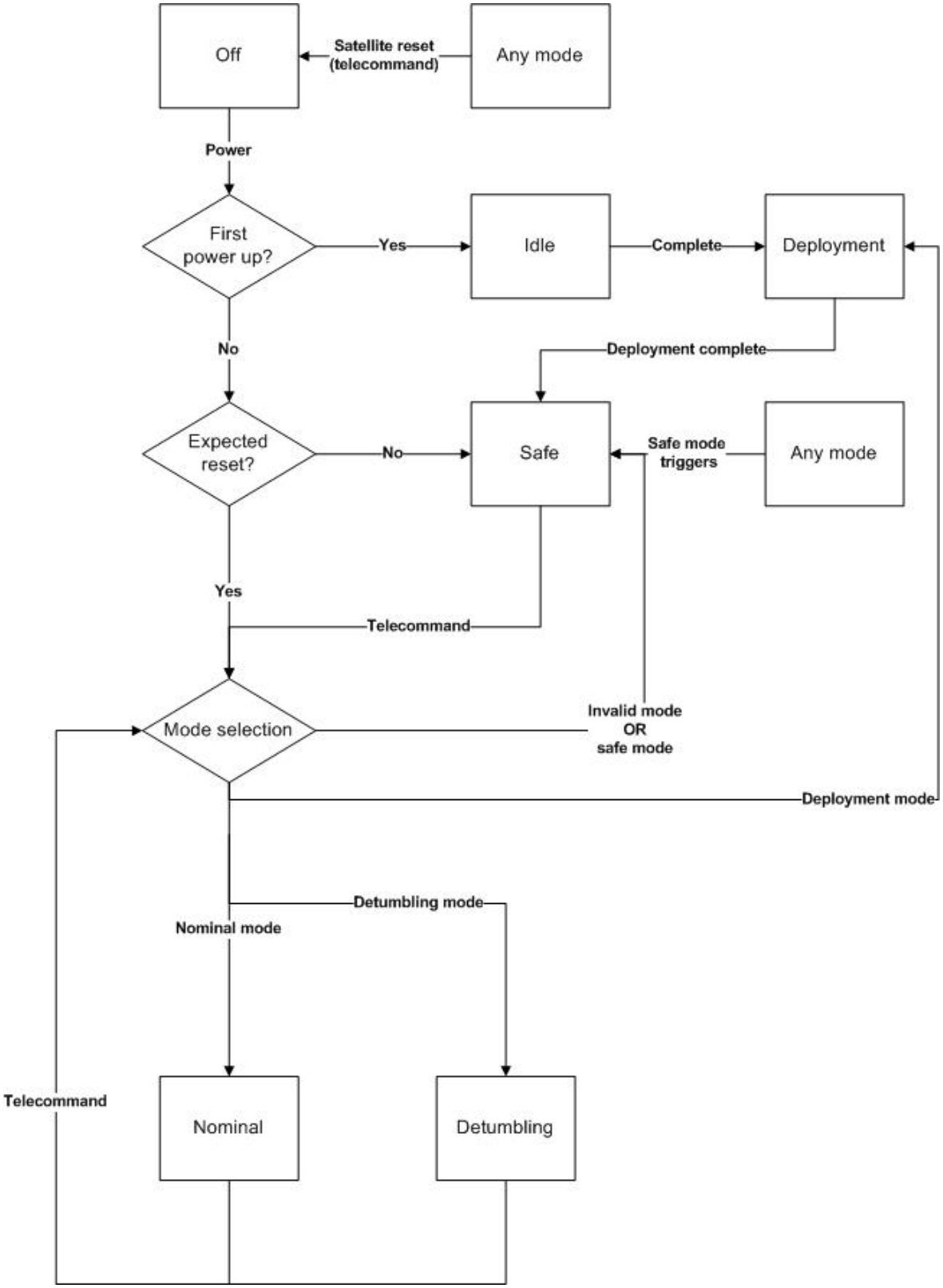

Figura 3.3 - Diagrama transição entre os estados.

Fonte: ISIS (2012)

4. HARDWARE IN THE LOOP

Para sistema onde a confiabilidade deve ser alta, ou seja, o sistema não é tolerante a falhas, deve-se utilizar uma abordagem que minimize o risco. Conhecida na engenharia de sistema como modelo em V, adota um sistema de testes onde as partes são testadas de maneira previamente estabelecida de forma a validar todas as possíveis condições do sistema. Neste trabalho foram utilizados testes do modelo, comparando a outros modelos, de forma a validar o sistema modelado (satélite e ambiente onde o mesmo irá operar). Durante os estudos foi avaliada a possibilidade de avançar no sentido de efetuar os testes do hardware de controle e sistema ambiental, sendo necessário efetuar modificações no software do simulador de maneira a separar as funções de controle e determinação de atitude, a serem embarcadas no satélite, as funções ambientais e modelagem dos sensores de forma a obter um simulador de ambiente, onde seria possível conectar o satélite e simular o ambiente , podendo ser testadas condições iniciais do sistema em órbitas previamente definidas, testando os componentes que seriam embarcados no modelo em voo e o comportamento dos módulos de software embarcados perante alterações possíveis e falhas do sistema.

No diagrama da Fig. 4.1 é apresentada esta separação em dois simuladores, o primeiro chamado de simulador de ambiente, onde todas as informações referentes ao sistema espacial são calculadas gerando a informação para os sensores do satélite, o segundo simulador fica responsável por simular as condições do computador de bordo, podendo ser substituído por um modelo real, assim que o modelo atinja um grau de maturidade e as características assim o permitirem.

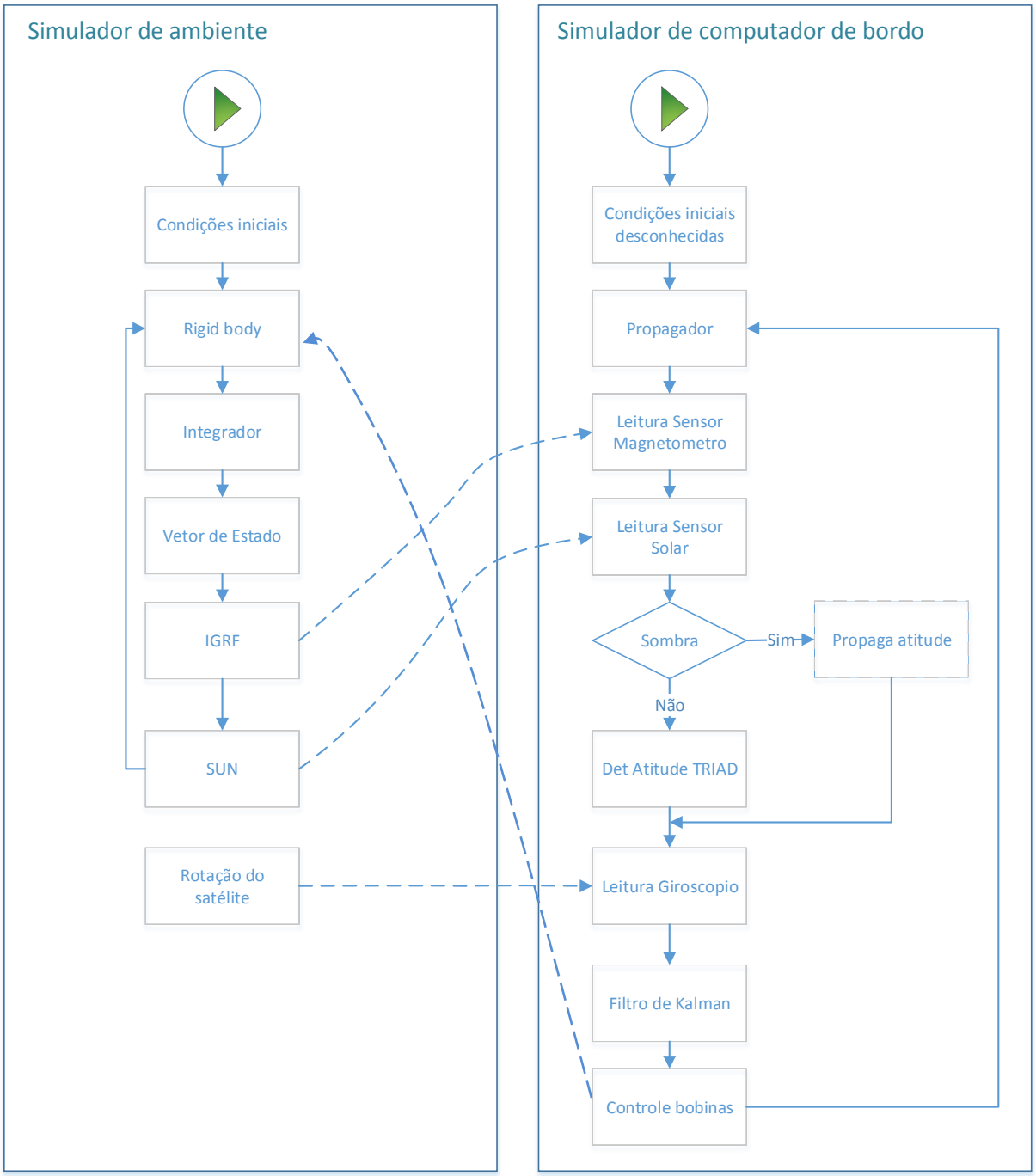

Figura 4.1- Separação dos ambientes de simulação

Utilizando dos dados disponíveis a respeito do computador de bordo, foi gerado um diagrama de interligação conforme Fig. 4.2, a fim de verificar se os requisitos seriam atendidos. Ao comparar as entradas e saídas do Nanomind às saídas e entradas do hardware utilizado para gerar/ receber as informações, pode-se verificar que existe uma perda de informação/ resolução quando comparamos a saída do Arduino à entrada de Foto Diodos do Nanomind 8 bits versus 12 bits, porém verificou-se ainda que se compararmos ao modelo do sensor “grosseiro” não teremos problemas.

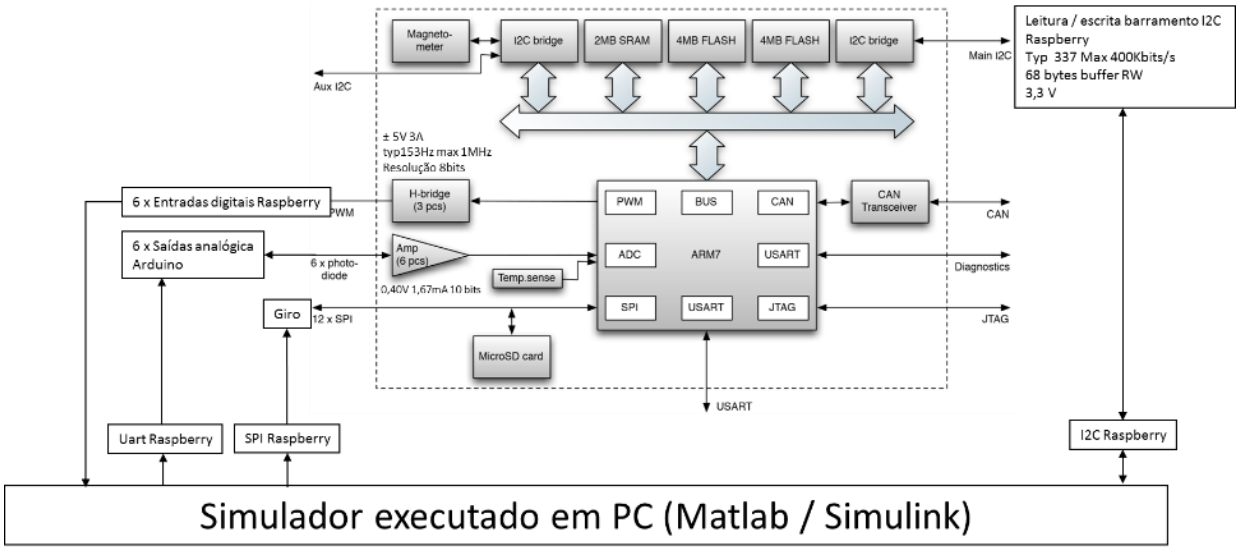

Figura 4.2- Diagrama de blocos do simulador

Na Fig.4.3, abaixo, ilustra os blocos utilizados no simulador, em Simulink, os blocos serão comentados a seguir:

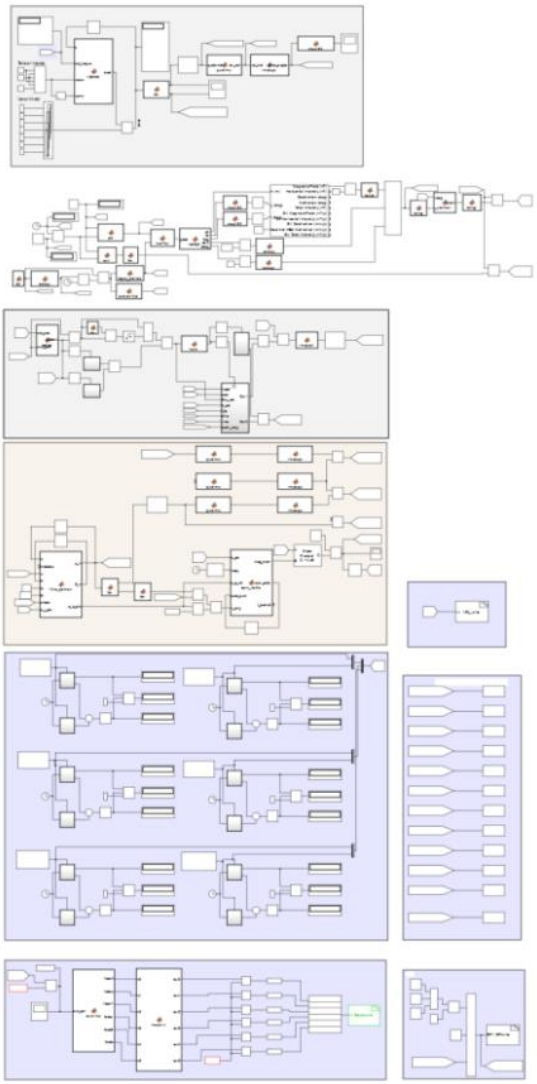

Figura 4.3 - Simulador desenvolvido em Simulink

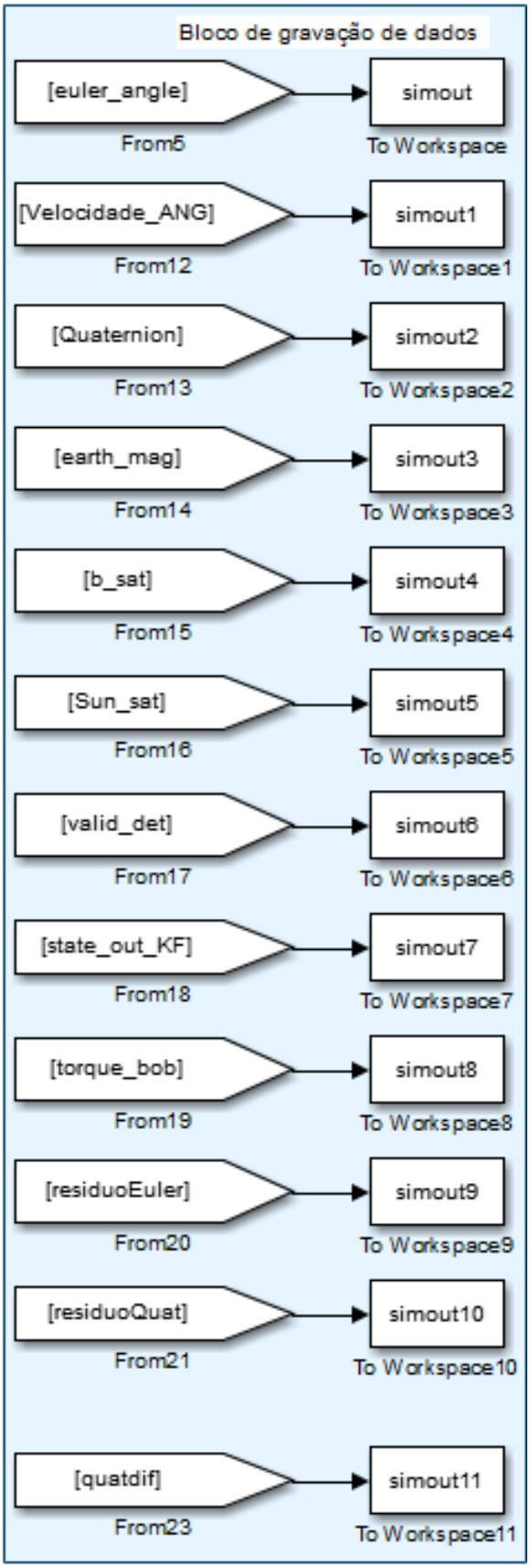

Figura 4.4 – Saídas do sistema simulado armazenadas

Neste simulador são armazenadas, as informações de atitude (em ângulos de Euler e quatérnio), velocidade angular, campo magnético da Terra no sistema geocêntrico e no sistema do satélite, o vetor Sol no sistema do satélite, flag de determinação de atitude completa (quando o satélite está na sombra, a atitude é propagada), o vetor de estado na saída do filtro de Kalman (Atitude e bias do giroscópio), o torque magnético gerado pela bobina e resíduos calculados em ângulos de Euler e quatérnios e a diferença entre o quatérnios simulados com e sem perturbações (modelo ideal e o modelo real).

No bloco 1, Fig. 4.5, são apresentadas as entradas de características do satélite, e condições iniciais, bem como o modelo de corpo rígido a ser integrado, gerando as informações de atitude do satélite em quatérnios que são convertidas para matriz de rotação e ângulos de Euler.

No bloco 2, Fig. 4.6, são calculadas a órbita, bem como sus variações e por meio destas informações são calculados os valores dos sensores solar e magnetômetro no sistema do satélite.

O bloco 3, Fig. 4.7, apresenta a determinação de atitude, realizada pela Triad.

O filtro estendido de Kalman, a função de controle e o cálculo dos erros são apresentados na Fig 4.8 – Bloco 4.

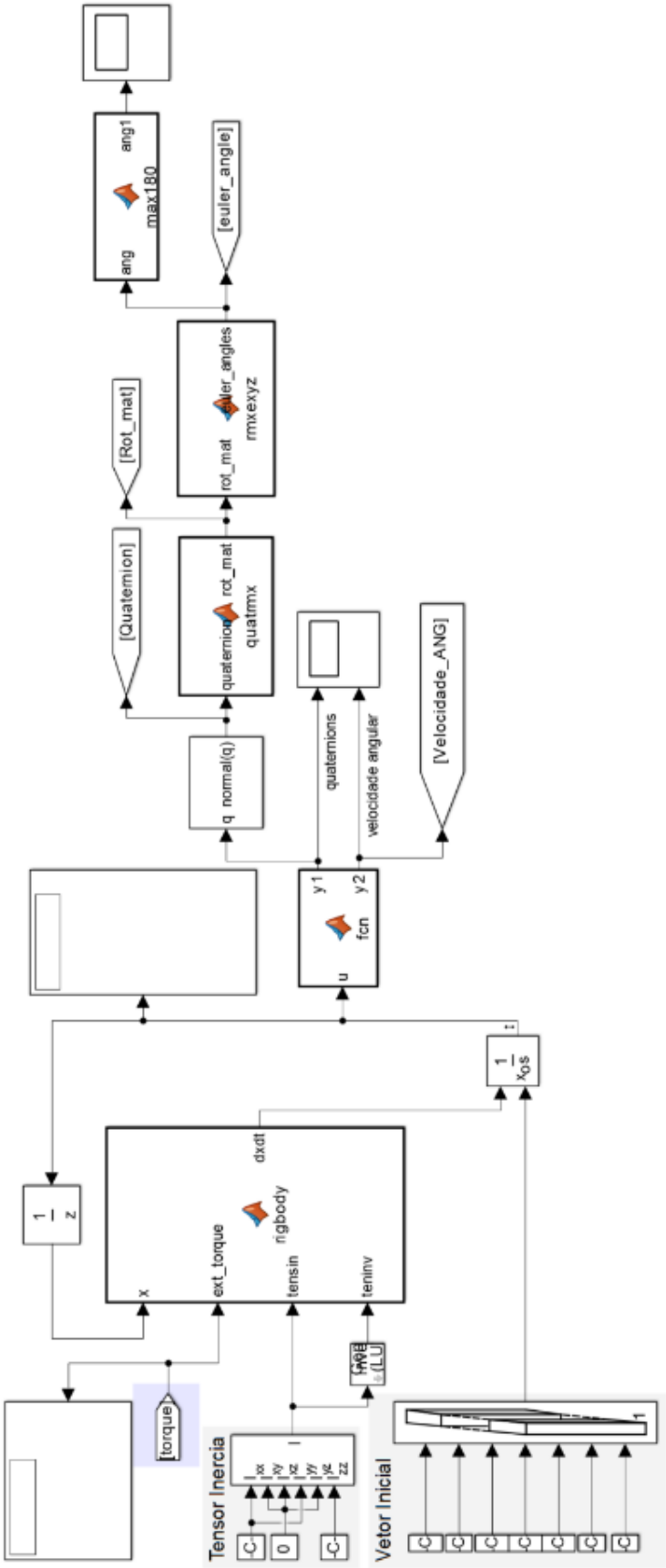

Figura 4.5 - Bloco 1

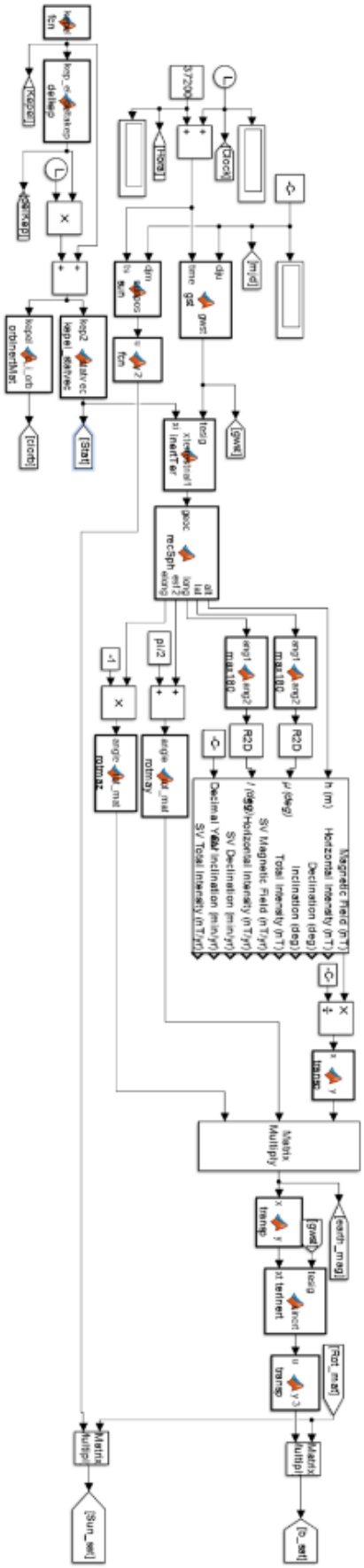

Figura 4.6 – Bloco 2

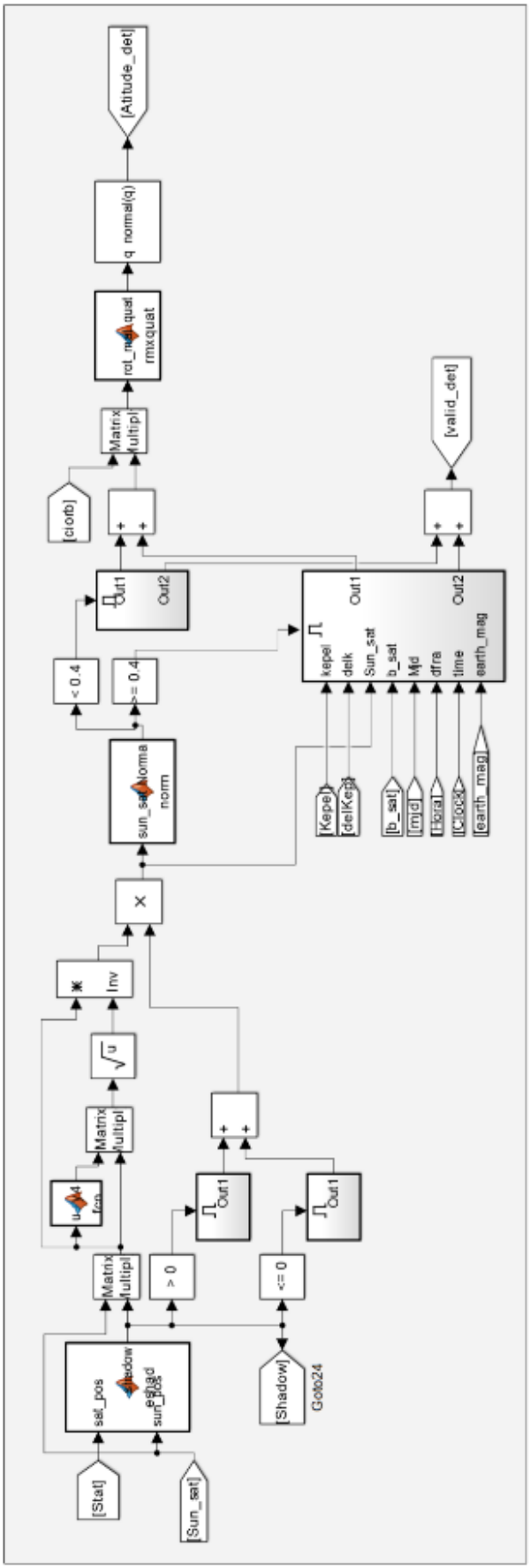

Figura 4.7 – Bloco 3

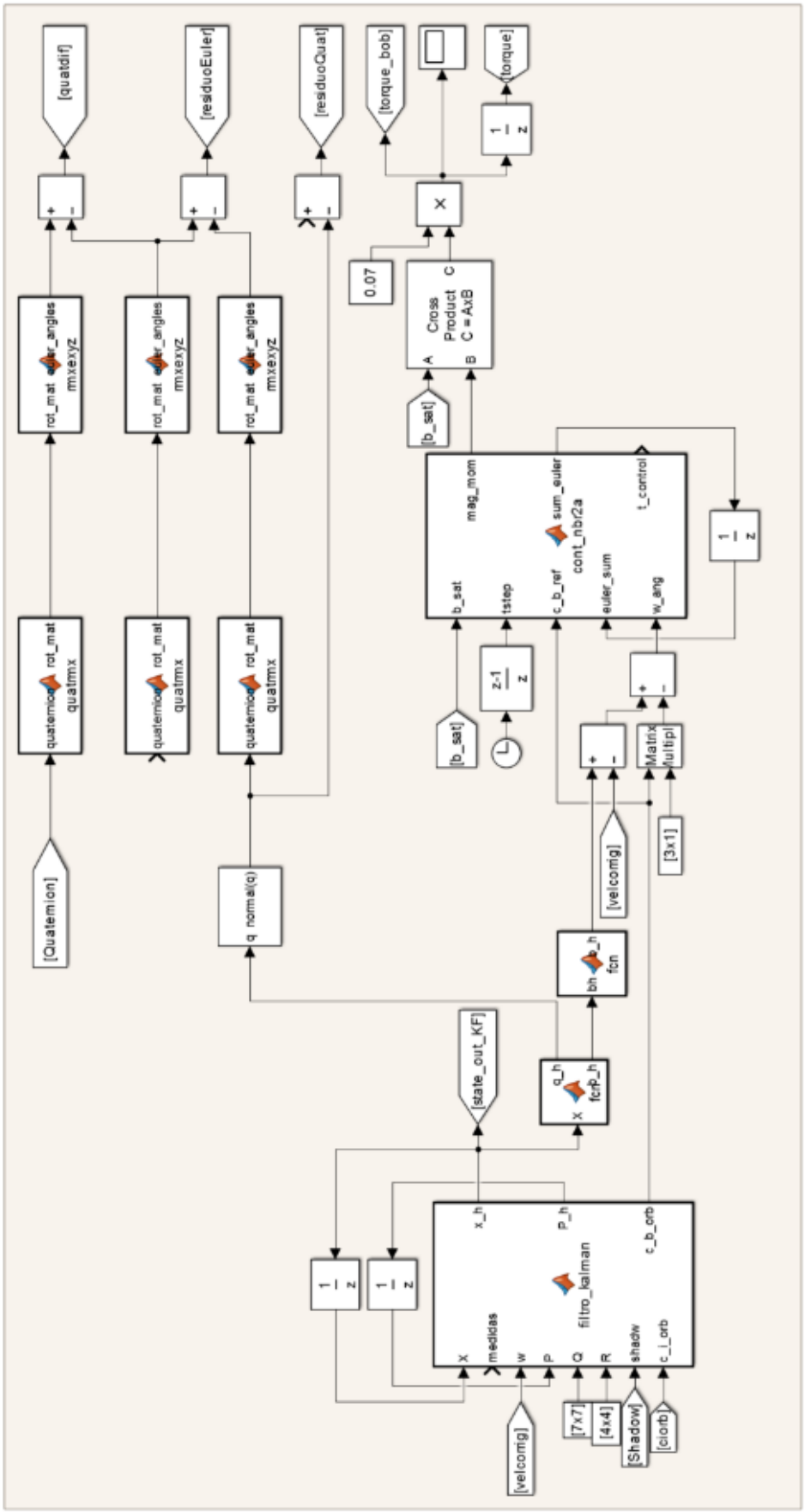

Figura 4.8 – Bloco 4

Na Fig. 4.10 é apresentada a aquisição de dados de tempos do acionamento das bobinas magneto torqueadoras. Com os tempos cronometrados é possível calcular o período de acionamento e duty cycle, que podem ser convertidos em torque realimentado o ciclo de simulação. A aquisição de dados é feita pelo circuito composto de dois fotoacopladores ligados em antiparalelo de forma a obtermos dois sinais de onda quadrada, conforme Fig. 4.9, o primeiro referente ao ciclo onde a bobina está com seus terminais a e b ligados ao positivo e negativo e o outro sinal referente ao oposto. Para cada bobina foi montado um circuito de forma a obtermos os dados dos três eixos.

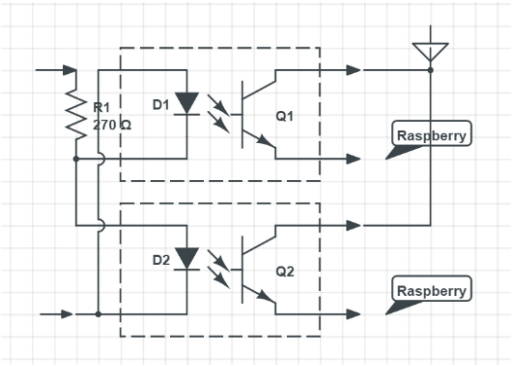

Figura 4.9- Foto acopladores em antiparalelo

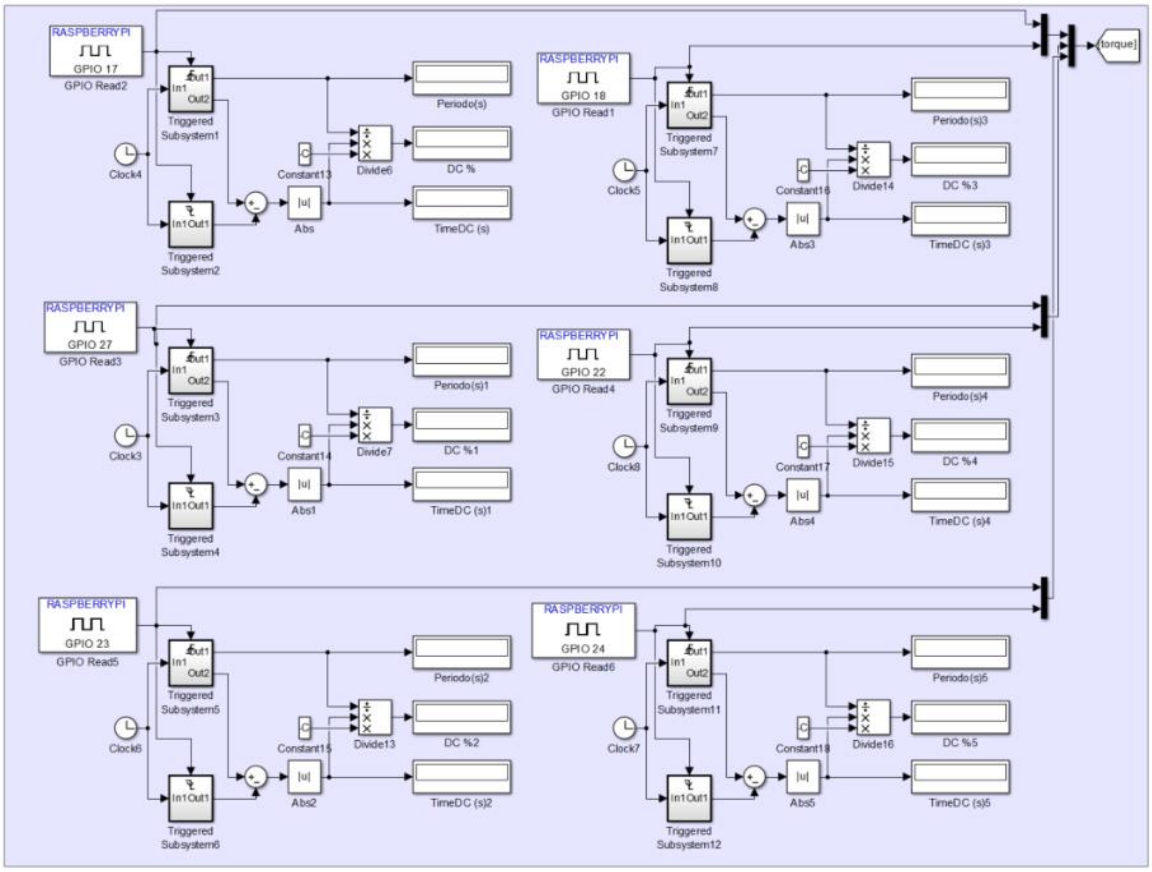

Figura 4.10 – Bloco 5

O sensor solar é excitado através da saída do Arduino, que recebe os dados da simulação do Raspberry via serial. O sinal da saída passa por um filtro passa baixa e um divisor resistivo de forma a adequarmos o sinal a entrada do Nanomind.

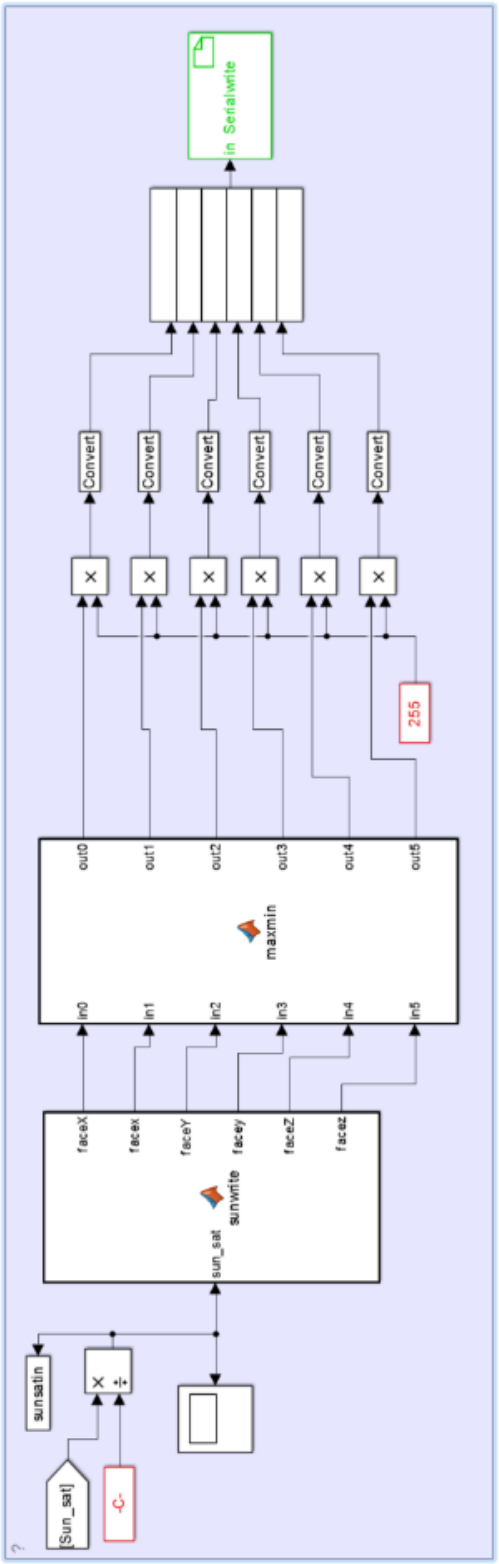

Figura 4.11 – Bloco 6

Na Fig. 4.11 é apresentada a conexão do Raspberry ao Nanomind, que envia os dados do giroscópio simulado utilizando-se da porta SPI

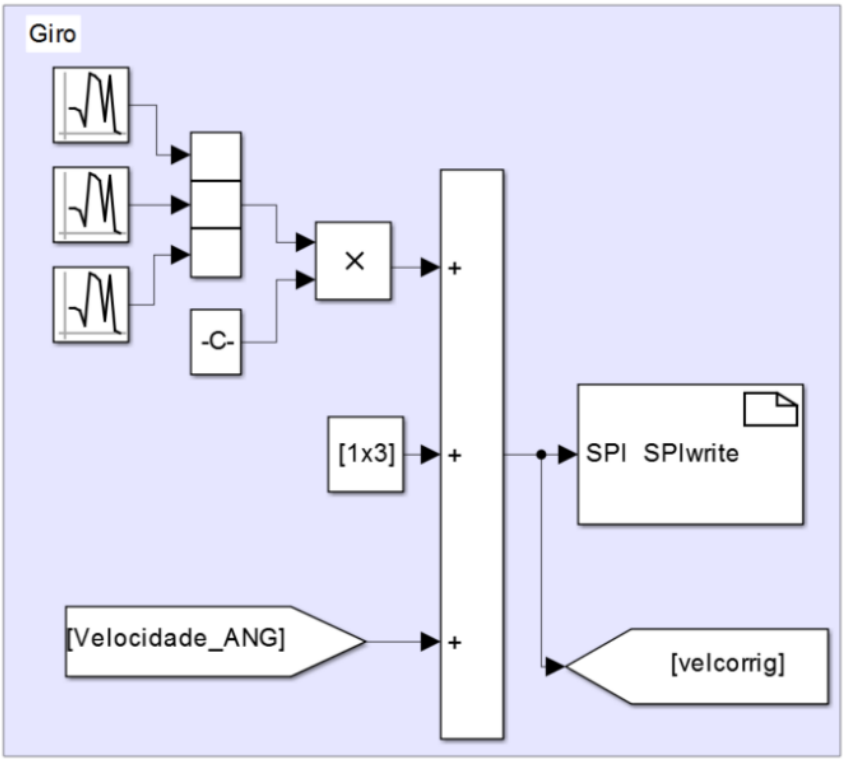

Figura 4.12 - Giroscópio simulado

O Magnetômetro simulado é escrito na porta I2C do Raspberry que será conectado ao EGSE da Isis no Barramento principal. A Fig. 4.12 apresenta a conexão de dados no simulador.

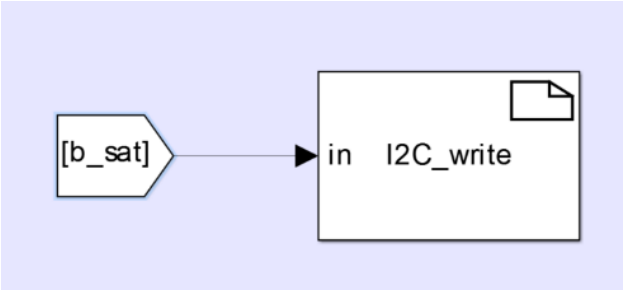

Figura 4.13 - Magnetômetro simulado

5. RESULTADOS SIMULAÇÃO

Nesta seção serão apresentados os resultados obtidos para a aplicação do filtro de Kalman em sistema de controle e determinação de atitude para os satélites NBR-1 e NBR2.

5.1 Resultados para o NBR-1

Os resultados aqui apresentados baseiam-se em um simulador criado no ambiente de programação Matlab com as ferramentas de simulação de atitude do PROPAT (CARRARA, 2009).

Utilizaram-se os elementos keplerianos (a, e, i, Ω, ω, M) para calcular o posicionamento do satélite. A Fig. 5.1 mostra que a órbita do satélite NBR-1 deverá ser polar, definida pelos seguintes elementos keplerianos: o semi-eixo maior da órbita a = 7008155 m, a excentricidade e = 0.01, a inclinação i = 98º, a ascensão do nodo ascendente Ω = 0, o argumento do perigeu ω= 0 e a anomalia média M = 0.

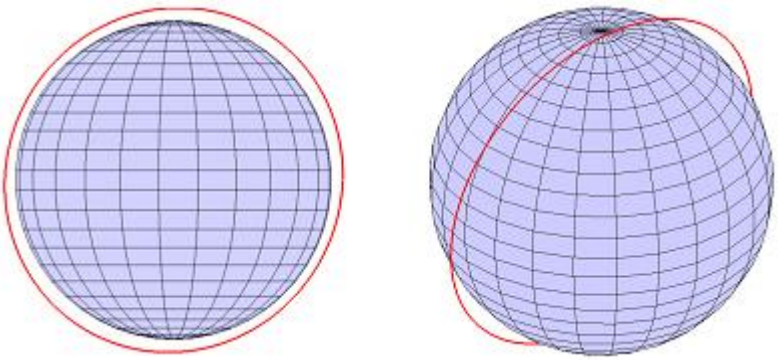

Figura 5.1 - Órbita prevista do NBR1

O modelo do magnetômetro implementado leva em consideração a posição do satélite relativo à Terra calculada pelo propagador orbital incluído no PROPAT. O modelo computacional IGRF11 (IGRF, 2014) do campo magnético terrestre foi utilizado para simular as leituras do magnetômetro. Com a posição advinda do simulador de órbitas convertido para o sistema terrestre em coordenadas esféricas, tem-se a colatitude, longitude e altura do satélite. Desta forma, pode-se calcular o campo magnético por meio da função `igrf_field`. Após o cálculo, o campo magnético deve ser convertido para o sistema de referência do satélite, adicionado de um viés e de um ruído gaussiano para obter características próximas às encontradas no sistema real. Como características de viés ou bias, foram adicionado 2,5% no eixo x, -1,5% no eixo y e adicionado 1% no eixo z, de 200 mili-Gauss. As características do ruído gaussiano utilizados foram média zero e desvio padrão de 5% de 200 mili-Gauss.

Para simular o sensor solar é utilizada a posição do Sol no sistema geocêntrico inercial calculada pela função `sun` do pacote de ferramentas PROPAT, e posteriormente convertida para o sistema do satélite e adicionada a um ruído branco, com média zero e desvio padrão de 5% de 0,05 radianos, para simular as características reais do sensor. Neste caso, devido à redundância de sensores solares no sistema considera-se que o mesmo não apresenta bias.

A dinâmica e cinemática do satélite são simuladas por meio da propagação dos estados por um integrador Runge Kutta de quarta ordem com passo de um segundo, sendo o tempo total de simulação igual ao tempo de uma órbita do satélite, que é igual a 6.000 segundos. Como o lançamento deste satélite será feito como carga auxiliar, não é possível garantir que o satélite terá a atitude e velocidade angular inicial estipuladas, e desta forma, para fins de simulação, estes parâmetros foram considerados nulos, sendo a pior condição de inicialização do filtro.

Os erros associados à estimação da atitude do satélite em ângulos de Euler são apresentados nas Figuras 5.2, 5.3 e 5.4. Tipicamente ficam abaixo de 1º em cada um dos eixos. A incerteza na estimativa do filtro fica dentro do limite dos erros reais; este fato é importante e mostra que a incerteza na estimativa realizada pelo filtro é uma representação condizente com o erro real do sensor.

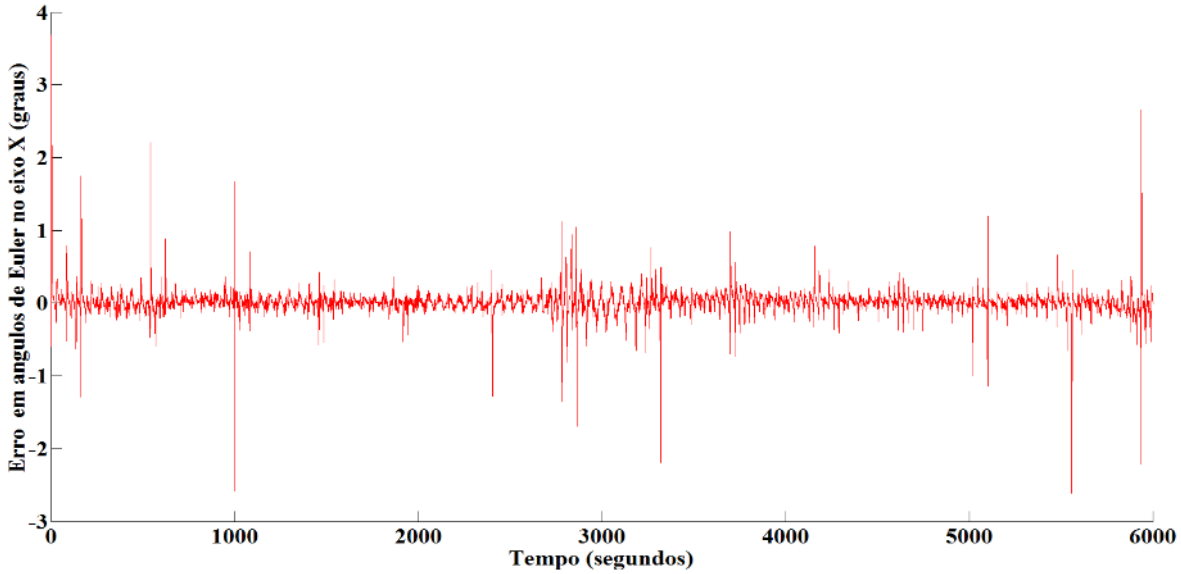

Figura 5.2 - Erro na estimativa da atitude do satélite (eixo X)

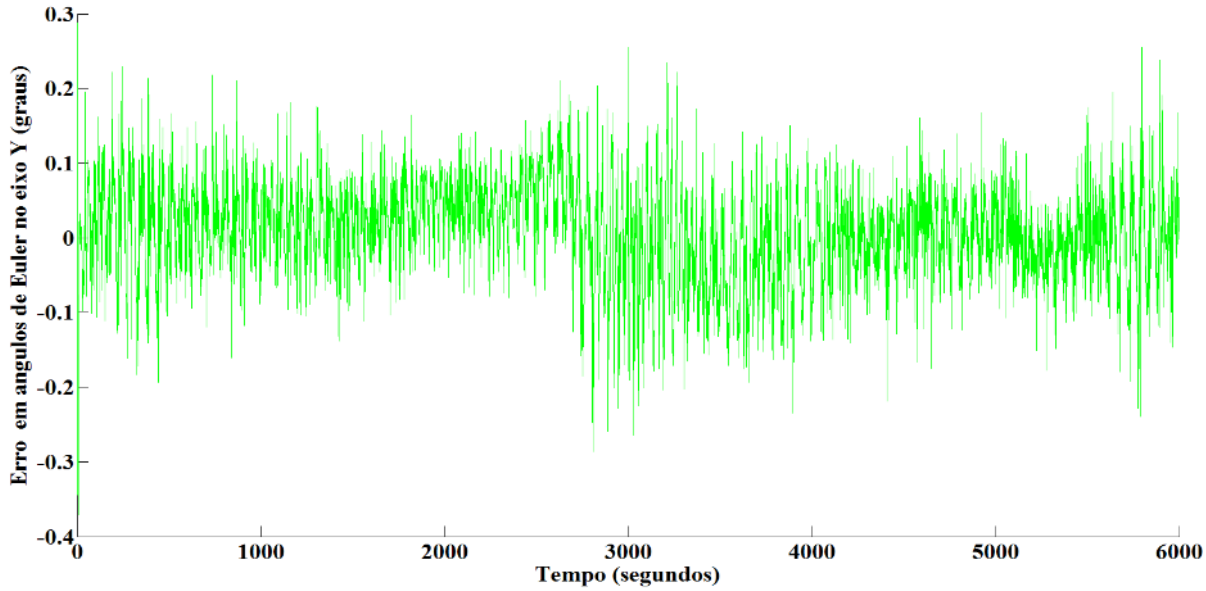

Figura 5.3 - Erro na estimativa da atitude do satélite (eixo Y)

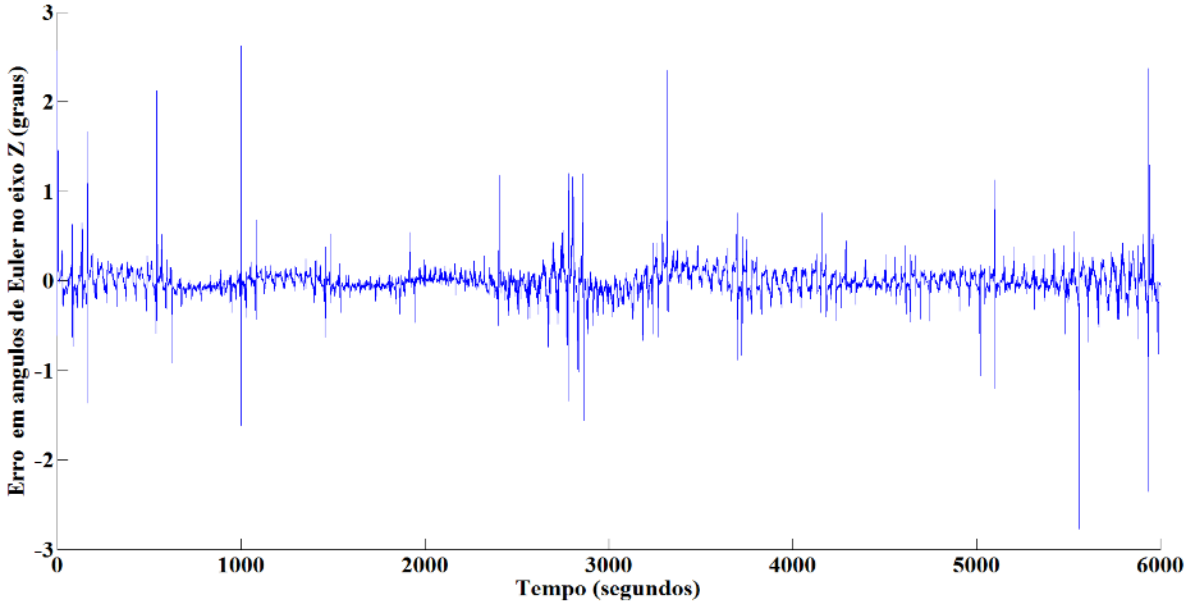

Figura 5.4 - Erro na estimativa da atitude do satélite (eixo Z)

Dado que a atitude do satélite depende da velocidade angular, é necessário verificar seu comportamento. A velocidade angular estimada, e a diferença entre a velocidade real e a estimada são mostradas nas Figuras 5.5 e 5.6 respectivamente. A convergência do filtro é mostrada na Fig. 5.7, com tempo de simulação de 20 segundos, onde é possível verificar que o erro inicial de estimação da velocidade angular é de 2,4 rpm e o tempo de convergência é de 16 segundos até atingir 5% (0,12rpm) do erro máximo. Em regime permanente o sistema opera com erros na velocidade angular melhor que 0,01 rpm, mostrados na Fig. 5.8.

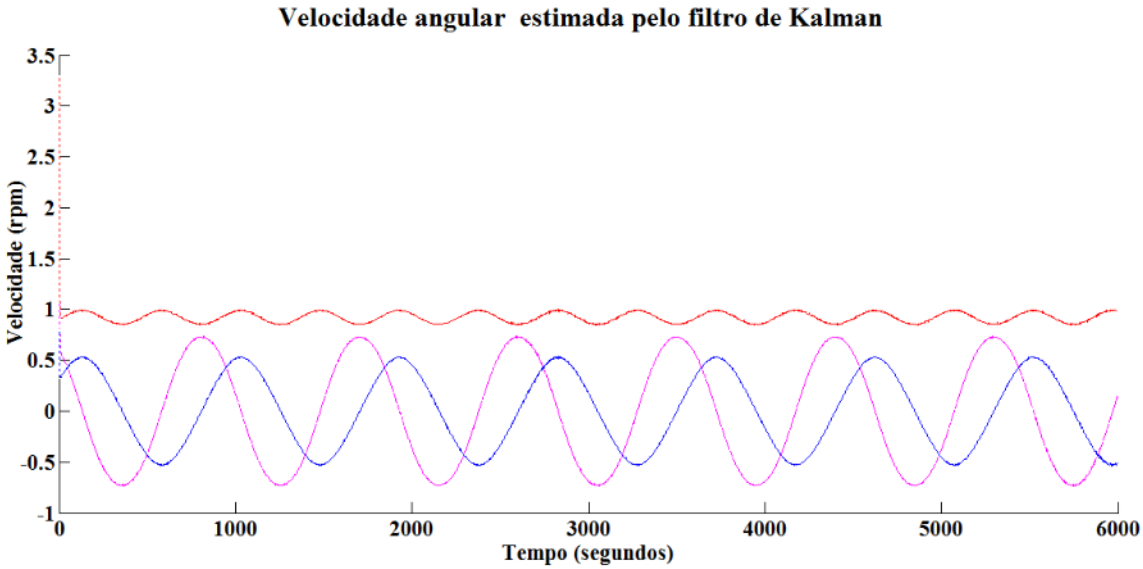

Figura 5.5 - Velocidade angular

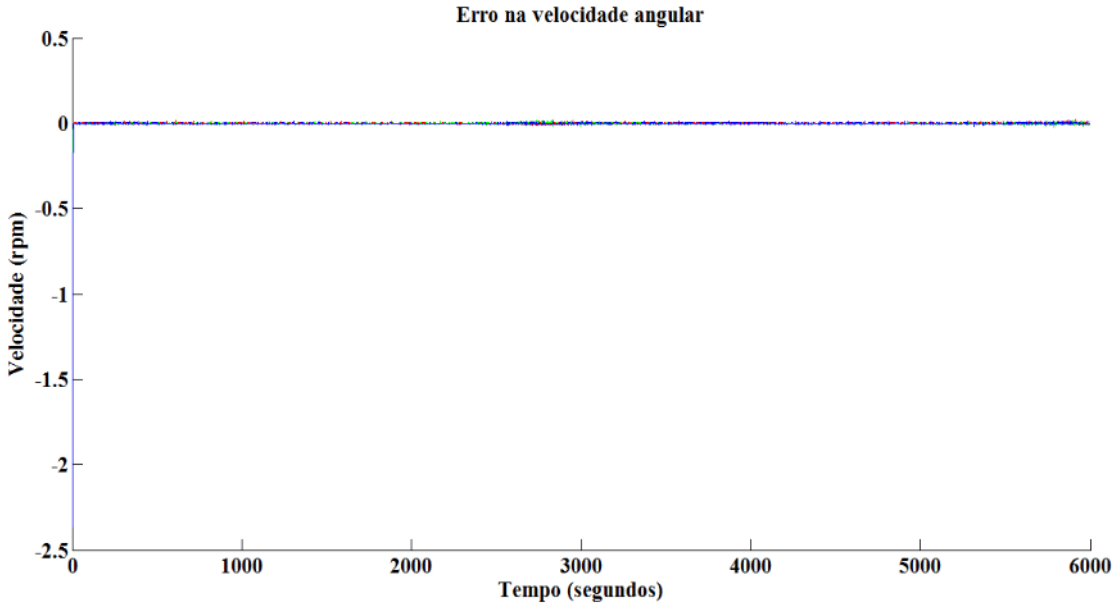

Figura 5.6 - Erro na estimativa da velocidade angular

A simulação realizada mostrou que é possível utilizar dados do magnetômetro e dos sensores solares para gerar a atitude do satélite, com erros de velocidade angular menor que 0.01rpm e erros de atitude menor que 1º.

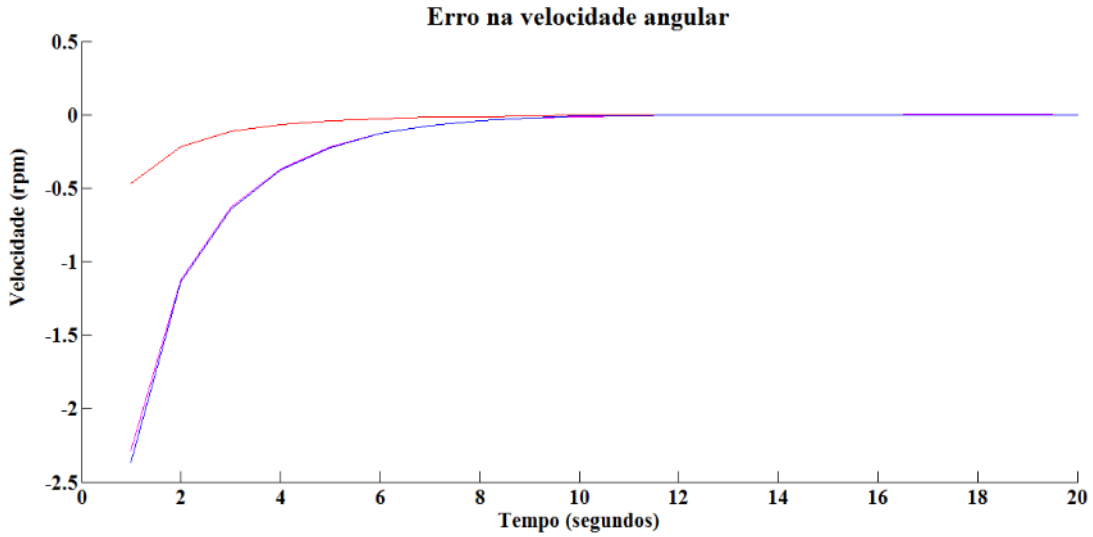

Figura 5.7 - Erro na estimativa da velocidade angular

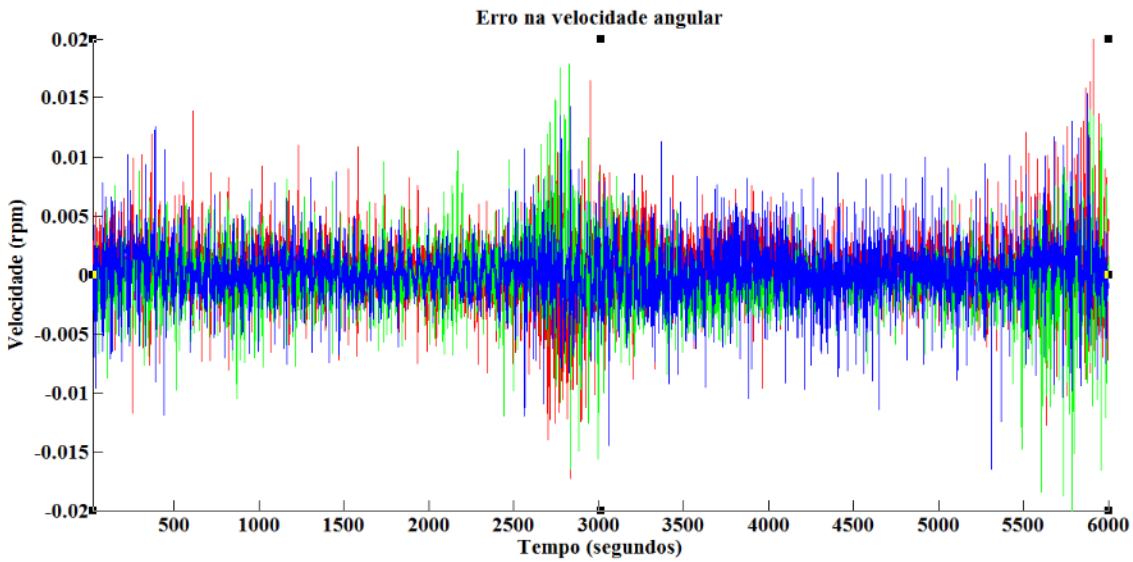

Figura 5.8 - Erro na estimativa da velocidade angular em regime permanente

5.2 Resultados para o NBR-2

Sendo o NBR-2 um Cubesat 2U, a massa considerada foi de 2,00 kg. Supondo então uma distribuição de massa homogênea no prisma, a matriz de inércia resulta diagonal, ou seja, com produtos de inércia nulos:

$$J = \begin{bmatrix} 0,0083 & 0 & 0 \\ 0 & 0,0083 & 0 \\ 0 & 0 & 0,0033 \end{bmatrix} \text{ kg m}^2. \tag{27}$$

A órbita prevista para o NBR-2 é a mesma do NBR-1, deverá ser polar, definida pelos seguintes elementos keplerianos: o semi-eixo maior da órbita a = 7008155 m, a excentricidade e = 0.01, a inclinação i = 98º, a ascensão do nodo ascendente Ω = 0, o argumento do perigeu ω= 0 e a anomalia média M = 0.

A dinâmica e cinemática do satélite são simuladas por meio da propagação dos estados por um integrador Runge Kutta de quarta ordem com passo de um segundo, sendo o tempo total de simulação igual ao tempo de uma órbita do satélite, que é igual a 6.000 segundos. Como o lançamento deste satélite será feito como carga auxiliar, não se conhece previamente a atitude inicial nem tampouco o viés do giroscópio. Por isso, adotou-se valores nulos para a atitude e para o viés na estimativa prévia do filtro.

O sistema de controle de atitude do NBR-2 deverá prover o apontamento de uma das faces do satélite para a Terra em seu modo nominal. Para que isso ocorra será implementado um controle proporcional - derivativo (PD), baseado em Carrara et al. (2014), onde o cálculo do torque demandado é calculado por:

$$\mathbf{u}_{\text{torque}} = -k_p \boldsymbol{\theta}_k - k_d \boldsymbol{\omega}_k \tag{28}$$

onde k_pe $k_d$são os ganhos proporcionais e derivativos respectivamente, $\boldsymbol{\theta}_k$ é a representação de atitude em ângulos de Euler no instante k e $\boldsymbol{\omega}_k$ é a velocidade angular.

O diagrama de blocos da Fig. 5.9 apresenta os principais blocos do sistema de determinação de atitude:

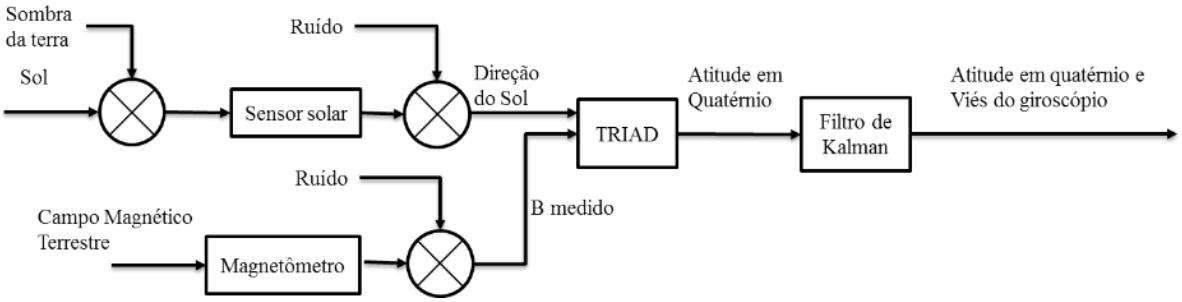

Figura 5.9 – Diagrama de blocos do sistema de determinação de atitude

A determinação da atitude é realizada pela TRIAD; como os dados de entrada estão contaminados com ruídos, o filtro de Kalman é responsável pela estimação do estado. Na Fig. 5.10 mostra-se em diagrama de blocos o sistema de controle de atitude.

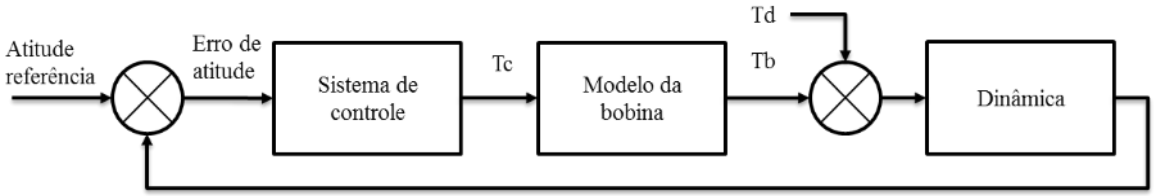

Figura 5.10 – Diagrama de blocos do sistema de controle de atitude

Os erros associados à estimação da atitude do satélite em ângulos de Euler são apresentados nas Figuras 5.11, 5.12 e 5.13; na primeira figura pode-se observar que o maior erro durante uma simulação de vinte e cinco órbitas é de cento e quarenta graus e ocorre logo após a saída da sombra da Terra, porém converge rapidamente para um valor de erro na ordem de cinco graus, conforme mostrado na Fig. 5.12. O erro em regime permanente fica na ordem de três graus, ocorrendo após a vigésima órbita.

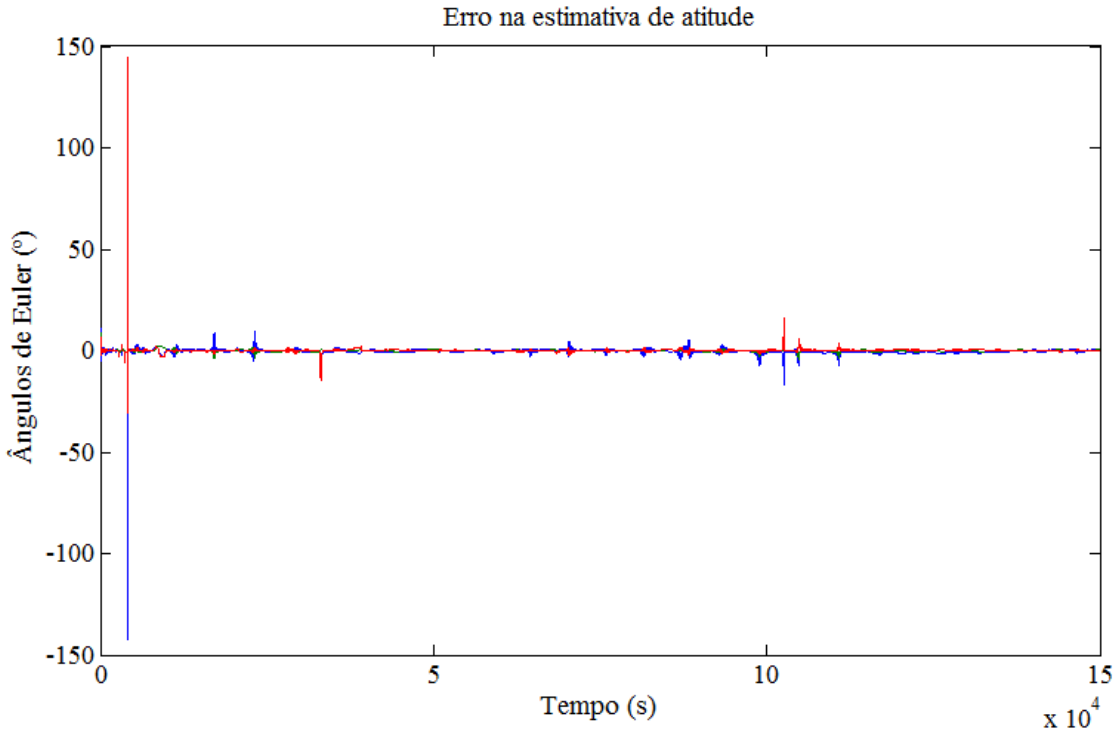

Figura 5.11 – Erro na estimativa da atitude do satélite

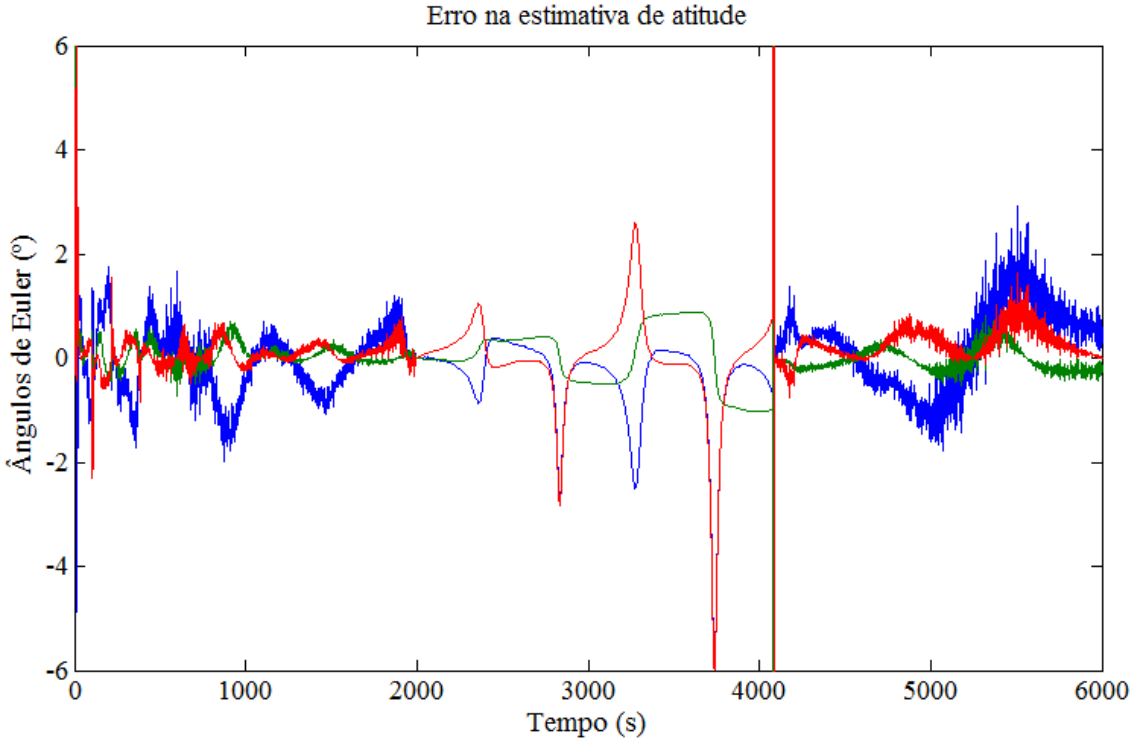

Figura 5.12 - Erro na estimativa de atitude do satélite durante a primeira órbita

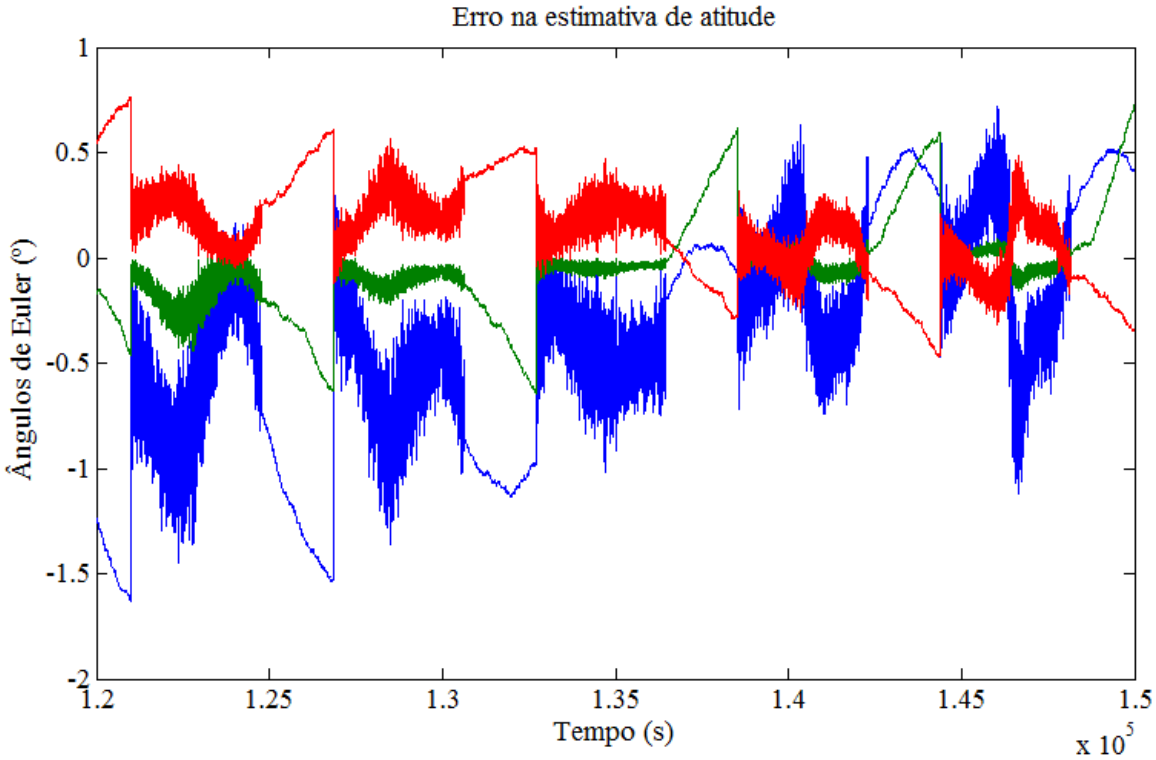

Figura 5.13 - Erro na estimativa da atitude do satélite em regime permanente

O controle atua nas bobinas gerando torque de forma a reduzir a velocidade angular do satélite em relação ao eixo z deixando a próxima da velocidade orbital, para o NBR-2 tem-

se algo em torno de 0,011 rpm, conforme mostrado na Fig. 5.15. Durante o período de sombra o controle é desligado, o que pode ser observado na Fig. 5.16 entre 2000 s e 4000 s.

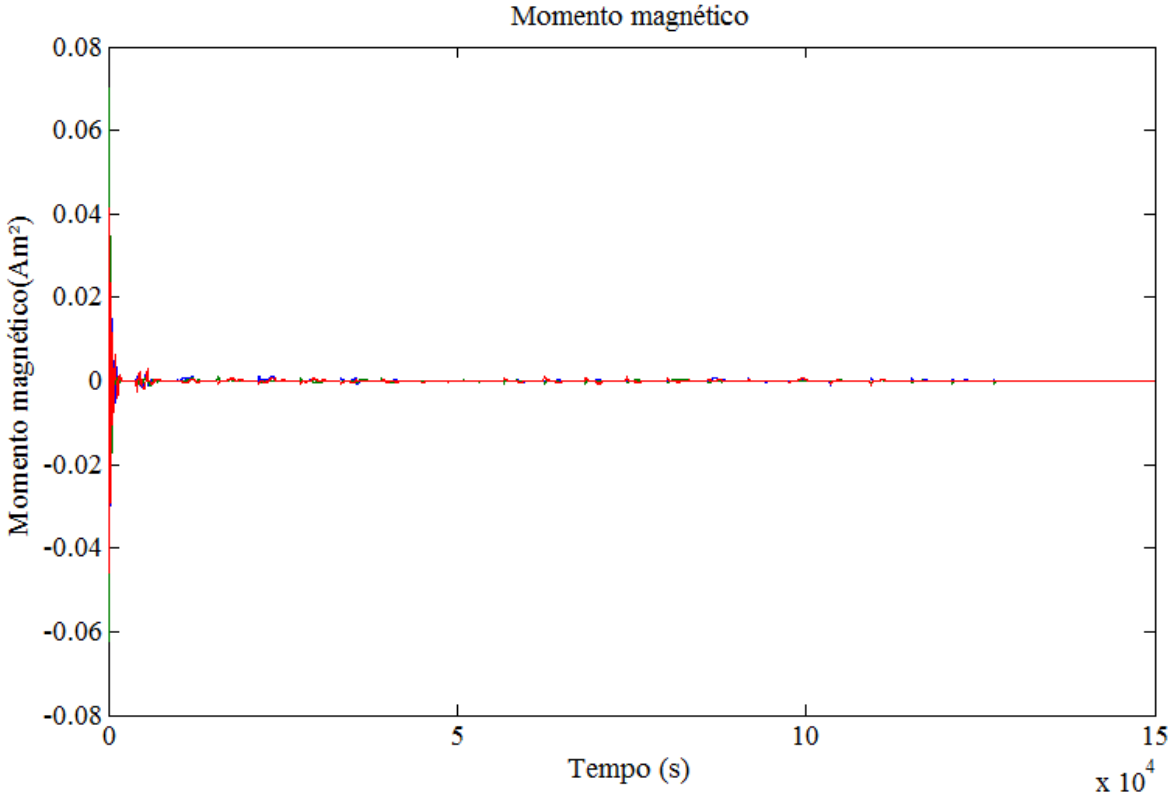

Figura 5.14 – Momento magnético

A velocidade angular é reduzida gradativamente, chegando na vigésima órbita ao valor da velocidade orbital; neste instante o sistema de controle começa a controlar o satélite de maneira a realizar o apontamento de uma das faces para a Terra. Isto pode ser observado nas Figuras 5.15, 5.16 e 5.17, onde é mostrada a simulação por 150000s, a primeira órbita e a velocidade em regime permanente respectivamente.

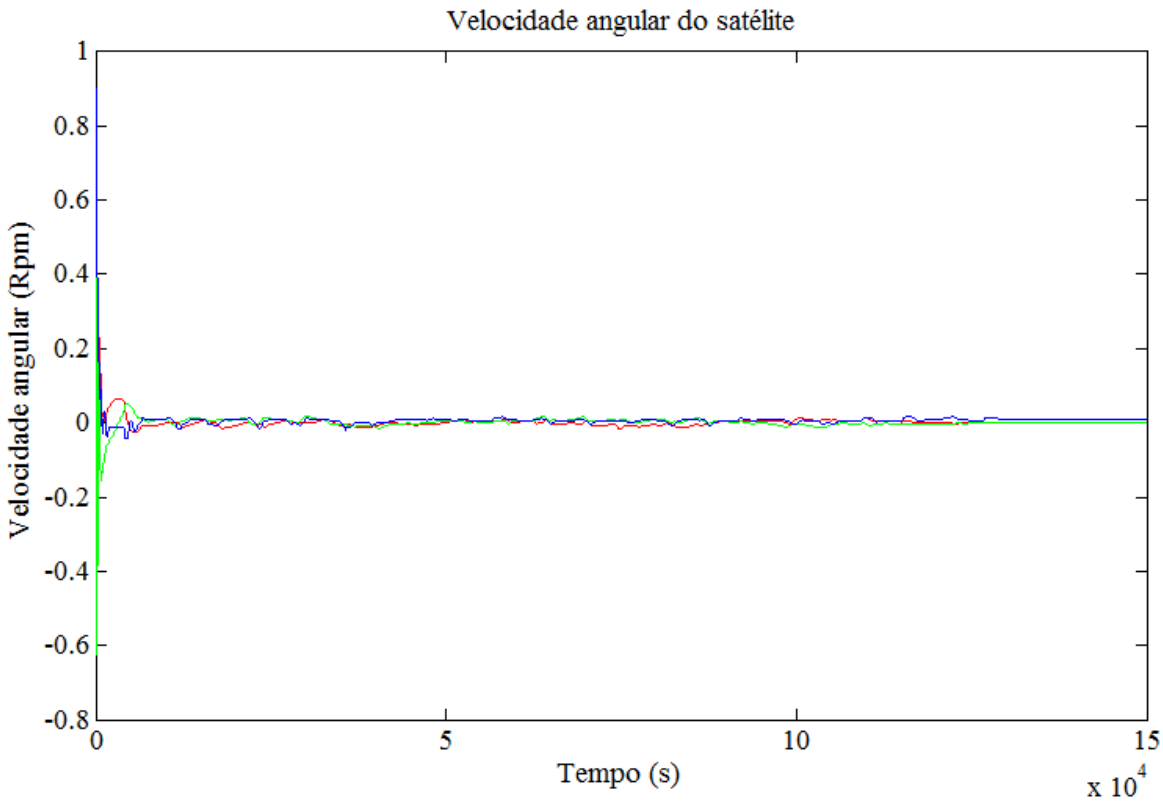

Figura 5.15 - Velocidade angular do satélite

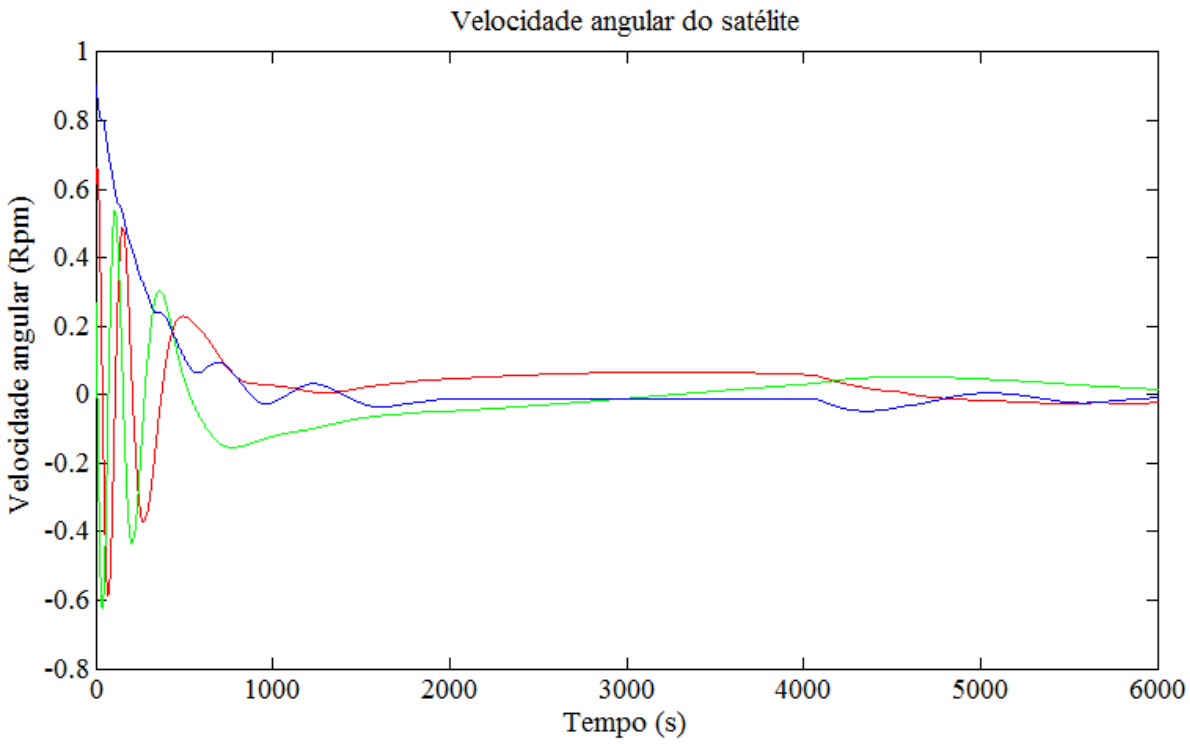

Figura 5.16 - Velocidade angular do satélite durante primeira órbita

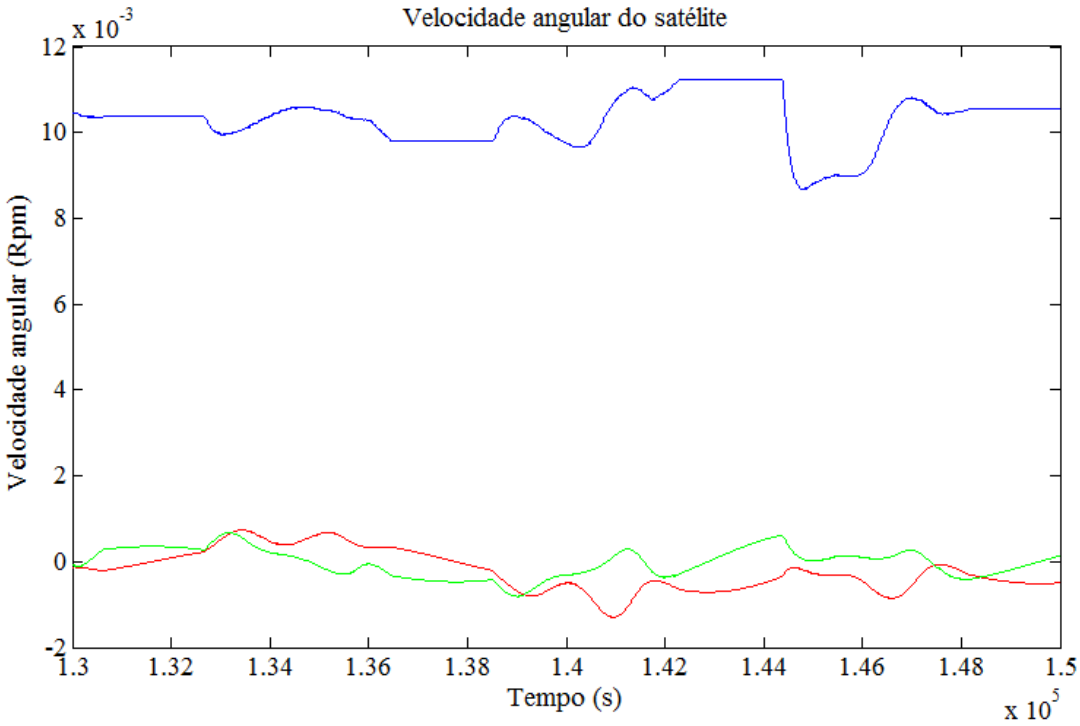

Figura 5.17 - Velocidade angular do satélite em regime permanente

O erro de atitude é mostrado na Fig. 5.18, numa simulação de 150000 s e o erro em regime permanente é mostrado na Fig. 5.19, onde se pode observar que o erro de apontamento é menor que cinco graus.

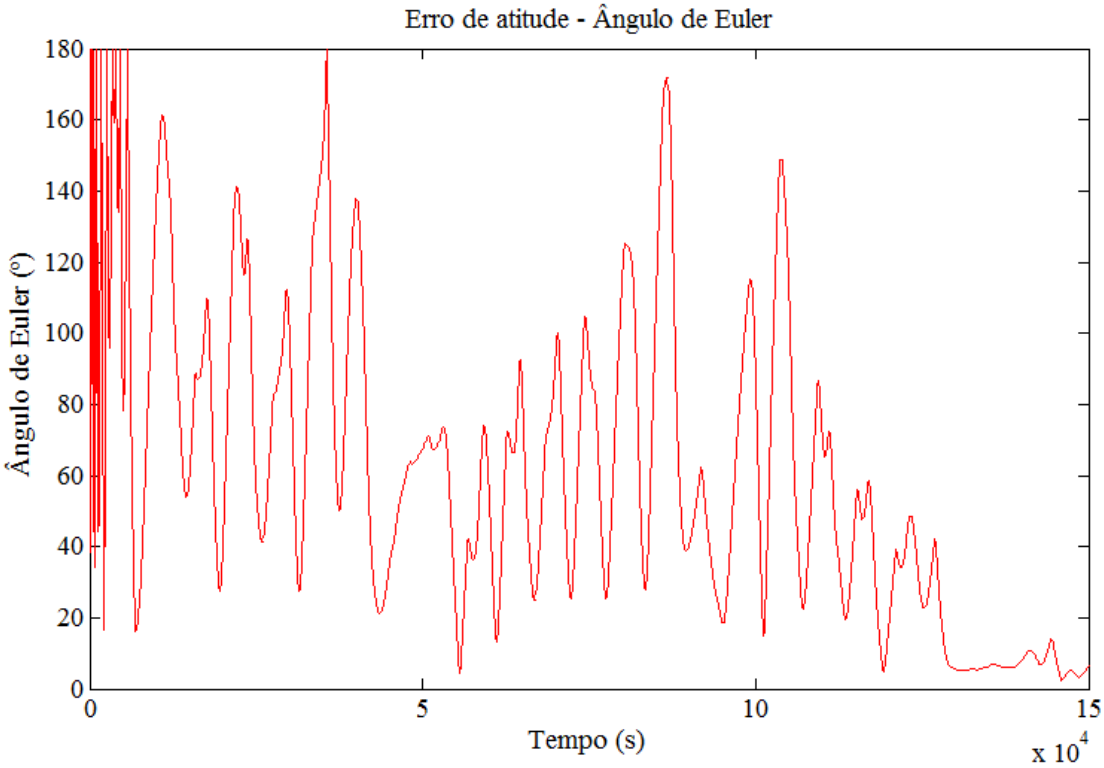

Figura 5.18 – Erro de atitude no sistema orbital

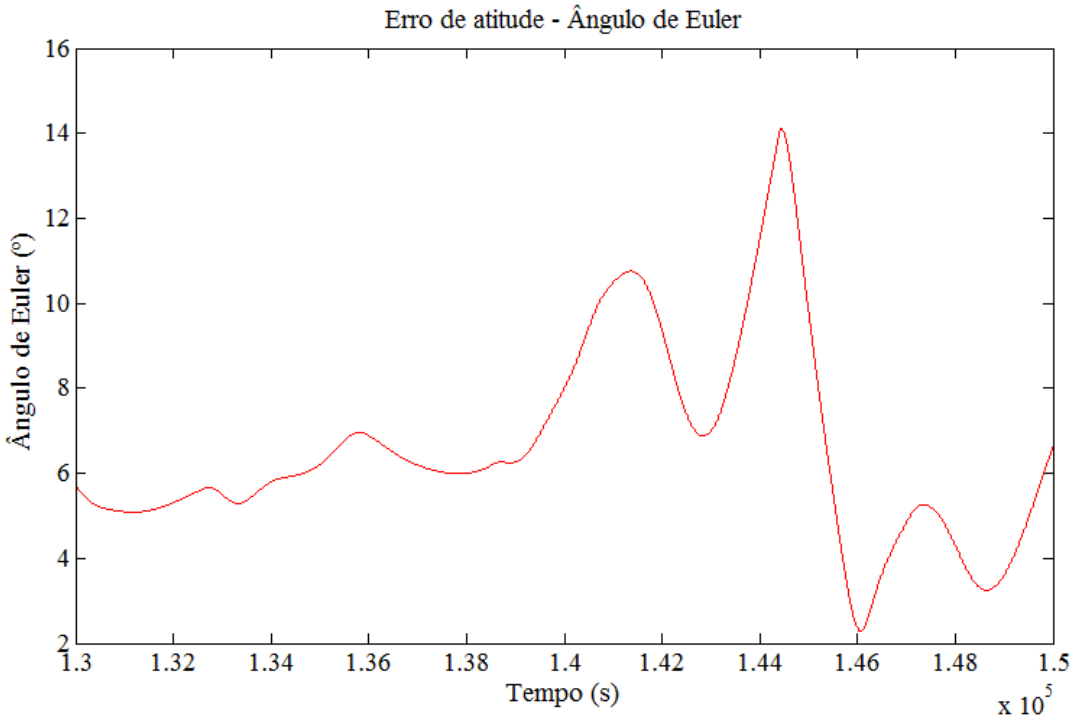

Figura 5.19 - Erro de atitude em regime permanente

O viés do giroscópio é estimado pelo filtro de Kalman e mostrado nas Figuras 5.20 e 5.21, sendo a primeira a simulação dos 150000 s e a segunda o valor em regime permanente.

Como a dinâmica do satélite é influenciada diretamente pela velocidade angular, o menor erro possível deve ser buscado.

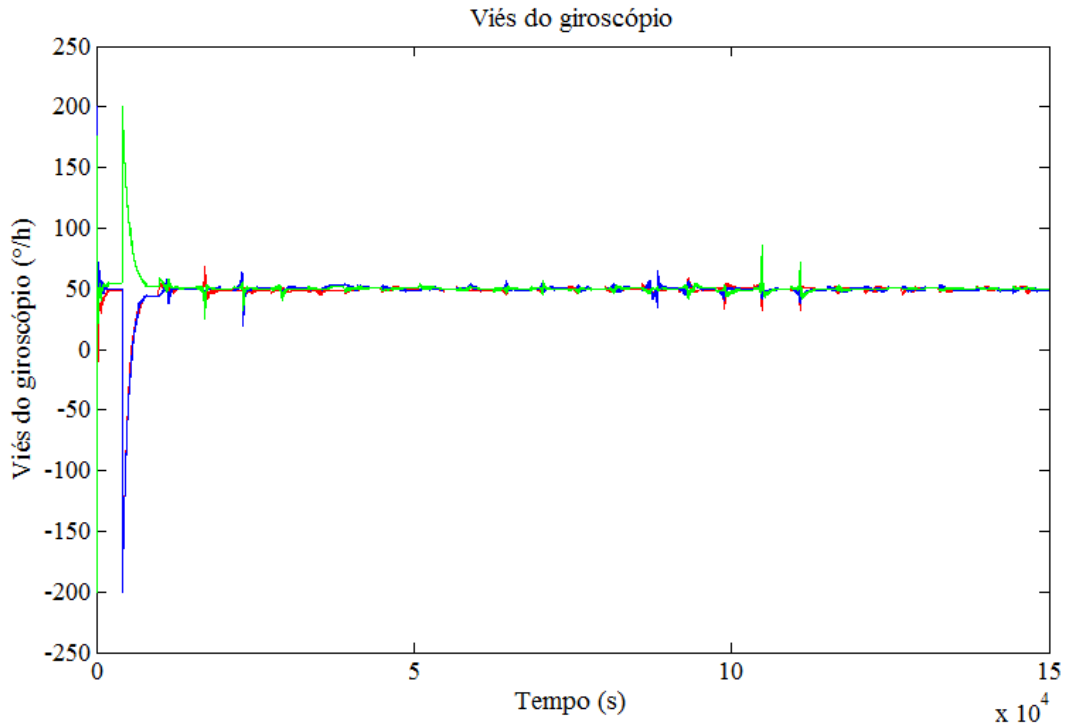

Figura 5.20 - Viés do giroscópio estimado pelo filtro de Kalman

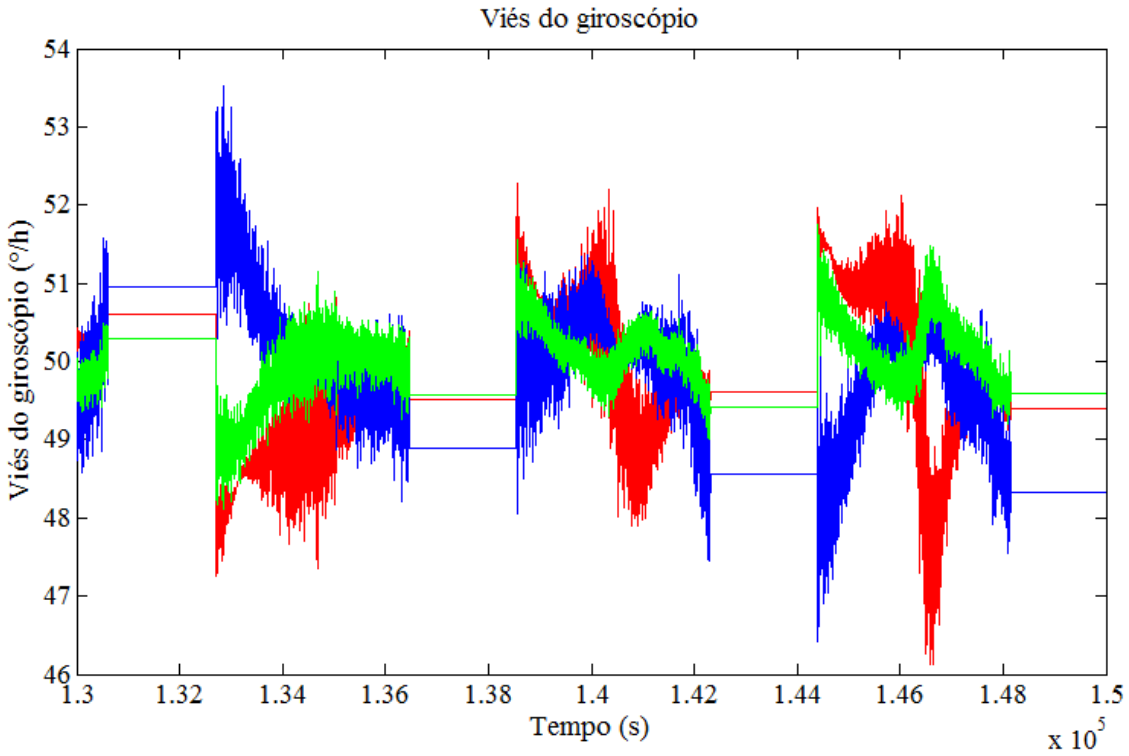

Figura 5.21 - Viés do giroscópio em regime permanente

A atitude do satélite é mostrada nas Figuras 5.22 e 5.23 respectivamente numa simulação dos 150000 s, e o valor em regime permanente. O erro de apontamento está na ordem de três graus, atendendo os requisitos iniciais para este projeto que é de cinco graus.

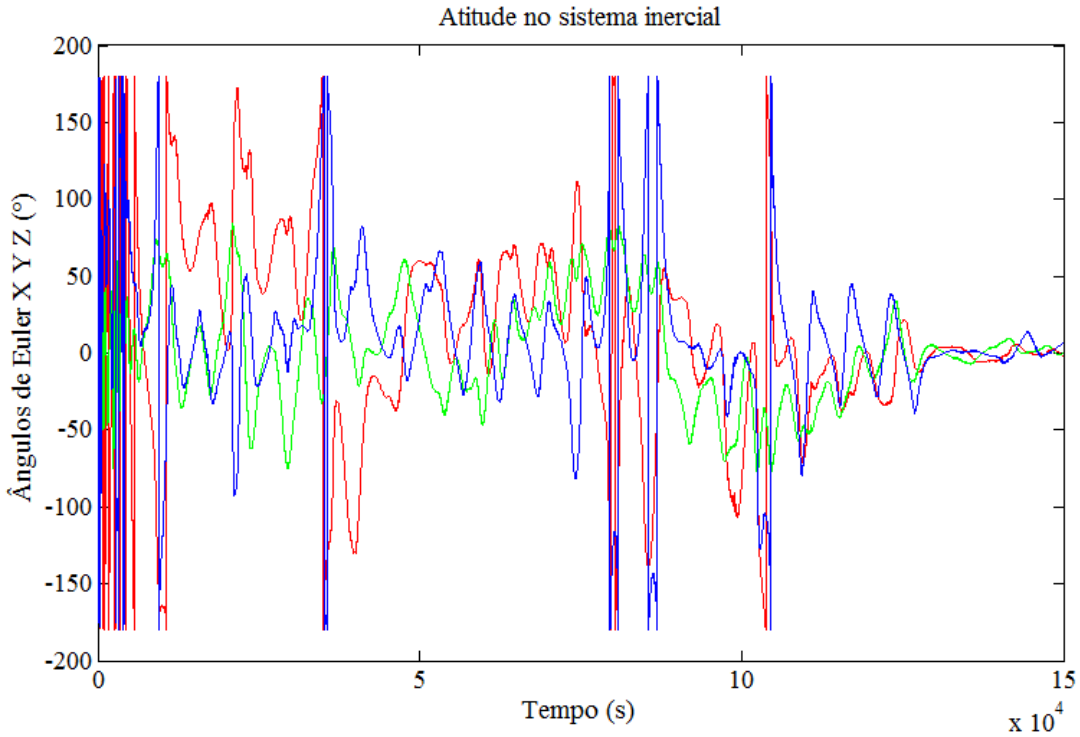

Figura 5.22 - Atitude inercial em ângulos de Euler

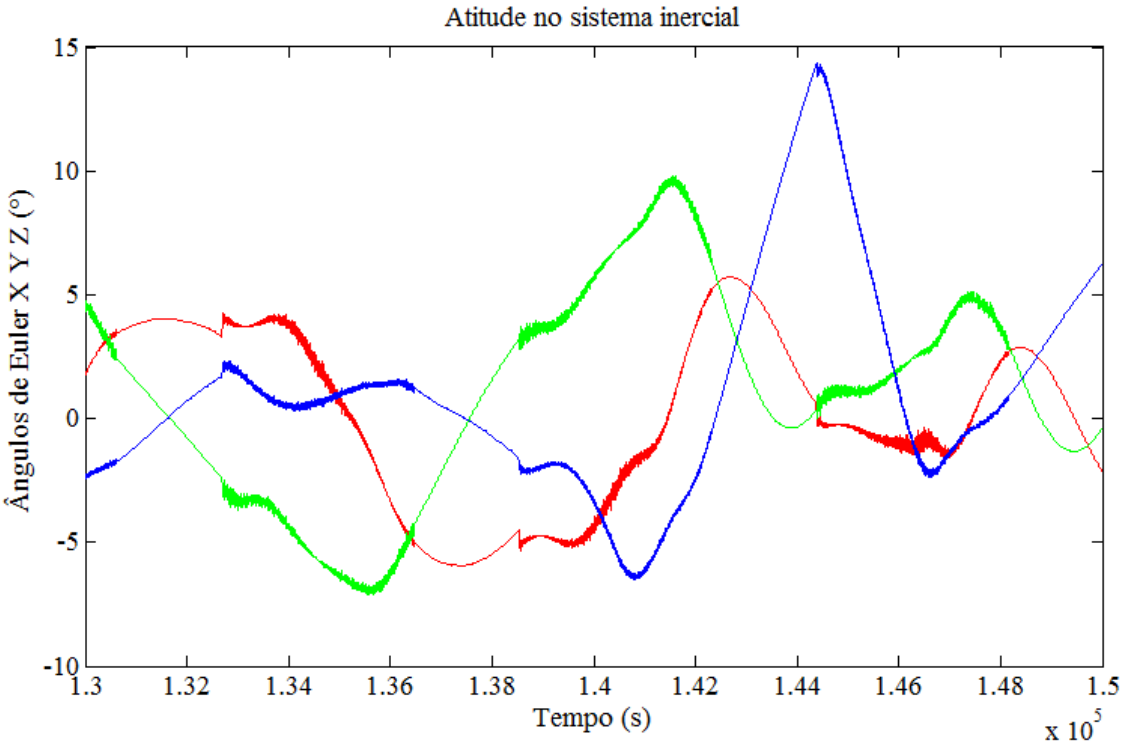

Figura 5.23 - Atitude inercial em ângulos de Euler em regime permanente

5.3 Resultados EGSE

Conforme descrito no capítulo 4 foi utilizada uma metodologia a fim de minimizar a possiblidade de ocorrência de erros, desta forma neste capítulo serão apresentados os testes realizados.

O primeiro teste realizado foi para avaliar o tempo de latência do Windows e simulador, de forma a verificar se seria necessária a utilização de sistemas de tempo real. Ao executar o simulador no Matlab, foi verificado que o Windows e Matlab otimizam a rotina de forma a melhorar a performance do processamento, fazendo com que nos primeiros ciclos ocorra uma diminuição significativa do tempo de processamento, em torno de 10 vezes, conforme pode-se observar na Fig. 5.24, o que a princípio não é um problema, dado que o satélite levará algumas órbitas para sair do modo detumbling.

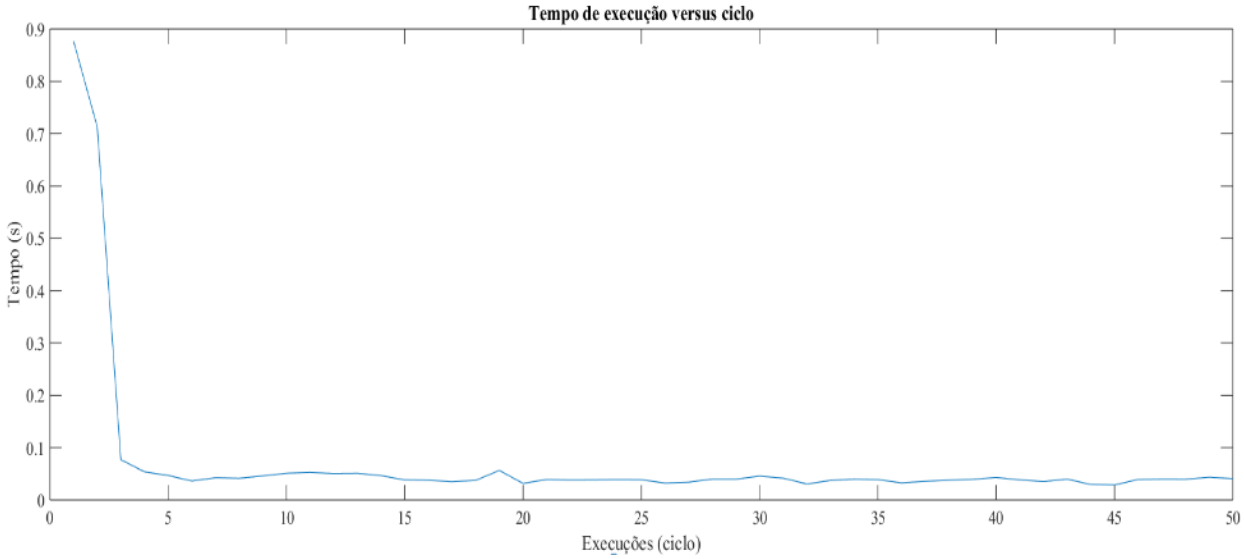

Figura 5.24 - Otimização da execução pelo Windows Matlab

No entanto o tempo de processamento também varia com o tipo de integrador, a tolerância selecionada para o integrador, se o equipamento está em modo economia de energia, entre outros aspectos. Na figura 5.25 é mostrado que o tempo de execução varia entre 0,02s e 0,18 quando utilizado o integrador ODE 45 no Matlab, ocasionando problemas de execução em tempo real, o sistema não é determinístico. Alterando o tipo de integrador para ODE23T obteve-se um aumento no tempo de ciclo, porem quando mudamos o modo de operação do computador para alto desempenho houve uma redução significativa do tempo.

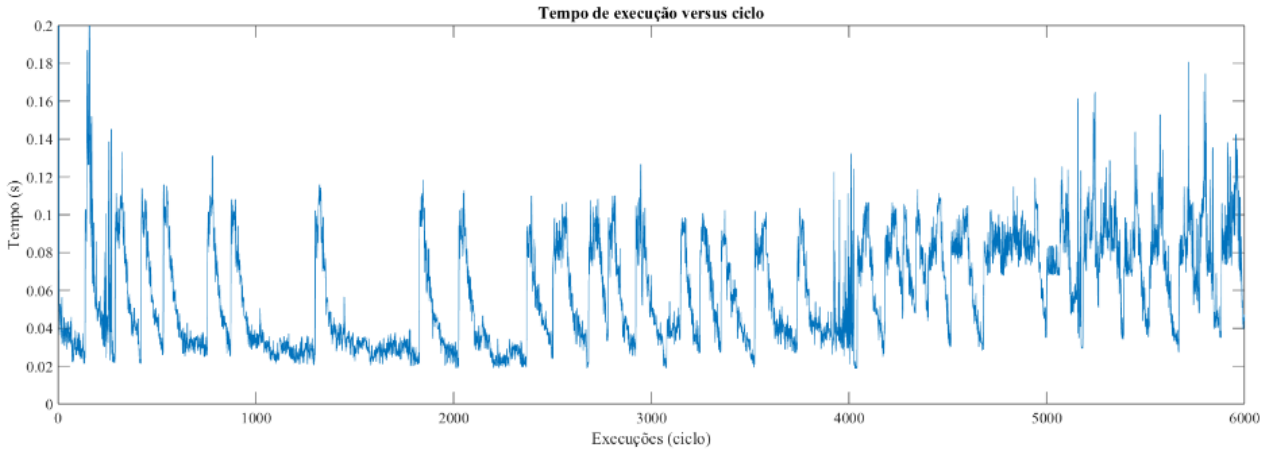

Figura 5.25- tempo de execução utilizando integrador Runge Kuta de passo variável - ODE45

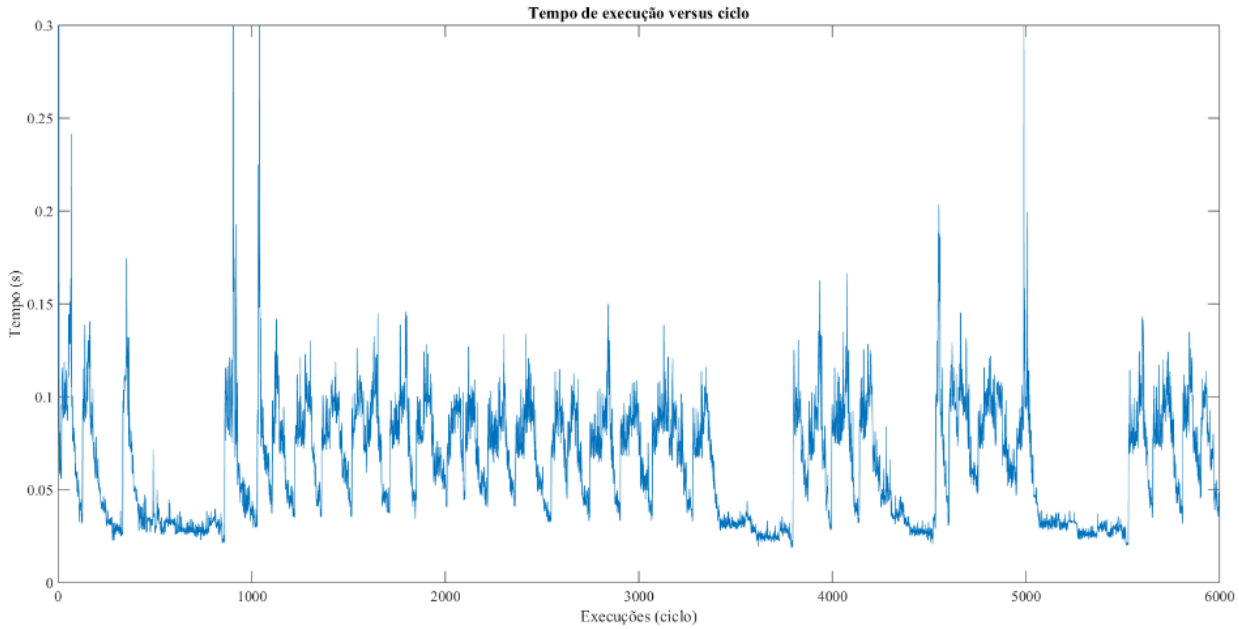

Figura 5.26 - teste integrador ODE23T

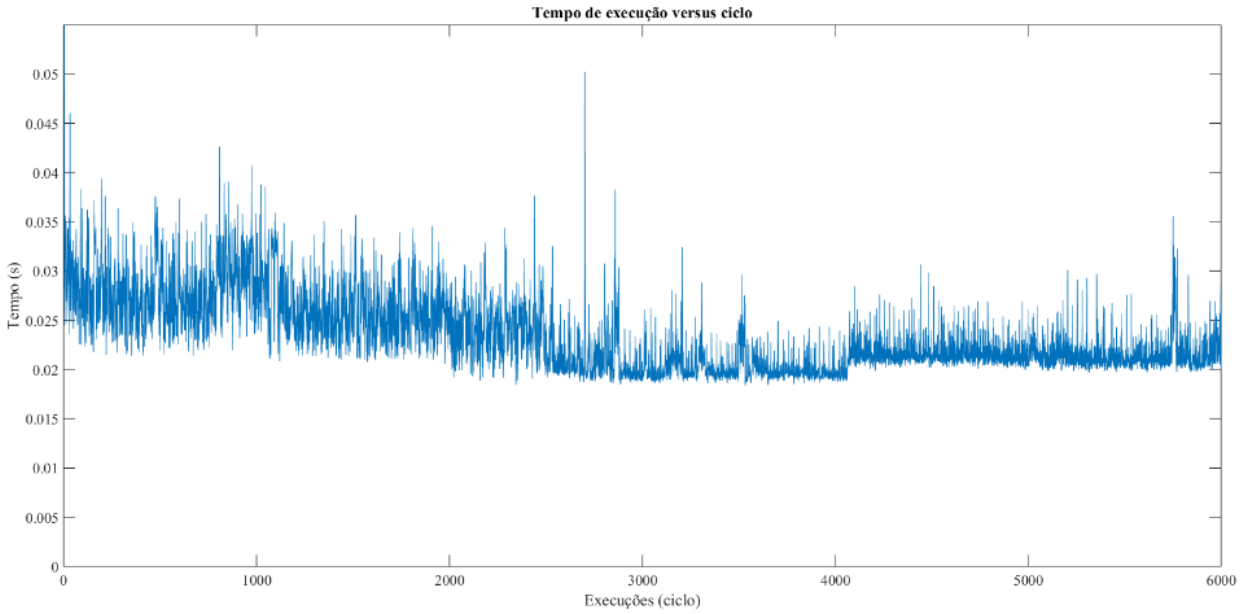

Figure 5.27- teste integrador ODE23T em modo alto desempenho

Durante os testes no simulador no Matlab ocorreram problemas de integrar o Raspberry e o Arduino em um único simulador, devido a este problema foi modificado para ser executado no Simulink, onde através da utilização de funções específicas pode-se integrar mais de um processador externo. A Fig. 5.28 apresenta o simulador em Simulink.

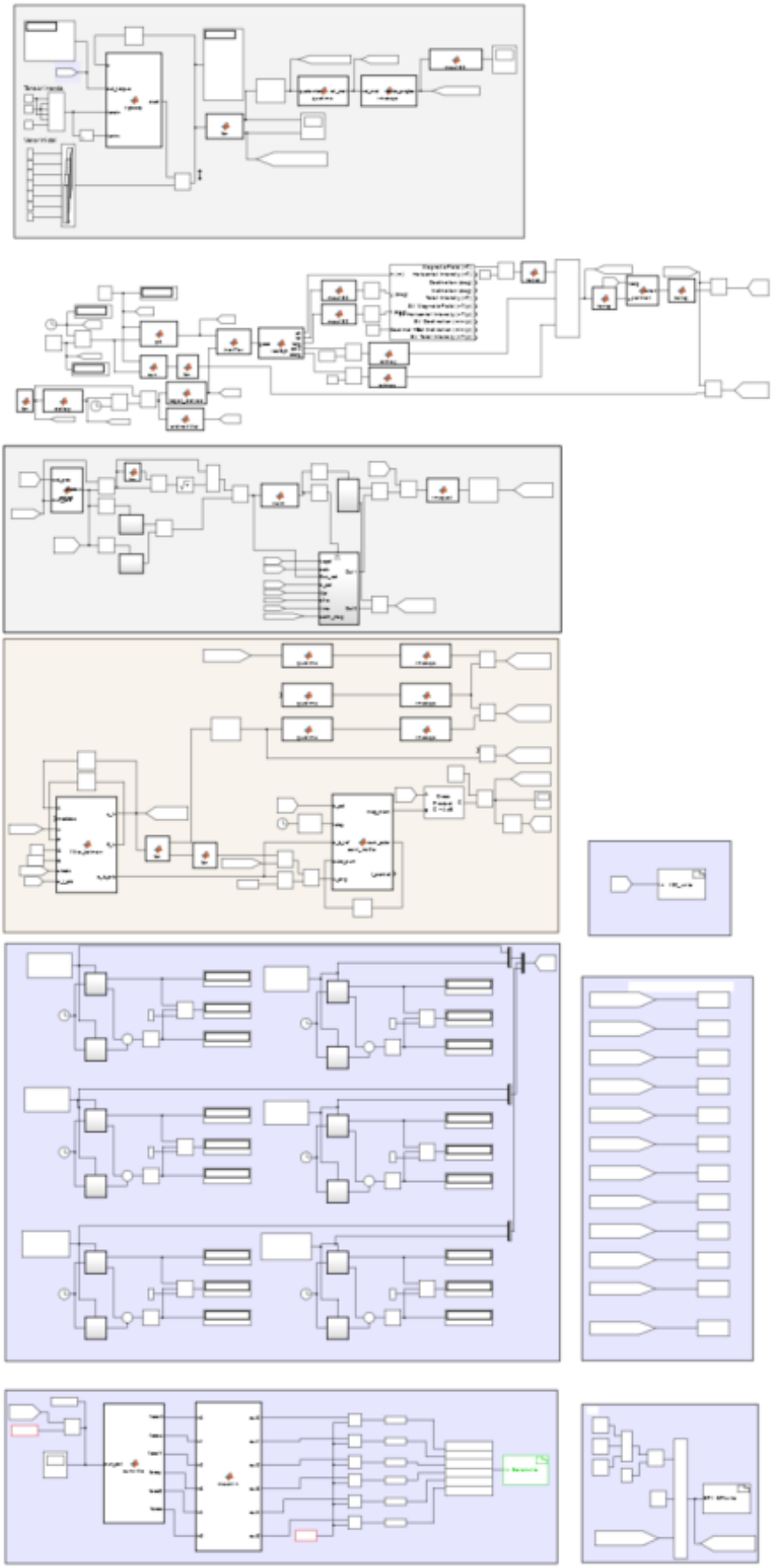

Figura 5.28 - Simulador em Simulink

Ao executar os testes no simulador verificou-se que em frequências acima de 40 hz ocorre um estouro do buffer de entrada e saída do simulador com o hardware real.

Para testar o sensor solar foi executada a escrita de valores do simulador no Raspberry Pi onde via serial escreve os valores no Arduino Pró Mini que via modulação por largura de pulso aplicada a um filtro passa baixa e divisor de tensão gera o sinal na entrada do Nanomind. A tabela 5.1 apresenta os valores e a Fig. 5.29 apresenta os valores em um gráfico.

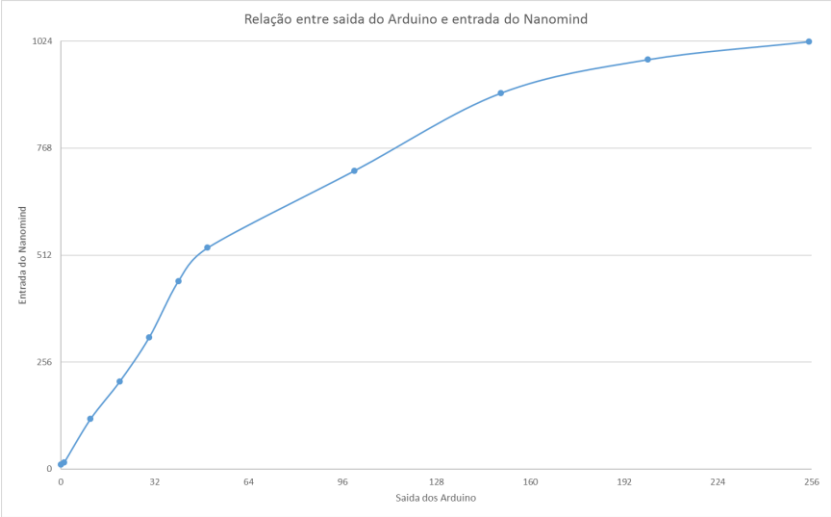

Figura 5.29 – Relação entre a saída do Arduino e entrada Nanomind

Tabela 5.1 – Comparativo Arduino versus Nanomind

Escrita Arduino	Leitura Nanomind	Tensão Nanomind (mV)
0	11	0,06
1	16	3
10	120	23,7
20	210	44,9
30	315	64,4
40	450	82,3
50	530	99
100	714	169,8
150	900	218,2
200	1023	240,7
255	1023	253,4

Realizado teste para verificação do sistema de leitura do tempo em que a bobina fica ligada em torque aplicado ao satélite, como se pode observar na Fig. 5.30.

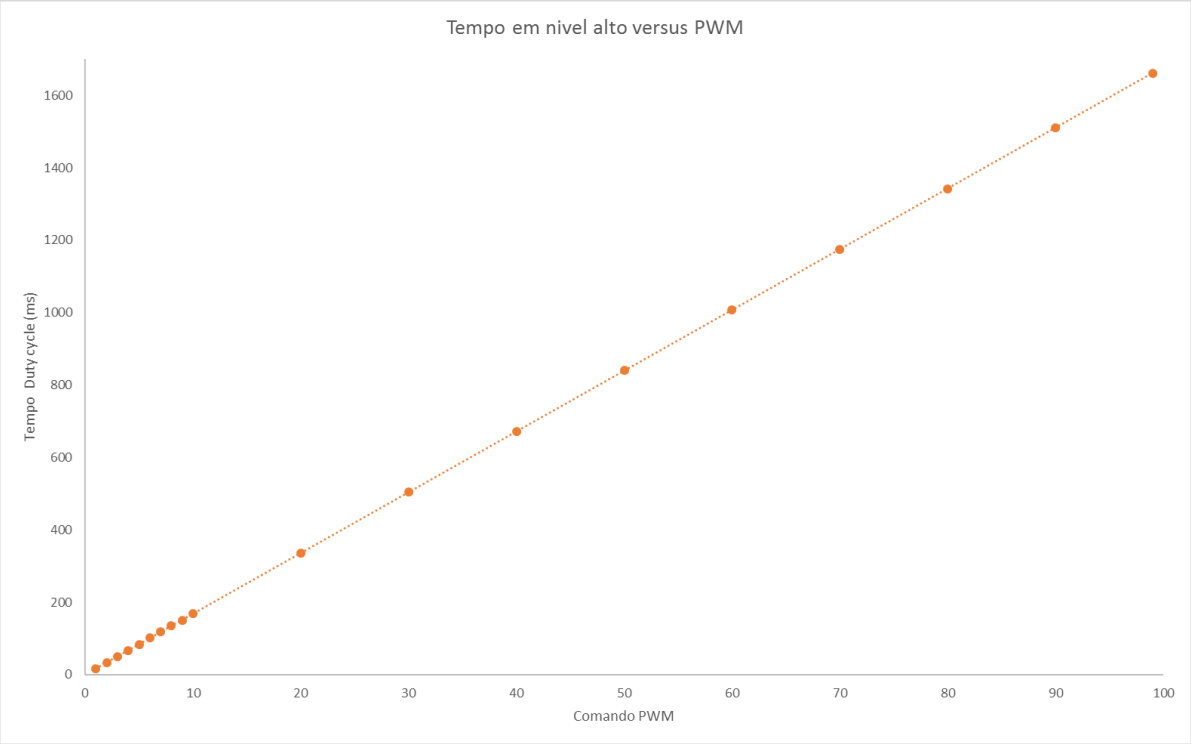

Figura 5.30 – Tempo em nível alto versus comando PWM

O magnetômetro não foi testado junto ao Nanomind devido à falta do código embarcado, para evitar a falta do teste foi adquirido um Magnetômetro HMC 5883 e seus dados foram adquiridos utilizando de um Arduino Due. De posse desses dados foi gerado o código no Raspberry, para verificar a fidelidade ao modelo foi rodado o código no Arduino Due e comparado ao Real. Abaixo, apresenta-se a Tabela 5.2 com valores reais e curva gerada pelo Raspberry.

Tabela 5.2 – Valores obtidos no Raspberrry

Valores reais	Valores gerados pelo Raspberry
0 Ga	0
0 Ga	0
0 Ga	0
1.1 Ga	1.1 Ga
1.1 Ga	1.05 Ga
1.1 Ga	1.04 Ga

6. CONCLUSÕES

Desta forma pode-se observar que os requisitos iniciais foram atendidos e o simulador do sistema de determinação e controle de atitude está funcionando. Como próximos passos, pretende-se criar estas funções no ambiente de programação Eclipse, com linguagem C, para que se possa criar uma biblioteca compatível com o FreeRTOS.

Tendo em vista que este trabalho se propõe a criar um sistema de determinação e controle de atitude a ser embarcado no computador de bordo NanoMind do NBR-2, deve-se analisar as funções que foram utilizadas no simulador de forma a existir compatibilidade. Como se pretende utilizar o FreeRTOS como sistema operacional a ser embarcado no NanoMind, e o FreeRTOS pode ser compilado com funções desenvolvidas em C++, pode-se afirmar que as funções utilizadas no simulador em Matlab podem ser utilizadas com pequenas alterações, como a definição do tipo da variável que está sendo manipulada. Todas as tarefas executadas pelo sistema são gerenciadas pelo sistema operacional, que executará as tarefas de maneira pré-ordenada pelo seletor de prioridades. O NanoMind pode ser programado para interromper uma tarefa e executar outra, voltando à primeira quando esta tiver maior prioridade no momento.

A memória será outro ponto a ser otimizado, uma vez que existem limitações, a saber: 2MB para memória RAM que será utilizada para efetuar os cálculos e manipulações de dados, 4MB para o código, o que não será o ponto mais crítico, uma vez que este será otimizado via software durante a geração do código em *assembler* e 4MB para os dados a serem armazenados para *upload* via *downlink* para estação terrena, o que ocorrerá no pior dos casos em duas conexões de dez minutos sendo uma em cada estação do INPE (RS e SP).

Neste trabalho, utilizou-se uma placa de desenvolvimento Raspberry Pi Modelo B+ programada com Matlab e Simulink para servir de interface com o hardware. As informações necessárias para a correta instalação e utilização são apresentadas no Apêndice A. Sua capacidade de conexão e interligação permite concluir que, utilizando de software simulado, é possível substituir hardwares complexos como mesas de mancais aerostáticos e bobinas de Helmholtz. O Matlab possui um pacote de apoio para gerar códigos para o Raspberry Pi, que permite acesso a todas as interfaces deste microprocessador. A implementação foi realizada utilizando modelos Simulink, conforme foi apresentado no Capítulo 4, cujos resultados foram mostrados no Capítulo 5. O modelo foi construído usando blocos especiais para o Raspberry Pi e funções Sfunction, onde foram carregados os scripts para acesso aos diversos dispositivos de maneira simultânea, e também para a implementação de funções não existentes no

Simulink. A criação de funções específicas permite simular componentes custosos, e avaliar o comportamento de hardware em situações de falha, erros de modelagem e condições definidas pelo operador.

Como sugestão de trabalhos futuros, recomenda-se a implementação de uma interface Gráfica (GUI) ao Matlab, na qual os parâmetros de simulação podem ser ajustados e modificados de forma a não ser necessária a alteração do código fonte. Outra sugestão seria integrar o EGSE apresentado aqui a um modelo com simulação física capaz de estimular os sensores, com mancais aerostáticos e bobina de Helmholtz, que permitem avaliar o grau de fidelidade dos modelos gerados. Ensaios de estabilidade também podem ser avaliados utilizando este trabalho como base.

REFERÊNCIAS BIBLIOGRÁFICAS

APTE, A. **Set up and Blink -** MATLAB and simulink with raspberry Pi. Adafruit Industries, Learning System, 2014

BAK, T; WISNIEWSKI, R; BLANKE, M. Autonomous attitude determination and control system for the ørsted satellite. In: IEEE AEROSPACE APPLICATION CONFERENCE, 1996, Colorado. **Proceedings…** Piscataway, NJ.: IEEE, 1996.

CARRARA, V; **PROPAT Satellite Attitude and Orbit Toolbox for Matlab**. São José dos Campos, INPE, 2009.

CARRARA, V. **Cinemática e dinâmica de satélites artificiais**. São José dos Campos, Instituto Nacional de Pesquisas Espaciais, 2012. 111 p. (sid.inpe.br/mtc-m19/2012/01.26.19.13-PUD). Disponível em: <http://urlib.net/8JMKD3MGP7W/3B96GD8>. Acesso em: 22 jul. 2016.

CARRARA, V; KUGA, H. K; BRINGHENTI, P. M; CARVALHO, M. J. M. Attitude Determination, Control and Operating Modes for CONASAT Cubesats. In: INTERNATIONAL SYMPOSIUM ON SPACE FLIGHT DYNAMICS (ISSFD), 24., 2014, Laurel, Maryland. **Proceedings...** 2014. DVD. Disponível em:<http://https://dnnpro.outer.jhuapl.edu/issfd2014/Agenda.aspx>. Acesso em: 22 jul. 2016.

FARREL, J. L. Attitude determination by kalman filtering. Automatica. **Journal of IFAC**, v. 6. n. 3, p. 419-430, 1970.

FERREIRA, A. J; CASTRO, J. C. V. MARTINS-FILHO, L. S.; DUARTE, R. O; PRADO, R. S; ROCHA, R. Procedimento experimental para a determinação de atitude de satélites artificiais. **Jornal Ciência & Engenharia**. v. 17, n. 1-2, 2008.

GARCIA, R. **Filtro não linear de Kalman Sigma-Ponto com algoritmo unscented aplicado a estimativa dinâmica da atitude de satélites artificiais.** 2011. 189 p. (sid.inpe.br/mtc-m19/2011/10.16.14.32-TDI). Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011. Disponível em: <http://urlib.net/8JMKD3MGP7W/3AKGTCP>. Acesso em: 22 jul. 2016.

HUGHES, P. C. **Spacecraft attitude dynamics**. Dover Books, 1986. Dover Books on Aeronautical Engineering (série).

OLIVEIRA JUNIOR, A. F; CARRARA, V; KUGA, H. K. Estimation of angular velocity and attitude quaternions by Kalman filter. In: IBERO LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING (CILANCE), 35., 2014, Fortaleza. **Anais...** Fortaleza: (PEC/UFC), 2014.

KUGA, H. K. **Noções práticas de técnicas de estimação**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2005.

LEFFERTS, E. J; MARKLEY, F. L; SHUSTER, M. D. Kalman filtering for spacecraft attitude estimation. **Journal of Guidance, Control and Dynamics**, v. 5, n. 5, p. 417, 1982.

LING, E. **Serial Communication between Raspberry Pi and Arduino**. Dashboard Project. Disponível em https://dashboardproject.wordpress.com/2015/12/14/serial-communication-between-raspberry-pi-and-arduino-in-simulink/ acessado em 02 de janeiro de 2015.

MCGEE, L. A; SCHMIDT, S. F. **Discovery of the Kalman filter as a practical tool for aerospace and industry**. Washington: NASA Technical Memorandum 86847, 1985.

POMYEN, S. **Signal and image processing with Matlab on Raspberry pi plataform**. Master of Science thesis. Tampere University of Technology, 2015.

QUADRINO, K. M; MILLER, D; KARROY, K. **Testing the attitude determination and control of a CubeSat with hardware-in-the-Loop**. Master of Science thesis. MIT, 2014

SANTOS, D. A. **Estimação de atitude e velocidade Angular de satélites utilizando medidas do campo geomagnético e da direção do Sol**. 2008. Dissertação (Mestrado em Engenharia Eletrônica e Computação) - ITA, São José dos Campos, SP, 2008.

SHUSTER, M. D; Oh, S. D. Three-axis attitude determination from vector observations. **Journal of Guidance and Control**. v. 4, n. 1, p. 70-77, 1981.

VIEGAS, W. V; WALDMANN, J. **Estimação e controle autônomos de atitude em 3 eixos de satélite de baixo custo estabilizado por roda de momentum com sensores de sol, magnetômetro e atuação magnética**. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA (CBA), 18., 2010, Bonito - MS. **Anais...** Campinas: SBA, 2010.

WALTER, J; FAKIH, M; GRUTTNER, K. Hardware-based real-time simulation on the Raspberry Pi. In: WORKSHOP ON HIGH-PERFORMANCE AND REAL-TIME EMBEDDED SYSTEMS (HiRES 2014), 2., 2014, Vienna. **Proceedings…** Vienna: OFFIS, 2014.

WERTZ, J. R. **Spacecraft attitude determination and control**. . London: D. Reidel, 1978.

Nanomind A712C. Disponível em: http://gomspace.com/index.php?p=products-a712c [acessado em 10 de janeiro de 2015].

Matlab support package for raspberry pi hardware [online], Disponível em: http://se.mathworks.com/help/supportpkg/raspberrypiio/ [acessado em 15 de Maio de 2015].

APENDICE A

Procedimento para utilização do EGSE (HIL NCBR-2)

Instalação do pacote de software Raspberry Pi

- No menu Home
- Clicar em AddOns
- Get Hardware Support Packages

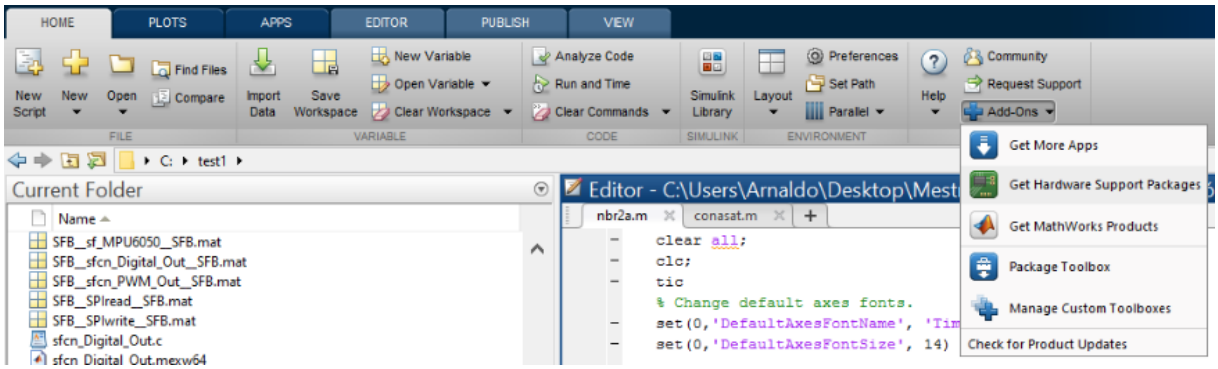

Figura A.1 – Menu

- Instalar da Internet
- Clicar em próximo

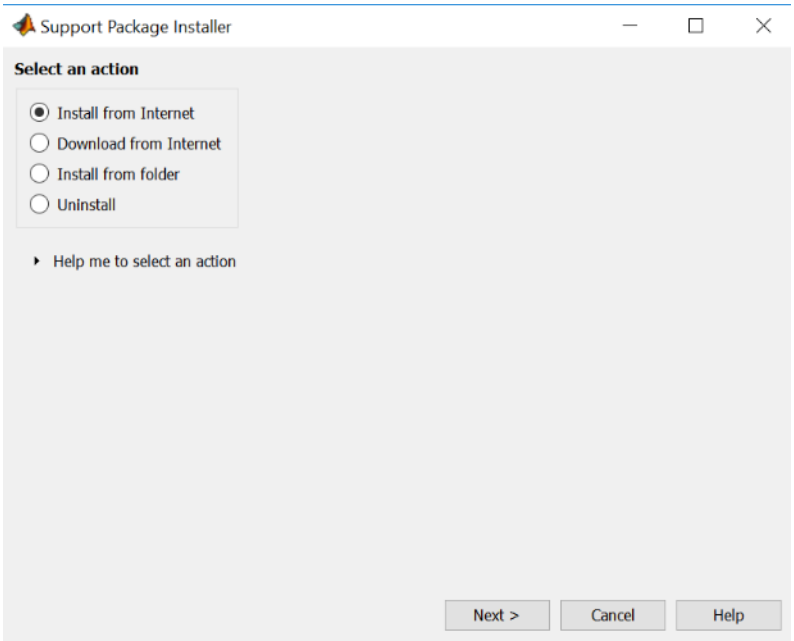

Figura A.2 - Support Pakage

- Selecione Raspberry Pi
- Clicar em Próximo

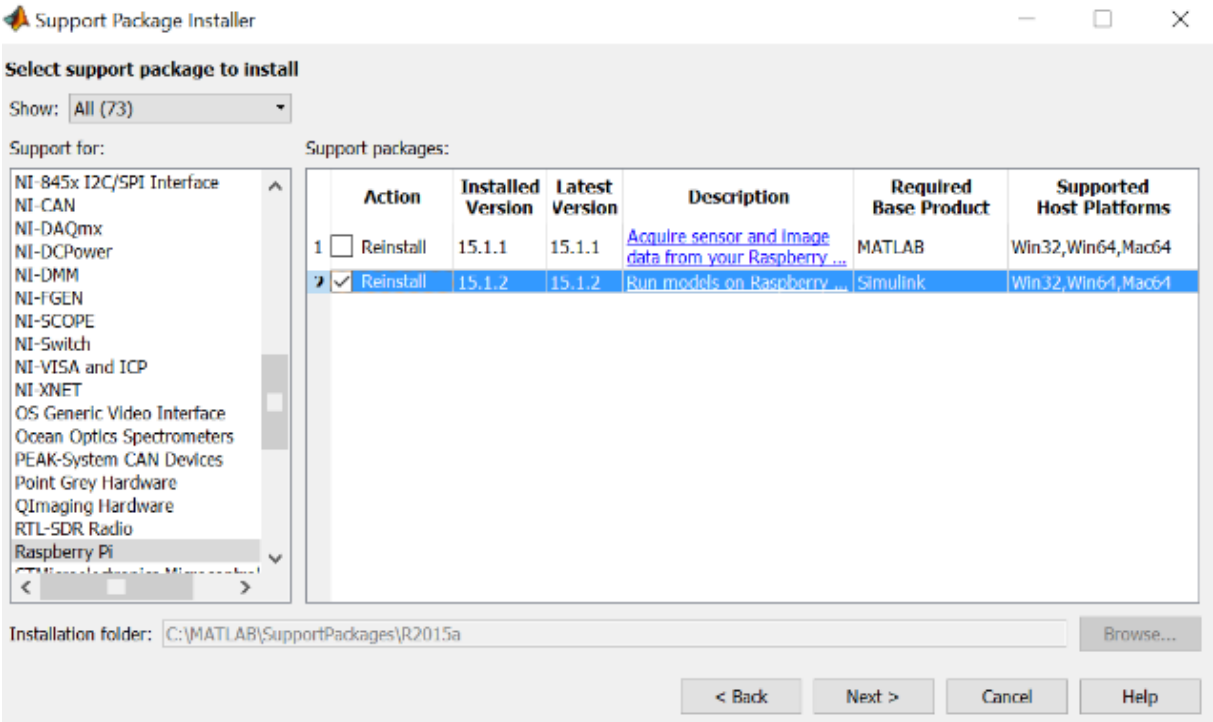

Figura A.3 Seleção dos pacotes

- Aceitar o contrato

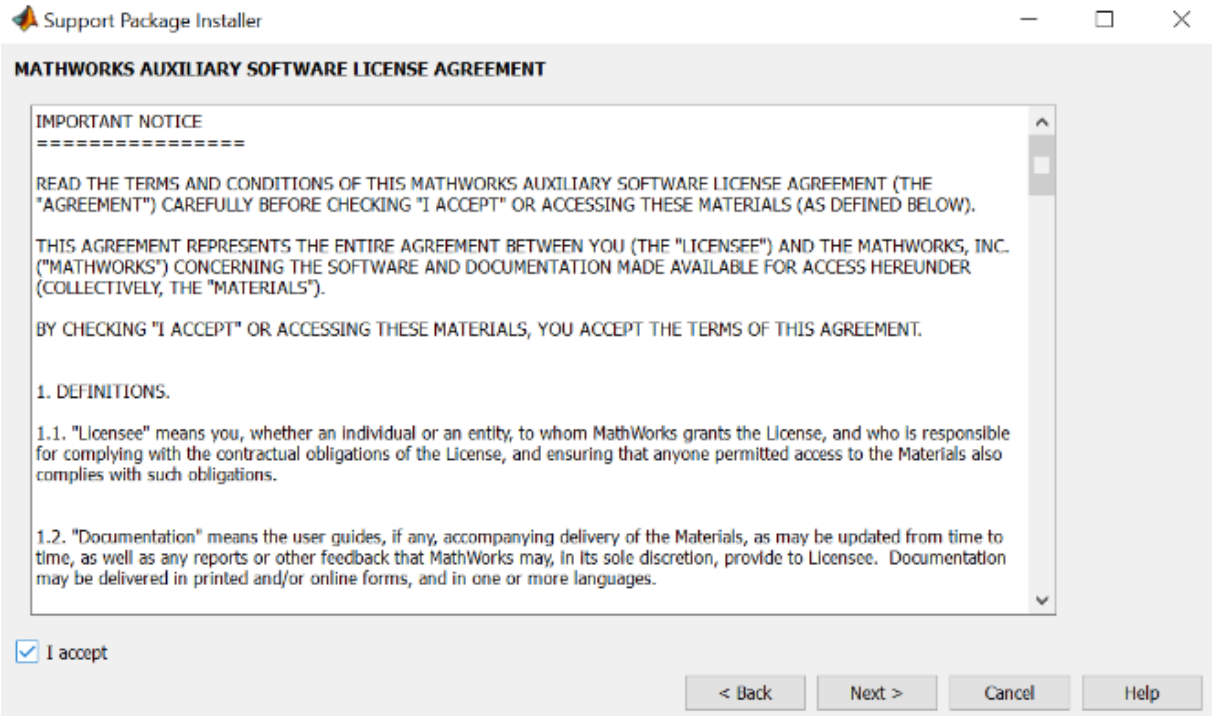

Figura A.4 – Contrato

- Aceitar o contrato de terceiros

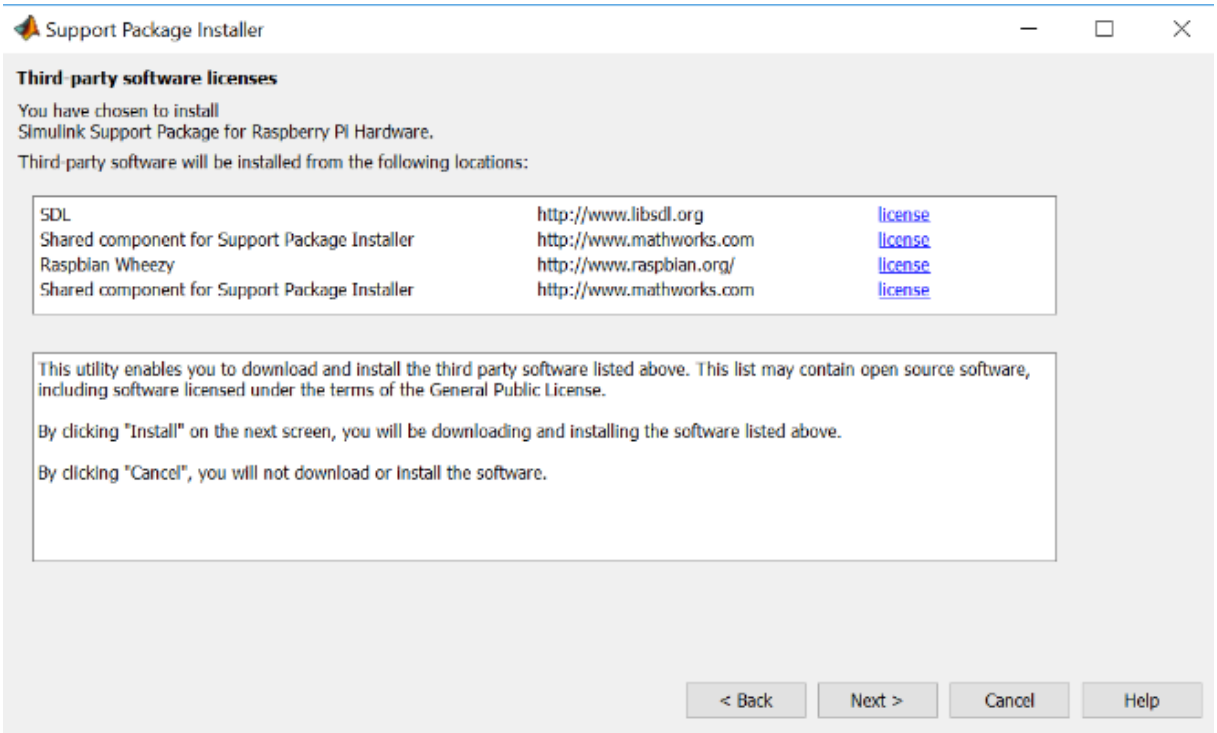

Figura A.5 - Licensa de terceiros

- Confirmar a instalação.

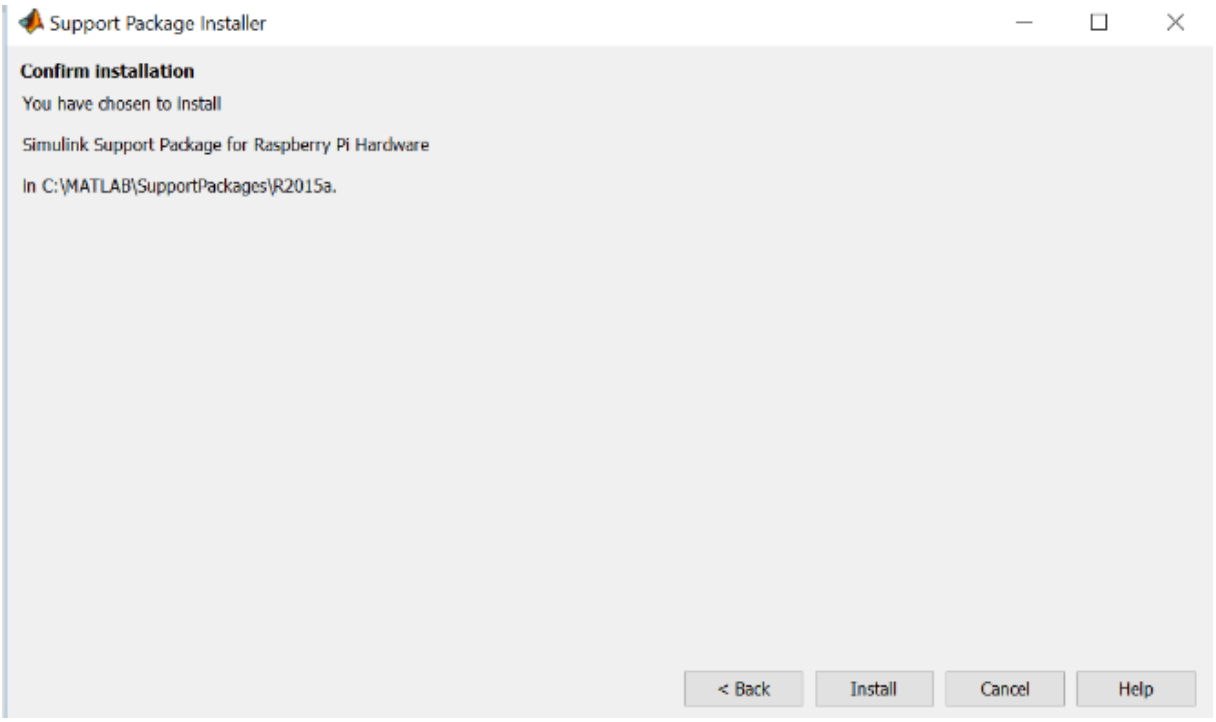

Figura A.6 – Confirmar instalação

- Aguardar o download

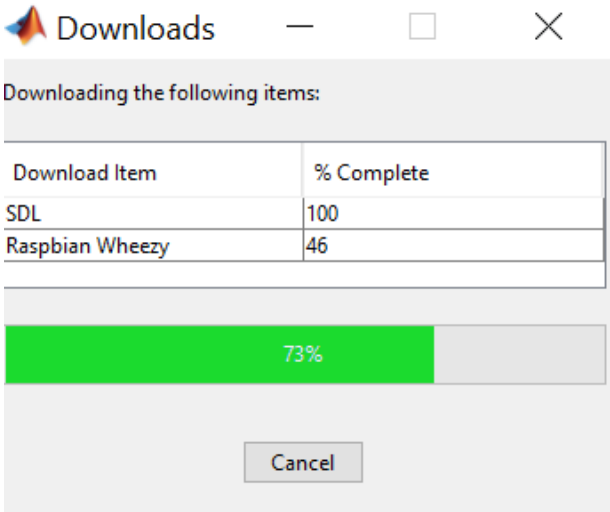

Figura A.7 – Download

- Clicar em continuar

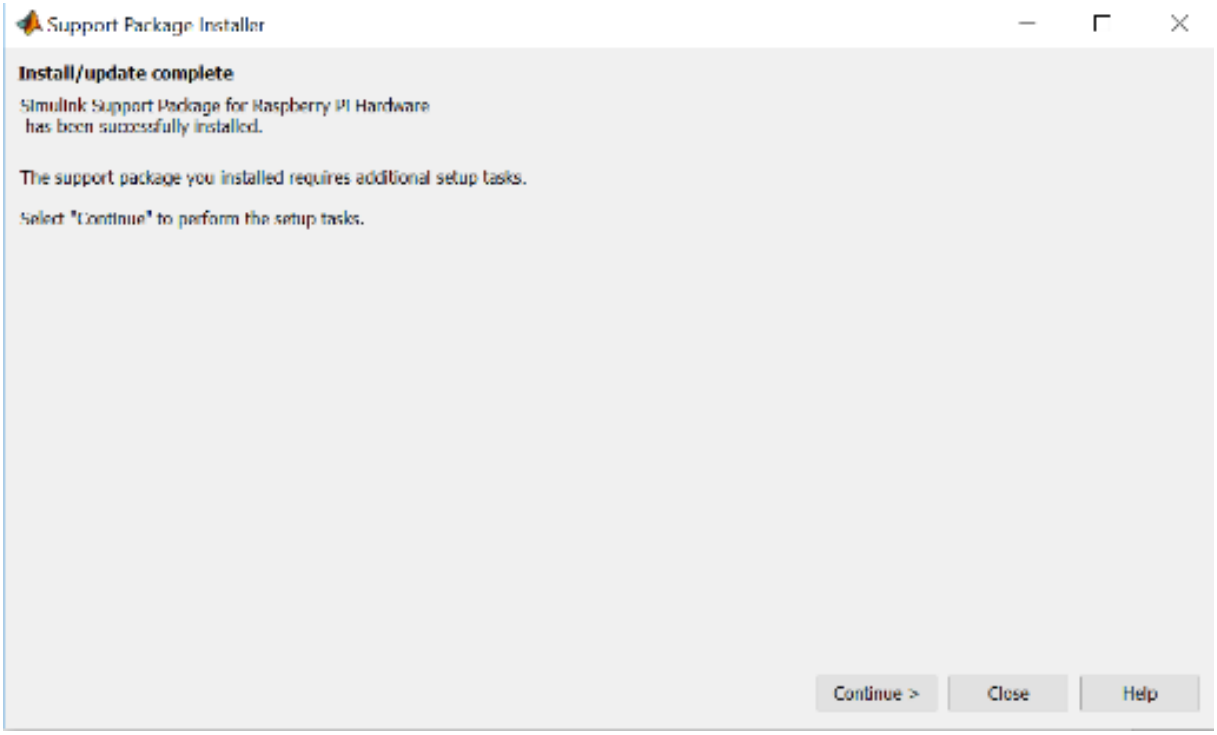

Figura A.8 – Instalação Completa

- Selecionar Raspberry Pi (Simulink) e clicar em continua

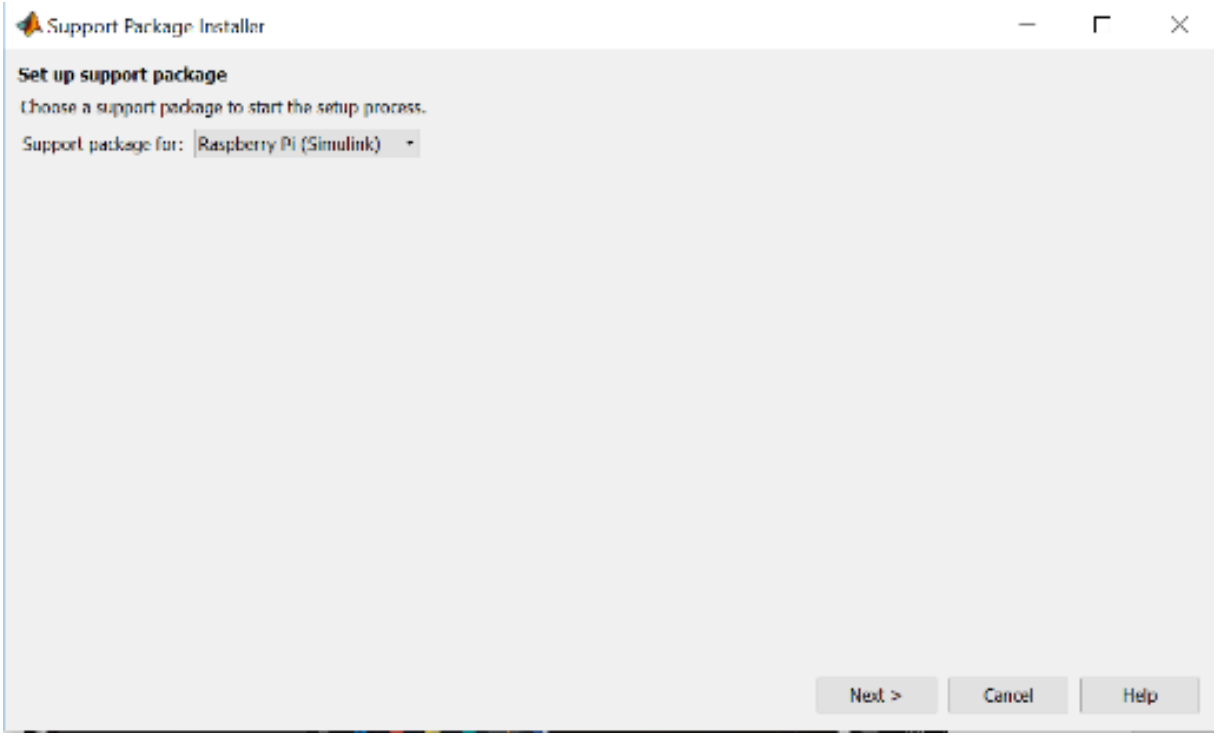

Figura A.9 – Setup

- Selecionar o modelo compatível com a sua placa

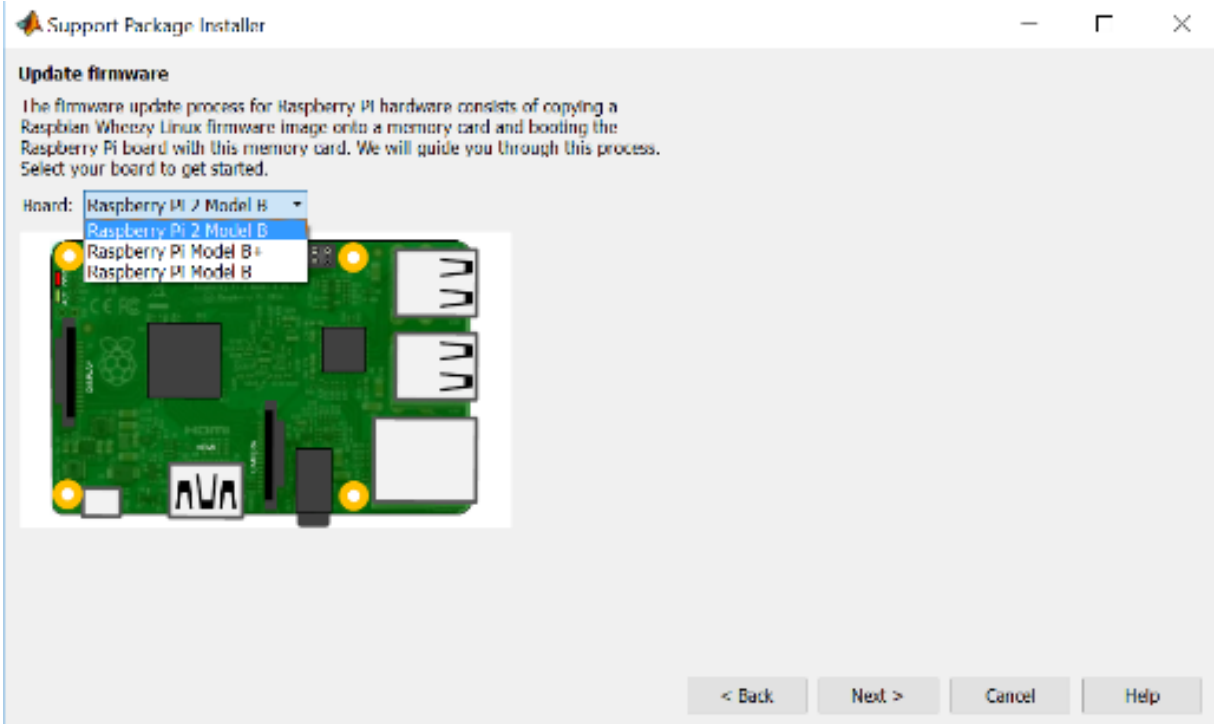

Figura A.10 – Atualizar o Firmware embarcado

- Selecionar o tipo de conexão, sugiro a utilização de um cabo de ethernet com ligação crossover

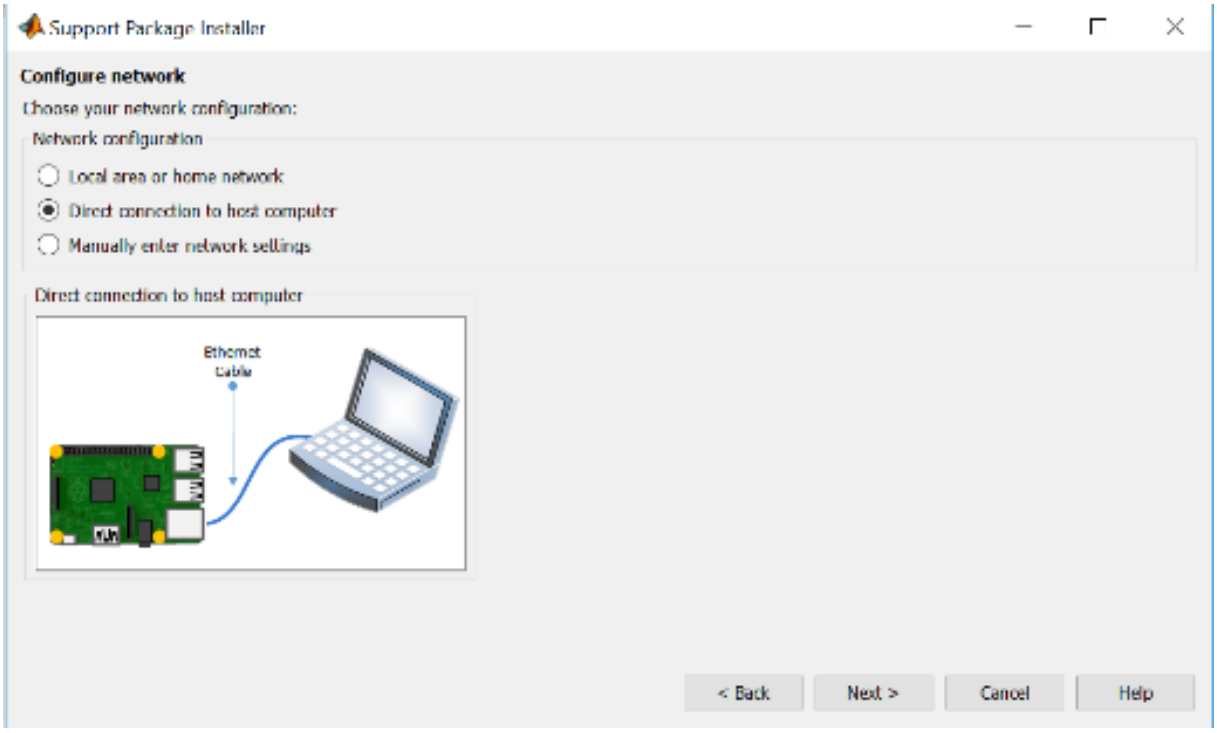

Figura A.11 – Configurar Rede

- Inserir um memory card com capacidade superior a 4 Gb, isto apagará todos os dados presentes no mesmo

Figura A.12 – Selecione o Drive para instalação

- Inserir o memory card e alimentar a placa, após alguns segundos clicar em próximo e aguardar a conclusão das configurações pelo Matlab

Figura A.13 – Conecte o Raspberry

- Após a instalação da imagem padrão, abrir o Simulink, selecionar Ferramentas(Tools), Run on target Hardware e Prepare to run para configurar a placa.

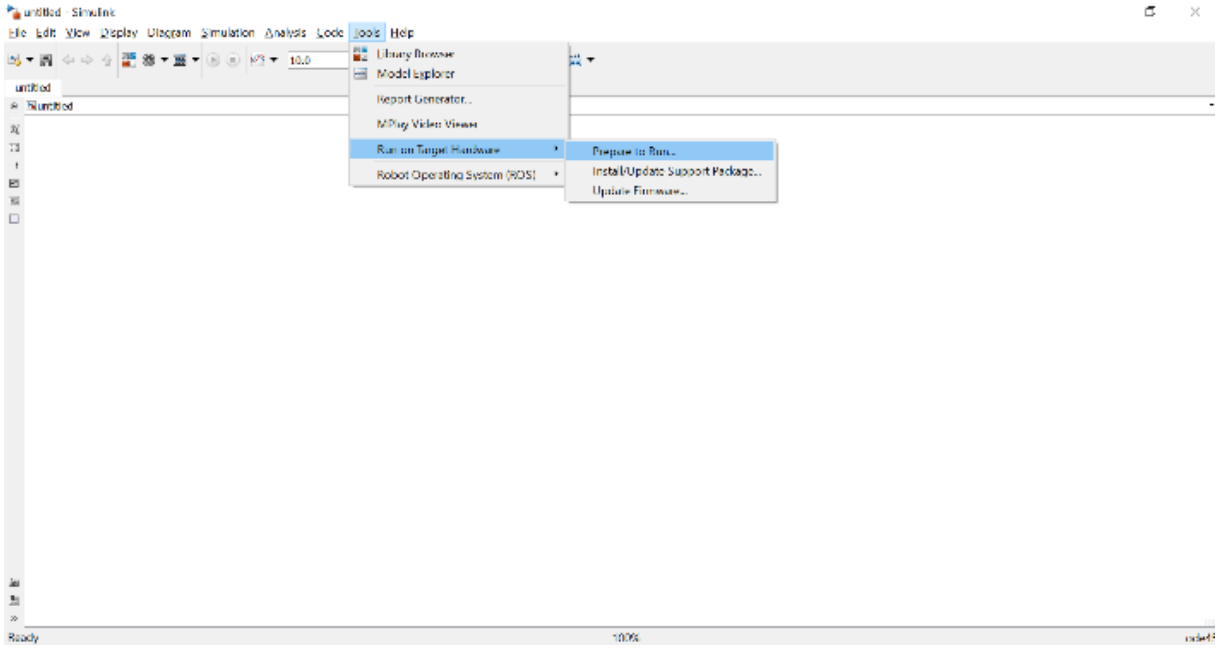

Figura A.14 – Menu

- Selecionar Raspberry Pi

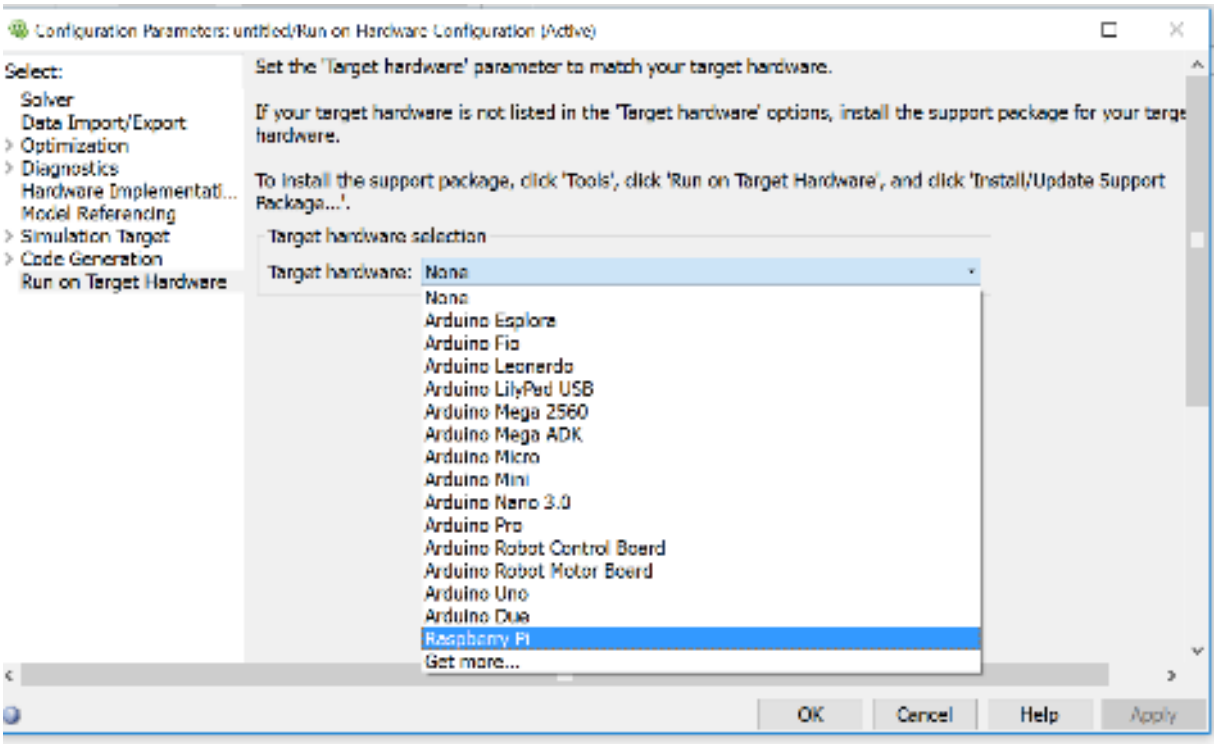

Figura A.15 – Configurações de hardware

- Selecionar *enable overrun detection*, para visualizar se não está ocorrendo colisões / perdas de dados.

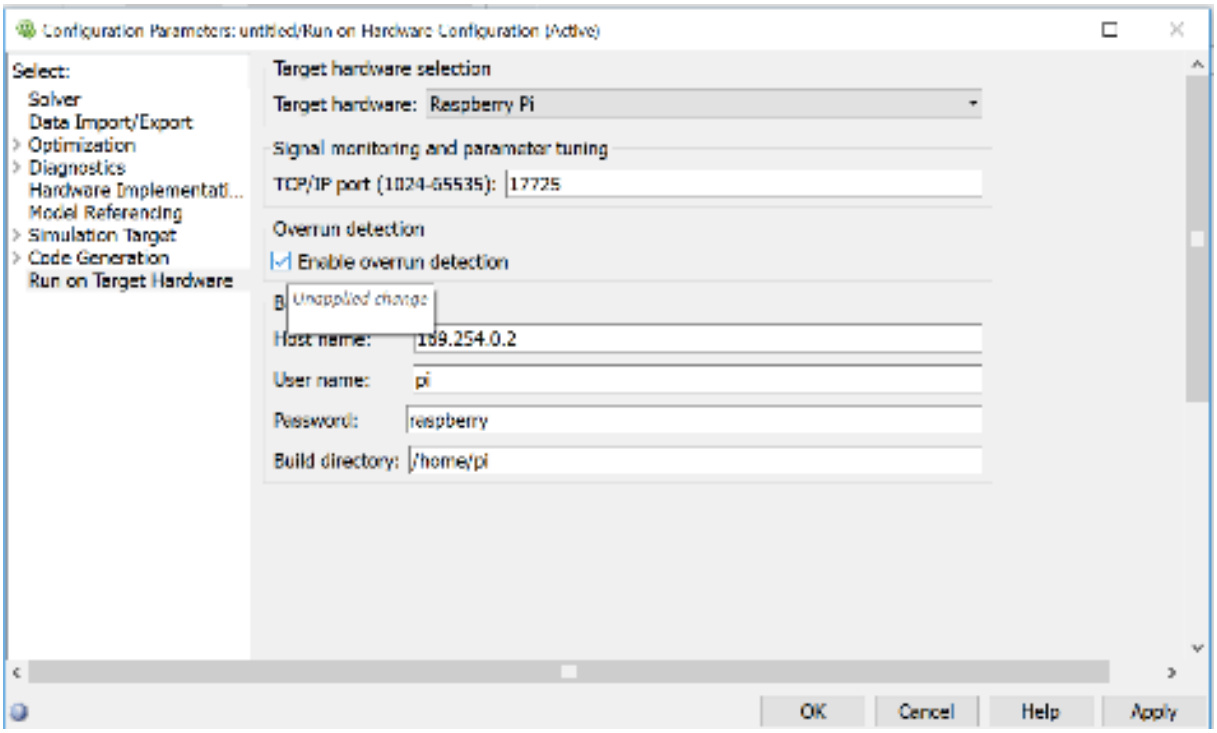

Figura A.16 – Configurações de IP

- Mude o usuário e password para root, necessário para elevar o nível de permissão para Super usuário

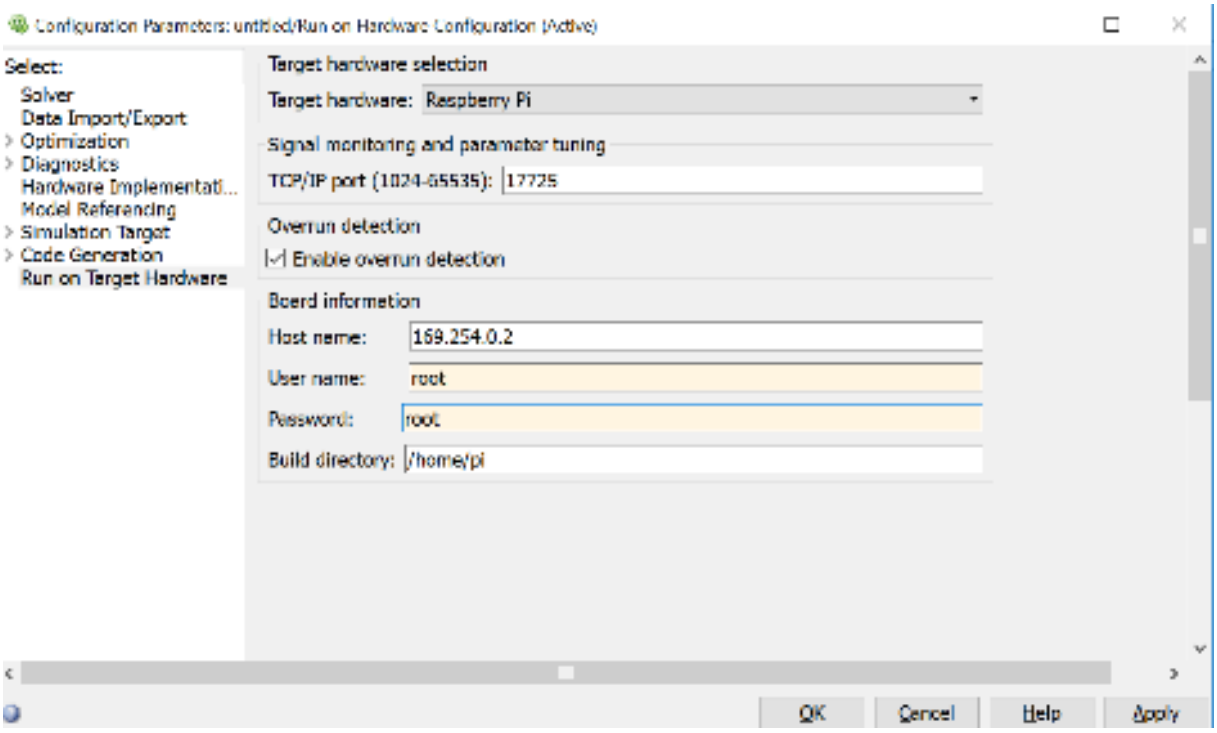

Figura A.17 – Elevação de nível de permissão

- Abrir o exemplo serialwrite
- Mude o tempo de 10 para inf (infinito)

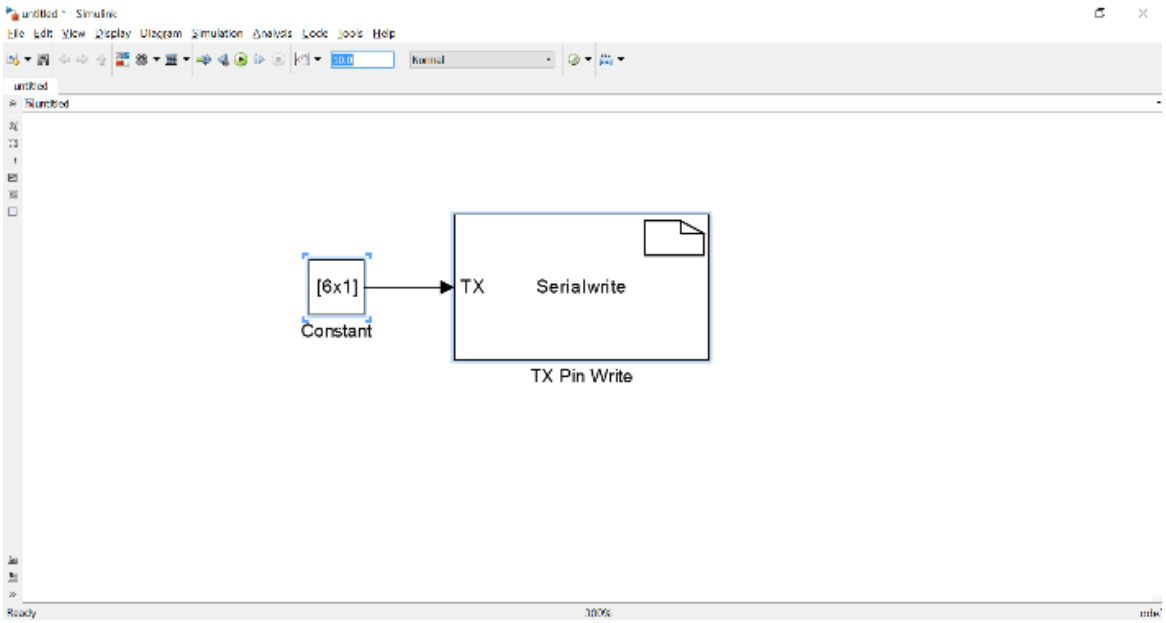

Figura A.18 – Exemplo de aplicação

- Mude o modo de operação de normal para external, desta forma toda a temporização ocorrerá no Raspberry, evitando o tempo de latência do Windows
- Entre no putty

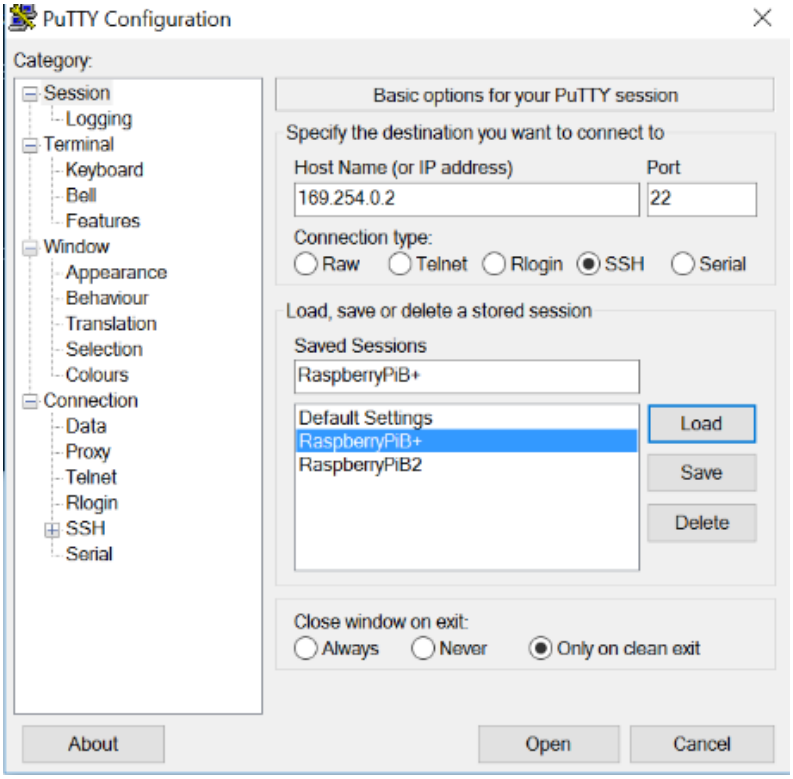

Figura A.19 – Configuração do Putty

- Ip configurado : 169.254.0.2
- Digite sudo password root, para mudar o password do usuário root
- Digite su e o password root para testar se o sistema carregou suas mudanças

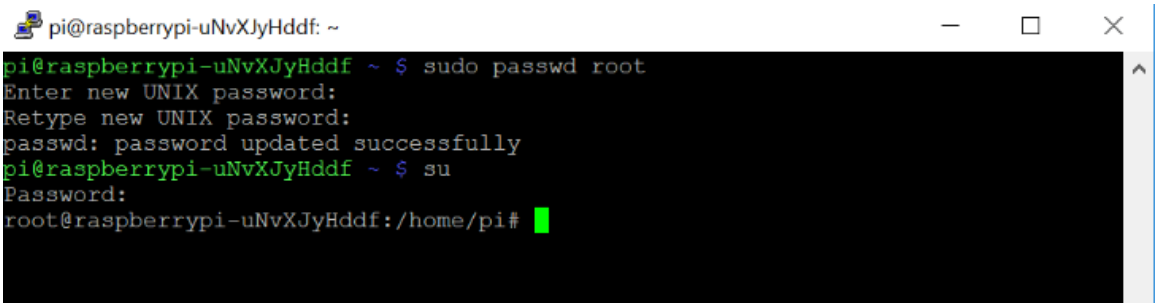

Figura A.20 – Inicializando o sistema

- Clique no botão play para inicializar o sistema
- Se o sistema retornar erro no diagnostico verificar
- Install wiringPi

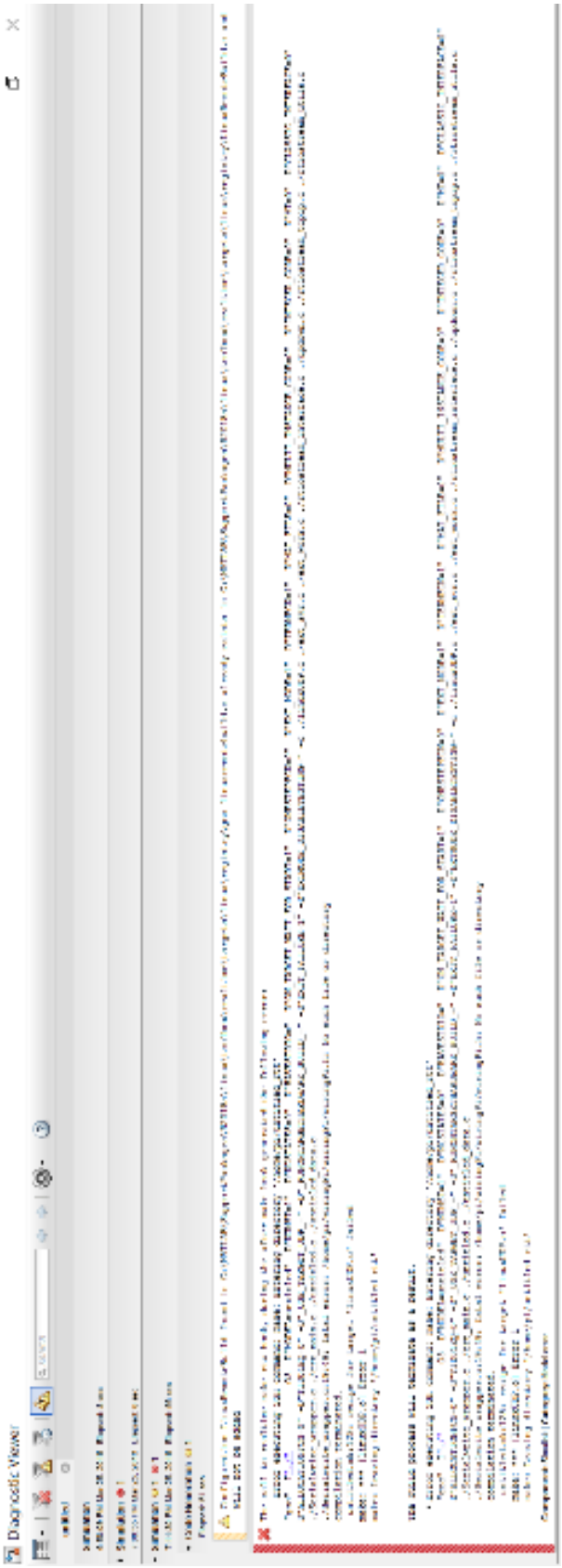

Figura A.21 – Erro no diagnóstico

- Download no seu PC a biblioteca wiringPi, sugestão Projeto Gordons https://projects.drogon.net/raspberry-pi/wiringpi/download-and-install/
- Coloque em um memory card para mover ao Raspberry
- Entre no Putty
- Montar o memory card no Raspberry
- http://www.pkill-9.com/mount-usb-micro-sd-card-linux-automount/
- Digite ls para verificar o que há no diretório
- Digite lsusb para verificar se o Raspberry reconheceu o memory card

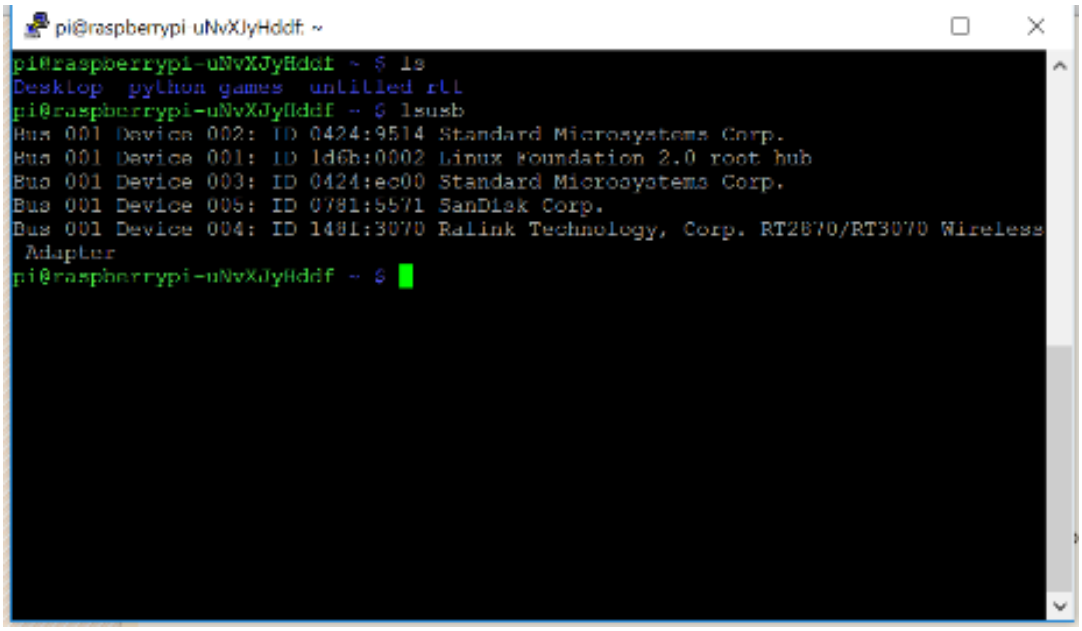

Figura A.22 – Montando uma USB no Linux

- `sudo blkid`
- `sudo mount -t vfat /dev/sda1 /mnt`
- `cd /mnt`
- `ls`
- `cp wiringPi-b0a60c3.tar.gz /home/pi`
- `cd /home/pi`
- `tar -xvzf wiringPi-b0a60c3.tar.gz`
- `mv wiringPi-b0a60c3 wiringPi`
- `cd wiringPi/`
- `sudo ./build`

Leitura sugerida:

http://www.mathworks.com/matlabcentral/fileexchange/54601-raspberry-pi-and-arduino-serial-communication

- Alterar configurações do Makefile no Matlab conforme o Engenheiro John

http://engineer.john-whittington.co.uk/2015/06/simulink-raspberry-pi-driver-blocks/

- Abrir o arquivo no simulink
- Digitar na linha de comando do Matlab :

xmakefilesetup

- Inserir no Linker arguments " -lwiringPi" (Não esquecer do espaço ' ') e clicar em Apply.

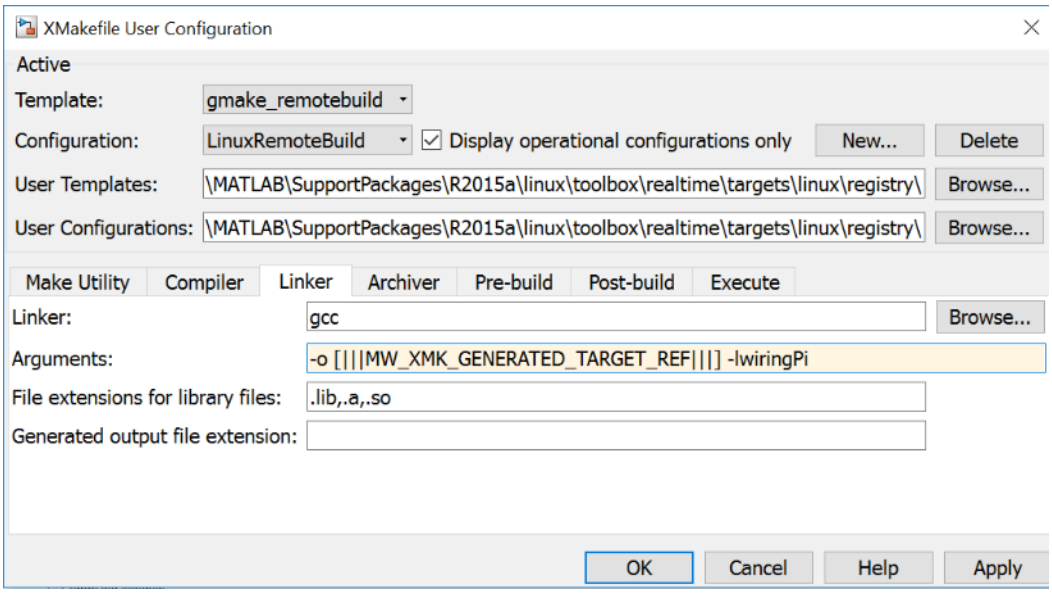

Figura A.23 – Configuração do Debuger

OBS: Por algum motivo eventualmente pode ocorrer a perda desta configuração o que gera o seguinte erro:

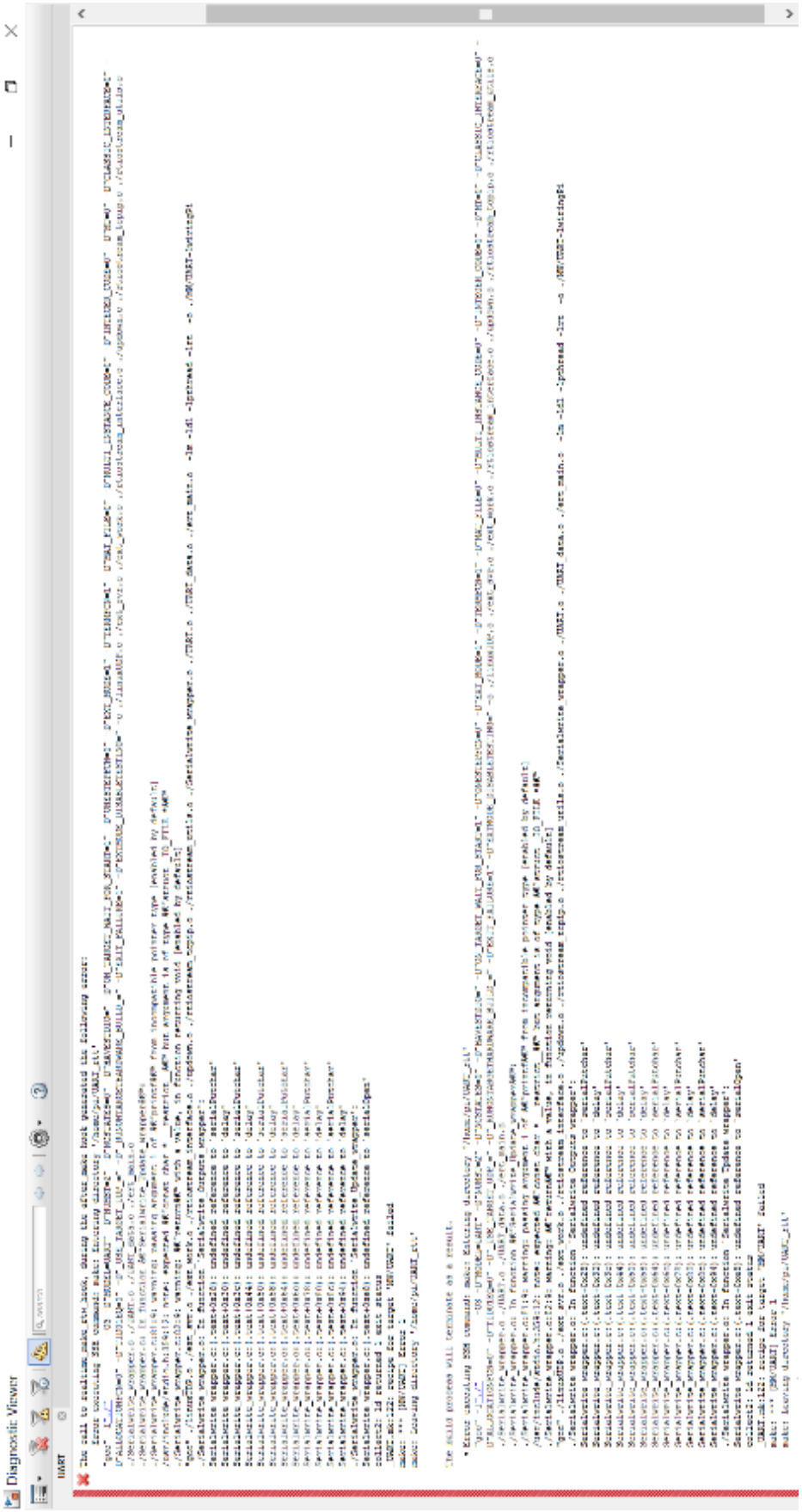

Figura A.24 – Erro de configuração debuger

Erros do tipo abaixo são gerados por não estar na pasta onde encontra-se a biblioteca “propat” (disponível com Valdemir Carrara)

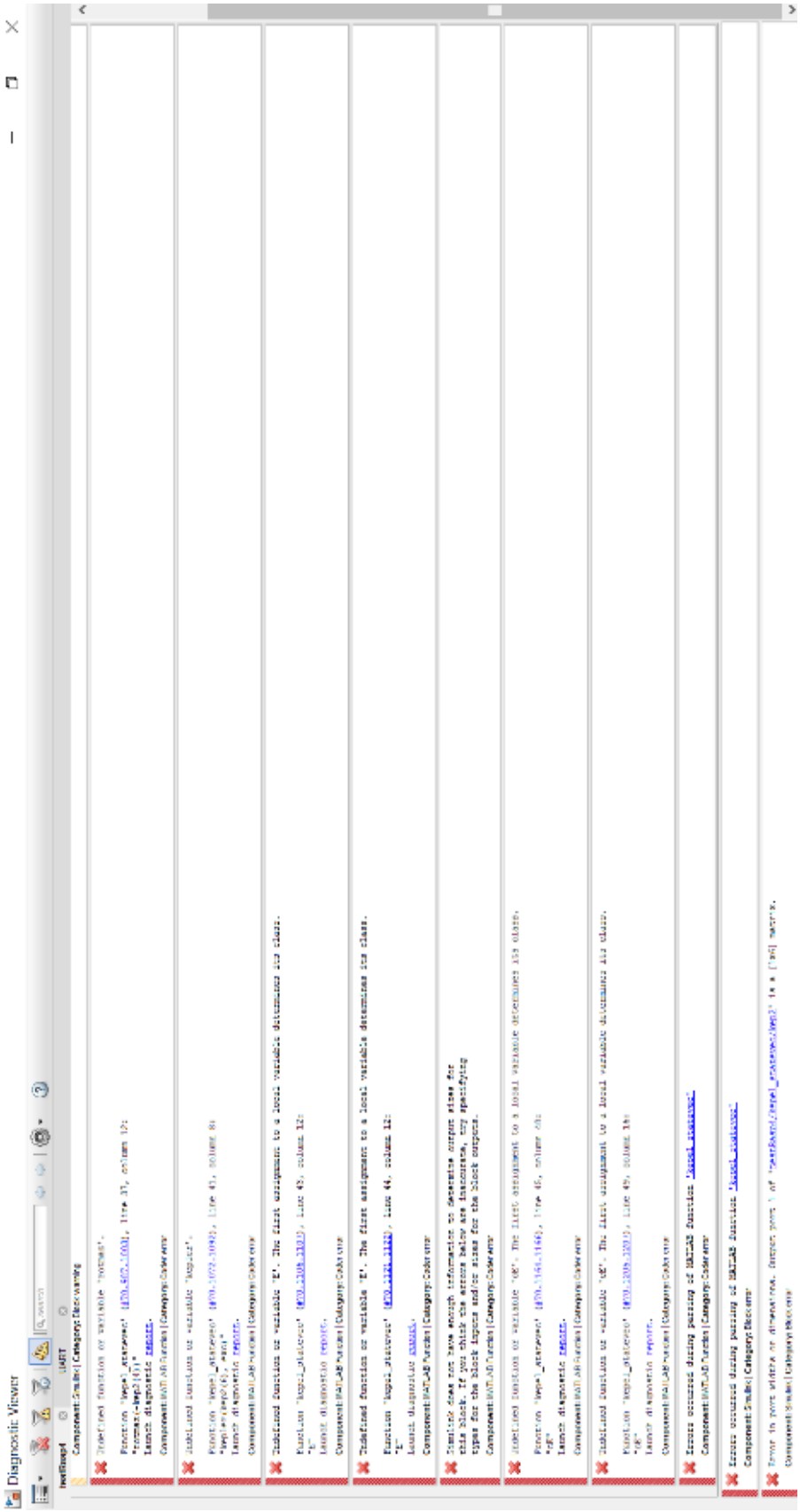

Figura A.25 – Erro de localização da pasta

Para corrigir , ir ao Matlab e alterar o caminho para a pasta onde encontra-se os arquivos da biblioteca Propat

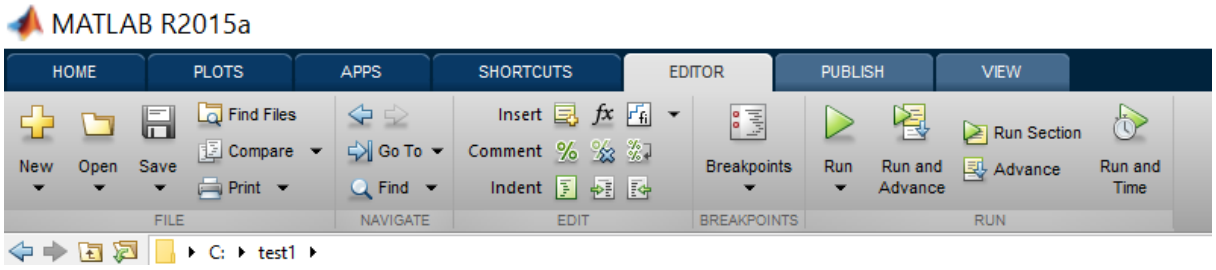

Figura A.26 – Mudando o caminho no Matlab

APENDICE B

Programas do Arduino para leitura de dados da serial e escrita do PWM:

```
/*
*   Leitura de dados da serial e escrita do pwm
*   Arnaldo Oliveira Junior
*   abril de 16
*  livre para utilização desde que cite o autor.
*/
int bytes = 1;
int buffer[6];
int Count = 6;
int outputValue[] = {
  0, 0, 0, 0, 0, 0
};
const int analogOutPin[] = {
  9, 6, 5, 3, 10, 11
};      //

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
}

void loop() {
  if (Serial.available() > 0) {

    // put your main code here, to run repeatedly:
    int bytes = Serial.available();
    for (int i = 0; i < Count; i++) {
      buffer[i] = Serial.read();
      Serial.println(buffer[i]);
      outputValue[i] = map(buffer[i], 0, 255, 0, 255);
      analogWrite(analogOutPin[i], outputValue[i]);
      delay (1);
    }
  }
}
```