



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2013/02.08.16.21-TDI

**ESTUDO DE UM CONTROLADOR DE ATITUDE EM
UM SIMULADOR DE AVIÔNICA MODULAR
INTEGRADA (IMA) APLICADO AO SATÉLITE
AMAZÔNIA-1**

Gitsuzo de Brito Siqueira Tagawa

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Contrôles, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 27 de maio de 2013.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3DGNKP2>>

INPE
São José dos Campos
2013

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@sid.inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):

Presidente:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Maria Tereza Smith de Brito - Serviço de Informação e Documentação (SID)

Luciana Manacero - Serviço de Informação e Documentação (SID)



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2013/02.08.16.21-TDI

**ESTUDO DE UM CONTROLADOR DE ATITUDE EM
UM SIMULADOR DE AVIÔNICA MODULAR
INTEGRADA (IMA) APLICADO AO SATÉLITE
AMAZÔNIA-1**

Gitsuzo de Brito Siqueira Tagawa

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Contrôles, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 27 de maio de 2013.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3DGNKP2>>

INPE
São José dos Campos
2013

Dados Internacionais de Catalogação na Publicação (CIP)

Tagawa, Gitsuzo de Brito Siqueira.
T126e Estudo de um controlador de atitude em um simulador de Avi-
ônica Modular Integrada (IMA) aplicado ao satélite Amazônia-
1 / Gitsuzo de Brito Siqueira Tagawa. – São José dos Campos :
INPE, 2013.
xliv + 257 p. ; (sid.inpe.br/mtc-m19/2013/02.08.16.21-TDI)

Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Contrôles) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2013.

Orientador : Dr. Marcelo Lopes de Oliveira e Souza.

1. Plataforma MultiMissão (PMM). 2. Sistema de Controle de Atitude e Órbita (AOCS). 3. Avionica Modular Integrada (IMA).
I.Título.

CDU 629.7.62.2



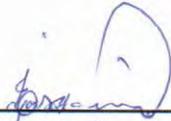
Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

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

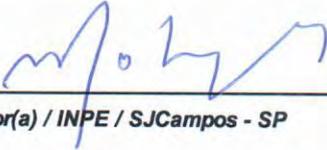
**Engenharia e Tecnologia Espaciais/Mecânica
Espacial e Controle**

Dr. Ijar Milagre da Fonseca



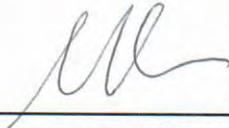
Presidente / INPE / SJC Campos - SP

Dr. Marcelo Lopes de Oliveira e Souza



Orientador(a) / INPE / SJC Campos - SP

Dr. Mario Cesar Ricci



Membro da Banca / INPE / SJC Campos - SP

Dr. Fernando José de Oliveira Moreira



Convidado(a) / EMBRAER / SJC Campos - SP

Dr. Henrique Mohallem Paiva



Convidado(a) / MECTRON / SJC Campos - SP

Este trabalho foi aprovado por:

maioria simples

unanimidade

Aluno (a): **Gitsuzo de Brito Siqueira Tagawa**

São José dos Campos, 27 de Maio de 2013

“A dúvida é o princípio da sabedoria”.

Aristóteles

À minha esposa Monia e a meu filho Kenji pelo apoio e incentivo para que eu completasse este trabalho, além da enorme compreensão pelas muitas horas nas quais estive ausente.

AGRADECIMENTOS

Ao meu orientador Dr. Marcelo Lopes de Oliveira e Souza por todas as lições e incentivos, especialmente nos últimos meses.

Ao INPE, pela oportunidade e infraestrutura necessária para o desenvolvimento deste trabalho.

À Coordenação Acadêmica do DMC e seus docentes, sem os quais nada disso teria sido possível.

À GMV Innovating Solutions, nas pessoas do seu presidente, Dr. Alberto de Pedro Crespo, e dos Engenheiros Tobias Schoofs, e Torre Fernão de Magalhães por disponibilizar o SIMA para nossas simulações.

Ao amigo Eloy Martins de Oliveira Junior pelas muitas horas despendidas me ajudando, principalmente com a programação em C e MatLab.

Aos amigos da Embraer, Eng. José Ricardo Parizi e Marcos Antônio de Moraes Brito, por disponibilizarem seu tempo me orientando e aconselhando e, Eng. Marcos Ortiz pelas dicas valiosas, no início deste trabalho.

Ao amigo Rafael Lopes Costa pelos puxões de orelha dia após dia para que eu finalizasse esse trabalho.

À amiga Paula Cristiane Pinto Mesquita Pardal pelos valiosos conselhos.

Ao meu chefe no LIT, Eng. Mauro Kakizaki por me disponibilizar um tempo para que eu pudesse me dedicar, especialmente durante os últimos meses deste trabalho.

Aos colegas de trabalho Eng. Adriano Moura e Eng. Paulo Vinicius Jeronimo que, durante a fase final deste trabalho, me pouparam de diversos contratemplos.

Aos demais amigos que porventura eu possa ter me esquecido de mencionar aqui, mas que não são menos importantes.

RESUMO

O conceito IMA (*Integrated Modular Avionics*) tem sido alvo de estudo em centros de renome como NASA e ESA, em grandes empresas como Boeing e Airbus etc. Com intuito de explorar esse conceito, este trabalho modela, implementa e simula um Sistema de Controle de Atitude (SCA) compatível com o modo nominal da Plataforma MultiMissão (PMM) do Instituto Nacional de Pesquisas Espaciais (INPE) em um Simulador de IMA (SIMA). As simulações iniciam-se com um sistema simples, o Massa-Mola-Amortecedor (MMA). Em seguida, é feita a modelagem, implementação e simulação do SCA. Todas as simulações são feitas em MatLab, linguagem C e SIMA, e os valores são comparados e concordam entre si. Além disso, simularam-se ambos os sistemas, MMA e SCA, em um mesmo módulo, dividido em seis partições e, inseriu-se uma falha em uma das partições do MMA. Apesar da falha desta partição, as demais partições do SCA continuaram seu funcionamento normalmente. Os resultados demonstram a viabilidade no uso da plataforma IMA e provam que o particionamento baseado no conceito IMA é robusto.

STUDY OF AN ATTITUDE CONTROLLER IN A SIMULATED INTEGRATED MODULAR AVIONICS (SIMA) APPLIED TO AMAZÔNIA-1 SATELLITE

ABSTRACT

The IMA (Integrated Modular Avionics) concept has been studied at NASA and ESA, in big companies such as Boeing and Airbus etc. In order to explore this concept, this work models, implements and simulates an Attitude Control System (ACS) compatible with the nominal mode of the MultiMission Platform (MMP) of INPE (National Institute for Space Research) in a Simulated Integrated Modular Avionics (SIMA). The simulations begin with a simple system, the Mass-Spring-Damper (MSD). After that, the modeling, implementation and simulation of the ACS are done. All the simulations are performed in MatLab, C language and SIMA, and their results are compared and agreed one with each other. Moreover, both systems, MSD and ACS, are simulated within the same module, divided in 6 partitions, and a fault was introduced in one of the MSD partitions. Despite the fault within this partition, the partitions, the ACS partitions remaining kept working properly. The results demonstrate the viability of the use of the IMA platform, and prove that the partitioning based on IMA concept is robust.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 2.1 - PMM (configuração em órbita).....	6
Figura 2.2 - Visão explodida da PMM (Carga útil).....	7
Figura 2.3 - Satélite Amazônia-1 (PMM+Módulo de Serviço).....	8
Figura 2.4 - Visão explodida do módulo de serviço.....	9
Figura 2.5 - Sensor de Estrelas ASTRO 15	12
Figura 2.6 - Sensor de Estrelas VST-41M.....	13
Figura 2.7 - Sensor Solar	14
Figura 2.8 - Sensor Solar	15
Figura 2.9 - Sensor de Horizonte	16
Figura 2.10 - Sensor de Horizonte	16
Figura 2.11 - Magnetômetro HMR2300.....	17
Figura 2.12 - Giroscópio mecânico convencional.....	18
Figura 2.13 - Giroscópio de Fibra Óptica	19
Figura 2.14 - Constelação dos satélites do sistema GPS	20
Figura 2.15 - Jato Propulsor	21
Figura 2.16 - Jato Propulsor	21
Figura 2.17 – Desenhos de Rodas de Reação.....	22
Figura 2.18 - Rodas de Reação	22
Figura 2.19 - Roda de Reação	23
Figura 2.20 - Bobina Eletromagnética	24
Figura 2.21 - Bobina Eletromagnética VMT-35	24
Figura 2.22 - Sistema de controle automático.....	25
Figura 2.23 - Arquitetura Federada	26
Figura 2.24 - Arquitetura Integrada	27
Figura 2.25 - Partições funcionando como múltiplos computadores virtuais....	28
Figura 3.1 - Agendador de Partições.....	30
Figura 3.2 - Sistema padrão ARINC 653.....	31

Figura 3.3 – Diagrama de estados do processo.....	33
Figura 3.4 - (a) Unicast; (b) Multicast; (c) Broadcast.....	35
Figura 4.1 - Subsistema de Controle de Atitude e Órbita do satélite Amazônia-1	41
Figura 4.2 - Modos de controle do SCAO do satélite Amazônia-1	43
Figura 5.1 – Tela do MatLab	50
Figura 5.2 – Entrando no modo ROOT	52
Figura 5.3 – Pasta do SIMA dentro da pasta <i>usr</i>	53
Figura 5.4 – Alteração do PATH do Linux.....	54
Figura 5.5 - Inclusão do SIMA_ROOT.....	55
Figura 5.6- Exemplos simples do SIMA	56
Figura 5.7 – Exemplos complexos do SIMA.....	57
Figura 6.0 – Diagrama de Blocos do sistema Massa-Mola-Amortecedor.....	60
Figura 6.1 – Massa-Mola-Amortecedor.	61
Figura 6.2 - Diagrama de Blocos em malha aberta.	63
Figura 6.3 - Diagrama de Blocos em malha fechada.	64
Figura 6.4- Resposta a um degrau unitário de um sistema de 2 ^a ordem subamortecido e sem zeros	65
Figura 6.5 – Sistema Massa-Mola-Amortecedor dividido em duas partições... ..	72
Figura 6.6 - Representação das portas e canais de comunicação do MMA no SIMA	73
Figura 6.7 - Representação da comunicação entre as partições do sistema MMA.....	74
Figura 6.8 - Configuração no arquivo a653.xml da partição 1	75
Figura 6.9 - Configuração no arquivo sima.xml da partição 1	76
Figura 6.10 - Configuração no arquivo a653.xml da partição 2.....	77
Figura 6.11 - Configuração no arquivo sima.xml da partição 2	78
Figura 6.12 – Agendamento no tempo das partições do sistema MMA	79
Figura 6.13 - Configuração no arquivo a653.xml das partições 1 e 2	79
Figura 6.14 – Sistema de Coordenadas Celestes.....	81
Figura 6.15 - Referencial Orbital.	83

Figura 6.16 - Referencial do Satélite.....	84
Figura 6.17 - Diagrama de Blocos do AOCS do satélite Amazônia-1	85
Figura 6.18 - Rotação em torno do eixo X.....	87
Figura 6.19 - Rotação em torno do eixo Y.....	89
Figura 6.20 - Rotação em torno do eixo Z.....	90
Figura 6.21 - Referencial Orbital (O) e Referencial do Satélite (S).....	92
Figura 6.22 - Sequência de Rotações 3-2-1 dos respectivos ângulos (ψ , θ , ϕ)	94
Figura 6.23 - Rotação em torno do eixo \hat{e} de um ângulo	96
Figura 6.24 - Vetor em um sistema de coordenadas girante.....	98
Figura 6.25 – Relação de i e $d\hat{i}$	99
Figura 6.26 – Sistema de coordenadas inerciais.....	101
Figura 6.27 - Aproximação linear da curva característica do servomotor e diagrama de blocos correspondente	105
Figura 6.28 - Diagrama de Blocos (Roll)	107
Figura 6.29 - Diagrama de Blocos (Pitch)	108
Figura 6.30 - Diagrama de Blocos (Yaw)	108
Figura 6.31 – Fluxograma do funcionamento do programa.....	112
Figura 6.32 - AOCS dividido em quatro partições	113
Figura 6.33 - Representação das portas e canais de comunicação do AOCS no SIMA	114
Figura 6.34 - Representação da comunicação entre as partições	115
Figura 6.35 – Configuração no arquivo a653.xml da Partição 1.....	116
Figura 6.36 - Configuração no arquivo sima.xml da Partição 1	117
Figura 6.37 - Configuração no arquivo a653.xml da Partição 2	118
Figura 6.38 - Configuração no arquivo sima.xml da Partição 2.....	119
Figura 6.39 - Configuração no arquivo a653.xml da Partição 3	120
Figura 6.40 - Configuração no arquivo sima.xml da Partição 3.....	121
Figura 6.41 - Configuração no arquivo a653.xml da Partição 4	122
Figura 6.42 - Configuração no arquivo sima.xml da Partição 4.....	123
Figura 6.43 - Agendamento no tempo das partições do AOCS.....	124

Figura 6.44 – Configuração de agendamento no arquivo a653.xml das Partições 1 a 4	125
Figura 6.45 - Fluxograma do funcionamento do programa	126
Figura 6.46 – SCA e MMA dividido em seis partições.....	127
Figura 6.47 - Representação das portas e canais de comunicação do SCA .	128
Figura 6.48 - Representação das portas e canais de comunicação do MM...	129
Figura 6.49 - Representação da comunicação entre as partições	130
Figura 6.50 - Representação da comunicação entre as partições	131
Figura 6.51 – Configuração no arquivo a653.xml da Partição 1.....	133
Figura 6.52 - Configuração no arquivo sima.xml da Partição 1	134
Figura 6.53 - Configuração no arquivo a653.xml da Partição 2	135
Figura 6.54 - Configuração no arquivo sima.xml da Partição 2.....	136
Figura 6.55 - Configuração no arquivo a653.xml da Partição 3	137
Figura 6.56 - Configuração no arquivo sima.xml da Partição 3.....	138
Figura 6.57 - Configuração no arquivo a653.xml da Partição 4	139
Figura 6.58 - Configuração no arquivo sima.xml da Partição 4.....	140
Figura 6.59 - Configuração no arquivo a653.xml da Partição 5	141
Figura 6.60 - Configuração no arquivo sima.xml da Partição 5.....	142
Figura 6.61 - Configuração no arquivo a653.xml da Partição 6	143
Figura 6.62 - Configuração no arquivo sima.xml da Partição 6.....	144
Figura 6.63 - Agendamento no tempo das partições do SCA e MM	145
Figura 6.64 – Configuração de agendamento no arquivo a653.xml (Partições 1 a 6).....	146
Figura 7.1 – Curvas dos resultados dos programas simulados em linguagem C, MatLab e SIMA.....	148
Figura 7.2 – Curvas dos resultados dos programas simulados em linguagem C e SIMA	149
Figura 7.3 – Diferença das curvas dos programas simulados em linguagem C e SIMA	149
Figura 7.4 - Curvas dos resultados dos programas simulados em MatLab e SIMA	150

Figura 7.5 - Diferença das curvas dos programas simulados em MatLab e SIMA	150
Figura 7.6 – Curva do ângulo psi utilizando passo de integração de 20 segundos.....	151
Figura 7.7 - Curva do ângulo psi utilizando passo de integração de 10 segundos.....	152
Figura 7.8 - Curva do ângulo psi utilizando passo de integração de 5 segundos	152
Figura 7.9 - Curva do ângulo psi utilizando passo de integração de 1 segundo	153
Figura 7.10 - Curva do ângulo psi utilizando passo de integração de 0.6 segundos.....	153
Figura 7.11 - Curva do ângulo psi utilizando passo de integração de 0.4 segundos.....	154
Figura 7.12 - Curva do ângulo psi utilizando passo de integração de 0.01 segundos.....	154
Figura 7.13 – Tempo do SIMA (ts) e tempo do incremento do passo de integração (t)	156
Figura 7.14 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	157
Figura 7.15 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	157
Figura 7.16 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	158
Figura 7.17 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	158
Figura 7.18 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	159
Figura 7.19 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	159

Figura 7.20 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	160
Figura 7.21 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	160
Figura 7.22 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	161
Figura 7.23 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	161
Figura 7.24 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	162
Figura 7.25 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	162
Figura 7.26 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	163
Figura 7.27 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	163
Figura 7.28 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)	164
Figura 7.29 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	166
Figura 7.30 - Diferença das curvas dos programas simulados em linguagem C e MatLab	166
Figura 7.31 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	167
Figura 7.32 - Diferença das curvas dos programas simulados em linguagem C e MatLab	167
Figura 7.33 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	168
Figura 7.34 - Diferença das curvas dos programas simulados em linguagem C e MatLab	168

Figura 7.35 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	169
Figura 7.36 - Diferença das curvas dos programas simulados em linguagem C e MatLab	169
Figura 7.37 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	170
Figura 7.38 - Diferença das curvas dos programas simulados em linguagem C e MatLab	170
Figura 7.39 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	171
Figura 7.40 - Diferença das curvas dos programas simulados em linguagem C e MatLab	171
Figura 7.41 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	172
Figura 7.42 - Diferença das curvas dos programas simulados em linguagem C e MatLab	172
Figura 7.43 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	173
Figura 7.44 - Diferença das curvas dos programas simulados em linguagem C e MatLab	173
Figura 7.45 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	174
Figura 7.46 - Diferença das curvas dos programas simulados em linguagem C e MatLab	174
Figura 7.47 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	175
Figura 7.48 - Diferença das curvas dos programas simulados em linguagem C e MatLab	175
Figura 7.49 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	176

Figura 7.50 - Diferença das curvas dos programas simulados em linguagem C e MatLab	176
Figura 7.51 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	177
Figura 7.52 - Diferença das curvas dos programas simulados em linguagem C e MatLab	177
Figura 7.53 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	178
Figura 7.54 - Diferença das curvas dos programas simulados em linguagem C e MatLab	178
Figura 7.55 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	179
Figura 7.56 - Diferença das curvas dos programas simulados em linguagem C e MatLab	179
Figura 7.57 - Curvas dos resultados dos programas simulados em linguagem C e MatLab	180
Figura 7.58- Diferença das curvas dos programas simulados em linguagem C e MatLab	180
Figura 7.59 - Curvas dos resultados dos programas simulados em SIMA e MatLab	181
Figura 7.60 - Diferença das curvas dos programas simulados em SIMA e MatLab	182
Figura 7.61 - Curvas dos resultados dos programas simulados em SIMA e MatLab	182
Figura 7.62 - Diferença das curvas dos programas simulados em SIMA e MatLab	183
Figura 7.63 - Curvas dos resultados dos programas simulados em SIMA e MatLab	183
Figura 7.64 - Diferença das curvas dos programas simulados em SIMA e MatLab	184

Figura 7.65 - Curvas dos resultados dos programas simulados em SIMA e MatLab	184
Figura 7.66 - Diferença das curvas dos programas simulados em SIMA e MatLab	185
Figura 7.67 - Curvas dos resultados dos programas simulados em SIMA e MatLab	185
Figura 7.68 - Diferença das curvas dos programas simulados em SIMA e MatLab	186
Figura 7.69 - Curvas dos resultados dos programas simulados em SIMA e MatLab	186
Figura 7.70 - Diferença das curvas dos programas simulados em SIMA e MatLab	187
Figura 7.71 - Curvas dos resultados dos programas simulados em SIMA e MatLab	187
Figura 7.72 - Diferença das curvas dos programas simulados em SIMA e MatLab	188
Figura 7.73 - Curvas dos resultados dos programas simulados em SIMA e MatLab	188
Figura 7.74 - Diferença das curvas dos programas simulados em SIMA e MatLab	189
Figura 7.75 - Curvas dos resultados dos programas simulados em SIMA e MatLab	189
Figura 7.76 - Diferença das curvas dos programas simulados em SIMA e MatLab	190
Figura 7.77 - Curvas dos resultados dos programas simulados em SIMA e MatLab	190
Figura 7.78 - Diferença das curvas dos programas simulados em SIMA e MatLab	191
Figura 7.79 - Curvas dos resultados dos programas simulados em SIMA e MatLab	191

Figura 7.80 - Diferença das curvas dos programas simulados em SIMA e MatLab	192
Figura 7.81 - Curvas dos resultados dos programas simulados em SIMA e MatLab	192
Figura 7.82 - Diferença das curvas dos programas simulados em SIMA e MatLab	193
Figura 7.83 - Curvas dos resultados dos programas simulados em SIMA e MatLab	193
Figura 7.84 - Diferença das curvas dos programas simulados em SIMA e MatLab	194
Figura 7.85 - Curvas dos resultados dos programas simulados em SIMA e MatLab	194
Figura 7.86 - Diferença das curvas dos programas simulados em SIMA e MatLab	195
Figura 7.87 - Curvas dos resultados dos programas simulados em SIMA e MatLab	195
Figura 7.88 - Diferença das curvas dos programas simulados em SIMA e MatLab	196
Figura 7.89 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	197
Figura 7.90- Diferença das curvas dos programas simulados em SIMA e linguagem C	197
Figura 7.91 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	198
Figura 7.92- Diferença das curvas dos programas simulados em SIMA e linguagem C	198
Figura 7.93 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	199
Figura 7.94- Diferença das curvas dos programas simulados em SIMA e linguagem C	199

Figura 7.95 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	200
Figura 7.96- Diferença das curvas dos programas simulados em SIMA e linguagem C	200
Figura 7.97 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	201
Figura 7.98 - Diferença das curvas dos programas simulados em SIMA e linguagem C	201
Figura 7.99 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	202
Figura 7.100- Diferença das curvas dos programas simulados em SIMA e linguagem C	202
Figura 7.101 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	203
Figura 7.102 - Diferença das curvas dos programas simulados em SIMA e linguagem C	203
Figura 7.103 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	204
Figura 7.104 - Diferença das curvas dos programas simulados em SIMA e linguagem C	204
Figura 7.105 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	205
Figura 7.106- Diferença das curvas dos programas simulados em SIMA e linguagem C	205
Figura 7.107 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	206
Figura 7.108- Diferença das curvas dos programas simulados em SIMA e linguagem C	206
Figura 7.109 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	207

Figura 7.110- Diferença das curvas dos programas simulados em SIMA e linguagem C	207
Figura 7.111 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	208
Figura 7.112 - Diferença das curvas dos programas simulados em SIMA e linguagem C	208
Figura 7.113 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	209
Figura 7.114- Diferença das curvas dos programas simulados em SIMA e linguagem C	209
Figura 7.115 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	210
Figura 7.116- Diferença das curvas dos programas simulados em SIMA e linguagem C	210
Figura 7.117 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	211
Figura 7.118- Diferença das curvas dos programas simulados em SIMA e linguagem C	211
Figura 7.119 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	213
Figura 7.200- Diferença das curvas dos programas simulados em SIMA e linguagem C	213
Figura 7.201 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	214
Figura 7.202- Diferença das curvas dos programas simulados em SIMA e linguagem C	214
Figura 7.203 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	215
Figura 7.204- Diferença das curvas dos programas simulados em SIMA e linguagem C	215

Figura 7.205 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	216
Figura 7.206- Diferença das curvas dos programas simulados em SIMA e linguagem C	216
Figura 7.207 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	217
Figura 7.208- Diferença das curvas dos programas simulados em SIMA e linguagem C	217
Figura 7.209 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	218
Figura 7.210- Diferença das curvas dos programas simulados em SIMA e linguagem C	218
Figura 7.211 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	219
Figura 7.212 - Diferença das curvas dos programas simulados em SIMA e linguagem C	219
Figura 7.213 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	220
Figura 7.214- Diferença das curvas dos programas simulados em SIMA e linguagem C	220
Figura 7.215 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	221
Figura 7.216- Diferença das curvas dos programas simulados em SIMA e linguagem C	221
Figura 7.217 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	222
Figura 7.218- Diferença das curvas dos programas simulados em SIMA e linguagem C	222
Figura 7.219 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	223

Figura 7.220- Diferença das curvas dos programas simulados em SIMA e linguagem C	223
Figura 7.221 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	224
Figura 7.222- Diferença das curvas dos programas simulados em SIMA e linguagem C	224
Figura 7.223 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	225
Figura 7.224- Diferença das curvas dos programas simulados em SIMA e linguagem C	225
Figura 7.225 - Tela com todas as partições funcionando	229
Figura 7.226 - Partição 1	230
Figura 7.227 – Partição 2	230
Figura 7.228 - Partição 3	231
Figura 7.229 – Partição 4	231
Figura 7.230 – Partição 5	232
Figura 7.231 – Partição 6	232
Figura 7.232 – Partição com a falha inserida	233
Figura 7.233 – Partição 5 logo que ocorre a falha da partição 6	233
Figura 7.234 - Partição 5 pouco tempo após falha da partição 6	234
Figura 7.235 - Partição 5 num tempo considerável após a falha da partição 6	235
Figura 7.236 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	236
Figura 7.237- Diferença das curvas dos programas simulados em SIMA e linguagem C	236
Figura 7.238 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	237
Figura 7.239- Diferença das curvas dos programas simulados em SIMA e linguagem C	237

Figura 7.240 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	238
Figura 7.241- Diferença das curvas dos programas simulados em SIMA e linguagem C	238
Figura 7.242 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	239
Figura 7.243- Diferença das curvas dos programas simulados em SIMA e linguagem C	239
Figura 7.244 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	240
Figura 7.245- Diferença das curvas dos programas simulados em SIMA e linguagem C	240
Figura 7.246 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	241
Figura 7.247- Diferença das curvas dos programas simulados em SIMA e linguagem C	241
Figura 7.248 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	242
Figura 7.249- Diferença das curvas dos programas simulados em SIMA e linguagem C	242
Figura 7.250 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	243
Figura 7.251- Diferença das curvas dos programas simulados em SIMA e linguagem C	243
Figura 7.252 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	244
Figura 7.253- Diferença das curvas dos programas simulados em SIMA e linguagem C	244
Figura 7.254 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	245

Figura 7.255- Diferença das curvas dos programas simulados em SIMA e linguagem C	245
Figura 7.256 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	246
Figura 7.257- Diferença das curvas dos programas simulados em SIMA e linguagem C	246
Figura 7.258 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	247
Figura 7.259- Diferença das curvas dos programas simulados em SIMA e linguagem C	247
Figura 7.260 - Curvas dos resultados dos programas simulados em SIMA e linguagem C	248
Figura 7.261- Diferença das curvas dos programas simulados em SIMA e linguagem C	248

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 6.1 – Parâmetros da roda de reação utilizada nesse trabalho	106
Tabela 6.2 – Ganhos do PID nos três eixos	108
Tabela 6.3 – Parâmetros orbitais	109
Tabela 6.4 – Dados do satélite e da roda de reação	109
Tabela 6.5 – Ganhos do PID nos três eixos	110
Tabela 6.6 – Condições iniciais	110
Tabela 6.7 – Casos das simulações do Modo Nominal do SCA da PMM	165

LISTA DE SIGLAS E ABREVIATURAS

AC	Attitude Control
ACDH	Attitude Control and Data Handling
ACE	Attitude Control Electronic
AEEC	Airlines Electronics Engineering Committee
APEX	Application Executive
API	Application Programming Interface
ARINC	Aeronautical Radio Incorporated
CBERS	China-Brazil Earth-Resources Satellite
CNAE	Comissão Nacional de Atividades Espaciais
COTS	Commercial Off-The-Shelf
DH	Data Handling
ECU	Electronic Control Unit
ESA	European Space Agency
FIFO	First-In-First-Out
FOG	Fiber Optical Gyro
GCC	GNU Compiler Collection
GOCNAE	Grupo de Organização da Comissão Nacional de Atividades Espaciais
GPS	Global Positioning System
HW	Hardware
IMA	Integrated Modular Avionics
INPE	Instituto Nacional de Pesquisas Espaciais
ISS	International Space Station
LEO	Low Earth Orbit
LRM	Linear Replaceable Module
LRU	Linear Replaceable Unit
MMA	Massa-Mola-Amortecedor
MOS	Module Operating System
NASA	National Aeronautics and Space Administration
OBDH	On-Board Data Handling

ORU	Orbital Replaceable Unit
PID	Proporcional-Integral-Derivativo
PMM	Plataforma MultiMissão
PNAE	Programa Nacional de Atividades Espaciais
POS	Partition Operating System
RTCA	Radio Technical Commission for Aeronautics
SACI	Satélites de Aplicações Científicas
SCA	Sistema de Controle de Atitude
SCAO	Sistema de Controle de Atitude e Órbita
SCD	Satélite de Coleta de Dados
SIMA	Simulated Integrated Modular Avionics
SW	Software
VLHL	Vertical Local e Horizontal Local

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
1.1. Contexto e motivação	1
1.2. Objetivo	2
1.3. Organização deste trabalho	2
2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA.....	5
2.1. Breve histórico do programa espacial brasileiro de satélites.....	5
2.2. Plataforma MultiMissão (PMM).....	6
2.3. Satélite Amazônia-1	8
2.4. Movimento de um satélite.....	10
2.4.1. Movimento de atitude	10
2.4.1.1. Determinação de atitude	10
2.4.1.2. Predição de atitude	10
2.4.1.3. Controle de atitude	10
2.5. Sensores	11
2.5.1. Sensor de estrelas	11
2.5.2. Sensor Solar	13
2.5.3. Sensor de horizonte	15
2.5.4. Magnetômetro	17
2.5.5. Giroscópio	18
2.5.6. GPS (Global Positioning System)	19
2.6. Atuadores	20
2.6.1. Propulsores	20
2.6.2. Rodas de reação	22

2.6.3.	Bobinas eletromagnéticas	23
2.7.	Controle	25
2.8.	Sistemas para aplicações distribuídas	25
2.8.1.	Arquitetura federada.....	26
2.8.2.	Arquitetura integrada.....	27
3	AVIÔNICA MODULAR INTEGRADA (<i>INTEGRATED MODULAR AVIONICS</i> - IMA)	29
3.1.	ARINC 653	29
3.2.	APEX - APPLICATION EXECUTIVE.....	31
3.2.1.	Gerenciamento de partição e memória	32
3.2.2.	Gerenciamento de processo	32
3.2.3.	Gerenciamento de tempo.....	34
3.2.4.	Comunicação interpartição.....	34
3.2.5.	Comunicação intrapartição.....	35
3.2.6.	Diagnose (<i>Health Monitor</i>)	36
3.3.	Vantagens do IMA	36
3.3.1.	Tamanho, massa e consumo de energia reduzidos.....	36
3.3.2.	Competitividade	36
3.3.3.	Modularidade, Portabilidade e Reuso	37
3.3.4.	Certificação Incremental.....	37
3.3.5.	Alocação de Recursos	37
3.3.6.	Reconfiguração e robustez	38
3.4.	Desvantagens.....	38
3.4.1.	Processo de Integração Complexo	38

4	FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA A SUA SOLUÇÃO.....	39
4.1.	Formulação do problema.....	39
4.2.	Abordagens para sua solução.....	39
4.3.	O SCAO do Satélite Amazônia-1.....	40
4.4.	Arquiteturas de desenvolvimento, modelagem e simulação.....	44
4.5.	Motivação para o uso da plataforma IMA.....	45
4.6.	A utilização da IMA nos satélites brasileiros.....	48
5	PROGRAMAS E AMBIENTES UTILIZADOS NA SIMULAÇÃO.....	49
5.1.	Ambiente Computacional.....	49
5.2.	MatLab.....	49
5.3.	Linguagem C.....	50
5.4.	O SIMULADOR DE IMA (SIMA).....	51
5.5.	Instalação.....	52
5.6.	Programas-exemplo.....	55
5.7.	Portas.....	57
6	MODELAGEM E SIMULAÇÃO.....	59
6.1.	Modelagem e Simulação do Sistema Massa-Mola-Amortecedor.....	59
6.1.1.	Modelagem utilizando Função de transferência.....	60
6.1.1.1.	Planta.....	60
6.1.1.2.	Controlador.....	63
6.1.1.3.	Planta com PID em malha aberta.....	63
6.1.1.4.	Planta com PID em malha fechada.....	64
6.1.2.	Resposta em regime transitório.....	64
6.1.3.	Obtendo as constantes proporcional, integral e derivativa do PID.....	66

6.1.4.	Modelagem utilizando Equações de estado.....	69
6.1.4.1.	Implementação computacional (Discretização).....	70
6.1.4.1.1.	Equações de estado	70
6.1.4.1.2.	Conversão do PID analógico para o PID digital	70
6.2.	Implementação do sistema Massa-Mola-Amortecedor no SIMA.....	72
6.2.1.	Visão geral	72
6.2.2.	Portas e canais	73
6.2.3.	Configuração.....	75
6.2.3.1.	Partição 1	75
6.2.3.1.1.	a653.xml	75
6.2.3.1.2.	sima.xml.....	76
6.2.3.2.	Partição 2.....	77
6.2.3.2.1.	a653.xml	77
6.2.3.2.2.	sima.xml.....	78
6.2.4.	Agendamento (Scheduling) das partições.....	78
6.2.4.1.	a653.xml.....	79
6.3.	Modelagem do Sistema de Controle de Atitude	80
6.3.1.	Sistema de Coordenadas de Referência.....	80
6.3.1.1.	Referencial Inercial.....	80
6.3.1.2.	Referencial Orbital.....	82
6.3.1.3.	Referencial do Satélite	83
6.3.2.	Planta	85
6.3.2.1.	Cinemática	85
6.3.2.1.1.	Representação da Atitude.....	85
6.3.2.1.1.1.	Matriz de Rotação ou Matriz dos Cossenos Diretores.....	85

6.3.2.1.1.2. Ângulos de Euler	91
6.3.2.1.1.3. Quatérnios	96
6.3.2.2. Dinâmica	97
6.3.2.2.1. Equações do movimento de rotação.....	98
6.3.2.2.1.1. Rotação de um sistema de coordenadas em relação a um sistema de coordenadas inerciais	98
6.3.2.2.1.2. Quantidade de movimento angular.....	101
6.3.2.2.2. Equações de movimento de Euler	103
6.3.3. Atuador.....	105
6.3.4. Controlador	107
6.3.5. Sensor.....	109
6.3.6. Implementação computacional (Programação).....	109
6.3.6.1. Parâmetros orbitais:	109
6.3.6.2. Dados do satélite e da roda de reação:.....	109
6.3.6.3. Dados do controlador PID:	110
6.3.6.4. Condições iniciais.....	110
6.3.6.5. Funcionamento do programa:	111
6.4. Implementação do SCA no SIMA utilizando 4 partições	113
6.4.1. Visão geral	113
6.4.2. Portas e canais	113
6.4.2.1. Prioridade.....	115
6.4.2.2. Configuração.....	116
6.4.2.2.1. Partição 1.....	116
6.4.2.2.1.1. a653.xml	116
6.4.2.2.1.2. sima.xml	117

6.4.2.2.2. Partição 2.....	118
6.4.2.2.2.1. a653.xml	118
6.4.2.2.2.2. sima.xml	119
6.4.2.2.3. Partição 3.....	120
6.4.2.2.3.1. a653.xml	120
6.4.2.2.3.2. sima.xml	121
6.4.2.2.4. Partição 4.....	122
6.4.2.2.4.1. a653.xml	122
6.4.2.2.4.2. sima.xml	123
6.4.3. Agendamento (<i>Scheduling</i>).....	123
6.4.3.1. a653.xml.....	125
6.4.3.2. Funcionamento do programa no SIMA:.....	126
6.5. Implementação do SCA e do MM no SIMA utilizando 6 partições.....	127
6.5.1. Visão geral	127
6.5.2. Portas e canais	128
6.5.2.1. Prioridade.....	131
6.5.2.2. Configuração.....	132
6.5.2.2.1. Partição 1.....	133
6.5.2.2.1.1. a653.xml	133
6.5.2.2.1.2. sima.xml	134
6.5.2.2.2. Partição 2.....	135
6.5.2.2.2.1. a653.xml	135
6.5.2.2.2.2. sima.xml	136
6.5.2.2.3. Partição 3.....	137
6.5.2.2.3.1. a653.xml	137

6.5.2.2.3.2. sima.xml	138
6.5.2.2.4. Partição 4	139
6.5.2.2.4.1. a653.xml	139
6.5.2.2.4.2. sima.xml	140
6.5.2.2.5. Partição 5	141
6.5.2.2.5.1. a653.xml	141
6.5.2.2.5.2. sima.xml	142
6.5.2.2.6. Partição 6	143
6.5.2.2.6.1. a653.xml	143
6.5.2.2.6.2. sima.xml	144
6.5.3. Agendamento (<i>Scheduling</i>) das partições	144
6.5.3.1. a653.xml	145
7 RESULTADOS	147
7.1. Massa-Mola-Amortecedor	147
7.2. Sistema de Controle de Atitude	151
7.2.1. Passo de integração	151
7.2.2. Tempo	155
7.2.2.1. Simulação do Modo Nominal do SCA da PMM utilizando t e t_s	156
7.2.2.1.1. Ângulos de atitude	157
7.2.2.1.2. Velocidade angular da PMM	158
7.2.2.1.3. Velocidade angular da Roda	160
7.2.2.1.4. Tensão	161
7.2.2.1.5. Torque	163
7.2.3. Casos	164

7.2.3.1. Caso 1: Simulação do Modo Nominal do SCA da PMM em linguagem C e em MatLab.....	165
7.2.3.1.1. Ângulos de atitude	166
7.2.3.1.2. Velocidade angular da PMM.....	169
7.2.3.1.3. Velocidade angular das rodas.....	172
7.2.3.1.4. Tensão.....	175
7.2.3.1.5. Torque da roda	178
7.2.3.2. Caso 2: Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e MatLab.....	181
7.2.3.2.1. Ângulos de atitude	181
7.2.3.2.2. Velocidade angular da PMM.....	184
7.2.3.2.3. Velocidade angular das rodas.....	187
7.2.3.2.4. Tensão.....	190
7.2.3.2.5. Torque da roda	193
7.2.3.3. Caso3: Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e linguagem C	196
7.2.3.3.1. Ângulos de atitude	197
7.2.3.3.2. Velocidade angular da PMM.....	200
7.2.3.3.3. Velocidade angular das rodas.....	203
7.2.3.3.4. Tensão.....	206
7.2.3.3.5. Torque da roda	209
7.2.3.4. Caso 4: Simulação do Modo Nominal do SCA da PMM e do MMA divididos em 6 partições.....	212
7.2.3.4.1. Ângulos de atitude	212
7.2.3.4.2. Velocidade angular da PMM.....	216
7.2.3.4.3. Velocidade angular das rodas.....	218

7.2.3.4.4. Tensão	221
7.2.3.4.5. Torque da roda	223
7.2.3.5. Caso 5: Simulação do Modo Nominal do SCA da PMM e do MMA divididos em 6 partições com falha inserida	226
7.2.3.5.1. Ângulos de atitude	236
7.2.3.5.2. Velocidade angular da PMM	239
7.2.3.5.3. Velocidade angular das rodas	241
7.2.3.5.4. Tensão	243
7.2.3.5.5. Torque da roda	247
8 COMENTÁRIOS E CONCLUSÕES	249
8.1. Geral	249
8.1.1. Massa-Mola Amortecedor	250
8.1.2. Sistema de controle de atitude	250
8.2. Sugestões para trabalhos futuros	253
REFERÊNCIAS BIBLIOGRÁFICAS	255

1 INTRODUÇÃO

1.1. Contexto e motivação

Na área espacial, o desenvolvimento dos sistemas de um veículo está atrelado aos requisitos da missão. Uma vez que sejam estabelecidos estes requisitos e transformados em especificações, são feitos estudos e análises e, em seguida, definidos o Software (SW) e o Hardware (HW) necessários para cumpri-los.

Particularmente, o desenvolvimento do software e hardware para o Sistema de Controle de Atitude e Órbita (SCAO) de um satélite tem que ser planejado e desenvolvido de forma a cumprir os requisitos da missão em termos de atitude e órbita. Nesse contexto são feitos trabalhos de modelagem (por exemplo: dinâmica e cinemática do satélite, torques e forças ambientais, torques e forças internas, modelagem de atuadores, sensores e ambientes externos associados como o campo magnético da Terra, etc.) e de simulação computacional.

Nessas modelagens e simulações podem ser utilizados: 1) linguagens de computação gerais, como por exemplo, FORTRAN, Pascal, C, C++, etc.; 2) ambientes de computação gerais, como por exemplo, o MatLab®, o MatrixX®, o ScilLab®, etc.; 3) pacotes computacionais gerais como o SIMA®, etc. existentes no mercado; 4) pacotes computacionais específicos, como por exemplo, o STK®, o PSS®, etc., A escolha dentre eles depende de inúmeros fatores incluindo prazos, custos, disponibilidade, manutenibilidade, acesso às informações, etc.

Os controladores dos sistemas de controle de atitude e de órbita simulados podem ser implementados analogicamente ou digitalmente em: 1) um computador físico para Controle de Atitude (“Attitude Control - AC”) e outro computador físico para Manipulação de Dados (“Data Handling - DH”); ou 2) em um computador físico compartilhado para Controle de Atitude e Manipulação de Dados (“Attitude Control and Data Handling – ACDH”) implementando um computador virtual para “Attitude Control” e outro

computador virtual para “Data Handling”. Estes prometem ser, respectivamente, os casos dos satélites Amazônia-1 e Lattes, em desenvolvimento no INPE, utilizando a Plataforma MultiMissão (PMM). Neste contexto, parece ser oportuno iniciar-se a exploração do estudo e implementação de um SCAO em ambas as arquiteturas e compará-las entre si e com os requisitos da missão.

1.2. Objetivo

A presente proposta de trabalho se insere neste contexto, pois objetiva estudar um Sistema de Controle de Atitude (SCA) compatível com o Modo Nominal = Modo de Controle da Missão (“*Mission Control Mode*”) do satélite Amazônia-1, em desenvolvimento no INPE, e o implementar em um Simulador (SIMA) de Aviônica Modular Integrada, *Integrated Modular Avionics* (IMA). O conceito IMA baseia-se no uso de um computador real dividido em várias partições ou computadores virtuais.

A análise dos resultados é baseada na comparação dos resultados entre si e com os requisitos da missão para o SCA nesse modo de operação.

1.3. Organização deste trabalho

Este trabalho está organizado da seguinte forma:

Capítulo 1: INTRODUÇÃO

Este capítulo apresenta uma breve introdução sobre a modelagem de sistemas de controle de atitude e órbita e sua implementação; o objetivo e a organização deste trabalho.

Capítulo 2: CONCEITOS BÁSICOS E REVISÃO DA LITERATURA

Neste capítulo é feito um breve histórico do programa espacial brasileiro de satélites; são apresentados a Plataforma MultiMissão (PMM) e o Satélite Amazônia-1; o movimento de atitude de um satélite; os tipos de sensores e atuadores; e uma visão geral de arquiteturas federadas e integradas.

Capítulo 3: AVÔNICA MODULAR INTEGRADA - *INTEGRATED MODULAR AVIONICS* (IMA).

Este capítulo apresenta o padrão ARINC 653, a interface de propósito geral APEX, além das vantagens e desvantagens do uso da IMA.

Capítulo 4: FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA SUA SOLUÇÃO

Neste capítulo, formula-se o problema e apresentam-se algumas abordagens para sua solução. Além disso, detalha-se o Sistema de Controle de Atitude e Órbita (SCAO) e os requisitos de missão do Satélite Amazônia-1. Abordam-se, também: arquiteturas de desenvolvimento, modelagem e simulação; motivação do uso do conceito IMA; e sua utilização nos satélites brasileiros.

Capítulo 5: PROGRAMAS E AMBIENTES UTILIZADOS NA SIMULAÇÃO

Inicia-se esse capítulo descrevendo o ambiente utilizado para as simulações. Apresenta-se o MatLab, a linguagem C e, mais detalhadamente, o SIMA (Simulated Integrated Modular Avionics).

Capítulo 6: MODELAGEM E SIMULAÇÃO

Neste capítulo é apresentada a modelagem do sistema Massa-Mola-Amortecedor (MMA) e o seu controlador (PID); em seguida, a modelagem do Sistema de Controle de Atitude (SCA) do Satélite Amazônia-1 e seu controlador (PID). Além disso, é apresentado como foi feita a implementação do MMA e do SCA no SIMA com a utilização de portas, canais, o agendamento das partições e os arquivos de configuração do SIMA. Também são tratados os

sistemas de coordenadas de referência utilizados no trabalho e os modos de representação de atitude. Mais próximo do final do capítulo, são apresentados os parâmetros orbitais, dados do satélite e da roda de reação, dados dos ganhos do PID, condições iniciais e um fluxograma do funcionamento do programa.

Capítulo 7: RESULTADOS

Este capítulo inicia-se com os resultados das simulações do sistema Massa-Mola-Amortecedor. É apresentado o deslocamento em função do tempo obtido nas simulações em MatLab, linguagem C e SIMA; a comparação entre C e SIMA e MatLab e SIMA, bem como as diferenças entre os gráficos. Em seguida, são apresentados os resultados das simulações do SCA. Primeiramente, mostra-se a justificativa da utilização do passo de integração escolhido e do tempo utilizado. Após isso, são simulados cinco casos particulares em MatLab, linguagem C e SIMA. Os casos são: 1) Simulação do Modo Nominal do SCA da PMM em MatLab e em linguagem C; 2) Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e MatLab; 3) Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e linguagem C; 4) Simulação do Modo Nominal do SCA da PMM e do MMA divididos em 6 partições; e 5) Simulação do Modo Nominal do SCA da PMM e do MMA divididos em 6 partições com falha inserida no MMA. Em seguida, os resultados são comparados entre si e apresentados.

Capítulo 8: COMENTÁRIOS E CONCLUSÕES

Este capítulo apresenta as conclusões sobre as simulações do sistema Massa-Mola-Amortecedor e dos cinco casos do Sistema de Controle de Atitude. Além disso, são apresentadas várias sugestões para trabalhos futuros.

2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA

2.1. Breve histórico do programa espacial brasileiro de satélites

De acordo com Fonseca (2005), o programa espacial brasileiro teve início em 1961 com a criação do GOCNAE (Grupo de Organização da Comissão Nacional de Atividades Espaciais) que passou a ser conhecido apenas como CNAE (Comissão Nacional de Atividades Espaciais). Em 1971 a CNAE passou a ser chamada de Instituto de Pesquisas Espaciais; e, mais recentemente, foi denominado Instituto Nacional de Pesquisas Espaciais (INPE).

Inicialmente, o INPE esteve envolvido em atividades de pesquisas relacionadas às Ciências Espaciais e da Atmosfera. Com o tempo, evoluiu para áreas de Aplicações Espaciais como Sensoriamento Remoto, Meteorologia; e depois, para a Engenharia e Tecnologia Espaciais, e Ciência do Sistema Terrestre.

O primeiro satélite brasileiro, o Satélite de Coleta de Dados (SCD-1), foi lançado ao espaço em 09 de fevereiro de 1993 da Flórida, EUA, pelo foguete Pegasus. Em outubro de 1998, o segundo Satélite de Coleta de Dados (SCD-2) foi lançado pelo mesmo tipo de foguete.

Com o PNAE (Programa Nacional de Atividades Espaciais), 1996-2007, deu-se início ao desenvolvimento da série de Satélites de Aplicações Científicas (SACIs). O SACI-1 foi colocado em órbita, no entanto, a missão falhou, pois as estações de Terra não conseguiram comunicação com o satélite. O SACI-2, também teve problemas, pois o segundo estágio do foguete que o colocaria em órbita não teve sucesso durante a ignição e o satélite foi perdido.

Com a parceria do Brasil com a China em 1987, deu-se início à série de satélites de sensoriamento remoto CBERS (China-Brazil Earth-Resources Satellite). Estes satélites são utilizados no monitoramento de desmatamentos e queimadas na Amazônia, monitoramento de recursos hídricos, agrícolas, crescimento urbano, etc.

2.2. Plataforma MultiMissão (PMM)

A Plataforma MultiMissão do INPE é uma plataforma genérica flexível para ser aplicada em diferentes missões espaciais que sejam estabilizadas em três eixos e satisfaçam requisitos de: massa, órbita, atitude, potência, térmica, estrutura, computação (um único computador para desempenhar as funções de manipulação de dados e de controle de atitude e órbita), etc. Assim, a única diferença entre os satélites utilizando esta plataforma será a carga útil dos mesmos.



Figura 2.1 - PMM (configuração em órbita)

Fonte: Projeto PMM (2009)

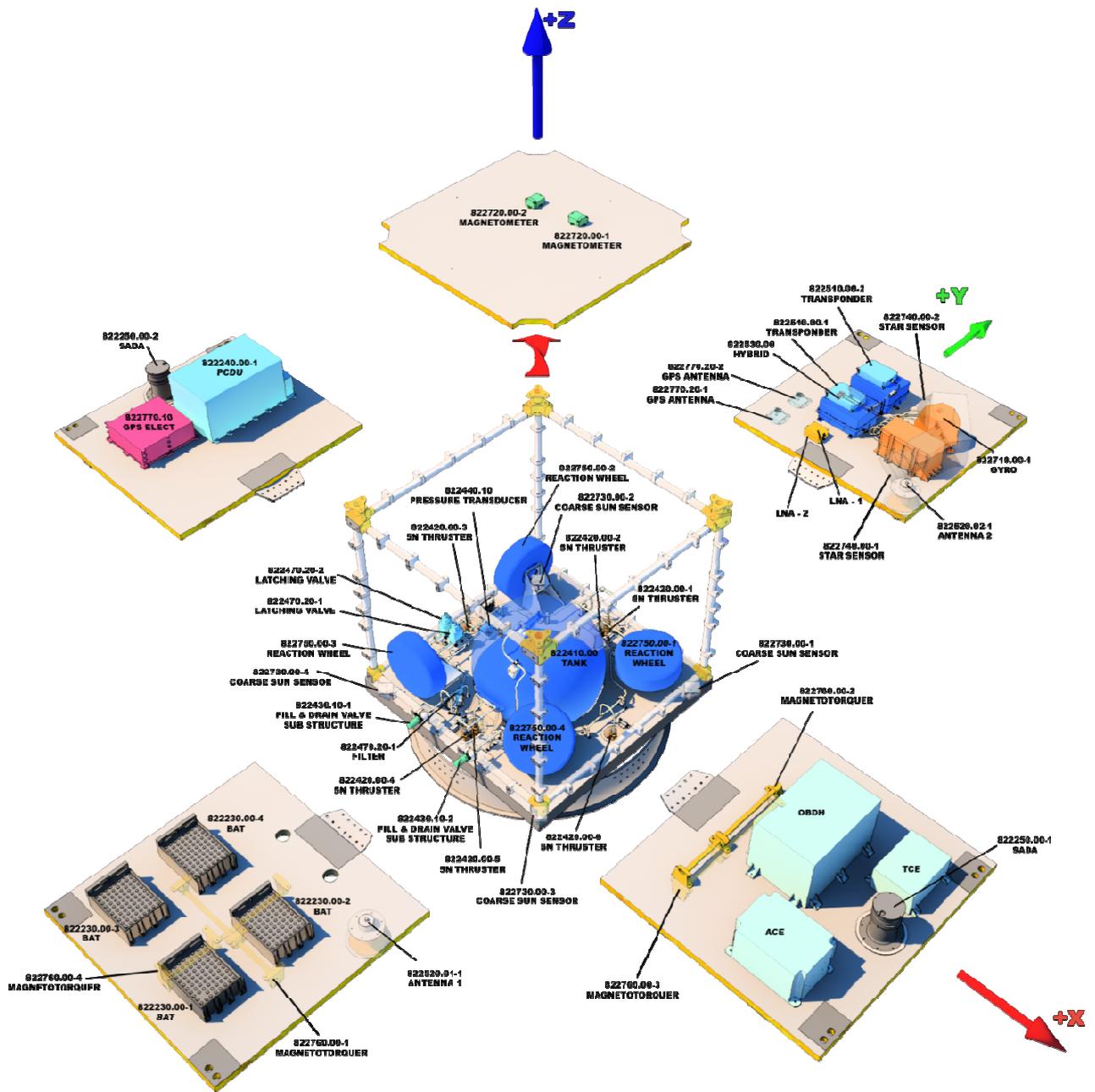


Figura 2.2 - Visão explodida da PMM (Carga útil)

Fonte: Projeto Amazônia-1 (2009)

2.3. Satélite Amazônia-1

O satélite Amazônia-1 será o primeiro satélite construído utilizando como base a Plataforma MultiMissão. No entanto, neste satélite, o SCAO e o DH são alocados em dois computadores. A figura 2.3 abaixo ilustra o Satélite Amazônia-1. O módulo inferior é o Módulo de Serviço e o módulo superior é a Carga Útil.

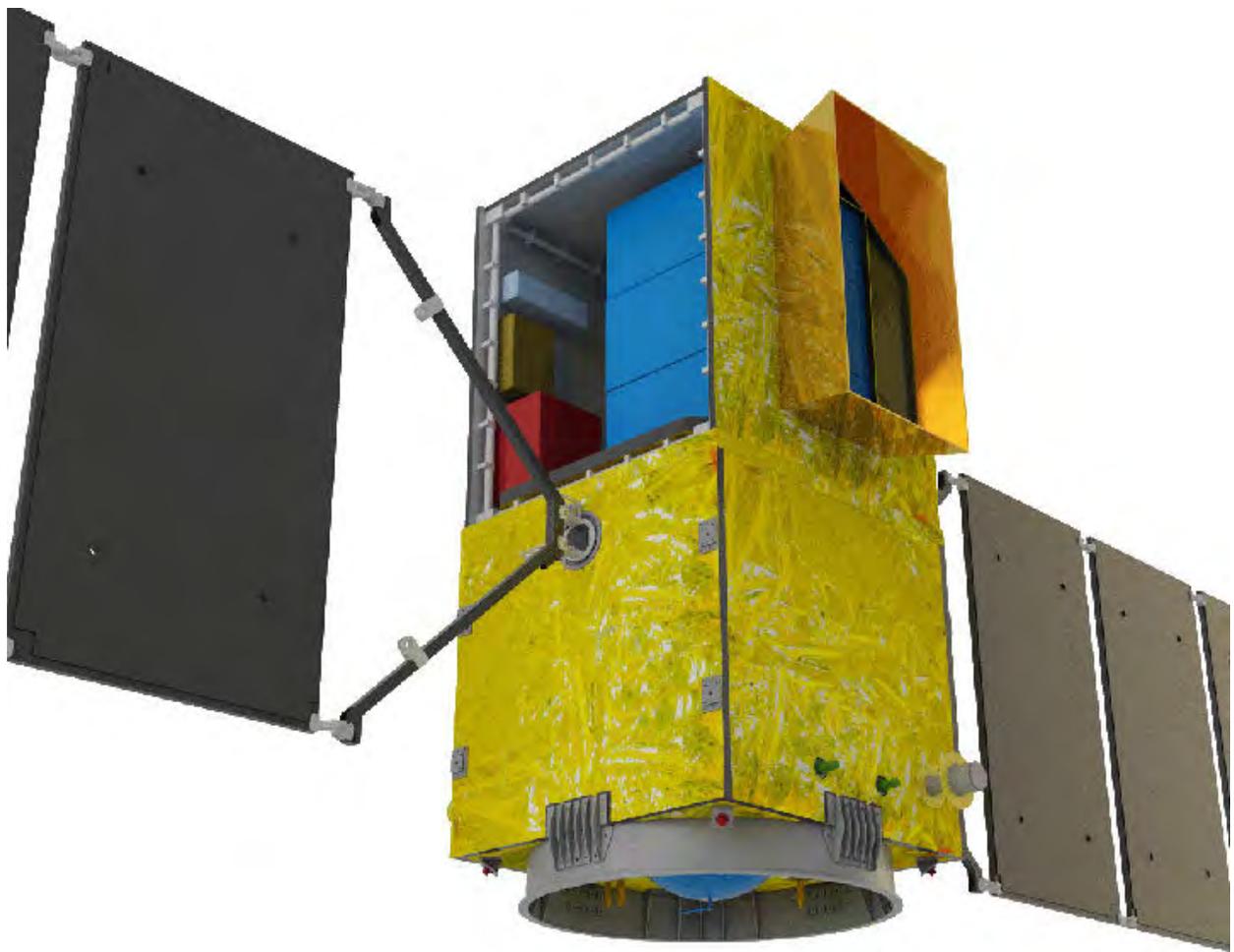


Figura 2.3 - Satélite Amazônia-1 (PMM+Módulo de Serviço)

Fonte: Projeto Amazônia -1 (2009)

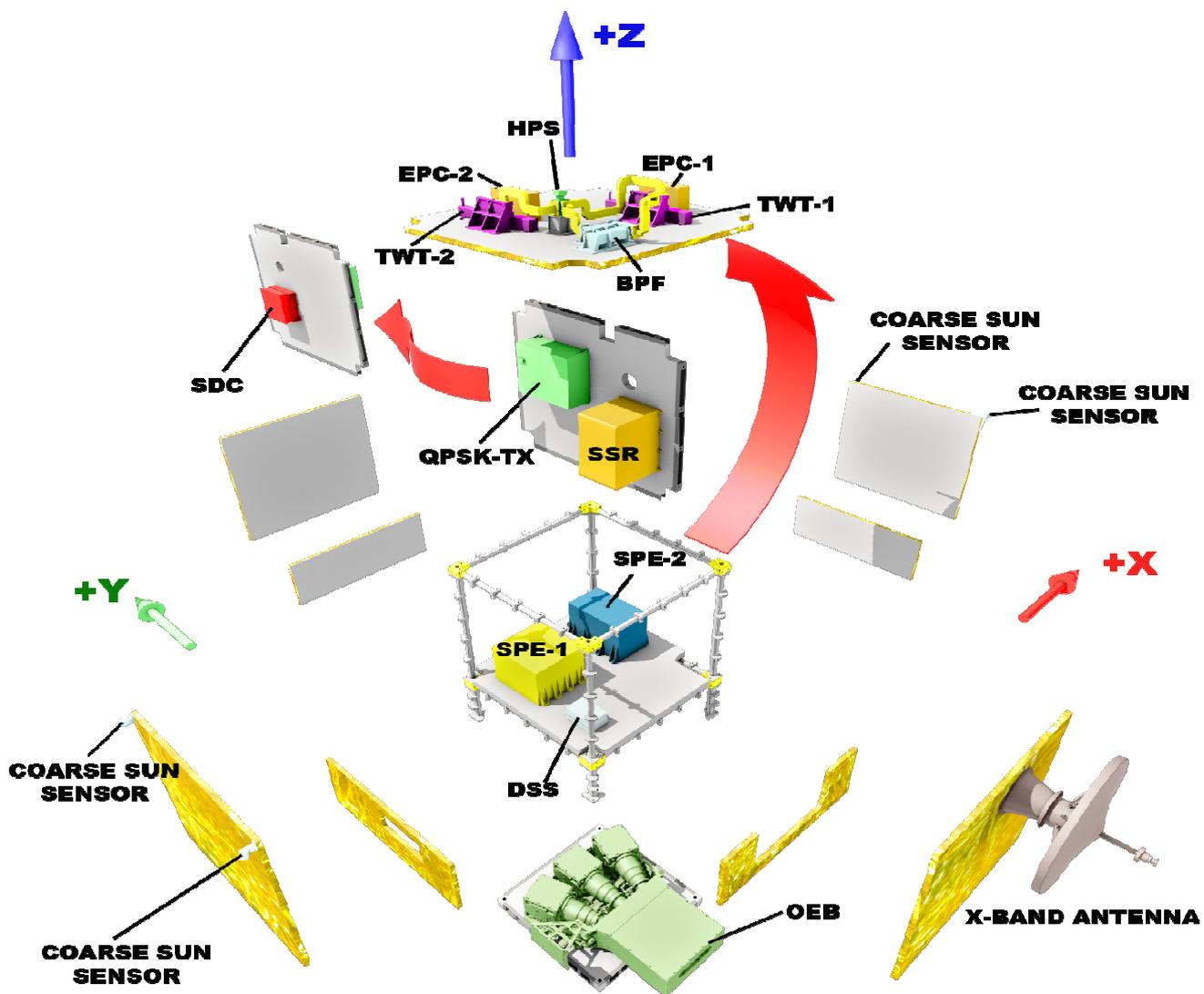


Figura 2.4 - Visão explodida do módulo de serviço

Fonte: Projeto Amazônia-1 (2009)

2.4. Movimento de um satélite

De acordo com Wertz (1978), o movimento de um satélite artificial é especificado por sua posição e velocidade lineares (órbita e movimento orbital), e por sua posição e velocidade angulares (atitude e movimento de atitude). A posição e velocidade lineares descrevem o movimento de translação do centro de massa do satélite. A posição e velocidade angulares descrevem o movimento de rotação do satélite em torno do seu centro de massa.

2.4.1. Movimento de atitude

A análise do movimento de atitude pode ser dividida em determinação, predição e controle de atitude (WERTZ, 1978).

2.4.1.1. Determinação de atitude

A determinação de atitude é o processo pelo qual se obtém a orientação do satélite relativa a um sistema inercial de referência ou a algum objeto de interesse, como a Terra. Utilizam-se diversos sensores para esse fim, tais como sensores de estrelas, sensores Solares, giroscópios, etc. (WERTZ, 1978).

2.4.1.2. Predição de atitude

A predição de atitude é o processo pelo qual se prevê a orientação do satélite utilizando modelos dinâmicos através da extrapolação do histórico da atitude (WERTZ, 1978).

2.4.1.3. Controle de atitude

O controle de atitude é o processo ativo de se reorientar o satélite para uma nova atitude ou manter uma atitude pré-determinada. Este processo é subdividido em manobra de atitude e estabilização de atitude. O primeiro consiste em reorientar o satélite de uma atitude qualquer para uma atitude pré-

determinada. O segundo consiste em manter o satélite em uma atitude pré-determinada (WERTZ, 1978).

Assim, para o controle de atitude de um satélite é necessário antes a determinação de atitude. A partir da leitura dos sensores, os valores lidos são comparados a valores de referência no controlador; este emite sinais de controle para os atuadores (propulsores a gás, rodas de reação, bobinas eletromagnéticas) agirem de modo a diminuir a diferença entre os valores lidos e os valores de referência.

O controle de atitude de um satélite é muito importante, pois normalmente, as especificações dos mesmos pedem o apontamento de uma ou mais faces para direções específicas.

Geralmente, em satélites de sensoriamento remoto, a câmera deve estar apontada para a Terra, enquanto que em satélites de comunicação, é a antena que deve estar apontada para a Terra. Além disso, diversos satélites são alimentados por energia Solar, portanto, seus painéis devem estar apontados para o Sol.

2.5. Sensores

2.5.1. Sensor de estrelas

Um sensor de estrelas é um dispositivo que mede a direção de uma estrela no sistema de coordenadas do veículo espacial. Baseado nas coordenadas desta estrela, armazenada em um catálogo de estrelas, a orientação do satélite pode ser determinada (SAMAAN, 2003). Deste modo, com sensores de estrelas é possível saber exatamente o apontamento do veículo em tempo real (DIAZ, 2006).

De acordo com Samaan (2003), a determinação de atitude através de sensores de estrelas é muito precisa, menor que um segundo de arco. No entanto, esses sensores são tradicionalmente caros, pesados e têm um consumo alto de

energia. Além disso, recursos computacionais (software/hardware) para processar seus dados são mais complexos e dispendiosos. Outro ponto negativo são a ocultação e as interferências de fontes de alto brilho, como o Sol e seu reflexo na Terra. Apesar dessas desvantagens, sua crescente precisão e versatilidade resultaram em seu uso numa grande variedade de veículos espaciais e experimentos.



Figura 2.5 - Sensor de Estrelas ASTRO 15

Fonte: Jena-Optronik GmbH (2011)



Figura 2.6 - Sensor de Estrelas VST-41M

Fonte: VECTRONIC Aerospace GmbH (2010a)

2.5.2. Sensor Solar

Os sensores solares determinam a orientação de um veículo espacial medindo a incidência solar sobre os mesmos (WINETRAUB et al., 2010).

Segundo Samaan (2003), esse tipo de sensor é o mais utilizado, para determinação de atitude, em quase todos os satélites já lançados. O desenvolvimento deste tipo de sensor é fácil e barato, além de ter um consumo reduzido, em relação a outros sensores. Existem basicamente três tipos de sensores: sensores analógicos, sensores de presença solar e sensores digitais.



Figura 2.7 - Sensor Solar

Fonte: SSBV Aerospace & Technology Group (2013)

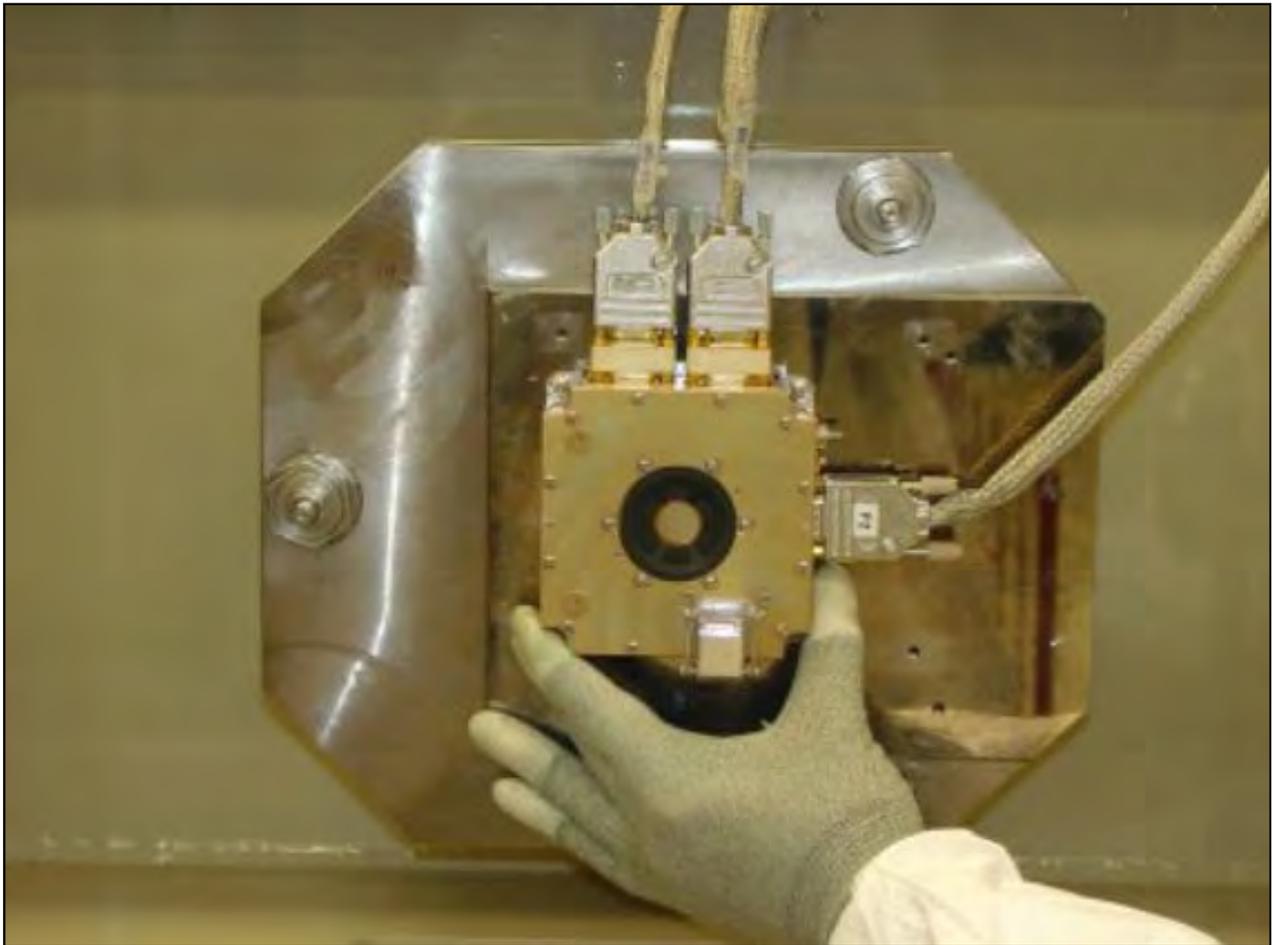


Figura 2.8 - Sensor Solar

Fonte: ESA Science & Technology (2007)

2.5.3. Sensor de horizonte

Sensores de horizonte são dispositivos óticos equipados com infravermelho que detectam o contraste entre o frio do espaço e o calor da Terra (SAMAAN, 2003). Estes sensores fornecem a orientação em relação à Terra sobre dois eixos ortogonais. Eles tendem a ser menos precisos que sensores baseados em observações estelares.



Figura 2.9 - Sensor de Horizonte

Fonte: SSBV Aerospace & Technology Group (2012)

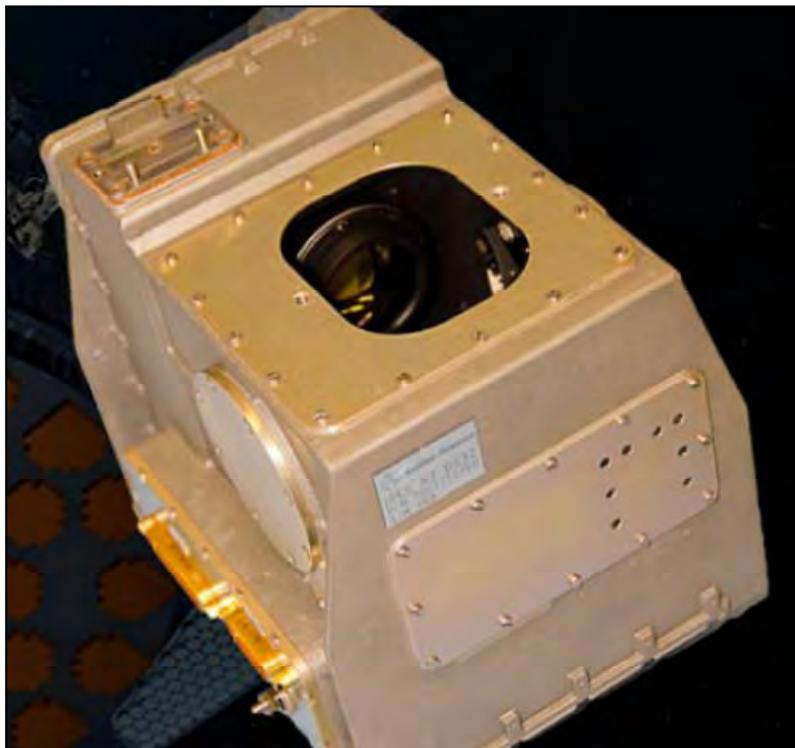


Figura 2.10 - Sensor de Horizonte

Fonte: Selex ES S.p.A (2013)

2.5.4. Magnetômetro

Os magnetômetros são sensores que medem a direção e o módulo do campo magnético terrestre. Têm um baixo consumo, são leves e podem operar em várias temperaturas. No entanto, como o campo magnético da Terra não é precisamente conhecido, os magnetômetros são susceptíveis a pequenos erros na determinação de atitude. Em particular, para altitudes acima de 1000 km onde o campo magnético é menos intenso, os erros se tornam consideráveis (SAMAAN, 2003).



Figura 2.11 - Magnetômetro HMR2300

Fonte: Honeywell International Inc.(2012)

2.5.5. Giroscópio

Giroscópios convencionais são baseados em uma massa girante montada sobre duas suspensões cardans. Esses cardans permitem que a massa gire nos três eixos. Devido à conservação do momento angular, a roda tende a resistir a mudanças de direção. Assim, quando o giroscópio é submetido a rotações, a massa se manterá em uma orientação global constante e os ângulos dos cardans sofrerão variação. A orientação é medida medindo-se estes ângulos (WOODMAN, 2007).

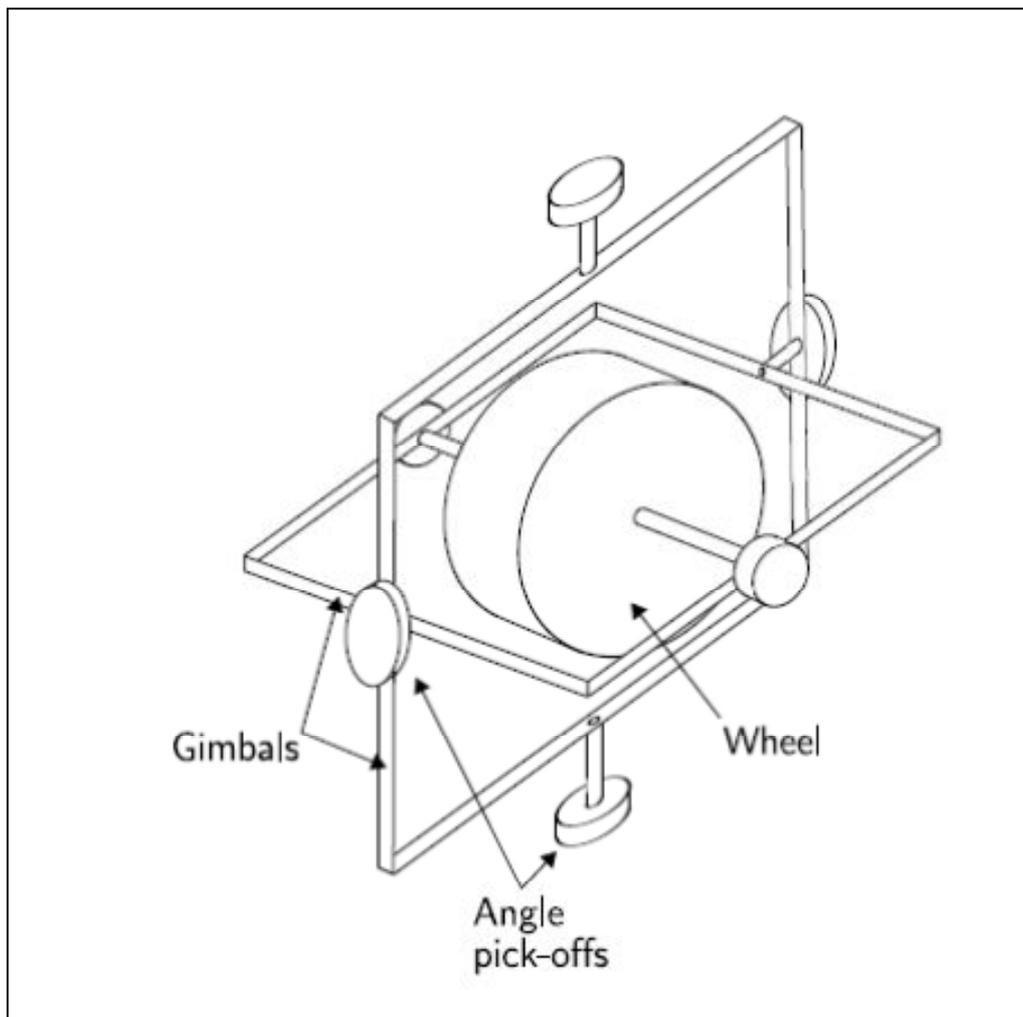


Figura 2.12 - Giroscópio mecânico convencional

Fonte: WOODMAN (2007)

Atualmente, há diversos tipos de giroscópios. Um tipo muito utilizado é o FOG '*Fiber Optical Gyro*'. Um giroscópio de fibra ótica consiste de uma grande bobina de fibra ótica, na qual dois raios de luz são disparados em posições opostas. Se o sensor estiver sujeito a uma rotação, o raio de luz que viaja na direção da rotação percorrerá um caminho mais longo que o raio contra a rotação (WOODMAN, 2007).

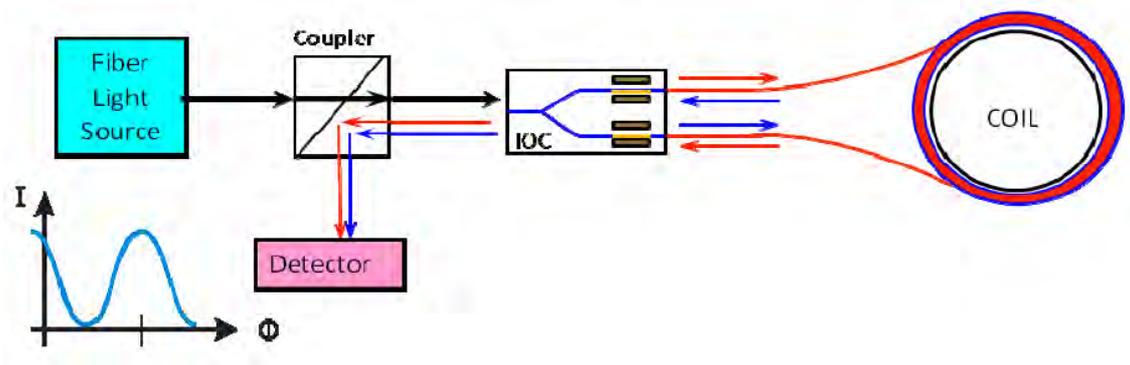


Figura 2.13 - Giroscópio de Fibra Ótica

Fonte: BARBOUR (2011)

2.5.6. GPS (Global Positioning System)

O GPS é um sistema de navegação composto por 24 satélites dispostos em uma configuração que garante a visualização, pelo receptor de, no mínimo, 4 satélites. Este sistema fornece as coordenadas, velocidade e direção do deslocamento entre pontos (ALBUQUERQUE e SANTOS, 2003).



Figura 2.14 - Constelação dos satélites do sistema GPS

Fonte: ALBUQUERQUE e SANTOS (2003)

2.6. Atuadores

2.6.1. Propulsores

Os jatos ou *thrusters* produzem empuxo em uma direção ao expelir propelente no sentido contrário. Os torques e forças resultantes têm como funções principais o controle de atitude, o controle da taxa de rotação, o controle de nutação, o controle da velocidade dos volantes de inércia, e o ajuste de órbitas (WERTZ, 1978).

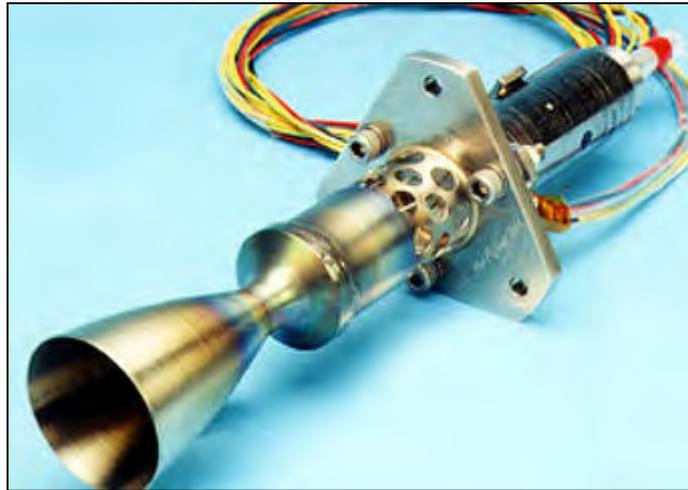


Figura 2.15 - Jato Propulsor
Fonte: SpaceRef Interactive Inc. (2012)



Figura 2.16 - Jato Propulsor
Fonte: University of Surrey (2012)

2.6.2. Rodas de reação

Roda de reação é basicamente um motor elétrico com um disco preso em sua extremidade. Ao se acionar o motor para girar o disco, devido à conservação do momento angular, o satélite sofre um torque no sentido contrário à rotação.

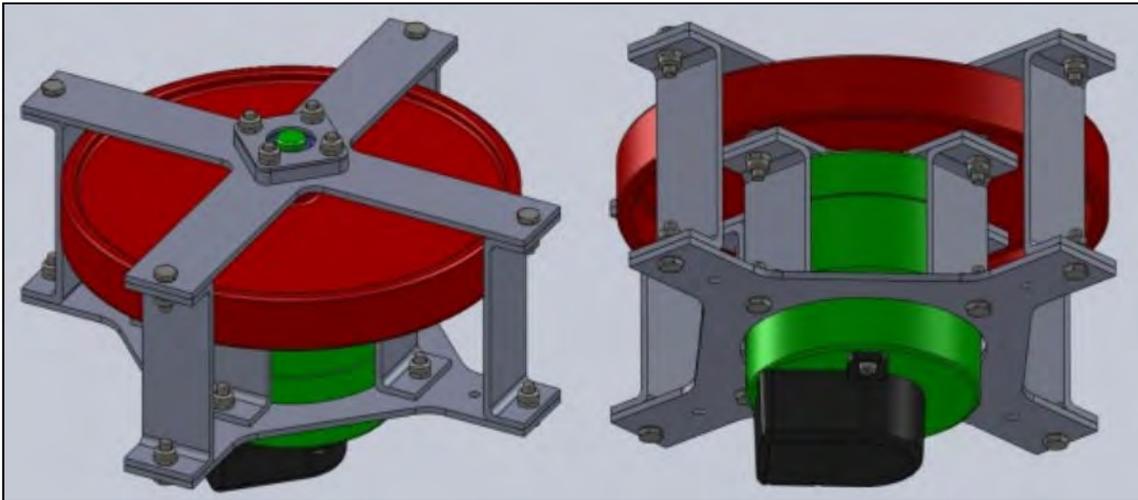


Figura 2.17 – Desenhos de Rodas de Reação

Fonte: Experiment ARCADE (2011)



Figura 2.18 - Rodas de Reação

Fonte: Rockwell Collins (2012)

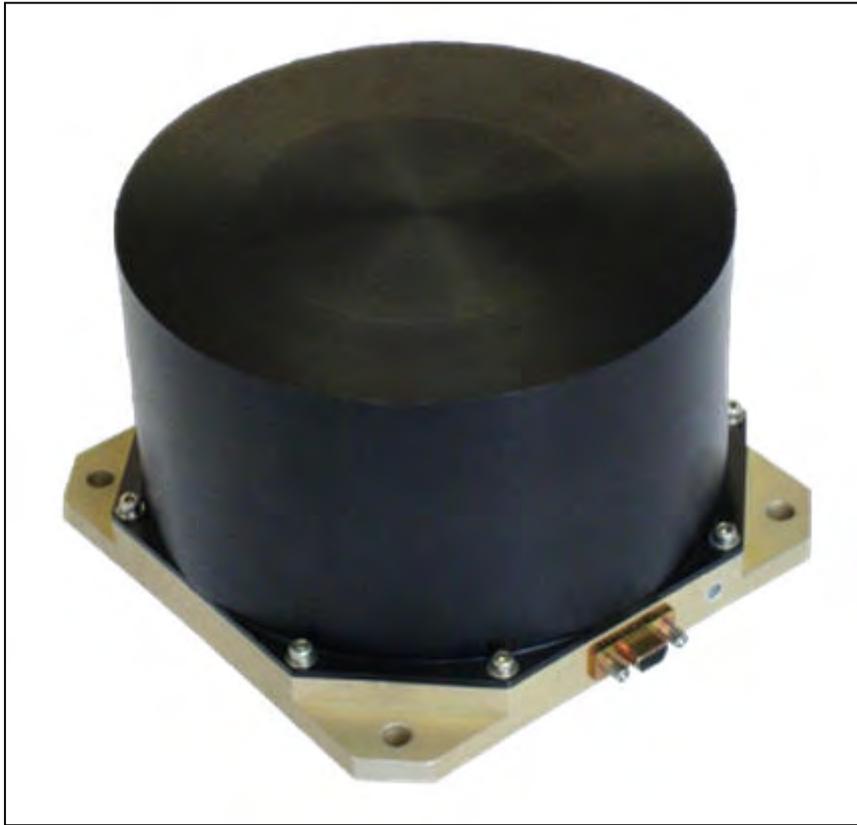


Figura 2.19 - Roda de Reação

Fonte: VECTRONIC Aerospace GmbH (2010b)

2.6.3. Bobinas eletromagnéticas

Uma bobina eletromagnética é essencialmente um longo fio de cobre encapado e enrolado em torno de um núcleo. O material do núcleo pode ser ligas especiais ou simplesmente ar (LEE; JOBANPUTRA, 2002).

Estes dispositivos são projetados para interagir com o campo magnético terrestre gerando forças e torques controlados, mudando assim a orientação do satélite.



Figura 2.20 - Bobina Eletromagnética

Fonte: Microcosm Inc (2008)



Figura 2.21 - Bobina Eletromagnética VMT-35

Fonte: VECTRONIC Aerospace GmbH (2010c)

2.7. Controle

Como dito na Seção 2.3.1.3, o controle da atitude de um satélite é feito em três etapas: leitura dos sensores; comparação destas com valores de referência nos controladores; e, em seguida, correção desses valores através dos atuadores.

O comando dos atuadores pode ser feito remotamente, através de comando do solo, ou pode-se controlar o sistema automaticamente utilizando-se um controlador automático (Figura 2.22).

Basicamente, o controlador compara os valores de saída da planta (processo) com um valor desejado, determina o erro e envia um sinal de controle para o atuador com o intuito de reduzir esse erro à zero ou a um valor muito pequeno.

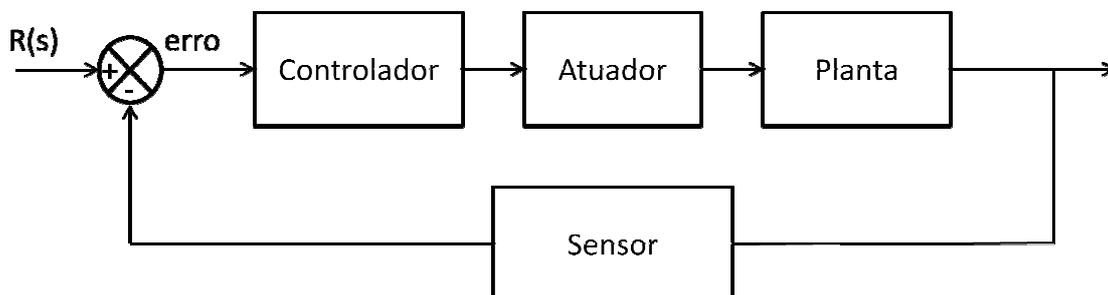


Figura 2.22 - Sistema de controle automático

2.8. Sistemas para aplicações distribuídas

De acordo com Obermaisser et al. (2007), é possível distinguir duas classes extremas de aplicações distribuídas: de um lado, as arquiteturas federadas; e de outro, as arquiteturas integradas. Ainda segundo o autor, sistemas reais estão frequentemente posicionados entre os dois extremos, inclinados mais para um lado ou para o outro.

2.8.1. Arquitetura federada

Arquiteturas federadas são baseadas em módulos computacionais dedicados distribuídos ao longo do veículo. Cada módulo é responsável pela execução de aplicações específicas de um determinado nível de criticalidade dentro do sistema, conforme a Figura 2.23.

As plataformas computacionais federadas recebem diversos nomes: alguns mais gerais como unidades de processamento ou módulos computacionais; até nomes mais específicos de acordo com o subsistema em que se encontram. Nos sistemas espaciais, são chamadas de *Orbital Replaceable Units* (ORUs), (ALENA et al ,2007); nos sistemas aeronáuticos, são chamadas de *Line Replaceable Units* (LRUs) ou *Line Replaceable Modules* (LRMs) (WATKINS e WALTER, 2007); e nos sistemas automotivos, são chamadas de *Electronic Control Units* (ECUs), (NATALE e SANGIOVANNI-VINCENNELLI, 2008).

De acordo com Alena et al. (2007), o sistema da Estação Espacial Internacional, (*International Space Station – ISS*), e do avião Boeing 767 são exemplos de arquiteturas federadas. A ISS utiliza um grande número de ORUs, enquanto o Boeing 767 utiliza diversas LRUs.

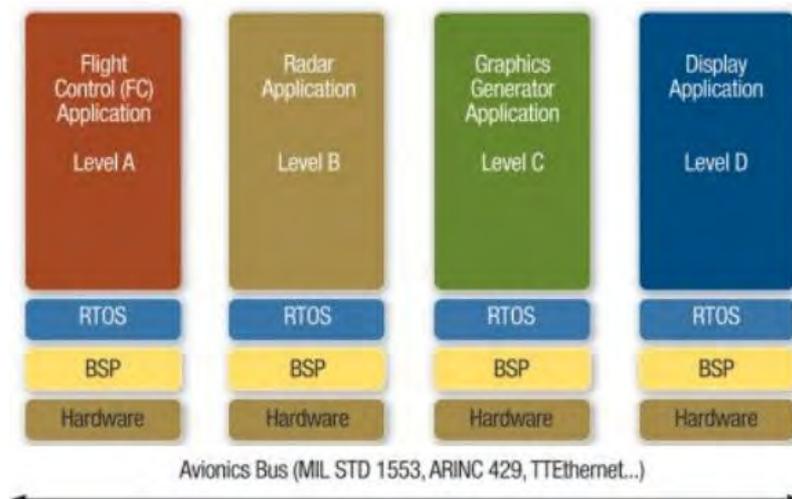


Figura 2.23 - Arquitetura Federada

Fonte: RTC Group (2010)

2.8.2. Arquitetura integrada

Arquiteturas integradas baseadas no conceito IMA têm como princípio o compartilhamento de recursos computacionais. Neste tipo de arquitetura, um único módulo suporta diversas aplicações, conforme a Figura 2.24. Para tanto, é necessário um sistema de particionamento robusto que divida o computador em n computadores virtuais, cada um alocando uma aplicação, (BLACK e FLETCHER, 2006). Este sistema de particionamento é padronizado pelo padrão ARINC 653.

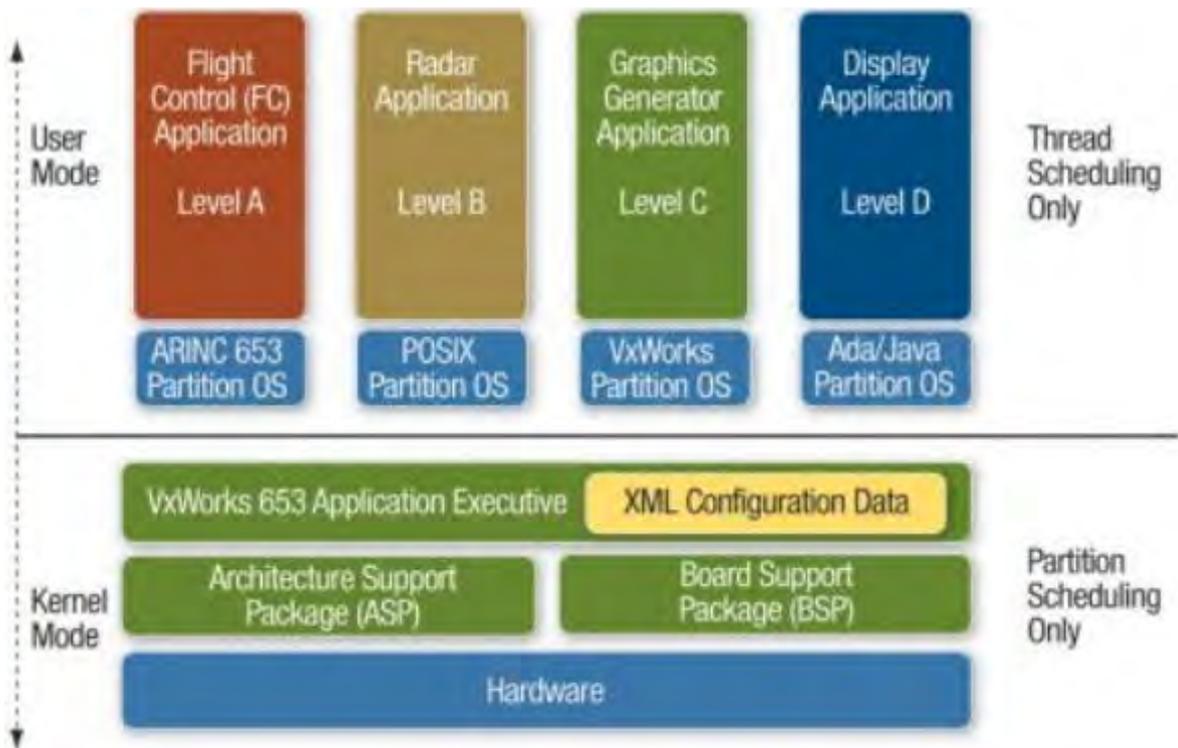


Figura 2.24 - Arquitetura Integrada

Fonte: RTC Group (2010)

A Figura 2.25 ilustra o particionamento utilizado no IMA. Com este sistema de particionamento o computador funciona como se fosse vários computadores virtuais.

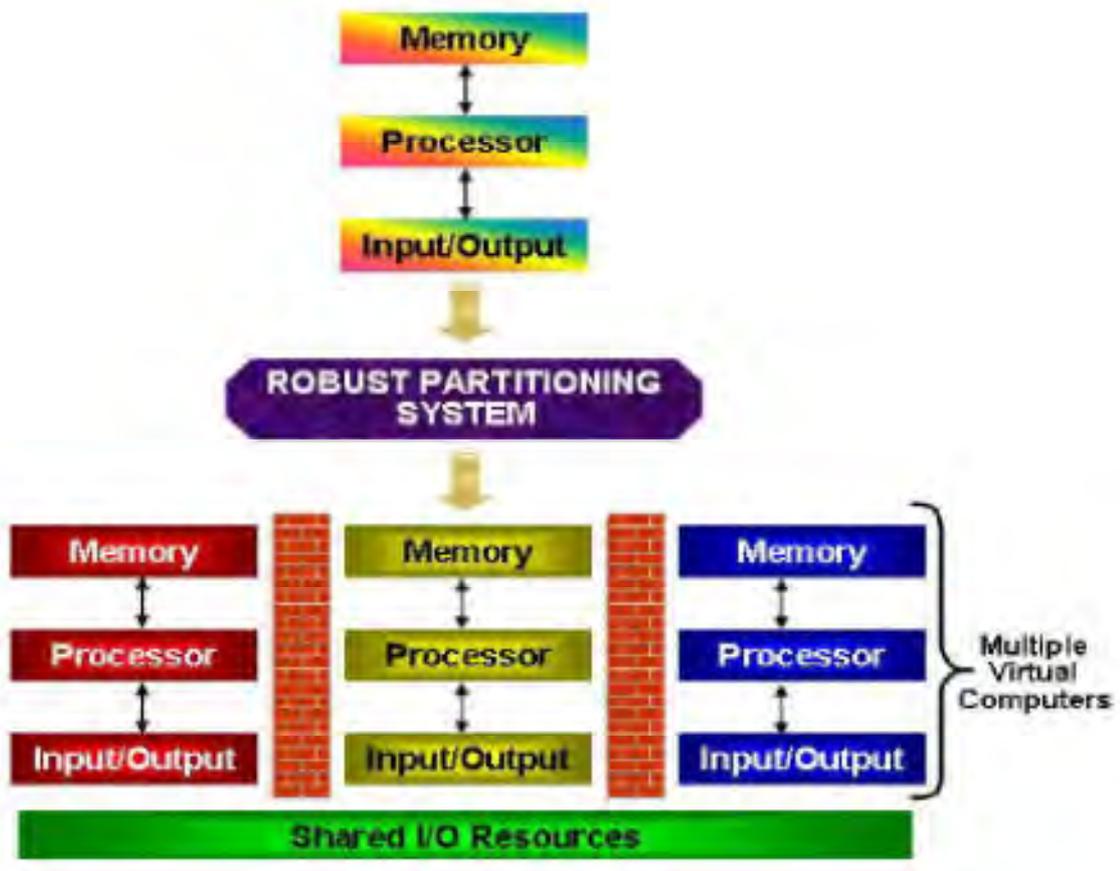


Figura 2.25 - Partições funcionando como múltiplos computadores virtuais

Fonte: BLACK e FLETCHER (2006)

3 AVIÔNICA MODULAR INTEGRADA (*INTEGRATED MODULAR AVIONICS* - IMA)

3.1. ARINC 653

Os padrões ARINC são preparados pelo Comitê de Engenharia Eletrônica das Linhas Aéreas, (“*Airlines Electronics Engineering Committee*” - AEEC), formado por diversas empresas aéreas. De acordo com Obermaisser et al. (2007), o padrão ARINC 651 (*Design Guidance For Integrated Modular Avionics*), define padrões gerais de projeto e implementação da IMA. Já o padrão ARINC 653 (*Avionics Application Standard Software Interface*) é o resultado de requisitos específicos mapeados dentro do ARINC 651 e define especificações de software para particionamentos de tempo e espaço em uma arquitetura IMA. Em outras palavras, ele define como um sistema operacional deve garantir particionamento temporal e espacial robustos.

De acordo com Schoofs et al. (2009), no particionamento temporal, somente uma aplicação, durante um determinado tempo, tem acesso aos recursos do sistema, incluindo o processador. Portanto, não há competição por recursos entre as aplicações. No particionamento espacial, a memória de uma partição é protegida, de modo que nenhuma outra aplicação possa acessar sua área de memória.

O suporte ao particionamento robusto definido pelo ARINC 653 pode ser corretamente alcançado se o Sistema Operacional do Módulo (“*Module Operating System – MOS*”) proporcionar a separação das aplicações, normalmente por contenção de falhas, de modo que a falha de uma função de uma partição não cause a falha de uma função de outra partição.

Os processos dentro de uma partição são agendados ou escalonados por um agendador preemptivo baseado em prioridades com ordem FIFO (“*First In, First Out*”) para processos com a mesma prioridade. Assim que a janela de execução atribuída a uma partição é inicializada, este agendador de segundo

nível é chamado e a partição ganha acesso ao processador. E, no momento em que a janela de execução é finalizada, o agendador de partição de primeiro nível suspende o processo, como mostra a Figura 3.1.

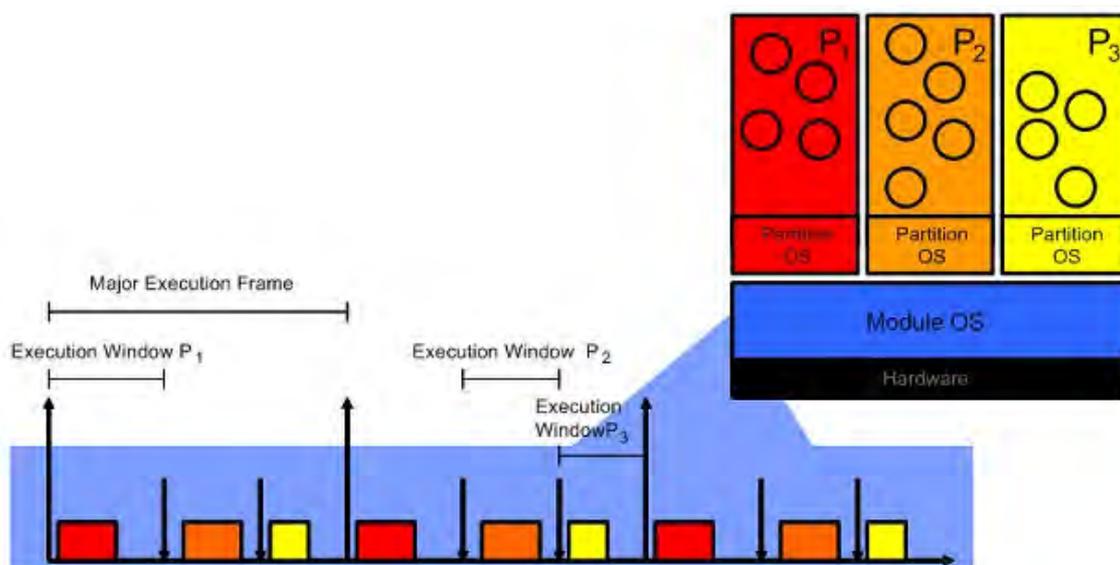


Figura 3.1 - Agendador de Partições

Fonte: GMV IMA Research Group (2011)

A aproximação de agendador de dois níveis é frequentemente associada à tecnologia de virtualização; o agendador de segundo nível é parte de um Sistema Operacional de Partições (“*Partition Operating System - POS*”) que é executado em uma camada virtual provida pelo MOS. O fato do POS não ser executado diretamente no hardware, proporciona a redução de componentes de hardware, pois máquinas virtuais substituem componentes reais utilizados nas LRUs, no sistema federado.

O ARINC 653 também padroniza a Interface de Programação de Aplicações (“*Application Programming Interface – API*”). A API definida pelo ARINC 653 é chamada de APEX, *APplication EXecutive*.

3.2. APEX - APPLICATION EXECUTIVE

A APEX é uma interface de propósito geral entre o sistema operacional e o software aplicativo. Com esta interface padronizada, a plataforma de hardware e o software aplicativo podem ser desenvolvidos separadamente, podendo ser integradas via APEX, na arquitetura IMA. Portanto, os desenvolvedores de subsistemas de controle têm mais flexibilidade na aquisição de subsistemas computadorizados de diferentes fornecedores. Além disso, atualizações ao longo da vida do sistema tornam-se mais baratas.

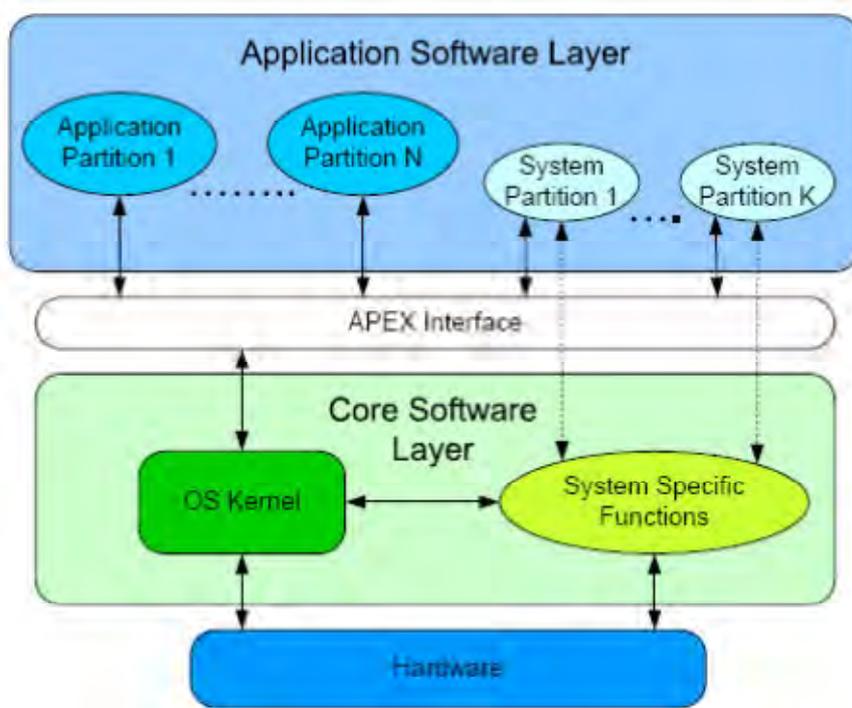


Figura 3.2 - Sistema padrão ARINC 653

Fonte: PASCOAL et al. (2011)

Na Figura 3.2, a *'System Partition'* é um conjunto de componentes opcionais que interagem com dispositivos de hardware específicos. Para tanto, o sistema deve ter uma série de funções específicas do sistema (*"System Specific Functions"*), (PASCOAL et al., 2011).

De acordo com Kopetz e Obermaisser (2009), a APEX provê serviços de gerenciamento de partição, gerenciamento de processo, gerenciamento de memória, gerenciamento de tempo, comunicação interpartição, comunicação intrapartição e diagnose (*Health Monitor*).

Em ARINC (2005), encontra-se a explicação de cada serviço, como mostrado a seguir:

3.2.1. Gerenciamento de partição e memória

Como já dito, no IMA, um sistema operacional é baseado no conceito de particionamento robusto. Este conceito consiste dos particionamentos temporal (acesso ao processador, I/O, etc.) e espacial (particionamento de memória).

No particionamento temporal, uma janela principal de tempo é definida e é executada repetidamente durante um tempo predeterminado. Dentro desta janela, são criadas subjanelas onde são alocadas as partições. A ordem de execução das partições e tempo de execução são definidas em tabelas de configuração. A janela principal pode ter diversas partições de diferentes tempos de execução.

No particionamento espacial, áreas únicas de memória são predeterminadas para cada partição. A identificação dessas áreas únicas é feita a partir de requisitos individuais de cada partição, e variam em tamanho e privilégios de acesso.

3.2.2. Gerenciamento de processo

Processo é uma unidade de programa contida dentro da partição que é executada concorrentemente com outros processos desta partição. Cada partição, no IMA, pode alocar diversos processos que se combinam dinamicamente para cumprir as funções associadas àquela partição. Para se tornar ativo, o processo precisa ser criado e inicializado.

O acesso às funções de gerenciamento de processos é feito utilizando os serviços da APEX. Cada processo tem um nível de prioridade, assim os processos podem sofrer preempção, ser finalizados etc. (Figura 3.3).

Deste modo, o sistema operacional pode ver o processo em diversos estados:

Dorment (*Dormant*) – processo inelegível para receber recursos. Um processo se encontra nesse estado antes de ser inicializado ou após ter sido terminado.

Pronto (*Ready*) – processo elegível para agendamento. Um processo se encontra nesse estado quando está pronto para ser executado.

Executando (*Running*) – processo sendo executado pelo processador. Um processo se encontra nesse estado quando é o processo em execução.

Esperando (*Waiting*) – processo não autorizado a receber recursos até que um evento particular ocorra (*delay, semaphore, event, etc.*).

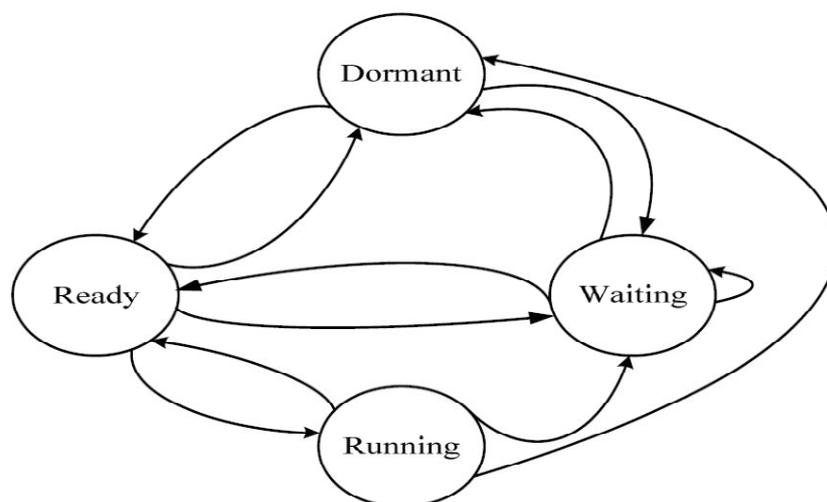


Figura 3.3 – Diagrama de estados do processo

Fonte: ARINC (2005)

3.2.3. Gerenciamento de tempo

No IMA, o tempo é único e independente da execução de partição dentro do módulo. Todos os valores e capacidades de tempo são relacionados com esse tempo único e não são relativos a nenhuma execução de partição. O sistema operacional provê divisão de tempo para agendamentos, prazos fatais (*deadlines*), periodicidade, atrasos (*delays*), etc.

3.2.4. Comunicação interpartição

A parte de maior importância deste padrão é a comunicação entre partições. Comunicação interpartição é um termo genérico utilizado neste padrão primeiramente para comunicação entre duas ou mais partições executadas no mesmo módulo ou em módulos diferentes. No entanto, essa comunicação pode ocorrer com equipamentos externos ao módulo.

Toda comunicação é feita através de mensagens, que é um bloco contínuo de dados de tamanho finito. No nível da aplicação, as mensagens são consideradas entidades atômicas, ou seja, a mensagem só é recebida se estiver inteira, caso contrário, nada é recebido. A comunicação interpartição faz uso de *Sampling ports* e *Queuing ports*. Nas *Sampling ports*, as mensagens são enviadas e não há enfileiramento de mensagens, pois, caso a mensagem não seja lida, a mensagem mais recente sobre-escreve a mais antiga. Isso permite que a partição de origem envie mensagens a qualquer momento, e que a partição de destino sempre leia a última mensagem enviada. Já nas *Queuing ports* as mensagens são organizadas em formato de fila. Desta forma, nenhuma informação é perdida, pois as mensagens enviadas pela partição de origem são armazenadas até que a partição de destino às acesse.

Para se comunicarem, as partições utilizam portas. Portas são áreas de memória, dentro da partição, onde mensagens são escritas ou lidas por um programa. Se as portas forem conectadas a um canal, as mensagens em uma porta de origem (*source port*) são copiadas para área de memória da porta de

destino (*destination port*). Este mecanismo de transporte é invisível ao software aplicativo. Também é transparente para o software aplicativo onde a outra porta está alocada, seja na partição no mesmo módulo ou em outro computador.

Os canais são definidos na configuração do ARINC 653 como uma relação entre uma porta de origem e uma ou mais portas de destino. O padrão ARINC 653 deixa em aberto se as mensagens são enviadas para um destino ou todos, no entanto, dá suporte a *unicast*, *multicast* e *broadcast*. No *unicast* a mensagem é enviada de uma porta de origem para uma porta de destino. No *multicast* a mensagem é enviada de uma porta de origem para mais de uma porta de destino. E no *broadcast* a mensagem é enviada de uma porta de origem para todas as portas de destino (Figura 29).

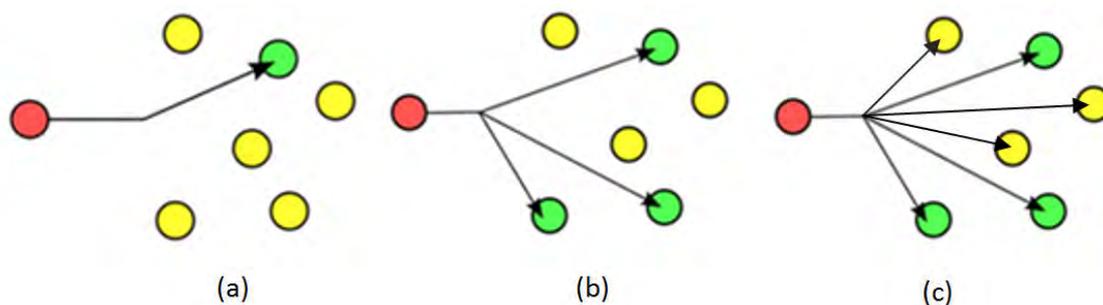


Figura 3.4 - (a) Unicast; (b) Multicast; (c) Broadcast

Fonte: Adaptado de Unicast (2013)

3.2.5. Comunicação intrapartição

Os mecanismos de comunicação intrapartição são *buffers*, *blackboards*, *semaphores* e *events*. Neste tipo de comunicação os *buffers* funcionam como as *queuing ports* da comunicação interpartição, enquanto que os *blackboards* funcionam como as *sampling ports*. Todas as mensagens são consideradas entidades atômicas, ou seja, a mensagem só é recebida se estiver inteira; caso contrário, nada é recebido. Já os *semaphores* e *events* são meios de

sincronização. Os *semaphores* possibilitam acesso controlado aos recursos, enquanto os *events* dão suporte controle de fluxo entre processos (notificando processos em estado de espera sobre a ocorrência de uma condição).

3.2.6. Diagnose (*Health Monitor*)

A diagnose é a função do sistema operacional responsável pelo monitoramento e comunicação de erros e falhas de hardware, da aplicação e do próprio sistema operacional. A diagnose ajuda a isolar erros e prevenir que falhas se propaguem.

A diagnose utiliza tabelas de configuração para lidar com os erros que possam ocorrer. Há tabelas para as partições, para o módulo e para todo o sistema.

3.3. Vantagens do IMA

3.3.1. Tamanho, massa e consumo de energia reduzidos

O conceito IMA é baseado no compartilhamento de recursos, de modo que o mesmo processador e sua infraestrutura associada são utilizados por várias aplicações de diferentes níveis de criticalidade. Hardware reduzido significa tamanho, massa e consumo de energia reduzidos. Conseqüentemente, os custos também são reduzidos.

3.3.2. Competitividade

Como confirma o RTC Group (2010), em arquiteturas federadas, o fornecedor de uma LRU é responsável por seu desenvolvimento completo (hardware e software), implementação, testes e certificação da unidade de acordo com as normas DO-178B (agora DO-178C) e DO-254. Portanto, uma LRU é tipicamente certificada para um único nível de segurança das normas DO-178B (agora DO-178C) e/ou DO-254. Deste modo, se uma nova funcionalidade é exigida, o problema é frequentemente resolvido pela substituição da LRU. Além

disso, se uma simples mudança em uma linha de código do software é feita, uma requalificação completa de toda a LRU deve ser feita.

Na IMA, as aplicações são separadas da plataforma computacional pela APEX. Assim, a plataforma e as aplicações podem ser escolhidas de diferentes fornecedores, gerando assim uma maior gama de opções. Além disso, a APEX permite o uso de aplicações de diferentes níveis de criticalidade no mesmo módulo.

3.3.3. Modularidade, Portabilidade e Reuso

Com a APEX intermediando as aplicações e plataformas computacionais, módulos padronizados para tarefas comuns que poderiam ser reusados por diferentes plataformas podem ser criados. Assim, como estes módulos seriam produzidos em grande escala, o custo seria reduzido. Além disso, os softwares das aplicações também são modulares permitindo sua reutilização em diferentes plataformas.

3.3.4. Certificação Incremental

A robustez da norma ARINC 653 permite a certificação incremental devido ao isolamento de hardware e software e à modularidade. Assim, diferentemente de arquiteturas federadas, em arquiteturas baseadas no conceito IMA não é necessário retestar e recertificar a plataforma inteira, mas somente retestar a alteração realizada. Além disso, aplicações de diferentes níveis de criticalidade podem ser certificadas no mesmo módulo.

3.3.5. Alocação de Recursos

Este tipo de arquitetura utiliza tabelas de configuração alocando os recursos computacionais para as devidas funções hospedadas. Manipulando estas tabelas de configuração, o integrador do sistema tem a flexibilidade de dinamicamente gerenciar recursos disponíveis.

Além disso, segundo Watkins e Walter (2007), o IMA tem a capacidade de reservar um conjunto de recursos para ser alocado por qualquer função hospedada. Assim, o integrador do sistema é capaz de variar, dentro de certos limites, a alocação de recursos para uma dada função no futuro; ou adicionar novas funções sem a necessidade de adicionar novos recursos computacionais. A alocação de recursos não é feita enquanto o sistema é utilizado devido a problemas de certificação. No entanto, isso pode ser feito dinamicamente através de dados de configuração do sistema atualizados; e então ser certificado para uso em serviço.

3.3.6. Reconfiguração e robustez

Segundo Alena et al (2007), neste tipo de arquitetura, os módulos são configurados por software e há um caminho em potencial entre qualquer um deles. Essa característica permite que os módulos respondam a falhas que possam ocorrer. Deste modo, eles podem se adaptar a mudanças no funcionamento da rede ou nos modos de operação. Portanto, em caso de falhas, o sistema se auto-reconfigura, em pré-determinadas configurações, o que torna o sistema muito robusto.

3.4. Desvantagens

3.4.1. Processo de Integração Complexo

Na IMA, a plataforma de base e a aplicação são separadas de modo que possam ser fornecidas por diferentes fornecedores, gerando mais competitividade, mas gerando também um processo de integração mais complexo. De acordo com Watkins e Walter (2007), antes da transição de arquiteturas federadas para arquiteturas integradas, o integrador do sistema deve estar confiante em suas habilidades de executar o processo de integração. Este processo inclui um maior gerenciamento e definição de interfaces, gerenciamento e alocação de recursos, além de geração e análise de configuração de sistema.

4 FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA A SUA SOLUÇÃO

4.1. Formulação do problema

O problema tratado neste trabalho é o estudo de um Sistema de Controle de Atitude (SCA) compatível com os satélites em desenvolvimento no INPE e sua implementação em uma arquitetura compatível com uma Aviônica Modular Integrada (*Integrated Modular Avionics* - IMA) em desenvolvimento e uso nos mais modernos programas aeroespaciais (presentes e futuros), para apreciar as vantagens e desvantagens da adoção de tal arquitetura em tais sistemas.

4.2. Abordagens para sua solução

Para estudar tal sistema podemos adotar as abordagens de: 1) análise; 2) modelagem e simulação; e 3) experimentos de um SCA: 1) genérico e novo; 2) específico e constante na literatura; e comparar os nossos resultados com resultados 1) genéricos e novos; 2) específicos e constantes na literatura, sob as mesmas condições.

Para implementar tal sistema podemos adotar abordagens compatíveis com as anteriores e que usem: 1) um só computador físico; 2) vários computadores físicos; dividido(s) em várias partições ou computadores virtuais realocáveis dinamicamente, mas idealmente independentes a ponto de serem imunes a falhas entre si.

Pela disponibilidade de resultados constantes na literatura, sob as mesmas condições, escolhemos modelar e simular o Sistema de Controle de Atitude compatível com o Modo Nominal = Modo de Controle da Missão (*"Mission Control Mode"*) do satélite Amazônia-1, em desenvolvimento no INPE.

Pela facilidade de implementação escolhemos implementar tal SCA em 1 computador modelado e simulado num Simulador de Aviônica Modular

Integrada (Simulated Integrated Modular Avionics - SIMA); e também em ambiente MatLab e na linguagem C (GCC+Linux), para comparações.

Para tanto, detalharemos: o SCAO do satélite Amazônia-1 e a Arquitetura IMA no resto deste Capítulo 4; no capítulo 5 trataremos dos programas e ambientes utilizados na simulação. Assim trataremos na seção: 5.1 o ambiente computacional utilizado; 5.2 o ambiente MatLab; 5.3 a linguagem C e o compilador GCC; e 5.4 o Simulador de IMA (SIMA).

4.3. O SCAO do Satélite Amazônia-1

Em torno de 2000, o INPE iniciou o estudo de uma Plataforma MultiMissão (PMM) para servir de modelo para os futuros satélites brasileiros. Trata-se de uma plataforma geral para satélites na classe de aproximadamente 500 kg (incluindo carga útil de aproximadamente 280 kg), na qual devem se basear uma série de futuros satélites do INPE, para operar em órbita baixa da Terra (*“Low Earth Orbit – LEO”*).

O satélite Amazônia-1, com lançamento previsto para 2015, será o primeiro satélite de recursos terrestres desenvolvido pelo Brasil, construído com base na Plataforma MultiMissão. O satélite Lattes será o segundo. A principal diferença entre os SCAOs do Amazônia-1 e do Lattes é que, o Amazônia-1 terá dois computadores dedicados para os subsistemas, ou seja, um computador para a Manipulação de Dados a Bordo (*“Onboard Data Handling – OBDH”*) e outro para a Eletrônica de Controle de Atitude (*“Attitude Control Electronics - ACE”*). O Sistema de Controle de Atitude e Órbita e o Sistema de Manipulação de Dados a Bordo foram encomendados à INVAP, empresa argentina qualificada pela NASA para o desenvolvimento de satélites. Já no Lattes, o Sistema de Controle de Atitude e Órbita e o Sistema de Manipulação de Dados a Bordo serão planejados de tal forma a se ter ambas as funções em um único

sistema; o ACDH (“Attitude Control and Data Handling”) hospedados em um único computador.

Consistente com a missão de monitoramento, o Amazônia-1 deverá produzir imagens com maior frequência e maior definição, adequadas para monitorar o ambiente e gerenciar recursos naturais. O satélite será estabilizado em três eixos com o apontamento de uma câmera para a Terra. O Sistema de Controle de Atitude e Órbita, cujo Modo de Operação Nominal é objeto do presente trabalho, conterà atuadores e sensores que permitirão satisfazer os requisitos da missão (Figura 4.1).

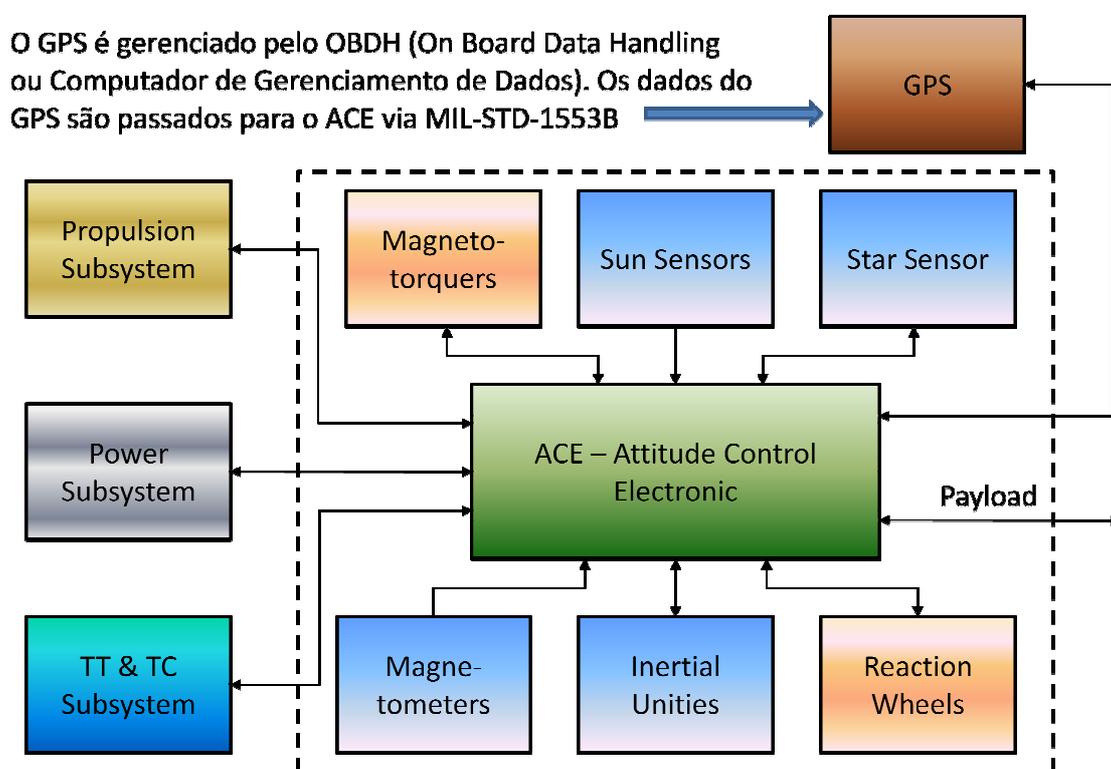


Figura 4.1 - Subsistema de Controle de Atitude e Órbita do satélite Amazônia-1

Fonte: Adaptado do Projeto Amazônia-1 (2009)

A Figura 4.1 ilustra o SCAO do satélite Amazônia-1 mostrando o HW e a interface com três outros subsistemas, além da interface com o OBDH.

O subsistema é composto de sensores (magnetômetros, unidades inerciais, sensores solares, sensores de estrelas), atuadores (bobinas magnéticas, rodas de reação), e o ACE.

O SCAO deve prover as seguintes funções de controle de atitude e órbita:

- Controle de atitude estabilizado em três eixos no Modo Nominal, permitindo apontamento para Terra;
- Determinação de atitude a bordo;
- Aquisição e manutenção segura de atitude após a fase de lançamento ou ocorrência de falha;
- Controle do posicionamento/velocidade dos painéis solares;
- Determinação a bordo da posição e velocidade do satélite;
- Controle dos propulsores para aquisição e manutenção de órbita;
- Dessaturação das rodas de reação.

Os Modos de Operação do SCAO do Amazônia-1 são (Figura 4.2):

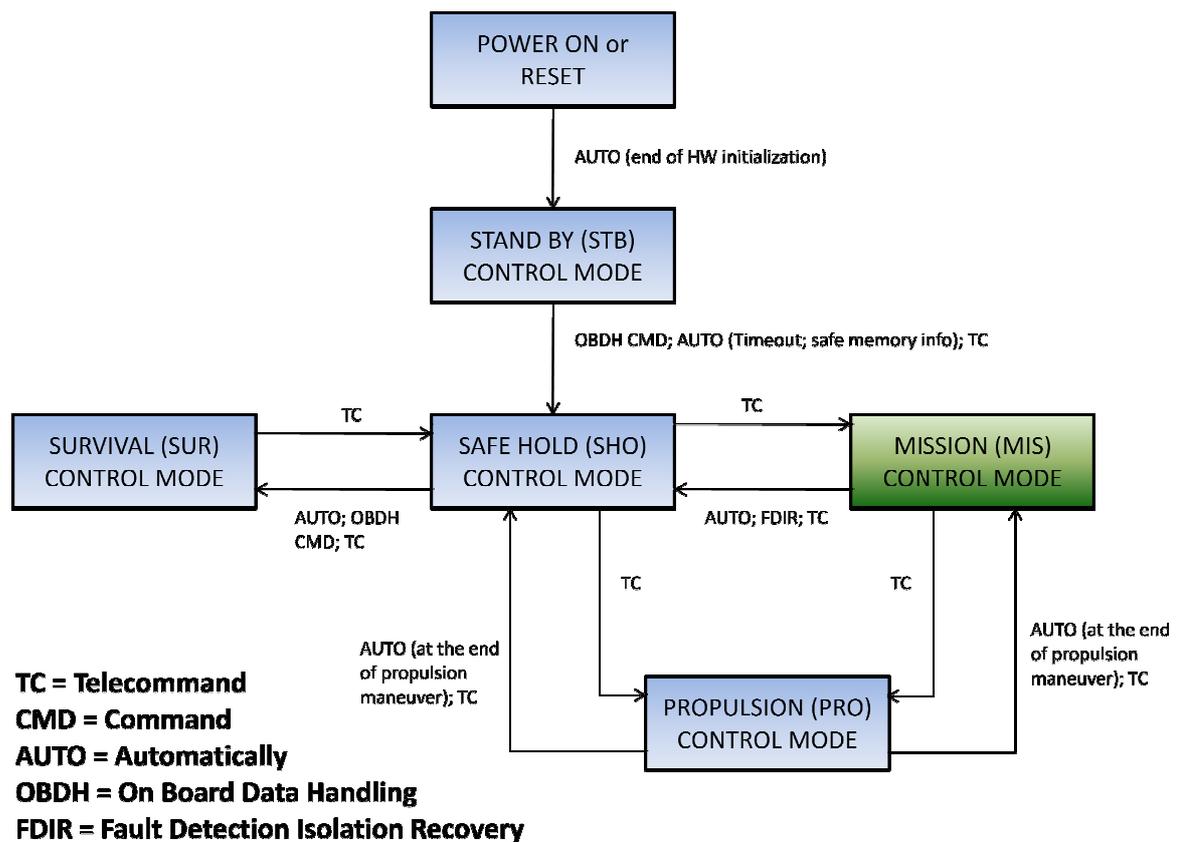


Figura 4.2 - Modos de controle do SCAO do satélite Amazônia-1

Fonte: Adaptado do Projeto Amazônia-1 (2009)

No Modo Nominal = Modo de Controle da Missão (“*Mission Control Mode*”), o satélite será configurado de forma que sua carga útil cumpra sua missão. A atitude do satélite é mantida em direção ao alvo.

Neste modo, a atitude do satélite bem como a taxa de variação da mesma deve ser controlada nos três eixos para cumprir com os seguintes requisitos:

- a) Precisão de apontamento: $< 0.05^\circ$ (3σ);
- b) Estabilidade:
- i) Deriva (“Drift”): $< 0.001^\circ/s$ com frequências até 2 Hz.

- ii) Variação pico a pico $< 0.005^\circ$ com frequências de 2 Hz a 10 Hz.
- c) Jitter $< 0.0005^\circ$ (1σ) com frequências acima de 10 Hz até 100 Hz.
- d) Determinação de atitude: $\leq 0.005^\circ$ (3σ).
- e) Desvio (“off pointing”) de até 30° em torno do eixo de rolamento em 180 segundos.

O trabalho de análise para confirmar a eficiência dos atuadores no controle da atitude requer a modelagem matemática e a simulação computacional do SCAO e seu ambiente. Trabalhos abordando o controle de atitude em três eixos podem ser vistos em (KAPLAN, 1976) e (WERTZ, 1978).

Para implementação das simulações podem ser utilizados ambientes computacionais existentes. Normalmente, utilizam-se ambientes com o software hospedado em um computador e, posteriormente, o software embarcado é testado utilizando-se sistemas federados com diferentes computadores comunicando-se com o computador de bordo onde está embarcado o software de bordo.

O intuito deste trabalho é o estudo de um sistema de controle de atitude compatível com o Modo Nominal do satélite Amazônia-1 (em verde na Figura 4.2) e sua implementação em um simulador de Aviônica Modular Integrada.

4.4. Arquiteturas de desenvolvimento, modelagem e simulação

De acordo com Schoofs et al. (2009), para o desenvolvimento de sistemas embarcados de tempo real são necessárias duas plataformas: a plataforma de desenvolvimento e a plataforma alvo. A plataforma de desenvolvimento é composta de um sistema operacional de propósito geral como o Windows ou

Linux. E seu hardware é basicamente um computador pessoal comum. Esta plataforma deve possuir funções adequadas para o desenvolvimento de aplicações embarcadas (depuração, execução passo a passo, indicadores, etc.). Já a plataforma alvo típica contém um sistema operacional capaz de gerenciar as aplicações embarcadas (ler, configurar e comandar sensores e atuadores através de interfaces de software) em tempo real. Além disso, o hardware da plataforma alvo tem que satisfazer os requisitos da missão espacial, como tamanho, massa, nível de vibração, variação de temperatura, resistência a choque, etc.

Uma aplicação em tempo real é uma aplicação em que a correção da aplicação depende não só da correção da avaliação das funções matemáticas e lógicas (correção funcional) bem como da correção do cumprimento das características temporais destas (correção temporal). Aplicações em tempo real fornecem uma ação ou uma resposta a um evento externo de uma forma previsível e apropriada de tempo. A plataforma de desenvolvimento não é capaz de prover comportamento de tempo real rígido (*“Hard Real Time”*), pois sua principal função é o desenvolvimento. Assim, na melhor das hipóteses, ela oferece comportamento de tempo real flexível (*“Soft Real Time”*).

Correntemente, uma plataforma alvo IMA é extremamente cara e, por esta razão, a substituímos pelo Simulador de IMA, SIMA (ambiente de SW), neste trabalho.

4.5. Motivação para o uso da plataforma IMA

Na área espacial, o IMA tem sido alvo de estudo intenso em diversos centros da NASA, da ESA, da indústria espacial, aeronáutica e automotiva. As tradicionais arquiteturas federadas estão sendo substituídas por este novo conceito.

Arquiteturas federadas são baseadas em plataformas computacionais dedicadas para aplicações específicas enquanto que, no IMA, diversas aplicações compartilham a mesma plataforma computacional. Tal configuração computacional associada a diversas aplicações gera uma redução considerável de massa, volume e consumo do subsistema, além de outras vantagens já mencionadas.

Considerando as vantagens do conceito IMA, as agências espaciais NASA, ESA e diversas empresas voltadas ao mundo espacial têm estudado profundamente o novo conceito.

De acordo com Di Vito (2002), o Centro de Pesquisas Langley (LaRC) da NASA, tem investigado problemas de particionamento de aviônica. O intuito destas pesquisas é de assegurar o particionamento seguro e a não interferência lógica entre aplicações sendo executadas separadamente em apenas uma unidade de processamento. Ainda segundo o autor, estes estudos foram fortemente influenciados pelos esforços de padronização da Comissão Radio-Técnica para a Aeronáutica (*“Radio Technical Commission for Aeronautics – RTCA”*), que resultaram no padrão ARINC 653.

Diniz e Rufino (2005) propõem o uso do padrão ARINC 653 no mundo espacial. Seus principais argumentos são que a maioria dos requisitos da aviação civil são também requisitos do mundo espacial. Além disso, eles destacam os benefícios em termos de certificação modular e o uso de componentes Comerciais de Prateleira (*“Commercial Off-The-Shelf – COTS”*).

De acordo com Alena et al (2007), o Centro de Pesquisas Ames (*“Ames Research Center”*) da NASA, apresentou um artigo onde os autores definem e analisam arquiteturas adequadas, padrões e projetos conceituais para módulos computacionais IMA para aplicabilidade em ônibus espaciais.

Segundo Santos et al. (2008), a Agência Espacial Europeia (*“European Space Agency – ESA”*), através da empresa Skysoft (agora parte do grupo GMV),

desenvolveu um ambiente capaz de executar e verificar aplicações de acordo com o padrão ARINC 653. Segundo os autores, com este ambiente é possível desenvolver e verificar aplicações espaciais. Além disso, como não é necessário um sistema operacional de tempo real pago e uma plataforma alvo, os custos são reduzidos.

De acordo com McCabe e Baggerman (2009), o Veículo Tripulado de Exploração Orion da NASA, (“*Orion Crew Exploration Vehicle – CEV*”), parte do “*Constellation Program*”, implementou uma arquitetura IMA, pois seus requisitos de tolerância a falhas, escalabilidade e disponibilidade não poderiam ser atingidos utilizando uma arquitetura federada tradicional.

De acordo com Mangieri e Vice (2010), o Laboratório de Análises de Engenharia Kedalion da NASA, juntamente com o principal contratante do Orion, validaram modernas técnicas baseadas no IMA adaptando-as a um veículo espacial.

Na área aeronáutica, onde o conceito IMA nasceu, é onde a arquitetura integrada alcançou seu maior amadurecimento. Projetos de grande visibilidade como o 787 da Boeing e o A380 da Airbus utilizam o conceito IMA. Deste modo, a Boeing conseguiu eliminar mais de 100 LRUs e diminuir 2000 libras de massa de sua aeronave, segundo o RTC Group (2010). Já a Airbus conseguiu reduzir pela metade o número de unidades de processamento, segundo a Avionics Magazine (2007).

Na área automotiva, arquiteturas federadas são constituídas de unidades funcionais contidas em plataformas de hardware denominadas *Electronic Control Units* (ECUs). De acordo com Di Natale e Sangiovanni-Vincentelli (2008), este tipo de arquitetura não é mais adequado aos sistemas automotivos modernos devido ao seu crescente número de funções e aplicações. Ainda segundo os autores, isto leva a uma proliferação de ECUs e cabeamentos ocupando mais espaço, além de aumentar massa, custos e riscos de falhas

devido ao grande número de conectores. Deste modo, os autores propõem o uso de arquiteturas integradas para superar estes problemas.

4.6. A utilização da IMA nos satélites brasileiros

Como dito anteriormente, o satélite Amazônia-1 terá o controlador de atitude (*Attitude Controller*) em um computador e a manipulação de dados (*Data Handling*) em outro. A utilização do IMA neste caso, a primeira vista, não parece vantajosa, pois a redução de massa não seria tão significativa. No entanto, pensando-se em redundância, o IMA é uma excelente opção, devido à sua capacidade de reconfiguração. Desta forma, para aumentar a segurança a falhas exponencialmente pode-se utilizar também dois computadores. No entanto, cada computador deve ser totalmente diferente do outro, pois dois computadores exatamente iguais têm chances de sofrerem as mesmas falhas. Cada computador hospeda ambos, controlador de atitude e manipulação de dados. Desta forma, se uma falha ocorrer em um computador o IMA reconfigura o sistema passando o comando para o outro computador.

5 PROGRAMAS E AMBIENTES UTILIZADOS NA SIMULAÇÃO

5.1. Ambiente Computacional

As simulações foram efetuadas utilizando-se um computador comum com processador Intel Core i7 de 2.93 Ghz e 8 GB de memória RAM. O sistema operacional utilizado para as simulações em MatLab foi o Microsoft Windows 7. E o MatLab utilizado foi a o MatLab 7.10 (2010a). Já para as simulações em linguagem C e SIMA utilizou-se o sistema operacional Linux Ubuntu 10.04, Kernel 2.6.32.24, GNOME 2.30.2. O compilador utilizado para a programação foi o GCC (GNU Compiler Collection) 4.4.6 do Linux.

5.2. MatLab

O MatLab (*Matrix Laboratory*) é um ambiente de computação numérica desenvolvido pela *MathWorks* (Figura 5.1). Este ambiente trabalha com a manipulação de matrizes, implementação de algoritmos, criação de interfaces de usuário etc. Além disso, o MatLab trabalha com *toolboxes*, que são módulos de códigos com funções específicas (MATLAB, 2013). Essas funções facilitam a simulação de diversos sistemas. Apesar da existência destas *toolboxes*, neste trabalho as mesmas não foram utilizadas e, fez-se a modelagem da mesma maneira que se faria em linguagem C, pois a modelagem seguinte seria nesta linguagem.

A modelagem no MatLab foi feita utilizando os parâmetros orbitais apresentados em **6.3.6.1**; os dados do satélite e da roda de reação apresentados em **6.3.6.2**; os ganhos do PID apresentados em **6.3.6.3**; e as condições iniciais apresentadas em **6.3.6.4**. O funcionamento do programa é apresentado no fluxograma apresentado em **6.3.6.5**.

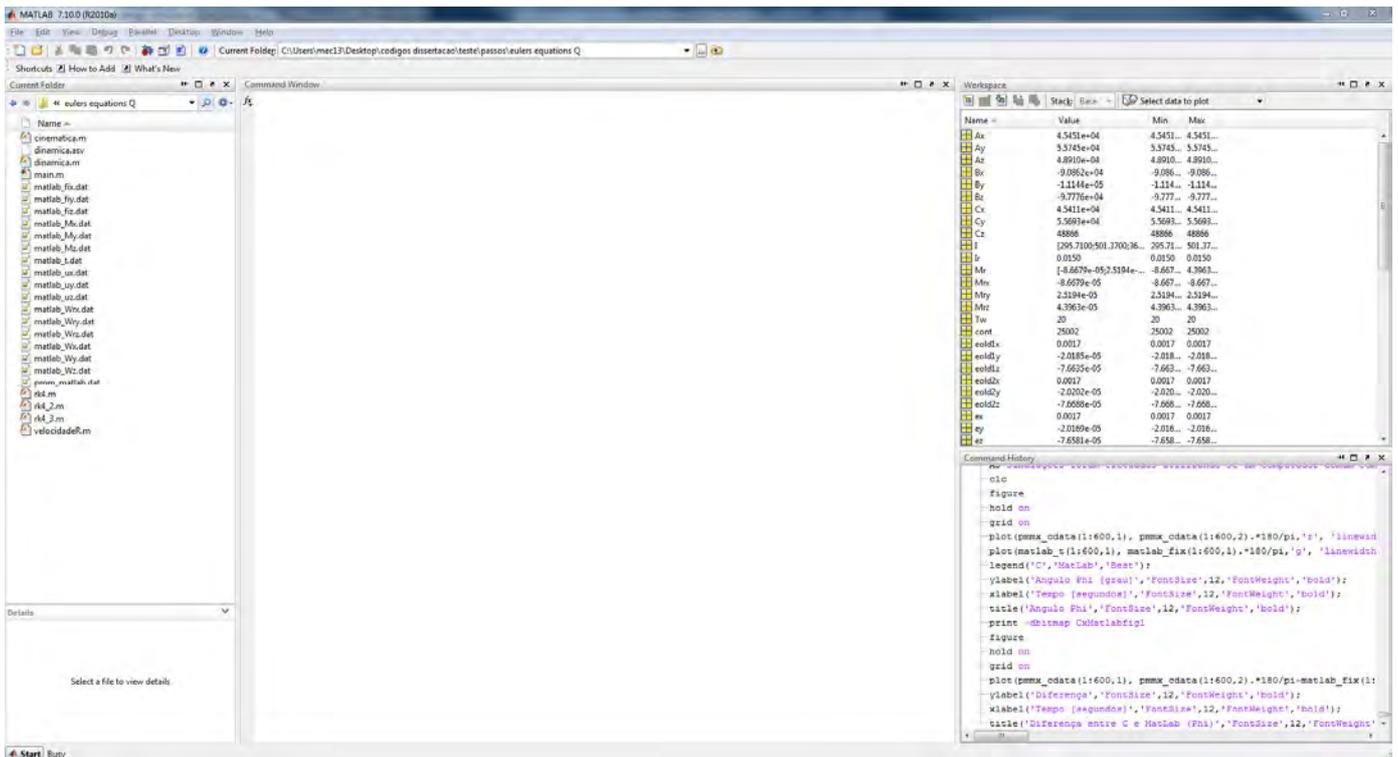


Figura 5.1 – Tela do MatLab

5.3. Linguagem C

O C é uma linguagem de programação de propósito geral largamente utilizada. Poucas são as arquiteturas que não contém um compilador C (C - LINGUAGEM DE PROGRAMAÇÃO).

Como já dito, o compilador utilizado foi o GCC. O GCC (GNU *Compiler Collection*) é um conjunto de compiladores de linguagens de programação que compilam o código fonte em binários executáveis (GNU *COMPILER COLLECTION*).

A modelagem em C foi feita utilizando os parâmetros orbitais apresentados em 6.3.6.1; os dados do satélite e da roda de reação apresentados em 6.3.6.2; os ganhos do PID apresentados em 6.3.6.3; e as condições iniciais apresentadas em 6.3.6.4. O funcionamento do programa é apresentado no fluxograma apresentado em 6.3.6.5.

5.4. O SIMULADOR DE IMA (SIMA)

De acordo com Schoofs et al. (2009) e com o GMV IMA *Research Group* (2011), o Simulador de IMA (“*Simulated Integrated Modular Avionics – SIMA*”) é um ambiente de execução que incorpora a API padronizada pela norma ARINC 653, APEX, a sistemas operacionais que não suportam este tipo de serviços. O propósito do SIMA é o desenvolvimento e a verificação de aplicações espaciais voltadas a plataformas integradas de acordo com o padrão ARINC 653. Com este ambiente de execução é possível desenvolver e verificar as aplicações espaciais a baixos custos.

Os serviços implementados no SIMA são: gerenciamento de partições, gerenciamento de processos, gerenciamento de tempo, comunicação nas partições, comunicação entre partições, “*health monitoring*”, agendamento de múltiplos módulos e sistema de “*logbook*”.

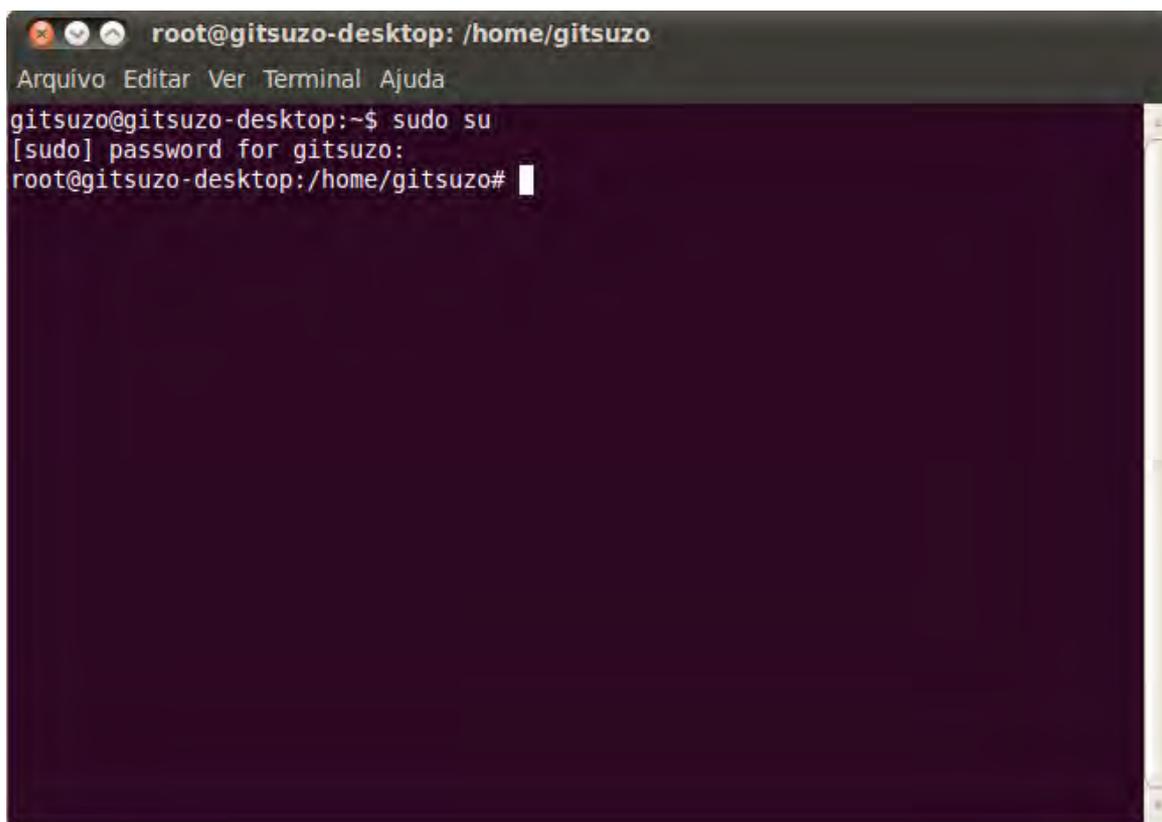
No SIMA, o Sistema Operacional das Partições (“*Partition Operating System - POS*”) é fornecido por uma biblioteca (pos.a), a qual contém a API ARINC 653. Da mesma forma, o Sistema Operacional dos Módulos (“*Module Operating System - MOS*”) é implementado por um programa chamado mos, o qual contém o agendador de partições, configurado nas tabelas de configuração de XML de acordo com os requisitos do ARINC 653.

O SIMA consegue simular diversos módulos em paralelo, contanto que o computador tenha capacidade de processamento suficiente para tanto.

As aplicações a serem simuladas no SIMA foram escritas em linguagem C.

5.5. Instalação

Para a instalação do SIMA no Linux foi necessário ativar o modo *ROOT* (superusuário) o qual tem todos os privilégios, ou seja, acesso total ao sistema (Figura 5.2).



```
root@gitsuzo-desktop: /home/gitsuzo
Arquivo Editar Ver Terminal Ajuda
gitsuzo@gitsuzo-desktop:~$ sudo su
[sudo] password for gitsuzo:
root@gitsuzo-desktop:/home/gitsuzo#
```

Figura 5.2 – Entrando no modo ROOT

O SIMA foi obtido em um arquivo compactado. O manual do SIMA sugere que se descompacte o arquivo em */usr*, */usr/local* ou */opt*. Neste trabalho, o arquivo foi descompactado em */usr* (Figura 5.3).

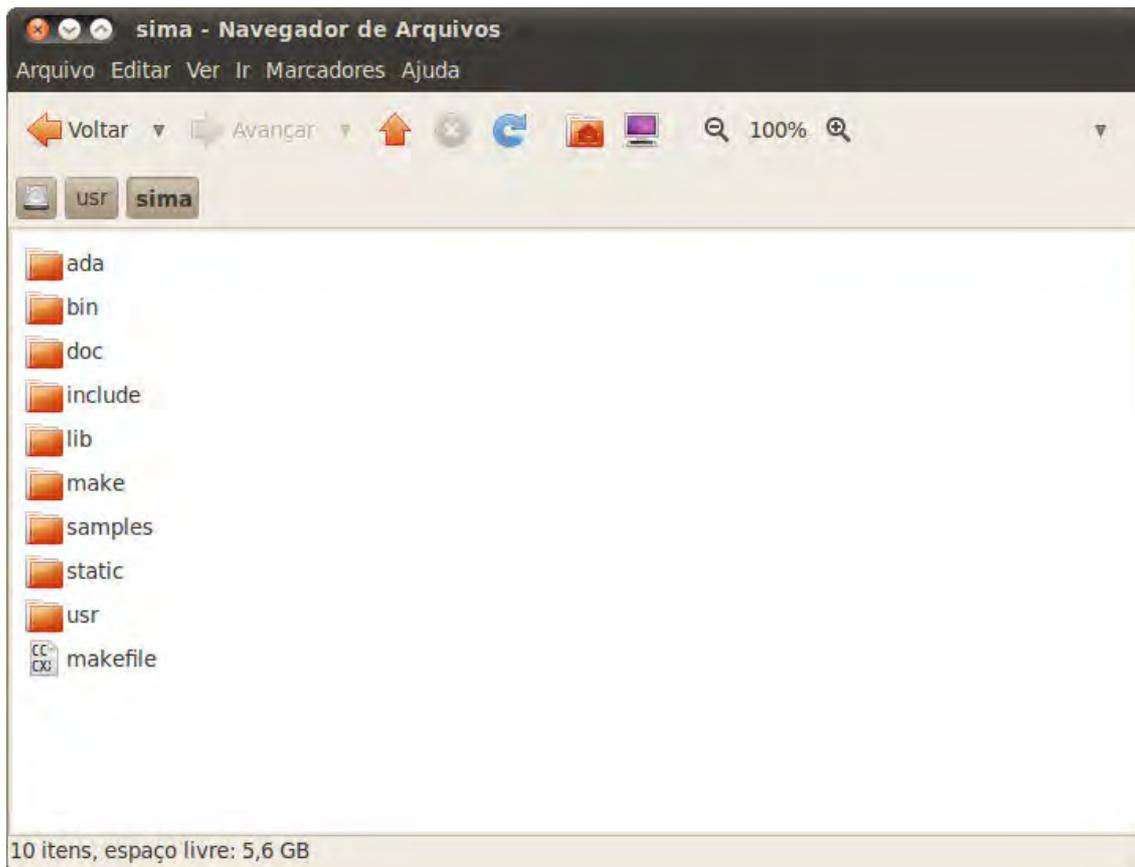


Figura 5.3 – Pasta do SIMA dentro da pasta *usr*

Também é de suma importância se adicionar a pasta *bin* do SIMA ao $\$PATH$ do Linux. $\$PATH$ é uma variável do Linux que indica a trajetória dos binários, de modo que os mesmos possam ser executados sem que se indique seu caminho. Para se alterar o $\$PATH$ foi necessário editar o arquivo */etc/environment*. O caminho adicionado ao $\$PATH$ neste trabalho foi: $PATH="/usr/sima/bin"$ (Figura 5.4).

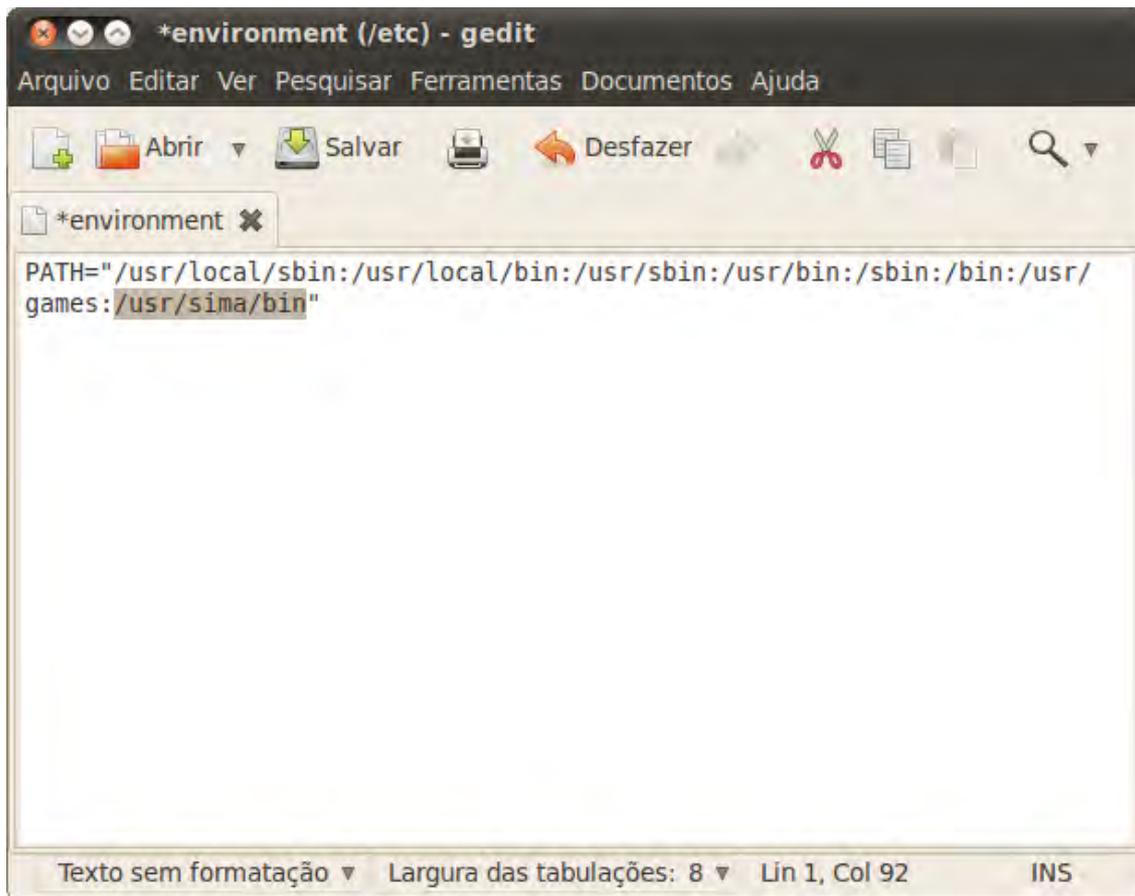


Figura 5.4 – Alteração do PATH do Linux

O SIMA trás alguns arquivos (*makefile*) prontos, que contém instruções, para a compilação de seus exemplos. No entanto, para que os mesmos funcionem é necessário que se adicione o caminho em que o SIMA foi instalado. Para tanto, deve-se defini-lo, também no */etc/environment*, da seguinte maneira: *SIMA_ROOT= "caminho"*. Assim, neste trabalho definiu-se: *SIMA_ROOT= "/usr/sima"* (Figura 5.5).

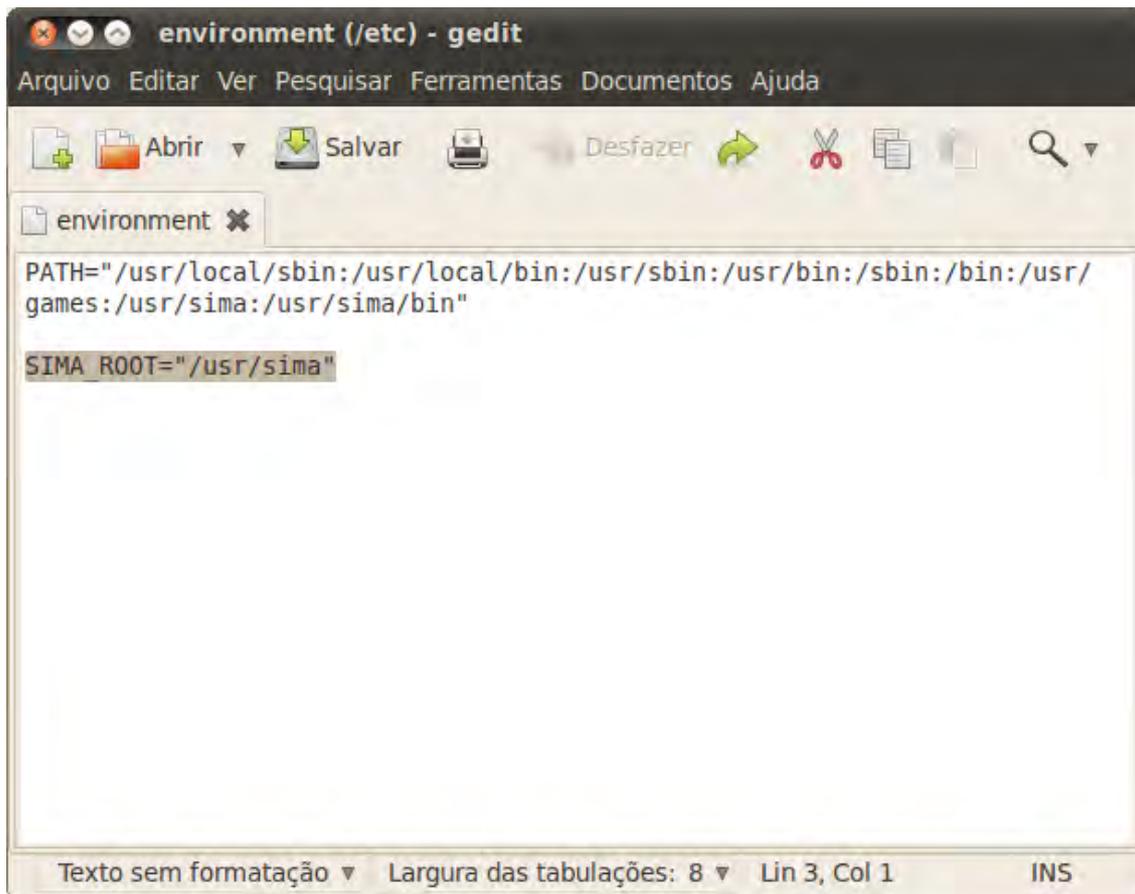


Figura 5.5 - Inclusão do SIMA_ROOT

5.6. Programas-exemplo

O SIMA fornece diversos programas-exemplo. Na pasta */usr/sima/samples/simple* encontram-se os exemplos simples. Já na pasta */usr/sima/samples* encontram-se os exemplos complexos.

Como exemplos simples têm-se: *base.c*, *board.c*, *buf.c*, *evt.c*, *frost.c*, *one.c*, *philos.c*, *prio.c*, *replenish.c*, *shut1.c*, *shut2.c* e *stress.c*. A referência GMV IMA Research Group (2011) detalha as funções de cada exemplo (Figura 5.6).

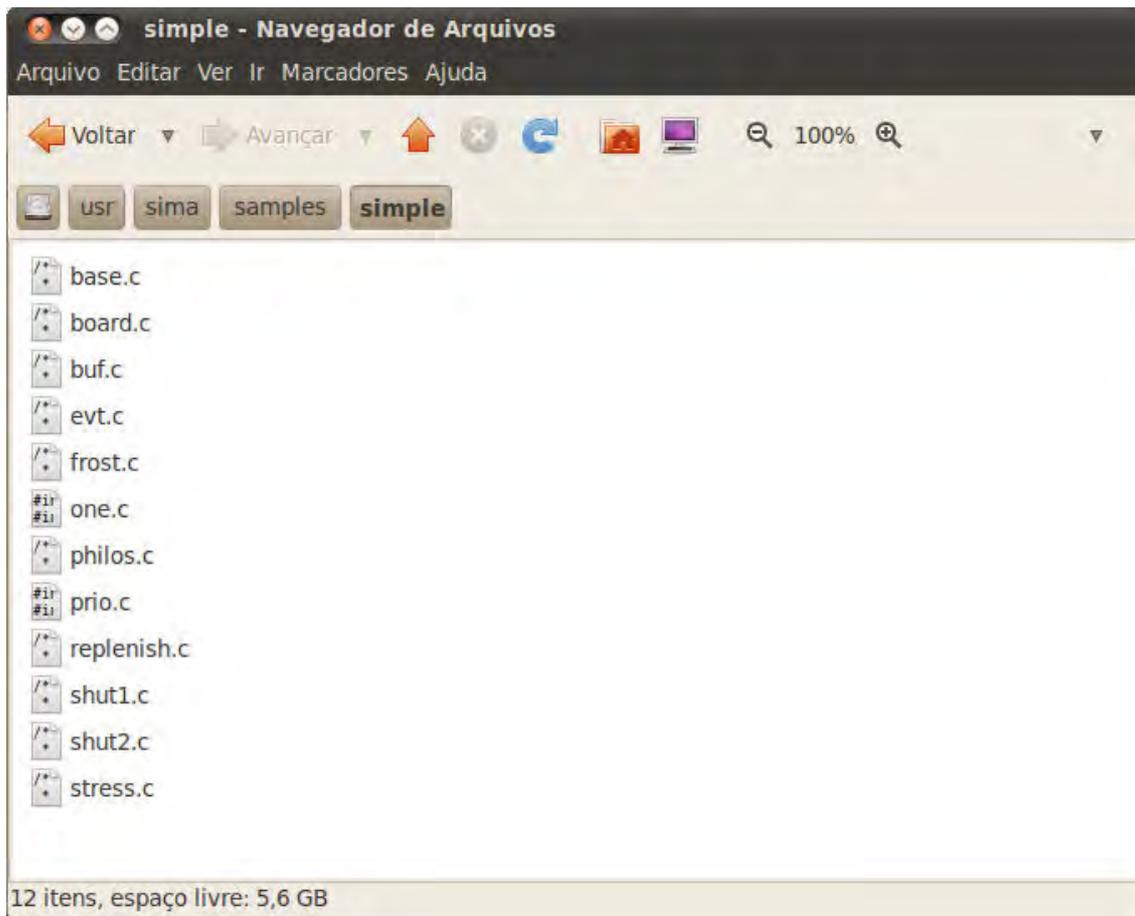


Figura 5.6- Exemplos simples do SIMA

Os exemplos complexos são: *a653*, *book*, *com*, *control*, *mms* e *ports*. Este trabalho foi feito alterando-se o exemplo *com* (Figura 5.7).

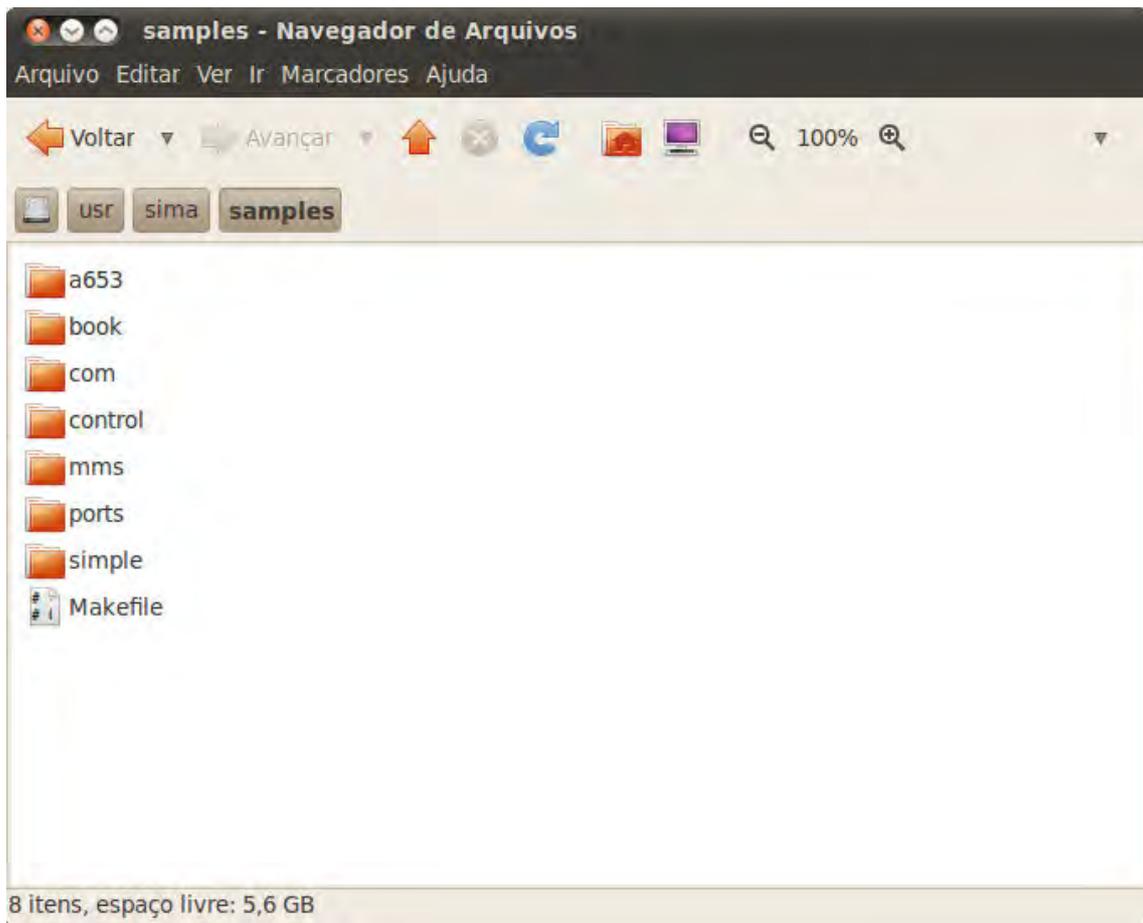


Figura 5.7 – Exemplos complexos do SIMA

5.7. Portas

Com já mencionado anteriormente, a comunicação interpartição faz uso de *Sampling ports* e *Queuing ports*. Já a comunicação intrapartição faz uso de *Buffer Services*, *Blackboard Services*, *Semaphore Services* e *Event Services*.

Segundo Diniz e Rufino (2005), o serviço de comunicação interpartição é independente de hardware e incluem *Sampling ports* e *Queuing ports*. Nas *Sampling ports*, as mensagens são gravadas e as mensagens seguintes a sobre-escrevem. Já nas *Queuing ports* as mensagens são organizadas em

formato de fila, ou seja, nenhuma informação é perdida. No serviço de comunicação intrapartição, o *Buffer Service* funciona como as *Queuing ports* da comunicação interpartição enquanto o *Blackboard Service* funciona como as *Sampling ports*. Ainda de acordo com o autor, os *Semaphore Services* e *Event Services* são meios de sincronização.

6 MODELAGEM E SIMULAÇÃO

Com o intuito de trilhar um caminho gradual, o trabalho inicia modelando e simulando um sistema mais simples e, em seguida, parte para a modelagem e simulação de um sistema mais complexo. Desta maneira, foi possível aprender a utilizar o ambiente SIMA gradualmente.

Como primeiro passo, optou-se por modelar e simular um sistema mais simples. Assim, o sistema escolhido foi o Massa-Mola-Amortecedor (MMA). Deste modo, a modelagem e a simulação foram divididas em três etapas. Na primeira etapa, o sistema Massa-Mola-Amortecedor foi modelado e simulado em MatLab. Em seguida, o mesmo sistema foi modelado e simulado na Linguagem C. E, como último passo, o modelo foi modelado e simulado no SIMA.

Terminado esse processo de amadurecimento, iniciou-se a modelagem e simulação do Sistema de Controle de Atitude da PMM no seu Modo Nominal. Como feito com o sistema Massa-Mola-Amortecedor, optou-se por dividir a modelagem e simulação em três etapas. Desta maneira, na primeira etapa, o SCA foi modelado e simulado em MatLab, em seguida, o mesmo sistema foi modelado e simulado na Linguagem C e, na etapa final, o modelo foi modelado e simulado no SIMA.

6.1. Modelagem e Simulação do Sistema Massa-Mola-Amortecedor

Para o controle adequado de um sistema é necessário que se conheça o comportamento dinâmico de cada uma das partes que o compõem. Desta forma, a modelagem consiste em se descrever matematicamente o funcionamento de todo o sistema. A partir dessa modelagem é possível projetar o controlador que atenda aos requisitos de desempenho do sistema em questão.

Portanto, para a modelagem do controle do sistema massa-mola-amortecedor foi necessário modelar controlador, atuador, planta e sensor (Figura 6.0).

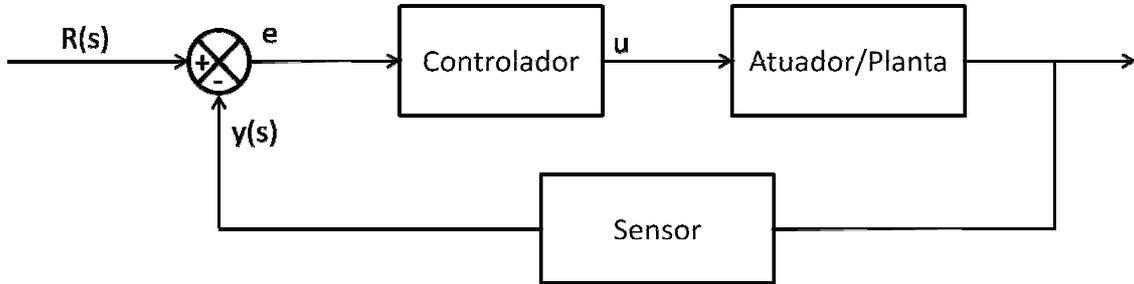


Figura 6.0 – Diagrama de Blocos do sistema Massa-Mola-Amortecedor

De acordo com Astrom (2002), o algoritmo de controle mais amplamente utilizado na indústria é o Proporcional-Integral-Derivativo (PID). Para as simulações deste trabalho, utilizou-se esse algoritmo. Desta forma, para a obtenção das constantes proporcional, integral e derivativa do PID julgou-se conveniente utilizar funções de transferência. E, para a implementação em linguagem C, utilizou-se equações de estado.

6.1.1. Modelagem utilizando Função de transferência

6.1.1.1. Planta

De acordo com Ogata (1997), uma maneira de representar o modelo Massa-Mola-Amortecedor de 2ª ordem é dado pelo diagrama que se segue (Figura 6.1):

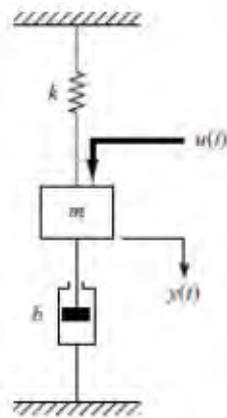


Figura 6.1 – Massa-Mola-Amortecedor.

Fonte: OGATA (1997).

E, pela seguinte equação:

$$m \ddot{y} + b \dot{y} + ky = u \quad (01)$$

Para modelagem utilizando-se funções de transferência foi necessário se aplicar a transformada de Laplace à equação (1). Lembrando que:

$$\mathcal{L}[y(t)] = Y(s) \quad (02)$$

$$\mathcal{L}[\dot{y}(t)] = sY(s) - y(0) \quad (03)$$

$$\mathcal{L}[\ddot{y}(t)] = s^2Y(s) - sy(0) - \dot{y}(0) \quad (04)$$

Deste modo, chega-se à equação:

$$[s^2Y(s) - sy(0) - \dot{y}(0)]m + [sY(s) - y(0)]b + [Y(s)]k = F(s) \quad (05)$$

Tem-se como condições iniciais:

$$y(0) = 1/3 m \quad (06)$$

$$\dot{y}(0) = 0 \quad (07)$$

Substituindo, chega-se a:

$$\left[s^2 Y(s) - \frac{1}{3} s \right] m + \left[s Y(s) - \frac{1}{3} \right] b + [Y(s)] k = F(s) \quad (08)$$

Como especificações temos:

$$m = 1 \text{ kg} \quad (09)$$

$$k = \frac{\pi^2}{4} \text{ N / m} \quad (10)$$

$$b = 0 \text{ N.s / m} \quad (11)$$

Logo:

$$\left(s^2 + \frac{\pi^2}{4} \right) Y(s) - \frac{1}{3} s = F(s) \quad (12)$$

Lembrando que:

$$G(s) = \frac{\mathcal{L}[\text{saída}]}{\mathcal{L}[\text{entrada}]} \Big|_{\text{condições iniciais nulas}} \quad (13)$$

Finalmente chegamos à função de transferência da planta:

$$\frac{Y(s)}{F(s)} = G(s) = \frac{1}{s^2 + 2,467} \quad (14)$$

6.1.1.2. Controlador

É apresentado em Ogata(1997) uma das maneira de se modelar o massa-mola-amortecedor:

$$\begin{aligned} C(s) &= Kp \left(1 + \frac{1}{Tis} + Tds \right) = Kp + \frac{Kp}{Tis} + KpTds = \\ &= \frac{KpTis + Kp + KpTdsTis}{Tis} \\ C(s) &= \frac{(KpTdTi)s^2 + (KpTi)s + Kp}{Tis} \end{aligned} \quad (15)$$

6.1.1.3. Planta com PID em malha aberta

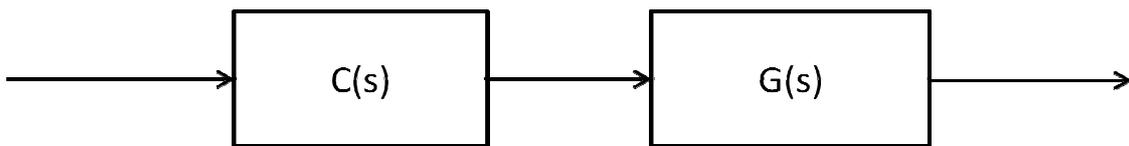


Figura 6.2 - Diagrama de Blocos em malha aberta.

Assim (Figura 6.2):

$$C(s).G(s) = \frac{(KpTdTi)s^2 + (KpTi)s + Kp}{(Ti)s^3 + (2,467Ti)s} \quad (16)$$

6.1.1.4. Planta com PID em malha fechada

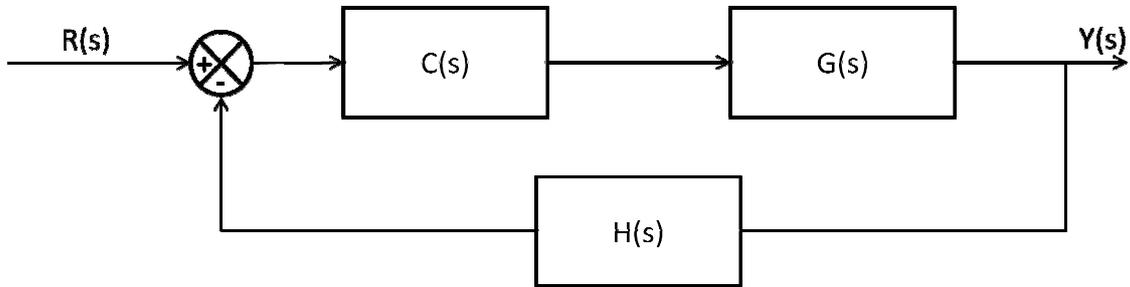


Figura 6.3 - Diagrama de Blocos em malha fechada.

Assim (Figura 6.3):

$$\frac{Y(s)}{R(s)} = \frac{C(s).G(s)}{1 + C(s).G(s).H(s)} \frac{(K_p T_d T_i) s^2 + (K_p T_i) s + K_p}{(T_i) s^3 + (2,467 T_i) s} \quad (17)$$

$$\frac{Y(s)}{R(s)} = \frac{(K_p T_d T_i) s^2 + (K_p T_i) s + K_p}{(T_i) s^3 + (K_p T_d T_i) s^2 + (2,467 T_i + K_p T_i) s + K_p}$$

Na função de transferência do PID em malha fechada, o denominador da equação é chamado de equação característica.

6.1.2. Resposta em regime transitório

De acordo com Ogata (1997), as características de desempenho de um sistema de controle podem ser especificadas em termos da resposta em regime transitório a um degrau unitário. Esta frequentemente exibe oscilações amortecidas antes de atingir o regime permanente como mostra a Figura 6.4 abaixo:

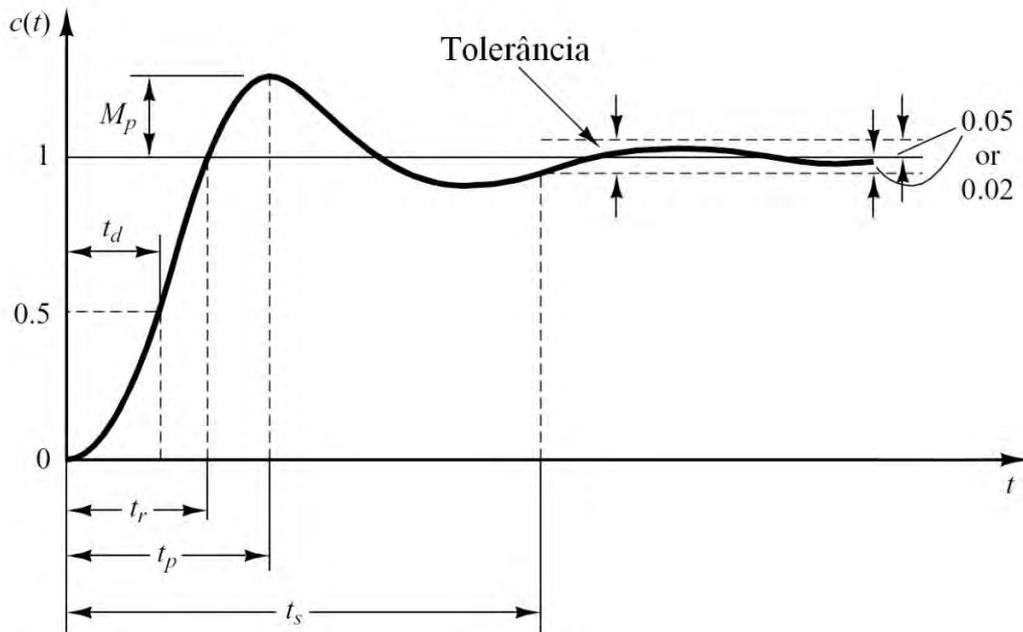


Figura 6.4- Resposta a um degrau unitário de um sistema de 2ª ordem subamortecido e sem zeros

Fonte: OGATA (1997).

Nela se definem os parâmetros de projeto:

- Tempo de atraso (“Delay time - t_d ”) é o tempo necessário para a resposta alcançar pela primeira vez a metade do valor final;
- Tempo de subida (“Rise time t_r ”) é o tempo necessário para a resposta passar de 10% a 90%, 5% a 95% ou 0% a 100% do seu valor final. Para sistemas de segunda ordem subamortecidos, usa-se normalmente o tempo de subida de 0% a 100%. Para sistemas sobreamortecidos, usa-se normalmente o tempo de subida de 10% a 90%;
- Instante do pico (“Peak time - t_p ”) é o tempo necessário para a resposta alcançar o primeiro pico de sobressinal;

- Sobressinal máximo (“Overshoot - M_p ”) é o máximo valor de pico da curva de resposta medido a partir do seu valor final. Também se usa o máximo sobressinal percentual. O sobressinal máximo percentual indica indiretamente a estabilidade relativa do sistema;

- Tempo de acomodação (“Settling time - t_s ”) é o tempo necessário para a curva de resposta alcançar e permanecer dentro de uma faixa em torno do seu valor final, faixa essa de magnitude especificada por uma porcentagem absoluta do valor final (geralmente 5% ou 2%). O tempo de acomodação está relacionado com a constante de tempo do sistema de controle. A escolha de que porcentagem usar no critério de erro pode ser determinada a partir dos objetivos do projeto do sistema em questão.

6.1.3. Obtendo as constantes proporcional, integral e derivativa do PID

De acordo com Ogata (1997), o comportamento dinâmico de um sistema de segunda ordem subamortecido e sem zeros pode ser descrito em termos de dois parâmetros ζ (fator de amortecimento) e ω_n (frequência natural não amortecida). A equação abaixo é a forma padrão de um sistema de segunda ordem:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (18)$$

Se $0 < \zeta < 1$, o sistema é subamortecido e a resposta transiente é oscilatória; se $\zeta = 1$, o sistema é criticamente amortecido; e se $\zeta > 1$ o sistema é sobreamortecido.

Assim, utilizando as especificações do sistema, calculou-se o ζ e ω_n .

Especificações definidas para o sistema:

$$\left\{ \begin{array}{l} \text{- tempo de subida } < 5\text{s} \rightarrow t_p \\ \text{- sobre-elevação } < 15\% \rightarrow M_p \\ \text{- tempo de acomodação } < 20\text{s} \rightarrow t_s \end{array} \right.$$

Assim, através da sobre-elevação obtemos o fator de amortecimento (ζ):

$$\begin{aligned} M_p &= e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)\pi} \\ 0.15 &= e^{-\pi\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)} \\ \zeta &= 0.5169 \end{aligned} \quad (19)$$

Através do tempo de acomodação obtemos a frequência natural não amortecida (ω_n):

$$\begin{aligned} t_s &= \frac{4}{\zeta\omega_n} \\ \omega_n &= 0.3869 \text{ rad/s} \end{aligned} \quad (20)$$

e

$$\zeta\omega_n = \sigma = 0.2 \text{ rad/s} \quad (21)$$

Para frequência natural amortecida temos:

$$\begin{aligned} \omega_d &= \omega_n \sqrt{1-\zeta^2} \\ \omega_d &= 0.3312 \text{ rad/s} \end{aligned} \quad (22)$$

Para se descobrir os valores das constantes proporcional, integral e derivativa do PID é necessário igualar-se a equação característica (denominador da equação 17) à equação do sistema de segunda ordem em sua forma padrão (equação 18). No entanto, a equação característica encontrada é de terceira ordem, deste modo, é necessário que a equação em sua forma padrão também

seja de terceiro grau. Uma maneira de resolver esse problema é adicionar uma raiz à equação, como mostrado abaixo:

$$\begin{aligned} & (s^2 + 2\zeta\omega_n s + \omega_n^2)(s + c) \\ & (s + \zeta\omega_n + j\omega_d)(s + \zeta\omega_n - j\omega_d)(s + c) \end{aligned} \quad (23)$$

Usualmente mantêm-se os polos dominantes do regime permanente $(\zeta\omega_n + j\omega_d)$ e $(\zeta\omega_n - j\omega_d)$ próximos do eixo imaginário, e aloca-se o polo dominante do regime transitório (c) de cinco a dez vezes mais distante que os anteriores. Neste trabalho adotamos $c=1$. Desta forma, igualando a equação (17) a equação (18) temos:

$$\begin{aligned} & (s + \zeta\omega_n + j\omega_d)(s + \zeta\omega_n - j\omega_d)(s + c) = \\ & (Ti)s^3 + (KpTdTi)s^2 + (2.467Ti + KpTi)s + Kp \end{aligned} \quad (24)$$

Substituindo os valores obtidos em (21) e (22), chegamos a:

$$\begin{aligned} & (s + 0.2 + 0.3312i)(s + 0.2 - 0.3312i)(s + 1) \\ & s^3 + 1.4s^2 + 0.5497s + 0.149 \end{aligned} \quad (25)$$

Igualando-se a equação (25) à equação característica (17), temos:

$$s^3 + 1.4s^2 + 0.5497s + 0.149 = (Ti)s^3 + (KpTdTi)s^2 + (2.467Ti + KpTi)s + Kp \quad (26)$$

Deste modo, chegamos às constantes proporcional, integral e derivativa do PID:

$$\begin{aligned} Kp &= 0.1497 \\ Ti &= 1 \\ Td &= 9.3960 \end{aligned} \quad (27)$$

6.1.4. Modelagem utilizando Equações de estado

De acordo com Ogata (1997), este sistema é de segunda ordem (Equação 01), portanto dois integradores são envolvidos. Definindo as variáveis de estado $x_1(t)$ e $x_2(t)$ como:

$$\begin{aligned}x_1(t) &= y(t) \\x_2(t) &= \dot{y}(t)\end{aligned}\tag{28}$$

Então

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{m}(-ky - b\dot{y}) + \frac{1}{m}u \\ \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k}{m}x_1 - \frac{b}{m}x_2 + \frac{1}{m}u\end{aligned}\tag{29}$$

e, como saída:

$$y = x_1(t)\tag{30}$$

Na sua forma matricial, temos:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u\tag{31}$$

E, como saída:

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\tag{32}$$

As equações (28) e (29) estão na forma padrão:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} + \mathbf{D}u\end{aligned}\quad (33)$$

onde:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 1 \\ m \end{bmatrix}, \mathbf{C} = [1 \quad 0], \mathbf{D} = 0 \quad (34)$$

6.1.4.1. Implementação computacional (Discretização)

6.1.4.1.1. Equações de estado

Para a solução utilizou-se a discretização, apresentada em Discretization (2013), conforme se segue:

$$\begin{aligned}x[k+1] &= A_d x(k) + B_d u(k) \\ y[k] &= C_d x[k] + D_d u[k]\end{aligned}\quad (35)$$

Onde

$$\begin{aligned}\mathbf{A}_d &= e^{\mathbf{A}T} = \mathcal{L}^{-1}\{(s\mathbf{I} - \mathbf{A})^{-1}\}_{t=T} \\ \mathbf{B}_d &= \left(\int_{\tau=0}^T e^{\mathbf{A}\tau} d\tau \right) \mathbf{B} = \mathbf{A}^{-1}(\mathbf{A}_d - \mathbf{I})\mathbf{B} \\ \mathbf{C}_d &= \mathbf{C} \\ \mathbf{D}_d &= \mathbf{D}\end{aligned}\quad (36)$$

6.1.4.1.2. Conversão do PID analógico para o PID digital

Para os cálculos numéricos fez-se necessário o uso de um controlador PID digital. Deste modo, discretizamos este controlador a partir de um controlador PID analógico:

Dada a equação do PID analógico:

$$u(t) = K_P e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dt} \quad (37)$$

Onde $u(t)$ é o sinal de atuação; $e(t)$ é o erro entre a saída e a referência; e K_p , K_i e K_d são as constantes proporcional, integral e derivativa do PID.

Considerando $t = nT$, onde T é o período de amostragem e n o número da amostra, obtemos uma equação discretizada:

$$u(nT) = K_P e(nT) + K_I T \sum_{j=0}^n e(jT) + K_D \frac{e(nT) - e((n-1)T)}{T} \quad (38)$$

Como nT representa a amostra número n do sistema, ela pode ser substituída por n , resultando em:

$$u(n) = K_P e(n) + K_I T \sum_{j=0}^n e(j) + K_D \frac{e(n) - e(n-1)}{T} \quad (39)$$

Simplificando:

$$u(n) = \left(K_P + \frac{K_D}{T} \right) e(n) + \frac{K_D}{T} e(n-1) + K_I T \sum_{j=0}^n e(j) \quad (40)$$

Fazendo

$$u(n-1) = \left(K_P + \frac{K_D}{T} \right) e(n-1) + \frac{K_D}{T} e(n-2) + K_I T \sum_{j=0}^{n-1} e(j) \quad (41)$$

Subtraindo a eq. (39) na eq. (38) chegamos a:

$$u(n) = u(n - 1) + \left(K_P + \frac{K_D}{T} + K_I T \right) e(n) + \left(-K_P - 2 \frac{K_D}{T} \right) e(n - 1) + \frac{K_D}{T} e(n - 2) \quad (42)$$

ou:

$$u(n) = u(n - 1) + Ae(n) + Be(n - 1) + Ce(n - 2) \quad (43)$$

6.2. Implementação do sistema Massa-Mola-Amortecedor no SIMA

6.2.1. Visão geral

Para a implementação do sistema Massa-Mola-Amortecedor no SIMA foram criadas 2 partições. Na primeira partição alocou-se o controlador e na segunda partição a planta, sensor e atuador, conforme ilustra a Figura 6.5:

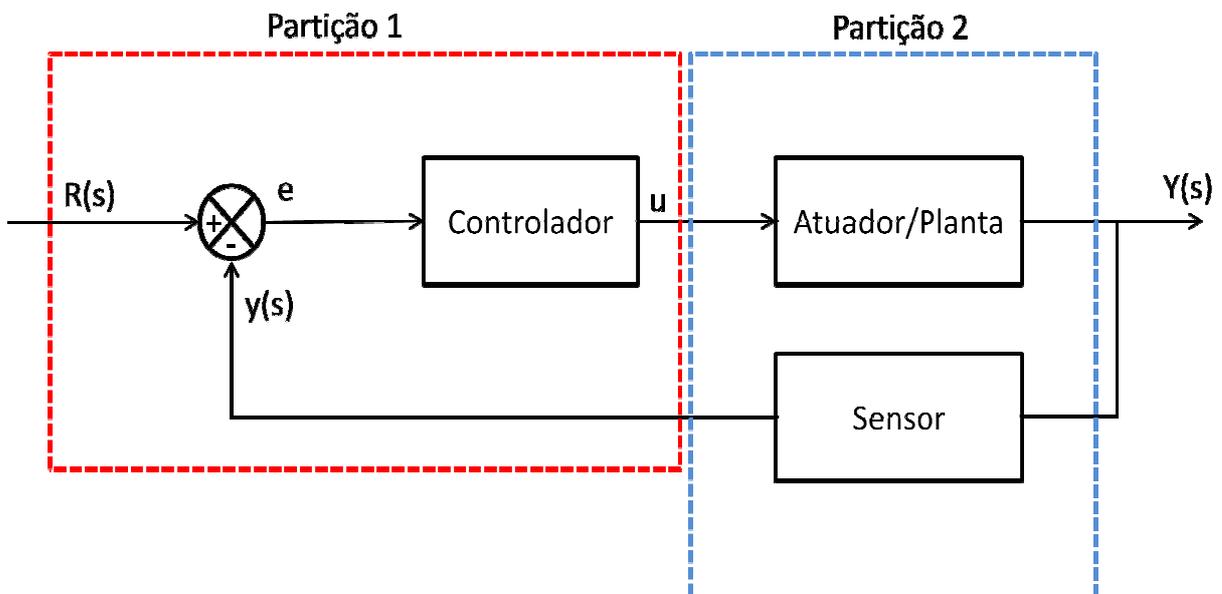


Figura 6.5 – Sistema Massa-Mola-Amortecedor dividido em duas partições

6.2.2. Portas e canais

Como já mencionado, a comunicação entre partições é feita a partir de portas e canais. Deste modo, é necessário se criar canais de comunicação para que as mensagens em uma porta de origem (*source port*) sejam copiadas para área de memória da porta de destino (*destination port*).

Deste modo, para a comunicação entre as partições foram criados 2 canais e 4 portas de comunicação (*queuing ports*), como mostrado na Figura 6.6:

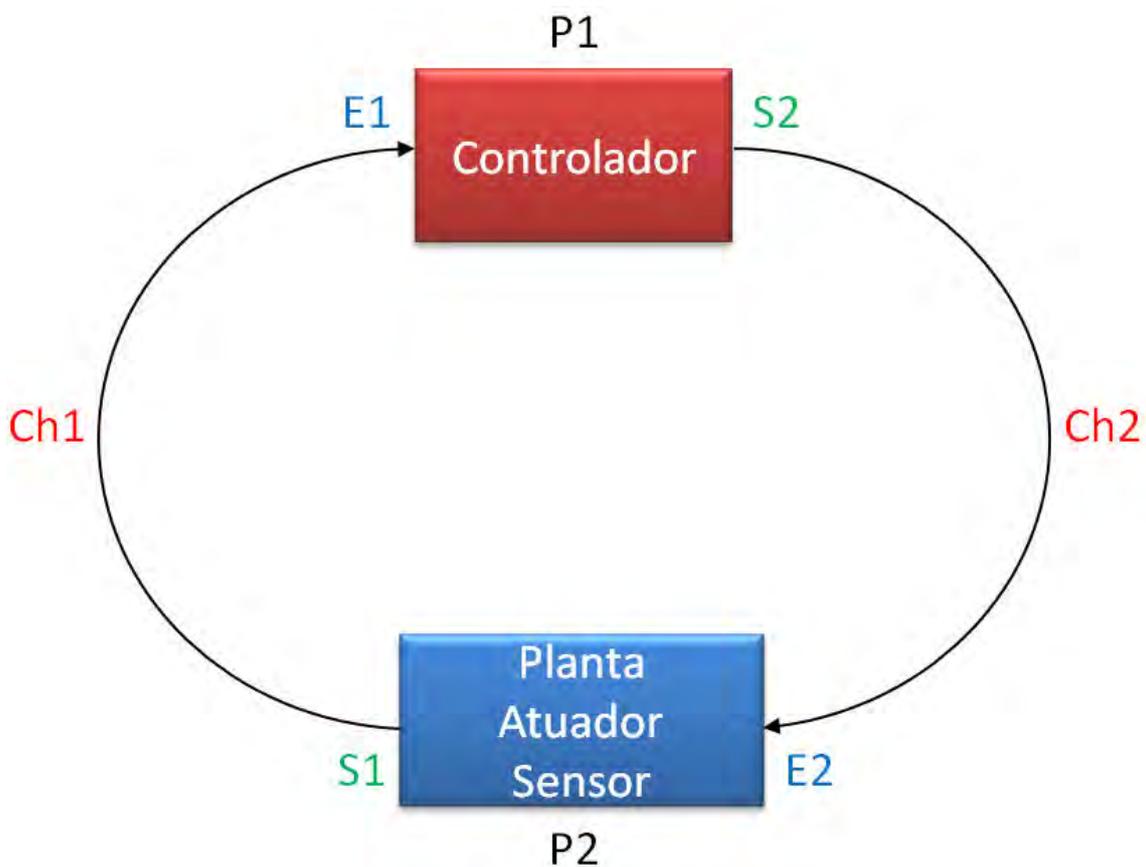


Figura 6.6 - Representação das portas e canais de comunicação do MMA no SIMA

E1 e E2 representam as portas de entrada de cada partição; S1 e S2 representam as portas de saída de cada partição.

Assim, a partição Atuador/Planta/Sensor envia, através do canal Ch1 o valor da saída, Y ; o valor é comparado com o valor de referência e o controlador envia um sinal de controle, U , para o atuador, através do Ch2, como mostrado na Figura 6.7:

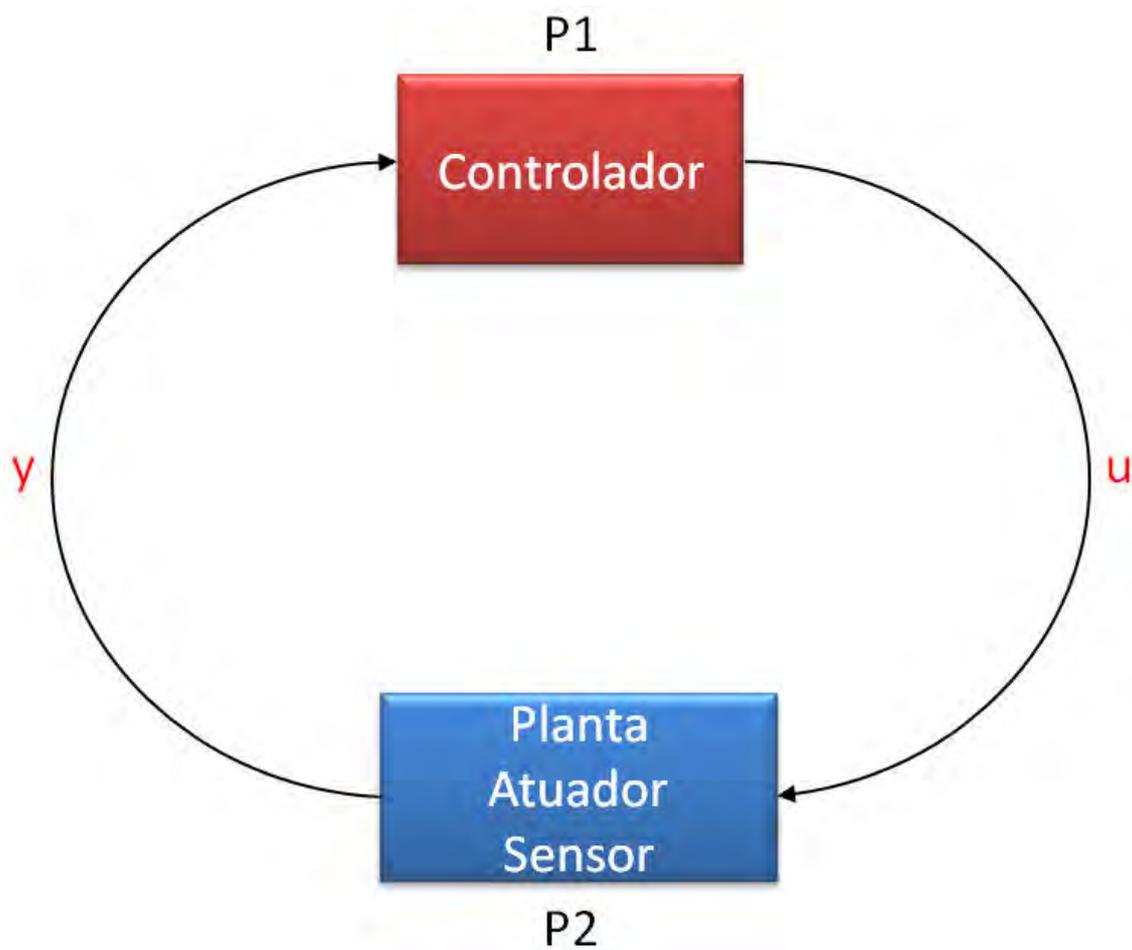


Figura 6.7 - Representação da comunicação entre as partições do sistema MMA

6.2.3. Configuração

No SIMA, os arquivos que devem ser alterados para a configuração das portas e canais são: a653.xml e sima.xml. A seguir são ilustradas as configurações utilizadas nas simulações (Figura 6.8 a 6.11):

6.2.3.1. Partição 1

6.2.3.1.1. a653.xml

```
<Partition PartitionIdentifier="1"
  PartitionName="controlador"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">
  <!-- entradas da planta-->
  <Queuing_Port Name="S2"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32" />

  <!-- entradas da planta-->
  <Queuing_Port Name="E1"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32" />
</Partition>
```

Figura 6.8 - Configuração no arquivo a653.xml da partição 1

6.2.3.1.2. sima.xml

```
</Partition>

<Partition PartitionIdentifier="1"
  PartitionName="controlador"
  SharedMemory="70040"
  Startup="/bin/sh ./samples/com/partitions/controlador/controlador.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/controlador/fifo_controlador"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.001"
    EndDurationSeconds="0.001"/>
  <Destination Port
    Name="E1"
    Type="UDP"
    IP="127.0.0.1"
    Port="12351"/>
</Partition>
```

Figura 6.9 - Configuração no arquivo sima.xml da partição 1

6.2.3.2. Partição 2

6.2.3.2.1. a653.xml

```
<Partition PartitionIdentifier="2"  
    PartitionName="planta"  
    Criticality="LEVEL_A"  
    SystemPartition="true" EntryPoint="Initial">  
  
    <!-- saídas da planta-->  
    <Queuing_Port Name="S1"  
        MaxMessageSize="1024"  
        Direction="SOURCE"  
        MaxNbMessages="32"/>  
  
    <!-- entradas da planta-->  
    <Queuing_Port Name="E2"  
        MaxMessageSize="1024"  
        Direction="DESTINATION"  
        MaxNbMessages="32"/>  
  
</Partition>
```

Figura 6.10 - Configuração no arquivo a653.xml da partição 2

6.2.3.2.2. sima.xml

```
<Partition PartitionIdentifier="2"
  PartitionName="planta"
  SharedMemory="70050"
  Startup="/bin/sh ./samples/com/partitions/planta/planta.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/planta/fifo_planta"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.001"
    EndDurationSeconds="0.001"/>

  <Destination_Port
    Name="E2"
    Type="UDP"
    IP="127.0.0.1"
    Port="12361"/>

</Partition>
```

Figura 6.11 - Configuração no arquivo sima.xml da partição 2

6.2.4. Agendamento (Scheduling) das partições

O agendamento das partições foi feito da seguinte forma: a janela de execução da partição 1 se inicia em 0,00 e tem um tempo de execução de 0,06 segundos. Finalizando este intervalo de tempo os processos desta partição sofrem preempção e a janela de execução da partição 2 ganha acesso ao processador, e esta, também com 0,06 segundos de intervalo, executa seus processos. Ao finalizar seu tempo, os processos da partição 1 que haviam sofrido preempção retomam sua execução a partir do ponto que tinham sido interrompidos. A Figura 6.12 ilustra agendamento das partições.

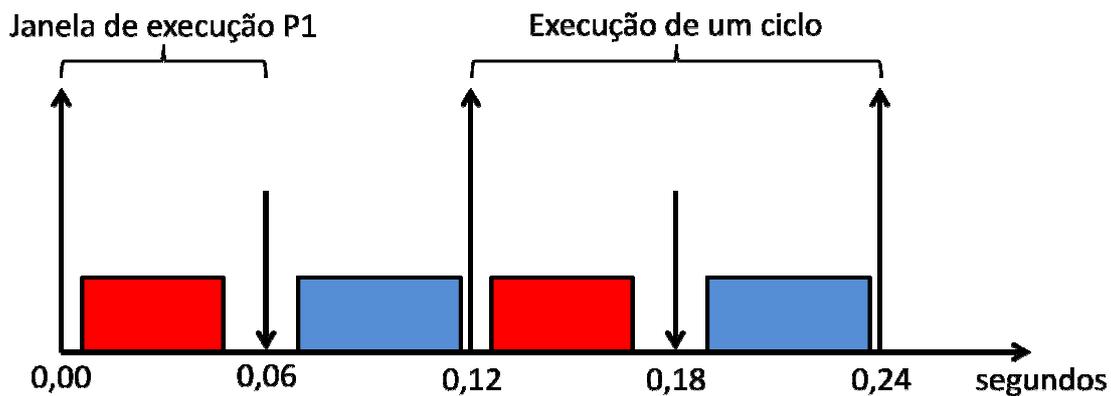


Figura 6.12 – Agendamento no tempo das partições do sistema MMA

As configurações de agendamento utilizadas nas simulações, também efetuadas no arquivo de configuração do SIMA (a653.xml) são ilustradas a seguir na Figura 6.13:

6.2.4.1. a653.xml

```

<Module_Schedule MajorFrameSeconds="0.12">
  <Partition_Schedule PartitionIdentifier="1"
    PartitionName="planta"
    PeriodSeconds="0.12"
    PeriodDurationSeconds="0.06">
    <Window_Schedule WindowIdentifier="101"
      WindowStartSeconds="0.0"
      WindowDurationSeconds="0.06"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>

  <Partition_Schedule PartitionIdentifier="2"
    PartitionName="controlador"
    PeriodSeconds="0.12"
    PeriodDurationSeconds="0.06">
    <Window_Schedule WindowIdentifier="201"
      WindowStartSeconds="0.06"
      WindowDurationSeconds="0.06"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
</Module_Schedule>

```

Figura 6.13 - Configuração no arquivo a653.xml das partições 1 e 2

6.3. Modelagem do Sistema de Controle de Atitude

Para a modelagem do sistema de controle de atitude da PMM é necessário antes um conhecimento prévio de sistemas de referência.

6.3.1. Sistema de Coordenadas de Referência

Como visto, na determinação de atitude, a orientação do satélite é expressa em relação a um sistema de referência. Segundo Wertz (1978), os sistemas de coordenadas mais importantes são os seguintes: o referencial fixo no espaço inercial (referencial inercial); o referencial definido em relação à órbita e não fixo em relação ao veículo nem ao espaço inercial (referencial orbital); e o referencial fixo em relação ao corpo do veículo espacial (referencial do veículo).

6.3.1.1. Referencial Inercial

De acordo com Wertz (1978), o sistema de coordenadas inerciais mais comum é o sistema de coordenadas celestes definido em relação ao eixo de rotação da Terra (Figura 6.14).

WERTZ (1978) afirma que o sistema de coordenadas celestes não é verdadeiramente inercial, pois ele não é fixo em relação às estrelas na proximidade do Sol. As forças gravitacionais da Lua e do Sol causam um torque que resulta em uma vagarosa rotação do eixo da Terra em relação à eclíptica. Esse fenômeno é conhecido como Precessão dos Equinócios e tem período de aproximadamente 26.000 anos. Isso representa, aproximadamente, uma rotação de 50 segundos de arco por ano.

No entanto, para este trabalho, o sistema de coordenadas celestes é considerado fixo em relação às estrelas, pois, de acordo com Kaplan (1976), situações práticas exigem somente que o referencial inercial seja um sistema

de coordenadas que garanta a precisão requerida sob um intervalo de tempo de interesse. Para a maioria dos problemas, basta escolher um sistema de referência que não esteja acelerando o suficiente para perturbar a solução do problema além da precisão requerida.

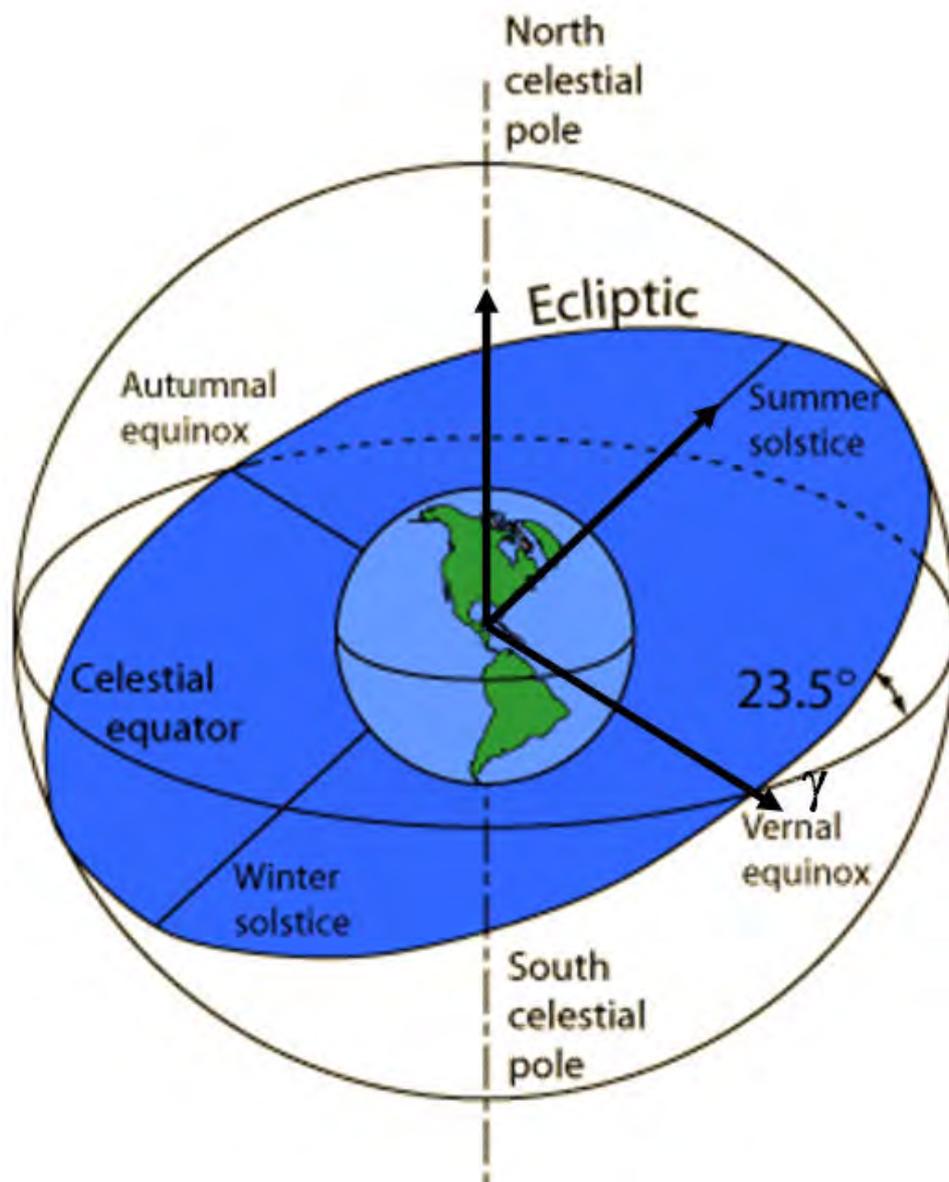


Figura 6.14 – Sistema de Coordenadas Celestes.

Fonte: GST (1997)

Relembrando que a Terra está no centro da esfera celeste. O eixo do sistema de coordenadas centralizado na Terra, aponta para os pólos norte e sul celestiais, é definido como superposto ao eixo de rotação da Terra. Assim, o pólo norte celestial está a aproximadamente 1 grau da estrela Polaris, a estrela mais brilhante da constelação da Ursa Menor. Para definir totalmente o sistema de coordenadas, nós também devemos definir o meridiano de referência ou ponto de referência. O ponto escolhido no Equador Celestial como referência é o ponto onde a Eclíptica, movimento aparente do Sol ao redor da Terra, cruza o Equador Terrestre indo do sul para o norte, conhecido como Equinócio Vernal. Esta é a direção paralela à linha do centro da Terra ao Sol no primeiro dia de primavera (no Hemisfério Norte).

O presente trabalho utiliza o sistema de coordenadas celestes (X, Y, Z), que tem origem no centro da Terra, o eixo X aponta na direção do Equinócio Vernal γ (primeiro ponto da Constelação de Áries), o eixo Z aponta para a direção do Polo Norte geográfico, e o eixo Y completa o sistema dextrógiro no plano do Equador Terrestre (Figura 6.14).

6.3.1.2. Referencial Orbital

O referencial orbital, também definido como Vertical Local e Horizontal Local (VLHL), é um sistema de coordenadas com origem definida no centro de massa do veículo espacial. Nesse sistema o eixo Z_0 é direcionado para o centro da Terra (Nadir), o eixo Y_0 é direcionado para o negativo da binormal ao plano de órbita, e o eixo de X_0 é perpendicular aos outros dois, e coincide com a direção do vetor velocidade orbital para o caso de uma órbita circular (Figura 6.15)(WERTZ, 1978).

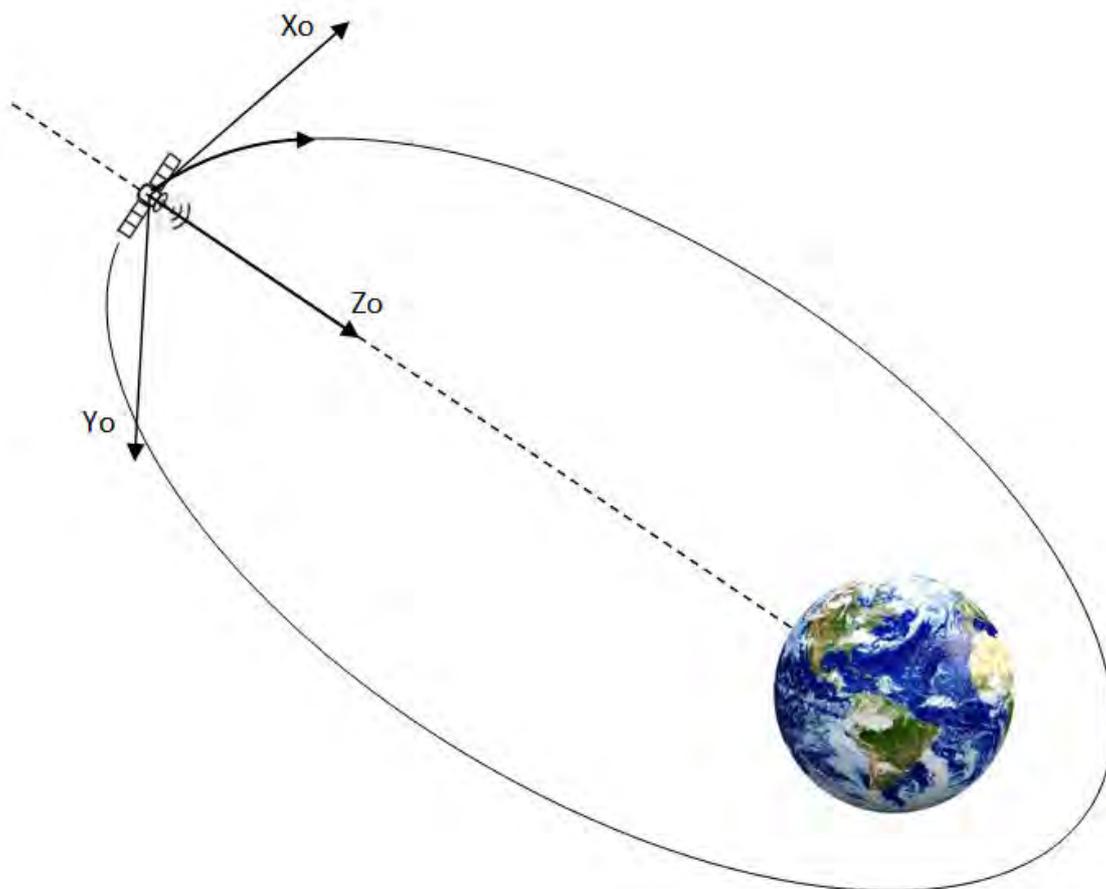


Figura 6.15 - Referencial Orbital.

6.3.1.3. Referencial do Satélite

Este sistema de coordenadas tem como origem o centro de massa do veículo e, seus eixos são coincidentes com os eixos principais de inércia do mesmo.

Foi utilizado o sistema de coordenadas definido como rolamento (“roll”), arfagem (“pitch”) e guinada (“yaw”). De acordo com SIDI (1997), é comum definir o ângulo de rolamento (ϕ) como a rotação em torno do eixo x do corpo, o ângulo de arfagem (θ) como a rotação em torno do eixo y do corpo, e o ângulo de guinada (ψ) como a rotação em torno do eixo z do corpo (Figura 6.16). Segundo Wertz (1978), esse sistema é o mais utilizado para satélites orientados para a Terra.

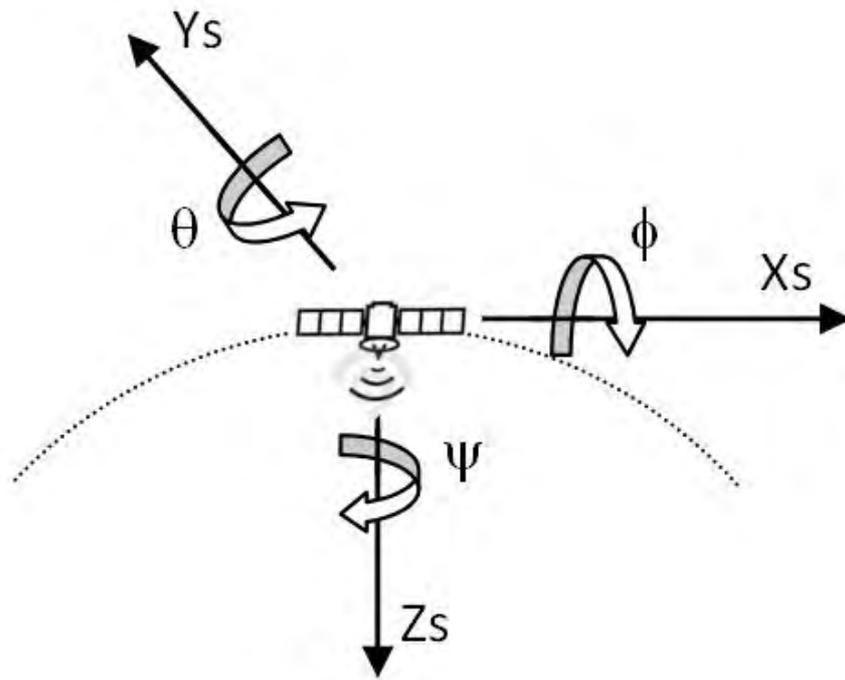


Figura 6.16 - Referencial do Satélite.

Para a modelagem do sistema de controle de atitude é necessário modelar controlador, atuador, planta e sensor (Figura 6.17).

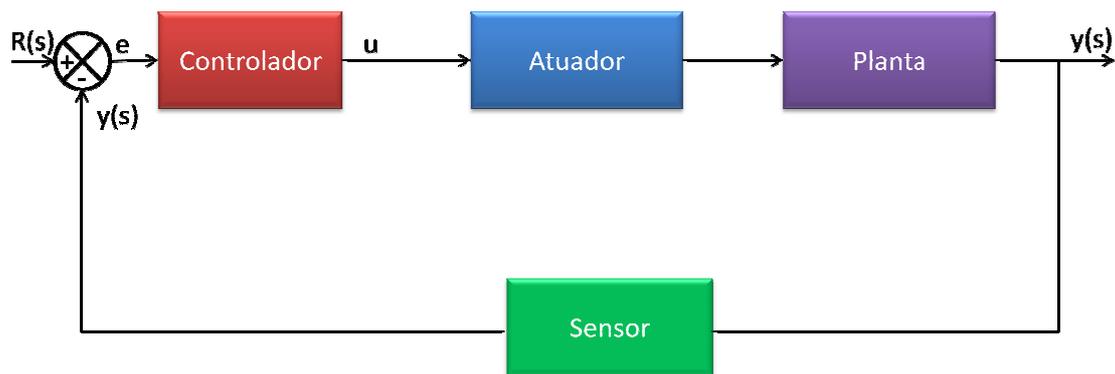


Figura 6.17 - Diagrama de Blocos do AOCS do satélite Amazônia-1

6.3.2. Planta

6.3.2.1. Cinemática

6.3.2.1.1. Representação da Atitude

Segundo Wertz (1978), a atitude em três eixos de um corpo no espaço é convenientemente representada através de uma transformação de coordenadas, na qual se transformam as coordenadas dos eixos de referência no espaço inercial em coordenadas dos eixos de referência fixadas ao satélite.

Os métodos comumente utilizados na transformação de coordenadas são: matriz de rotação, também chamada de matriz de atitude; ângulos de Euler; quaternions, também chamados de parâmetros de Euler e o ângulo do eixo equivalente.

6.3.2.1.1.1. Matriz de Rotação ou Matriz dos Cossenos Diretores

Para a dedução da matriz de rotação, Tewari (2006) considera um vetor, A , expresso em termos de suas componentes no sistema de coordenadas original e no sistema rotacionado:

$$\mathbf{A} = A_x \mathbf{i} + A_y \mathbf{j} + A_z \mathbf{k} = A'_x \mathbf{i}' + A'_y \mathbf{j}' + A'_z \mathbf{k}' \quad (44)$$

Em sua forma matricial:

$$\mathbf{A} = (\mathbf{i} \ \mathbf{j} \ \mathbf{k}) \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = (\mathbf{i}' \ \mathbf{j}' \ \mathbf{k}') \begin{Bmatrix} A'_x \\ A'_y \\ A'_z \end{Bmatrix} \quad (45)$$

Isolando-se $\begin{Bmatrix} A'_x \\ A'_y \\ A'_z \end{Bmatrix}$ chega-se a:

$$\begin{Bmatrix} A'_x \\ A'_y \\ A'_z \end{Bmatrix} = \begin{pmatrix} \mathbf{i}' \cdot \mathbf{i} & \mathbf{i}' \cdot \mathbf{j} & \mathbf{i}' \cdot \mathbf{k} \\ \mathbf{j}' \cdot \mathbf{i} & \mathbf{j}' \cdot \mathbf{j} & \mathbf{j}' \cdot \mathbf{k} \\ \mathbf{k}' \cdot \mathbf{i} & \mathbf{k}' \cdot \mathbf{j} & \mathbf{k}' \cdot \mathbf{k} \end{pmatrix} \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} \quad (46)$$

Da mesma forma, isolando-se $\begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix}$ chega-se a:

$$\begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{pmatrix} \mathbf{i} \cdot \mathbf{i}' & \mathbf{i} \cdot \mathbf{j}' & \mathbf{i} \cdot \mathbf{k}' \\ \mathbf{j} \cdot \mathbf{i}' & \mathbf{j} \cdot \mathbf{j}' & \mathbf{j} \cdot \mathbf{k}' \\ \mathbf{k} \cdot \mathbf{i}' & \mathbf{k} \cdot \mathbf{j}' & \mathbf{k} \cdot \mathbf{k}' \end{pmatrix} \begin{Bmatrix} A'_x \\ A'_y \\ A'_z \end{Bmatrix} \quad (47)$$

Reescrevendo a equação (47), temos:

$$\begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = C \begin{Bmatrix} A'_x \\ A'_y \\ A'_z \end{Bmatrix} \quad (48)$$

Onde C é a matriz dos cossenos diretores. Ela recebe este nome, pois cada termo da matriz é o cosseno dos ângulos entre os eixos dos dois sistemas de coordenadas, como é deduzido a seguir:

$$C \doteq \begin{pmatrix} \mathbf{i} \cdot \mathbf{i}' & \mathbf{i} \cdot \mathbf{j}' & \mathbf{i} \cdot \mathbf{k}' \\ \mathbf{j} \cdot \mathbf{i}' & \mathbf{j} \cdot \mathbf{j}' & \mathbf{j} \cdot \mathbf{k}' \\ \mathbf{k} \cdot \mathbf{i}' & \mathbf{k} \cdot \mathbf{j}' & \mathbf{k} \cdot \mathbf{k}' \end{pmatrix} \quad (49)$$

Utilizando a matriz C pode-se obter a matriz de rotação de cada eixo. Na Figura 6.18 é executada uma rotação em torno do eixo X :

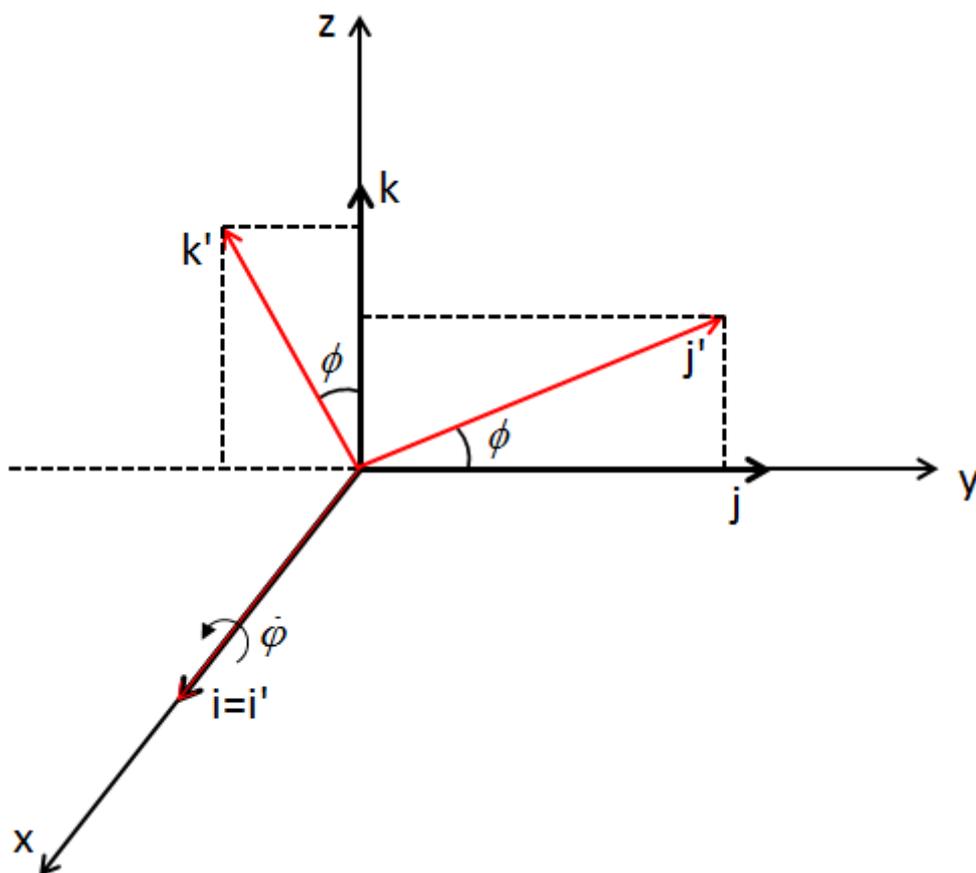


Figura 6.18 - Rotação em torno do eixo X

Temos que:

$$\begin{aligned}i.i' &= 1; j.i' = 0; k.i' = 0; \\i.j' &= 0; j.j' = \cos \phi; k.j' = \cos(90 - \phi); \\i.k' &= 0; j.k' = \cos(90 + \phi); k.k' = \cos \phi;\end{aligned}\tag{50}$$

Lembrando que:

$$\begin{aligned}\cos(x + y) &= \cos x \cdot \cos y - \operatorname{sen} x \cdot \operatorname{sen} y \\ \cos(x - y) &= \cos x \cdot \cos y + \operatorname{sen} x \cdot \operatorname{sen} y\end{aligned}\tag{51}$$

Portanto:

$$\begin{aligned}\cos(90 + \phi) &= \underbrace{\cos 90}_0 \cdot \cos \phi - \underbrace{\operatorname{sen} 90}_1 \cdot \operatorname{sen} \phi \\ \cos(90 + \phi) &= -\operatorname{sen} \phi\end{aligned}\tag{52}$$

e

$$\begin{aligned}\cos(90 - \phi) &= \underbrace{\cos 90}_0 \cdot \cos \phi + \underbrace{\operatorname{sen} 90}_1 \cdot \operatorname{sen} \phi \\ \cos(90 - \phi) &= \operatorname{sen} \phi\end{aligned}\tag{53}$$

Assim:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \operatorname{sen} \phi \\ 0 & -\operatorname{sen} \phi & \cos \phi \end{bmatrix}\tag{54}$$

Portanto:

$$\begin{bmatrix} Ax' \\ Ay' \\ Az' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \operatorname{sen} \phi \\ 0 & -\operatorname{sen} \phi & \cos \phi \end{bmatrix} \cdot \begin{bmatrix} Ax \\ Ay \\ Az \end{bmatrix}\tag{55}$$

Da mesma maneira, são obtidas as matrizes de rotação em torno dos demais eixos. Assim, chega-se à matriz de rotação em torno do eixo Y:

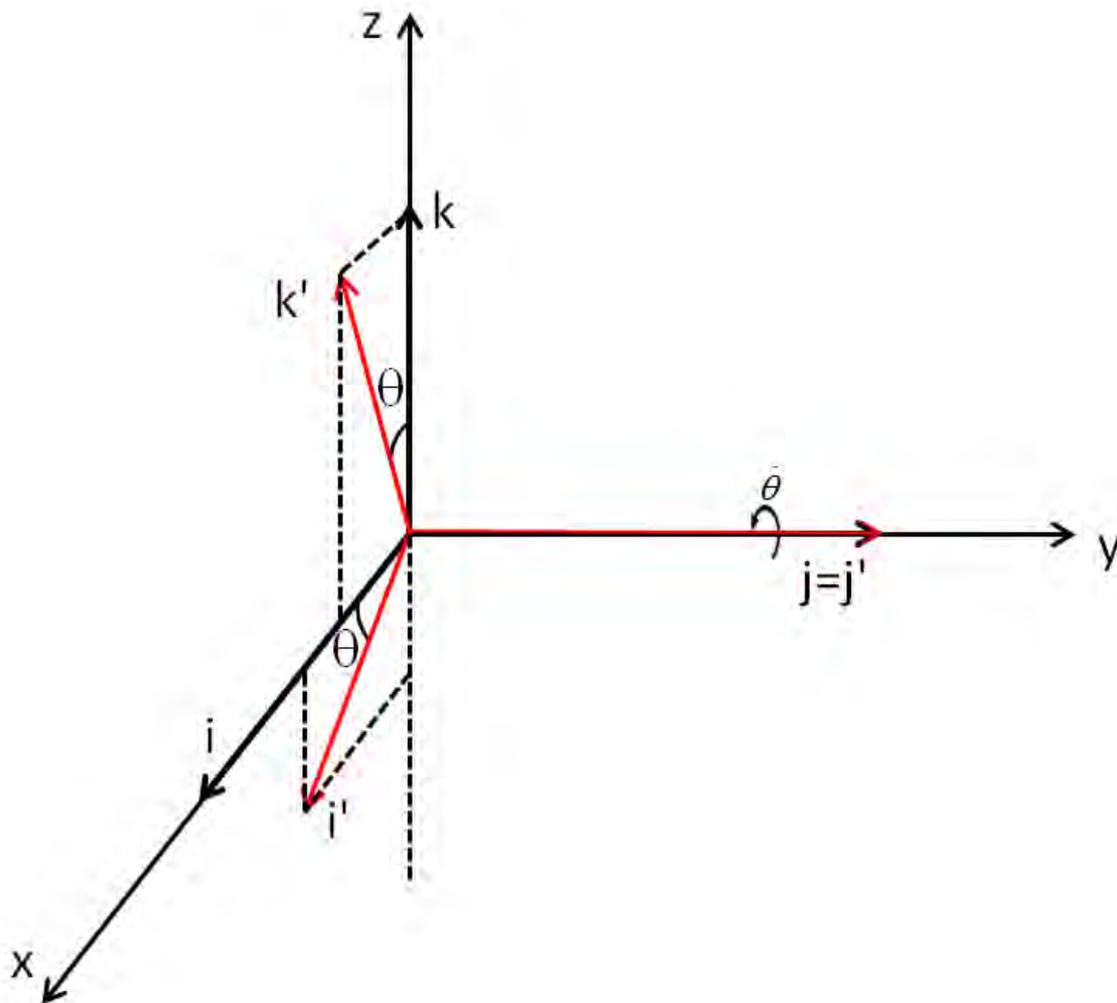


Figura 6.19 - Rotação em torno do eixo Y

$$C_2 = \begin{bmatrix} \cos\theta & 0 & -\text{sen}\theta \\ 0 & 1 & 0 \\ \text{sen}\theta & 0 & \cos\theta \end{bmatrix} \quad (56)$$

Portanto:

$$\begin{bmatrix} Ax' \\ Ay' \\ Az' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\text{sen}\theta \\ 0 & 1 & 0 \\ \text{sen}\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} Ax \\ Ay \\ Az \end{bmatrix} \quad (57)$$

Do mesmo modo, obtém-se a matriz de rotação em torno do eixo z:

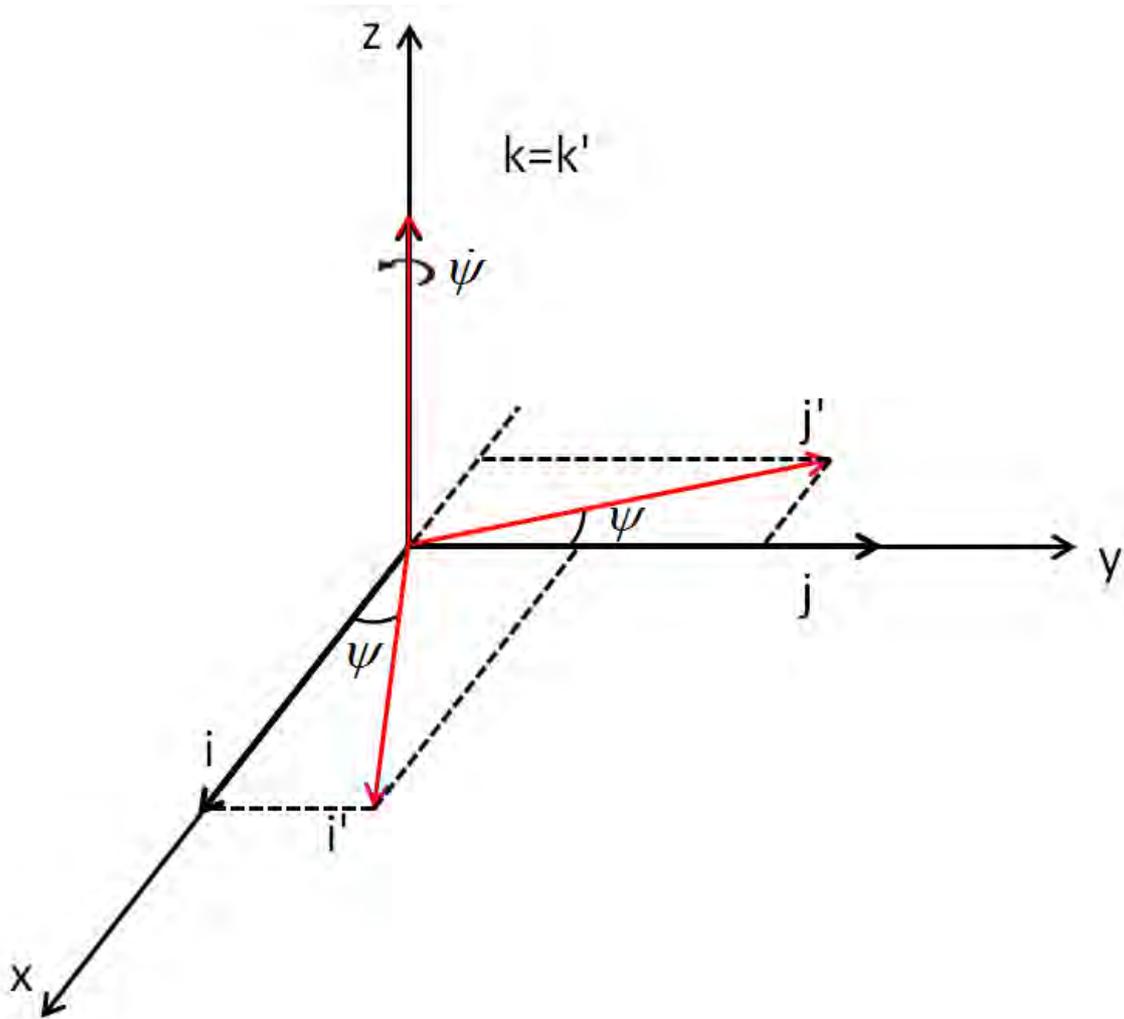


Figura 6.20 - Rotação em torno do eixo Z

$$C_3 = \begin{bmatrix} \cos\psi & \text{sen}\psi & 0 \\ -\text{sen}\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (58)$$

Portanto:

$$\begin{bmatrix} Ax' \\ Ay' \\ Az' \end{bmatrix} = \begin{bmatrix} \cos\psi & \text{sen}\psi & 0 \\ -\text{sen}\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Ax \\ Ay \\ Az \end{bmatrix} \quad (59)$$

6.3.2.1.1.2. Ângulos de Euler

Uma das maneiras utilizadas para se representar a orientação ou atitude de um sistema de coordenadas em relação a outro é baseada no uso de três ângulos, cada um em torno de um eixo particular. Esta representação de atitude pelos três ângulos é conhecida como representação dos ângulos de Euler, e os ângulos propriamente ditos como ângulos de Euler (TEWARI, 2006).

Há diversas sequências de rotações, e a ordem escolhida depende da situação em questão. De acordo com Sidi (1997), é comum definir o ângulo de rolamento (ϕ) como a rotação em torno do eixo x do corpo, o ângulo de arfagem (θ) em torno do eixo y do corpo, e o ângulo de guinada (ψ) em torno do eixo z do corpo. No entanto, qualquer outra definição é aceita contanto que se mantenha consistente com o desenvolvimento analítico.

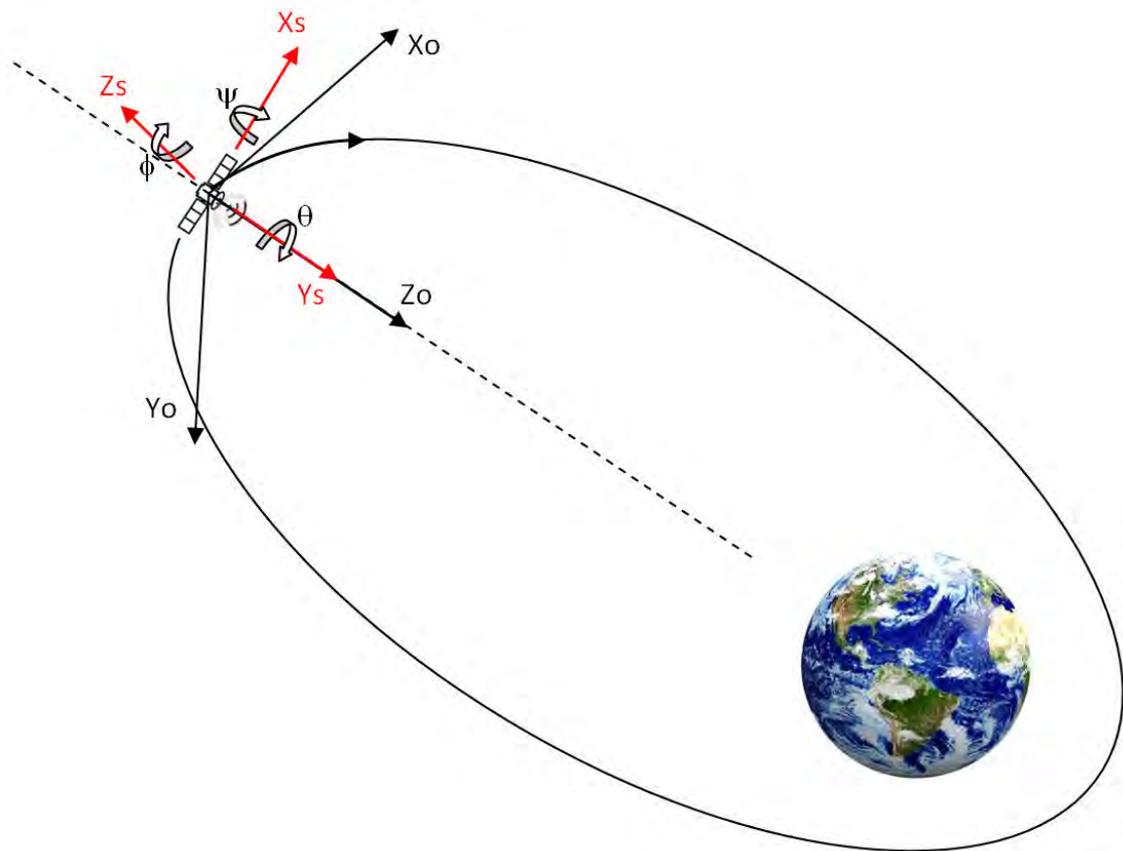


Figura 6.21 - Referencial Orbital (O) e Referencial do Satélite (S)

Para descrever a orientação de um referencial fixo ao corpo do satélite (S) em relação ao Referencial Orbital (O) em termos dos três ângulos de Euler (ϕ , θ , ψ), considera-se a sequência de rotações 3-2-1 ($C1(\phi) \leftarrow C2(\theta) \leftarrow C3(\psi)$) de O para S. Note que as setas indicam a ordem das rotações; e a disposição das matrizes C_i , $i = 1, 2, 3$, corresponde à posição das mesmas para a multiplicação de matrizes que resulta na matriz de rotação $R_{O/S}$ (Figura 6.21). Para essa sequência temos:

$$\vec{s} = R_{O/S} \vec{o} \quad (60)$$

$$\begin{aligned}
R_{O/S} &= C_1(\phi)C_2(\theta)C_3(\psi) \\
R_{O/S} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
R_{O/S} &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\psi\sin\theta\sin\phi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}
\end{aligned} \tag{61}$$

Portanto, temos:

$$\begin{bmatrix} \vec{s}_1 \\ \vec{s}_2 \\ \vec{s}_3 \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\psi\sin\theta\sin\phi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \vec{o}_1 \\ \vec{o}_2 \\ \vec{o}_3 \end{bmatrix} \tag{62}$$

Fisicamente, a sequência de rotações 3-2-1, partindo do referencial X, Y, Z para o referencial X3, Y3, Z3 pode ser descrita como (Figura 6.22):

- a) Rotação em torno do eixo Z de um ângulo ψ que leva ao sistema X1, Y1, Z1.
- b) Rotação em torno do eixo Y1 de um ângulo θ que leva ao sistema X2, Y2, Z2.
- c) Rotação em torno do eixo X2 de um ângulo ϕ que leva ao sistema X3, Y3, Z3.

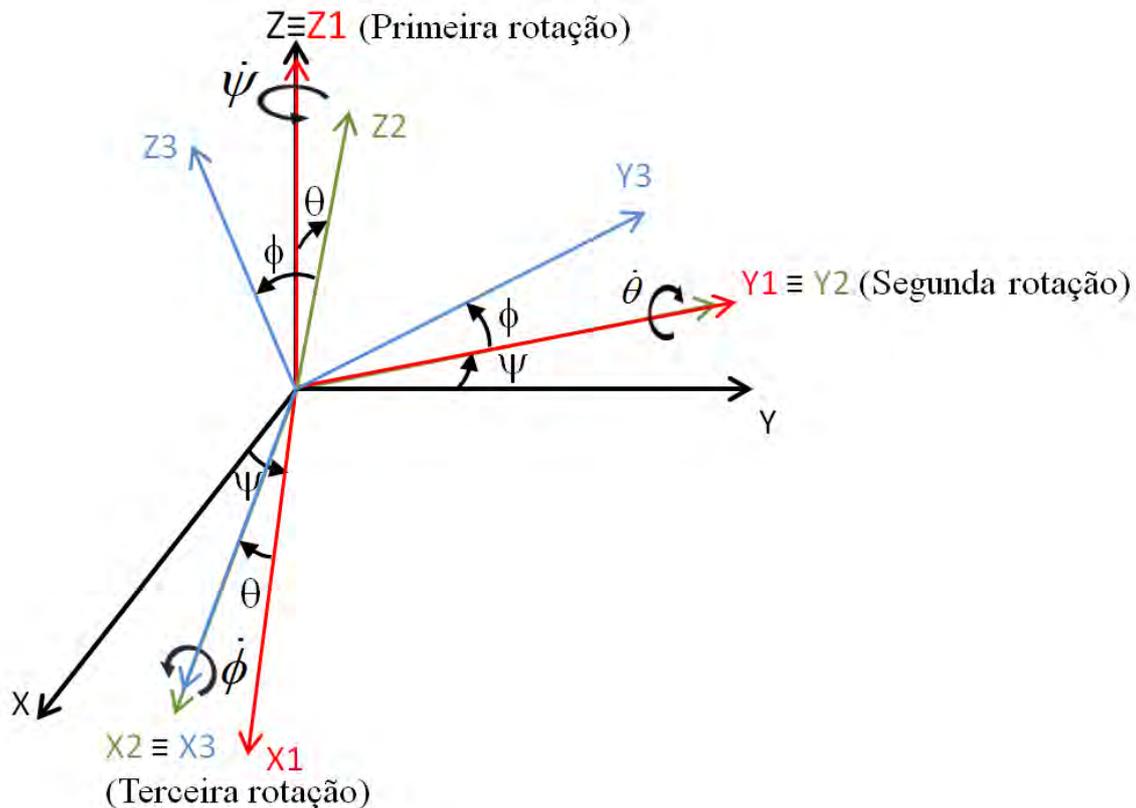


Figura 6.22 - Sequência de Rotações 3-2-1 dos respectivos ângulos (ψ , θ , ϕ)

O sistema de equações da Cinemática, descritos em ângulos de Euler é dependente da sequência de rotações escolhida para mudar de um sistema de referência para outro. Deste modo, para encontrar a relação dos ângulos de Euler, sua variação e a velocidade angular do corpo em rotação devemos analisar esse movimento de rotação. Como visto, na sequência 3-2-1 fez-se uma rotação em torno do eixo Z de um ângulo ψ que leva ao sistema X_1, Y_1, Z_1 ; uma rotação em torno do eixo Y_1 de um ângulo θ que leva ao sistema X_2, Y_2, Z_2 e, finalmente uma rotação em torno do eixo X_2 de um ângulo ϕ que leva ao sistema X_3, Y_3, Z_3 . Se imaginarmos o caminho inverso, para se voltar a posição inicial de ψ deve-se executar três rotações; para a posição inicial de θ , duas rotações e, para a posição inicial de ϕ , apenas uma rotação. Deste modo, para a sequência de rotações ($C_1(\psi) \leftarrow C_2(\theta) \leftarrow C_3(\phi)$), a velocidade angular de S relativa a O, $\omega_{O/S}$ é representada por:

$$\begin{bmatrix} \vec{\omega}_x \\ \vec{\omega}_y \\ \vec{\omega}_z \end{bmatrix} = C_1(\phi)C_2(\theta)C_3(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + C_1(\phi)C_2(\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_1(\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (63)$$

Fazendo-se a multiplicação de matrizes chega-se a:

$$\begin{aligned} \vec{\omega}_x &= \dot{\phi} - \dot{\psi} \operatorname{sen} \theta \\ \vec{\omega}_y &= \dot{\theta} \cos \phi + \dot{\psi} \operatorname{sen} \phi \cos \theta \\ \vec{\omega}_z &= \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \operatorname{sen} \phi \end{aligned} \quad (64)$$

De acordo com Sidi (1997), a primeira e última das equações (64) apresentam singularidade quando $\theta=90^\circ$.

Equação (64) em sua forma matricial:

$$\begin{bmatrix} \vec{\omega}_x \\ \vec{\omega}_y \\ \vec{\omega}_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\operatorname{sen} \theta \\ 0 & \cos \phi & \operatorname{sen} \phi \cos \theta \\ 0 & -\operatorname{sen} \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (65)$$

Como $\omega = \omega_{O/S} - \omega_0 o_2$, onde ω_0 é a velocidade orbital do satélite e, conseqüentemente, a velocidade angular com que o referencial VLHL (O) gira em relação ao referencial inercial (considerando-se uma órbita circular), então:

$$\begin{bmatrix} \vec{\omega}_x \\ \vec{\omega}_y \\ \vec{\omega}_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\operatorname{sen} \theta \\ 0 & \cos \psi & \operatorname{sen} \psi \cos \theta \\ 0 & -\operatorname{sen} \psi & \cos \psi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \omega_0 \begin{bmatrix} \cos \theta \operatorname{sen} \varphi \\ \operatorname{sen} \psi \operatorname{sen} \theta \operatorname{sen} \varphi + \cos \psi \cos \varphi \\ \cos \psi \operatorname{sen} \theta \operatorname{sen} \varphi - \operatorname{sen} \psi \cos \varphi \end{bmatrix} \quad (66)$$

Com isso, a equação diferencial da cinemática de um corpo rígido em órbita, representando a sequência de rotações 3-2-1, e que também pode ser vista em (ARANTES JÚNIOR, 2005), fica na forma apresentada:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos \theta} \begin{bmatrix} \cos \theta & \sin \varphi \sin \theta & \cos \varphi \sin \theta \\ 0 & \cos \varphi \cos \theta & -\sin \varphi \cos \theta \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \frac{\omega_0}{\cos \theta} \begin{bmatrix} \sin \psi \\ \cos \theta \cos \psi \\ \sin \theta \sin \psi \end{bmatrix} \quad (67)$$

Com foi dito anteriormente, os ângulos de Euler possuem singularidades. Deste modo, sua utilização é aconselhada nos casos que se garanta que a atitude estará distante dos pontos de singularidade.

6.3.2.1.1.3. Quatérnios

A representação por quatérnios, ou parâmetros simétricos de Euler, provê uma parametrização muito conveniente da atitude. Ela é mais compacta que a matrix de cossenos diretores, porque somente quatro parâmetros, ao invés de nove, são necessários. Ela é mais conveniente que os ângulos de Euler porque não apresenta funções trigonométricas, as quais requerem tempo de processamento na computação (WERTZ, 1978).

Esta representação é baseada na rotação de um ângulo Φ , ao longo de um eixo \hat{e} , com componentes (e_1, e_2, e_3) em relação a um sistema de referência. Normalmente, o eixo de rotação não é nenhum dos eixos do sistema de referência. A Figura 6.23 representa essa rotação.

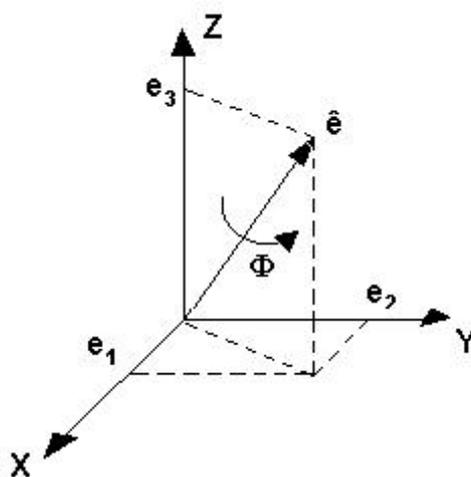


Figura 6.23 - Rotação em torno do eixo \hat{e} de um ângulo

O vetor de quatérnios (q_1, q_2, q_3, q_4) é definido a partir dos parâmetros descritos na Figura 6.23, da seguinte forma:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} e_1 \sin\left(\frac{\phi}{2}\right) \\ e_2 \sin\left(\frac{\phi}{2}\right) \\ e_3 \sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} \quad (68)$$

A equação cinemática representada por quatérnios, para levar do referencial orbital para o referencial do corpo é dada por:

$$\dot{q} = \frac{1}{2} \begin{bmatrix} 0 & w_x & -w_y + w_0 & w_x \\ -w_z & 0 & w_x & w_y + w_0 \\ w_y - w_0 & -w_x & 0 & w_z \\ w_x & -w_y - w_0 & -w_z & 0 \end{bmatrix} q \quad (69)$$

6.3.2.2. Dinâmica

Como já dito, o movimento de um satélite artificial é especificado por sua posição e velocidades orbitais, atitude e movimento de atitude. A posição e velocidade orbitais descrevem o movimento de translação do centro de massa do satélite, ou seja, seu movimento orbital. A atitude e o movimento de atitude descrevem o movimento rotacional do satélite em torno do seu centro de massa. O foco deste trabalho é o movimento de atitude, desta forma, é apresentada a seguir a modelagem do movimento de rotação.

6.3.2.2.1. Equações do movimento de rotação

6.3.2.2.1.1. Rotação de um sistema de coordenadas em relação a um sistema de coordenadas inerciais

Para este trabalho é necessário o desenvolvimento de uma equação que represente a rotação de um sistema de coordenadas em relação a um sistema de coordenadas inerciais. Assim, de acordo com Hibbeler (2010):

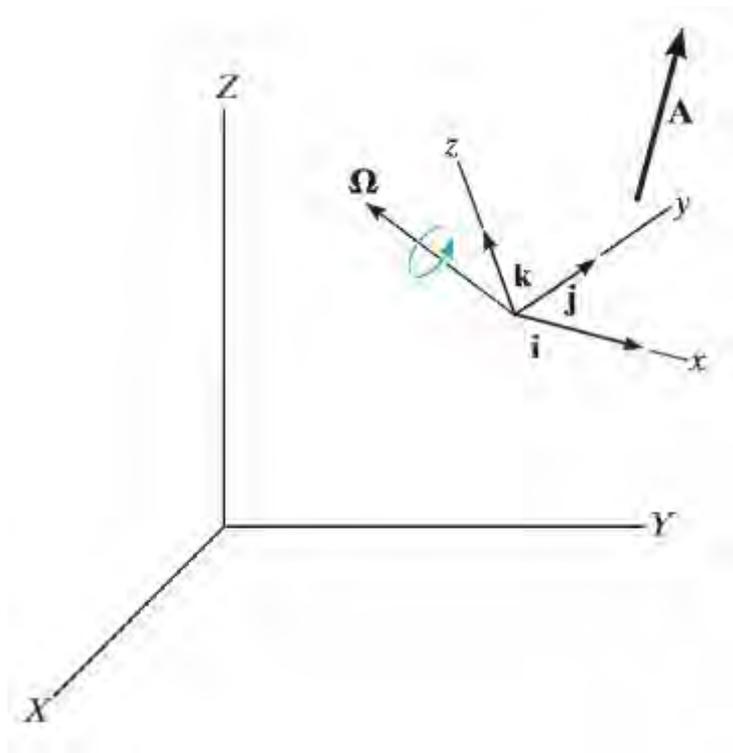


Figura 6.24 - Vetor em um sistema de coordenadas girante

Fonte: HIBBELER (2010)

A Figura 6.24 ilustra um vetor \mathbf{A} em um sistema de coordenadas x,y,z em rotação. A velocidade angular Ω deste sistema de coordenadas é medida em relação a um sistema de coordenadas inerciais.

O vetor \mathbf{A} expresso em função de suas componentes no sistema de coordenadas em rotação é:

$$\mathbf{A} = A_x \mathbf{i} + A_y \mathbf{j} + A_z \mathbf{k} \quad (70)$$

A derivada no tempo de \mathbf{A} em relação ao sistema de coordenadas inerciais resulta na equação (71). Nota-se que também há variação das direções (\mathbf{i} , \mathbf{j} , \mathbf{k}), pois este sistema está em rotação em relação ao sistema de coordenadas inerciais:

$$\dot{\mathbf{A}} = \dot{A}_x \mathbf{i} + \dot{A}_y \mathbf{j} + \dot{A}_z \mathbf{k} + A_x \dot{\mathbf{i}} + A_y \dot{\mathbf{j}} + A_z \dot{\mathbf{k}} \quad (71)$$

Como mostrado na Figura 6.25, $\dot{\mathbf{i}}$ representa apenas a mudança na direção de \mathbf{i} em relação ao tempo, pois \mathbf{i} tem magnitude 1.

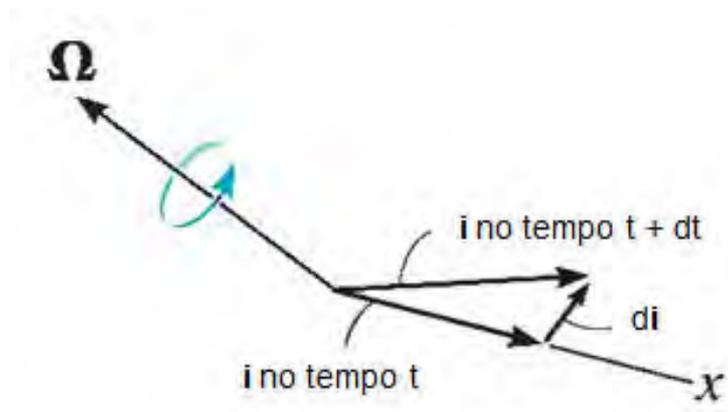


Figura 6.25 – Relação de \mathbf{i} e $d\mathbf{i}$

Fonte: HIBBELER (2010)

Como $d\mathbf{i}$ é perpendicular a \mathbf{i} , podemos definir:

$$\dot{\mathbf{i}} = \boldsymbol{\Omega} \times \mathbf{i} \quad (72)$$

Da mesma maneira, temos:

$$\dot{\mathbf{j}} = \boldsymbol{\Omega} \times \mathbf{j} \quad (73)$$

e

$$\dot{\mathbf{k}} = \boldsymbol{\Omega} \times \mathbf{k} \quad (74)$$

Substituindo:

$$\dot{\mathbf{A}} = \dot{A}_x \mathbf{i} + \dot{A}_y \mathbf{j} + \dot{A}_z \mathbf{k} + A_x (\boldsymbol{\Omega} \times \mathbf{i}) + A_y (\boldsymbol{\Omega} \times \mathbf{j}) + A_z (\boldsymbol{\Omega} \times \mathbf{k}) \quad (75)$$

Utilizando a seguinte propriedade de vetores:

$$(l\mathbf{a}) \times \mathbf{b} = l(\mathbf{a} \times \mathbf{b}) = \mathbf{a} \times (l\mathbf{b}) \quad (76)$$

Tem-se que:

$$A_x (\boldsymbol{\Omega} \times \mathbf{i}) = \boldsymbol{\Omega} \times (A_x \mathbf{i}) \quad (77)$$

$$A_y (\boldsymbol{\Omega} \times \mathbf{j}) = \boldsymbol{\Omega} \times (A_y \mathbf{j}) \quad (78)$$

$$A_z (\boldsymbol{\Omega} \times \mathbf{k}) = \boldsymbol{\Omega} \times (A_z \mathbf{k}) \quad (79)$$

Assim, chega-se a:

$$\dot{\mathbf{A}} = (\dot{\mathbf{A}})_{xyz} + \boldsymbol{\Omega} \times \mathbf{A} \quad (80)$$

6.3.2.2.1.2. Quantidade de movimento angular

Segundo Hibbeler (2010), considerando o corpo rígido da figura (6.26) o qual tem massa m e centro de massa em G , tem-se como formulação da quantidade de movimento angular da partícula i em torno do ponto A :

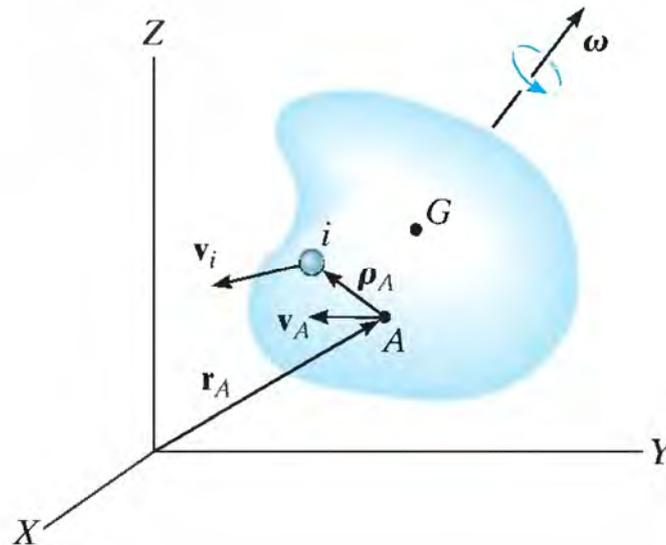


Figura 6.26 – Sistema de coordenadas inerciais

Fonte: HIBBELER (2010).

$$(\mathbf{H}_A)_i = \boldsymbol{\rho}_A \times m_i \mathbf{v}_i \quad (81)$$

Onde \mathbf{v}_i representa a velocidade da partícula medida a partir do sistema de coordenadas inerciais (X, Y, Z). Se o corpo tem uma velocidade angular $\boldsymbol{\omega}$ no instante considerado, \mathbf{v}_i deve ser relacionado com a velocidade de A utilizando-se a equação (80).

$$\mathbf{v}_i = \mathbf{v}_A + \boldsymbol{\omega} \times \boldsymbol{\rho}_A \quad (82)$$

Assim,

$$(\mathbf{H}_A)_i = \boldsymbol{\rho}_A \times m_i (\mathbf{v}_A + \boldsymbol{\omega} \times \boldsymbol{\rho}_A) \quad (83)$$

$$(\mathbf{H}_A)_i = (\boldsymbol{\rho}_A m_i) \times \mathbf{v}_A + \boldsymbol{\rho}_A \times (\boldsymbol{\omega} \times \boldsymbol{\rho}_A) m_i$$

Para a soma da quantidade de movimento de todas as partículas do corpo é preciso uma integração. Como $m_i \rightarrow dm$, temos:

$$\mathbf{H}_A = \left(\int_m \boldsymbol{\rho}_A dm \right) \times \mathbf{v}_A + \int_m \boldsymbol{\rho}_A \times (\boldsymbol{\omega} \times \boldsymbol{\rho}_A) dm \quad (84)$$

Para o trabalho, considerou-se que A está localizado no centro de massa (G) do corpo. Deste modo, $\int_m \boldsymbol{\rho}_A dm = 0$, portanto chega-se a:

$$\mathbf{H} = \int_m \boldsymbol{\rho} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}) dm \quad (85)$$

Expresso em componentes de x,y,z tem-se:

$$H_x \mathbf{i} + H_y \mathbf{j} + H_z \mathbf{k} = \int_m (x\mathbf{i} + y\mathbf{j} + z\mathbf{k}) \times [(\omega_x \mathbf{i} + \omega_y \mathbf{j} + \omega_z \mathbf{k}) \times (x\mathbf{i} + y\mathbf{j} + z\mathbf{k})] dm \quad (86)$$

Expandindo o produto vetorial e combinando os termos chega-se a:

$$\begin{aligned} H_x \mathbf{i} + H_y \mathbf{j} + H_z \mathbf{k} &= [\omega_x \int_m (y^2 + z^2) dm - \omega_y \int_m xy dm - \omega_z \int_m xz dm] \mathbf{i} \\ &+ [-\omega_x \int_m xy dm + \omega_y \int_m (x^2 + z^2) dm - \omega_z \int_m yz dm] \mathbf{j} \\ &+ [-\omega_x \int_m zx dm - \omega_y \int_m yz dm + \omega_z \int_m (x^2 + y^2) dm] \mathbf{k} \end{aligned} \quad (87)$$

Identificando os momentos e produtos de inércia chega-se a:

$$\begin{aligned}
H_x &= I_{xx}\omega_x - I_{xy}\omega_y - I_{xz}\omega_z \\
H_y &= -I_{yx}\omega_x + I_{yy}\omega_y - I_{yz}\omega_z \\
H_z &= -I_{zx}\omega_x - I_{zy}\omega_y + I_{zz}\omega_z
\end{aligned} \tag{88}$$

Se os eixos de referência forem alinhados de maneira que coincidam com os eixos principais de inércia, os produtos de inércia tornam-se nulos, e definindo-se $I_{xx} \equiv I_x$, $I_{yy} \equiv I_y$ e $I_{zz} \equiv I_z$ chega-se ao momento angular de cada eixo:

$$\begin{aligned}
H_x &= I_x\omega_x \\
H_y &= I_y\omega_y \\
H_z &= I_z\omega_z
\end{aligned} \tag{89}$$

Ainda de acordo com Hibbeler, a somatória de todos os torques externos aplicados em um corpo é igual a taxa de variação da quantidade de movimento angular deste corpo:

$$\sum \mathbf{M} = \dot{\mathbf{H}} \tag{90}$$

6.3.2.2.2. Equações de movimento de Euler

Como mostrado na eq. (90), a equação que representa a variação da quantidade de movimento angular, de um corpo no espaço, devido aos torques externos que agem nesse corpo é:

$$\sum \mathbf{M} = \dot{\mathbf{H}} \tag{91}$$

No entanto, como foi dito, o sistema de coordenadas do satélite gira em relação ao sistema de coordenadas inerciais. Desta forma, é necessário considerar

esta rotação. Para tanto, utiliza-se a equação (80). Assim, chega-se a equação do movimento de rotação de um corpo no espaço:

$$\sum \mathbf{M} = (\dot{\mathbf{H}})_{xyz} + \boldsymbol{\Omega} \times \mathbf{H} \quad (92)$$

Como os eixos do sistema em rotação estão fixados no corpo, tem-se que a velocidade angular do sistema de coordenadas em rotação $\boldsymbol{\Omega}$ é igual à velocidade angular do corpo $\boldsymbol{\omega}$. Logo:

$$\sum \mathbf{M} = (\dot{\mathbf{H}})_{xyz} + \boldsymbol{\omega} \times \mathbf{H} \quad (93)$$

Escrevendo esta equação em sua forma escalar e a relacionando com momentos e produtos de inércia, temos:

$$\begin{aligned} \sum M_x &= I_{xx} \dot{\omega}_x - (I_{yy} - I_{zz}) \omega_y \omega_z - I_{xy} (\dot{\omega}_y - \omega_z \omega_x) - I_{yz} (\omega_y^2 - \omega_z^2) - I_{zx} (\dot{\omega}_z - \omega_x \omega_y) \\ \sum M_y &= I_{yy} \dot{\omega}_y - (I_{zz} - I_{xx}) \omega_z \omega_x - I_{yz} (\dot{\omega}_z - \omega_x \omega_y) - I_{zx} (\omega_z^2 - \omega_x^2) - I_{xy} (\dot{\omega}_x - \omega_y \omega_z) \\ \sum M_z &= I_{zz} \dot{\omega}_z - (I_{xx} - I_{yy}) \omega_x \omega_y - I_{zx} (\dot{\omega}_x - \omega_y \omega_z) - I_{xy} (\omega_x^2 - \omega_y^2) - I_{yz} (\dot{\omega}_y - \omega_z \omega_x) \end{aligned} \quad (99)$$

Como os eixos do referencial do corpo estão alinhados com os eixos principais de inércia, os produtos de inércia são nulos, e definindo $I_{xx} \equiv I_x$, $I_{yy} \equiv I_y$ e $I_{zz} \equiv I_z$ chegamos às equações:

$$\begin{aligned} M_x &= I_x \dot{\omega}_x - (I_y - I_z) \omega_y \omega_z \\ M_y &= I_y \dot{\omega}_y - (I_z - I_x) \omega_z \omega_x \\ M_z &= I_z \dot{\omega}_z - (I_x - I_y) \omega_x \omega_y \end{aligned} \quad (100)$$

Esse conjunto de equações é conhecido como equações de movimento de Euler.

6.3.3. Atuador

O modelo do atuador utilizado nesse trabalho foi obtido em Souza (1980) e Gobato (2006) e é baseado em uma aproximação linear da curva característica de um servomotor CC e seu respectivo diagrama de blocos como mostrado abaixo:

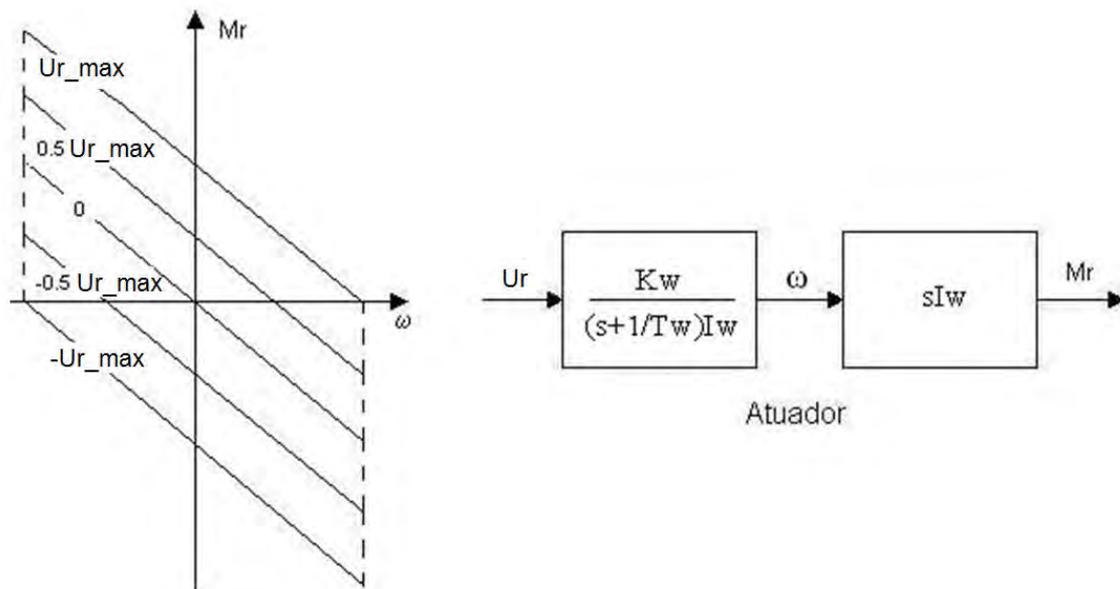


Figura 6.27 - Aproximação linear da curva característica do servomotor e diagrama de blocos correspondente

Fonte: Gobato (2006)

A saída é o torque (em Nm) produzido pela roda e a entrada é o valor de tensão comandado (em Volts).

De acordo com Souza (1980), o cálculo dos parâmetros da roda pode ser efetuado utilizando as equações abaixo:

$$T_w = \frac{I_R \cdot \omega_{Rmax}}{M_{Rmax}} \quad (103)$$

$$K_w = \frac{M_{R \max}}{U_{R \max}} \quad (104)$$

Onde T_w é a constante de tempo da roda; K_w é o ganho da roda; I_R é a inércia da roda; $\omega_{R \max}$ velocidade angular da roda; $M_{R \max}$ o torque da roda; e $U_{R \max}$ a tensão da roda.

Tabela 6.1 – Parâmetros da roda de reação utilizada nesse trabalho

I_R	0.015 [kg.m ²]	$T_w = 20[s]$	$K_w = 0.06$
$\omega_{R \max}$	7500 [r.p.m]		
$M_{R \max}$	0.6 [N.m]		
$U_{R \max}$	10 [V]		

São apresentadas abaixo as equações em funções de transferência da figura 6.27:

$$\frac{\omega_r}{U_r} = \frac{K_w}{(s + \frac{1}{T_w})I_w} \quad (105)$$

$$\frac{M_r}{U_r} = sI_w \left(\frac{K_w}{(s + \frac{1}{T_w})I_w} \right) \quad (106)$$

A partir das equações 105 e 106 chega-se a:

$$\dot{\omega}_r(t) = \frac{K_w}{I_w} u_r(t) - \frac{\omega_r(t)}{T_w} \quad (107)$$

$$M_r(t) = I_w \cdot \dot{\omega}_r(t) \quad (108)$$

Assim, para os três eixos têm-se:

$$\begin{aligned}\dot{\omega}_{rx}(t) &= \frac{k_w}{I_r} u_r(t) - \frac{\omega_{rx}(t)}{T_w} \\ \dot{\omega}_{ry}(t) &= \frac{k_w}{I_r} u_r(t) - \frac{\omega_{ry}(t)}{T_w} \\ \dot{\omega}_{rz}(t) &= \frac{k_w}{I_r} u_r(t) - \frac{\omega_{rz}(t)}{T_w}\end{aligned}\quad (109)$$

$$\begin{aligned}M_{rx}(t) &= I_w \cdot \dot{\omega}_{rx}(t) \\ M_{ry}(t) &= I_w \cdot \dot{\omega}_{ry}(t) \\ M_{rz}(t) &= I_w \cdot \dot{\omega}_{rz}(t)\end{aligned}\quad (110)$$

6.3.4. Controlador

O controlador utilizado foi o mesmo que o utilizado para o sistema Massa-Mola-Amortecedor, ou seja, PID (Proporcional-Integral-Derivativo). Também como no Massa-Mola-Amortecedor, o comportamento dinâmico desse sistema foi descrito em termos de dois parâmetros ζ (fator de amortecimento) e ω_n (frequência natural não amortecida) para obtenção dos ganhos (Kp, Td, Ti) do PID. A modelagem foi feita em cada eixo. As Figuras 6.28, 6.29 e 6.30 ilustram o diagrama de blocos do PID em cada eixo.

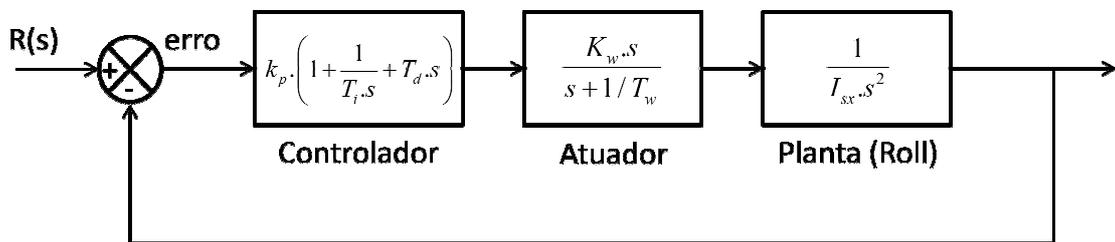


Figura 6.28 - Diagrama de Blocos (Roll)

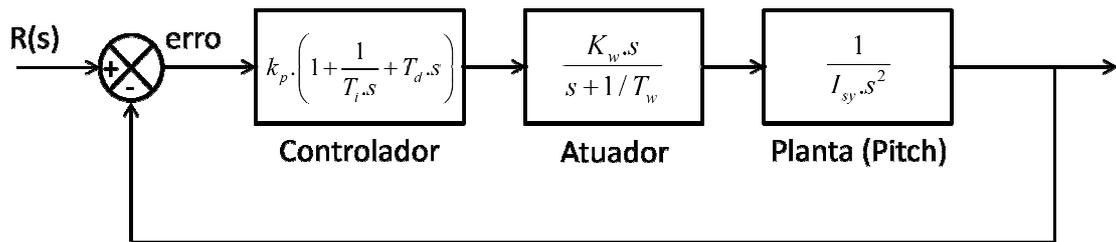


Figura 6.29 - Diagrama de Blocos (Pitch)

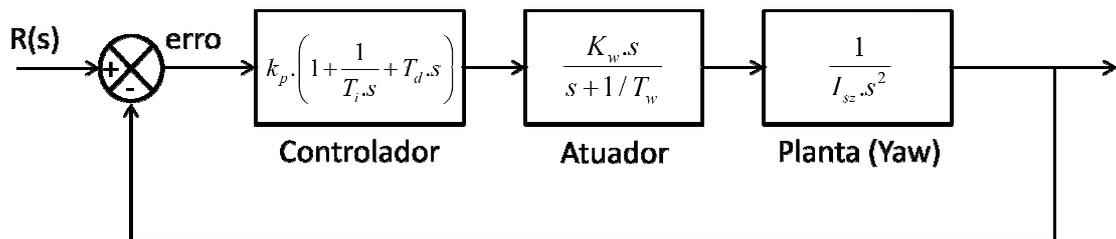


Figura 6.30 - Diagrama de Blocos (Yaw)

Os ganhos utilizados foram obtidos em Gobato (2006) e estão na Tabela 2:

Tabela 6.2 – Ganhos do PID nos três eixos

Eixo	Ganhos	
X	Kp	40.5931
	Td	454.1050
	Ti	1.0000
Y	Kp	51.7854
	Td	556.9350
	Ti	1.0000
Z	Kp	44.3541
	Td	488.6600
	Ti	1.0000

6.3.5. Sensor

Na simulação dos sensores, foi utilizado um sensor ideal, ou seja, os ângulos ϕ , θ e ψ foram lidos e reenviados.

6.3.6. Implementação computacional (Programação)

Para implementação computacional utilizaram-se os seguintes parâmetros (Tabelas 3 e 6) e o fluxograma da Figura 6.31:

6.3.6.1. Parâmetros orbitais:

Tabela 6.3 – Parâmetros orbitais

Altitude (h)	905 [km]
Excentricidade (e)	0
Inclinação (i)	0°
Velocidade Orbital Média (ω_0)	0.001016 [rad/s]
Período Orbital (T_0)	6184 [s]

6.3.6.2. Dados do satélite e da roda de reação:

Tabela 6.4 – Dados do satélite e da roda de reação

Momento de Inércia das Rodas (I_R)	0.015 [kg.m ²]
Velocidade Máxima das Rodas ($\omega_{Rm\acute{a}x}$)	7500 [r.p.m]
Torque Máximo das Rodas ($M_{Rm\acute{a}x}$)	0.6 [N.m]
Tensão Máxima na Saída do Controlador ($U_{Rm\acute{a}x}$)	10 [V]
Ganho das Rodas (K_W)	0.06
Constante de Tempo das Rodas (T_W)	20 [s]
Momento de Inércia do Satélite em X (I_{Sx})	295.71 [kg.m ²]
Momento de Inércia do Satélite em Y (I_{Sy})	501.37 [kg.m ²]
Momento de Inércia do Satélite em Z (I_{Sz})	364.82 [kg.m ²]

6.3.6.3. Dados do controlador PID:

Tabela 6.5 – Ganhos do PID nos três eixos

Eixo	Ganhos	
X	Kp	40.5931
	Td	454.1050
	Ti	1.0000
Y	Kp	51.7854
	Td	556.9350
	Ti	1.0000
Z	Kp	44.3541
	Td	488.6600
	Ti	1.0000

6.3.6.4. Condições iniciais

Tabela 6.6 – Condições iniciais

Ângulo em X (ϕ)	30 [°]
Ângulo em Y (θ)	0 [°]
Ângulo em Z (ψ)	0 [°]
Velocidade angular do satélite em X (ω_x)	0 [rad/s]
Velocidade angular do satélite em Y (ω_y)	0.001016 [rad/s]
Velocidade angular do satélite em Z (ω_z)	0 [rad/s]
Torque da roda em X (M_{Rx})	0 [N.m]
Torque da roda em Y (M_{Ry})	0 [N.m]
Torque da roda em Z (M_{Rz})	0 [N.m]
Velocidade angular da roda em X (ω_x)	0 [rad/s]
Velocidade angular da roda em Y (ω_y)	0 [rad/s]
Velocidade angular da roda em Z (ω_z)	0 [rad/s]
Referência em X (ϕ)	0 [°]
Referência em Y (θ)	0 [°]
Referência em Z (ψ)	0 [°]

6.3.6.5. Funcionamento do programa:

O programa inicia-se com as condições iniciais apresentadas na tabela 6. Em seguida, as acelerações angulares ($\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z$) obtidas a partir da modelagem da dinâmica (equação 100) são integradas para se obter as velocidades angulares ($\omega_x, \omega_y, \omega_z$). As velocidades angulares obtidas são inseridas na modelagem da cinemática (equação 67) e os $\dot{\phi}_x, \dot{\theta}_y, \dot{\psi}_z$ são integrados para se obter os ângulos de Euler, ϕ_x, θ_y, ψ_z . Os ângulos são lidos pelos sensores e enviados (sensor perfeito) para o controlador. O controlador compara os ângulos lidos com os ângulos de referência e, com intuito de zerar a diferença entre eles, envia sinais de controle (u_x, u_y, u_z) para os atuadores. A partir da modelagem dos atuadores (equação 109) é possível se obter as acelerações angulares ($\dot{\omega}_{rx}, \dot{\omega}_{ry}, \dot{\omega}_{rz}$) dos atuadores (rodas de reação). Integrando-se $\dot{\omega}_{rx}, \dot{\omega}_{ry}, \dot{\omega}_{rz}$, obtém-se as velocidades angulares ($\omega_{rx}, \omega_{ry}, \omega_{rz}$). E, a partir da equação 110, chega-se aos torques aplicados pelas rodas de reação (M_{rx}, M_{ry}, M_{rz}).

O fluxograma abaixo ilustra o funcionamento do programa em linguagem C e MatLab.

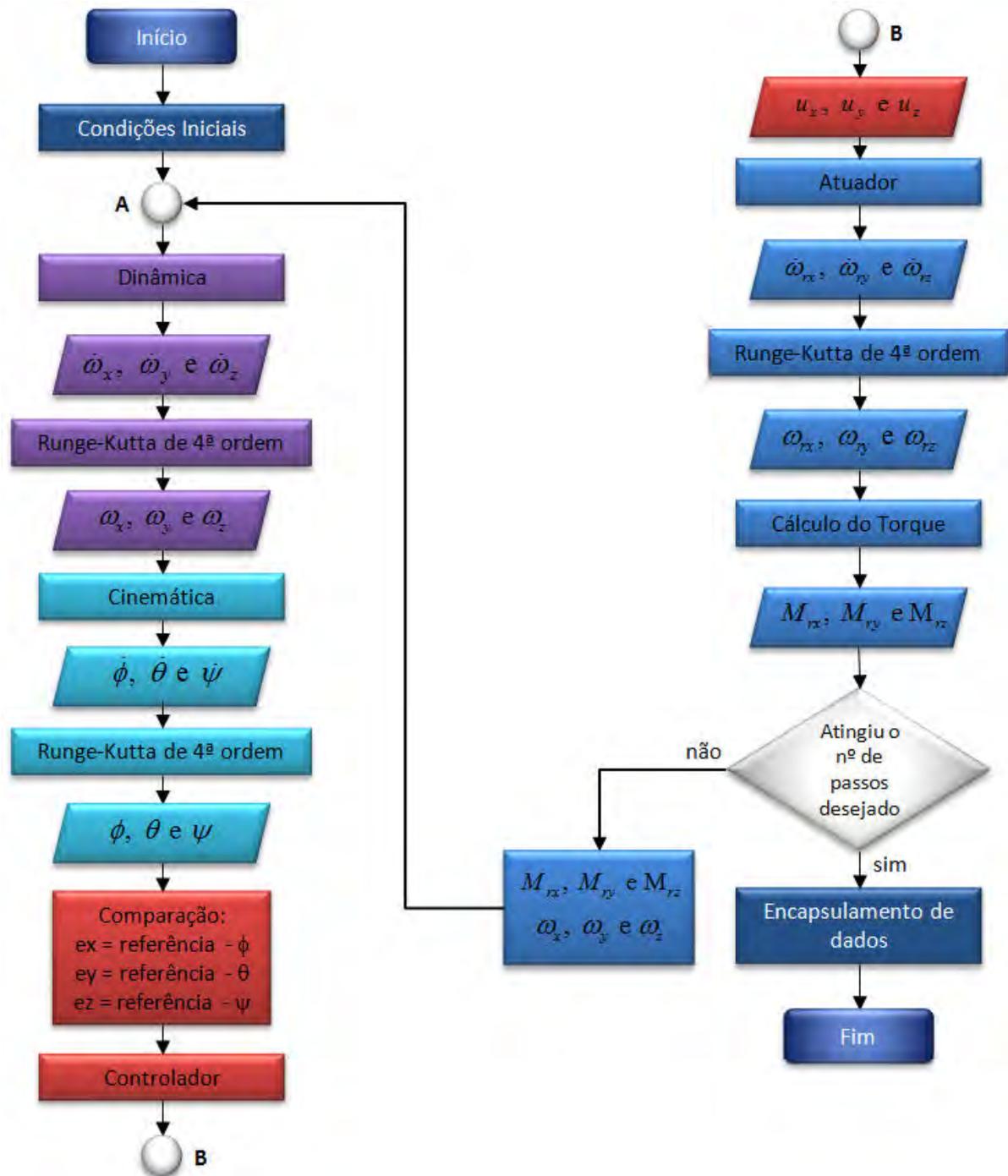


Figura 6.31 – Fluxograma do funcionamento do programa

6.4. Implementação do SCA no SIMA utilizando 4 partições

6.4.1. Visão geral

Para implementação do Sistema de Controle de Atitude da PMM no SIMA foram criadas 4 partições. Na primeira partição alocou-se o controlador, na segunda o atuador, na terceira a planta e na quarta o sensor, como mostrado na Figura 6.32:

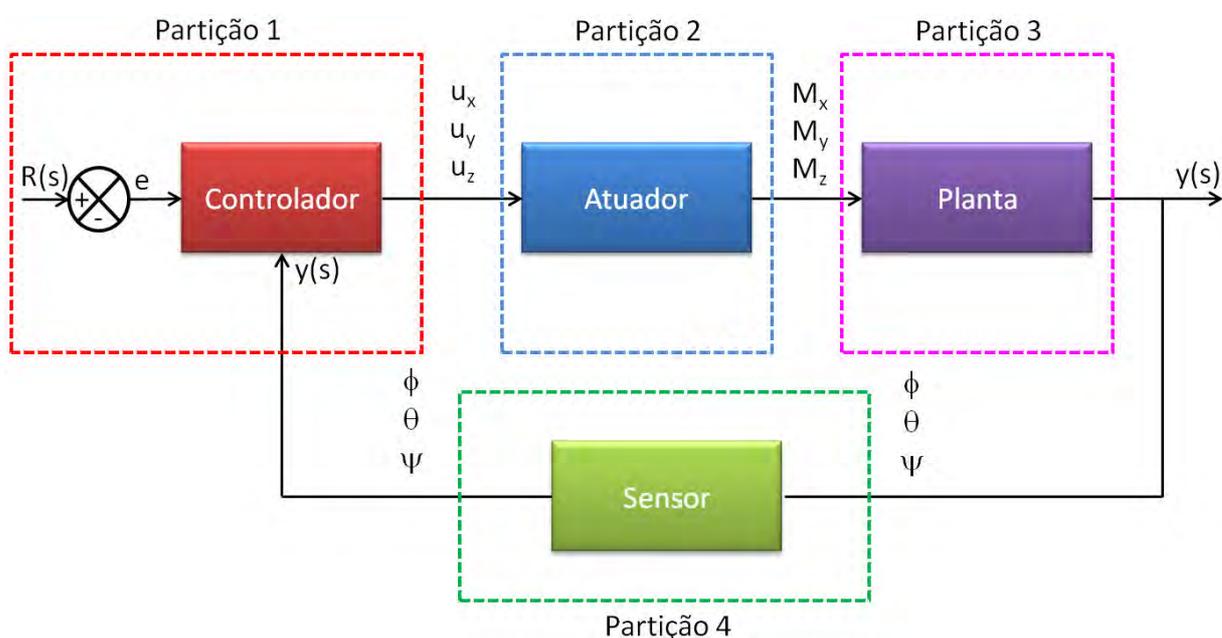


Figura 6.32 - AOCS dividido em quatro partições

6.4.2. Portas e canais

Como já mencionado, a comunicação entre partições é feita a partir de portas e canais. Deste modo, é necessário se criar canais de comunicação para que as mensagens em uma porta de origem (*source port*) sejam copiadas para área de memória da porta de destino (*destination port*).

Deste modo, para a comunicação entre as partições foram criados 12 canais e 24 portas de comunicação (*queuing ports*), como ilustrado na Figura 6.33:

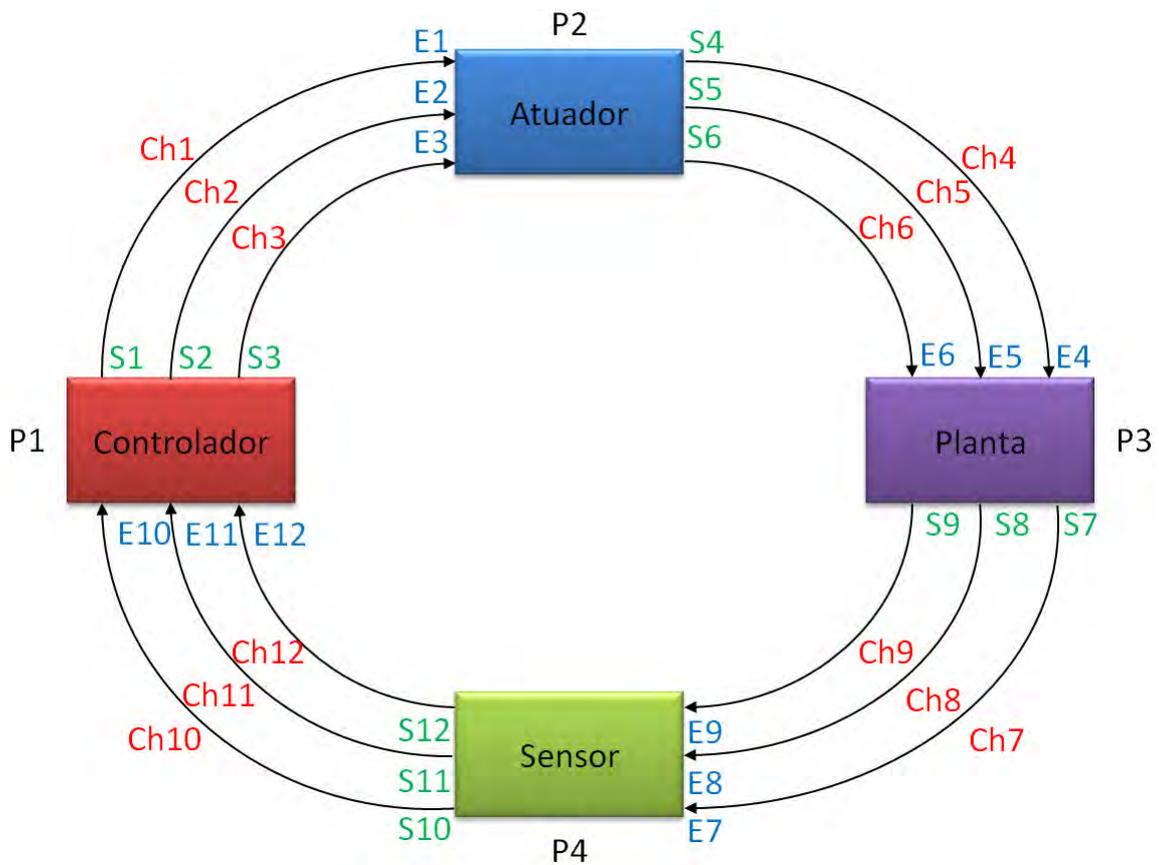


Figura 6.33 - Representação das portas e canais de comunicação do AOCS no SIMA

Nela, E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11 e E12 representam as portas de entrada de cada partição, e S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11 e S12 representam as portas de saída de cada partição.

Assim, o controlador envia os Controles U_x , U_y e U_z a serem executados pelo atuador através dos canais Ch1, Ch2 e Ch3. O atuador recebe os controles e envia os momentos M_x , M_y e M_z para a planta através dos canais Ch4, Ch5 e Ch6. A planta recebe os momentos e envia os ângulos ψ , θ e ϕ para os sensores através dos canais Ch7, Ch8 e Ch9. Fechando o ciclo, os sensores enviam ψ , θ e ϕ ao controlador através dos canais Ch10, Ch11, Ch12, como mostra a Figura 6.34:

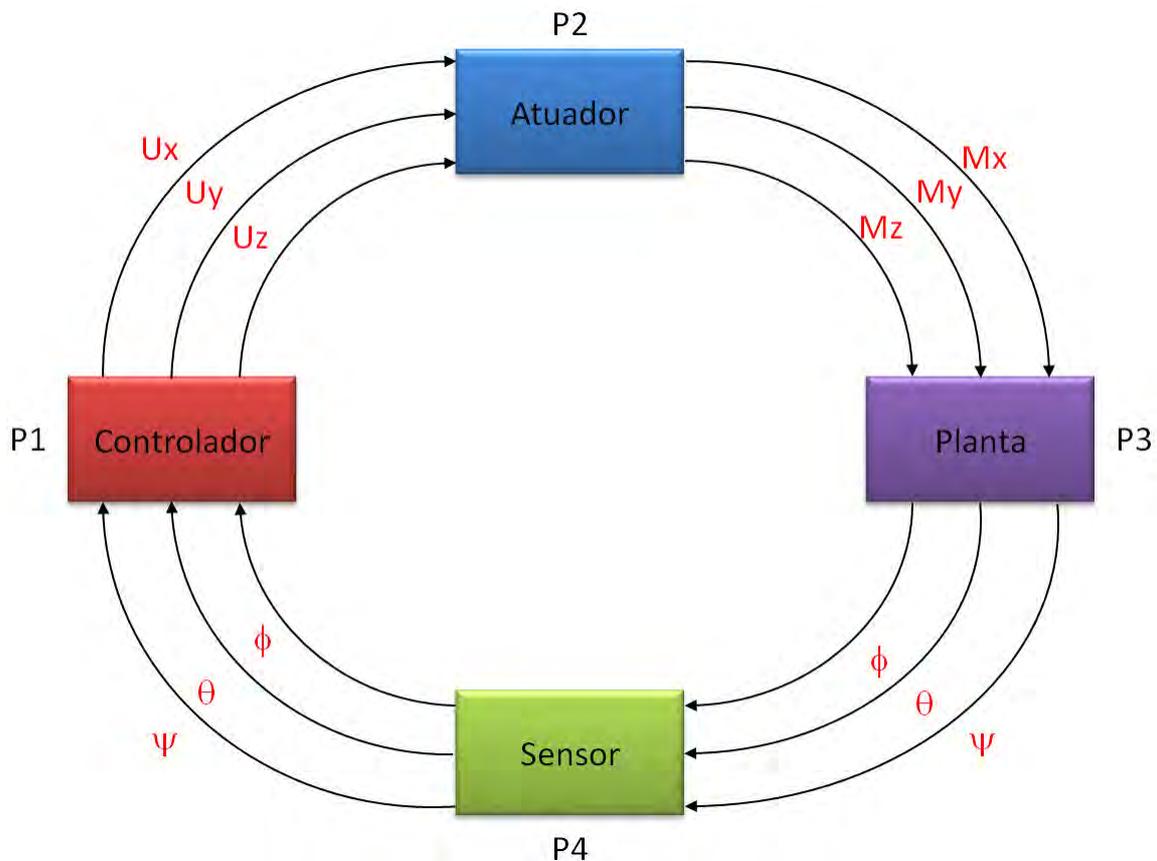


Figura 6.34 - Representação da comunicação entre as partições

6.4.2.1. Prioridade

Dentro de cada partição há diversos processos sendo executados. Para que haja uma sequência lógica, algumas prioridades foram definidas. Deste modo, em cada partição os processos de recebimento e leitura de dados tinham a maior prioridade, seguidos por cálculos executados a partir desses valores e, em seguida, os processos de envio de resultados. Além disso, foram definidas prioridades internas para os processos de leitura, cálculo e envio de dados. Portanto, as leituras no eixo x tinham maior prioridade que as leituras no eixo y que, por sua vez, tinham maior prioridade que as leituras no eixo z. As prioridades para os cálculos e envio de dados foram definidas da mesma maneira.

Como o controlador foi a primeira partição a ser executada, não havia resultado a ser lido, portanto, esta partição se inicia enviando os dados iniciais. Já as demais se iniciam lendo os resultados das partições anteriores.

6.4.2.2. Configuração

Nas Figuras 6.35 a 6.42 são apresentados os arquivos de configuração, a653.xml e sima.xml, das portas e canais utilizados no SIMA.

6.4.2.2.1. Partição 1

6.4.2.2.1.1. a653.xml

```
<Partition PartitionIdentifier="1"
  PartitionName="controlador"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do controlador-->
  <Queuing_Port Name="S1"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S2"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S3"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <!-- entradas do controlador-->
  <Queuing_Port Name="E10"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E11"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E12"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

</Partition>
```

Figura 6.35 – Configuração no arquivo a653.xml da Partição 1

6.4.2.2.1.2. sima.xml

```
<Partition PartitionIdentifier="1"
  PartitionName="controlador"
  SharedMemory="70010"
  Startup="/bin/sh ./samples/com/partitions/controlador/controlador.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/controlador/fifo_controlador"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>

  <Destination_Port
    Name="E10"
    Type="UDP"
    IP="127.0.0.1"
    Port="12311"/>

  <Destination_Port
    Name="E11"
    Type="UDP"
    IP="127.0.0.1"
    Port="12321"/>

  <Destination_Port
    Name="E12"
    Type="UDP"
    IP="127.0.0.1"
    Port="12331"/>
</Partition>
```

Figura 6.36 - Configuração no arquivo sima.xml da Partição 1

6.4.2.2.2. Partição 2

6.4.2.2.2.1. a653.xml

```
<Partition PartitionIdentifier="2"
  PartitionName="atuador"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do atuador-->
  <Queuing_Port Name="S4"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S5"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S6"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <!-- entradas do atuador-->
  <Queuing_Port Name="E1"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E2"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E3"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

</Partition>
```

Figura 6.37 - Configuração no arquivo a653.xml da Partição 2

6.4.2.2.2. sima.xml

```
<Partition PartitionIdentifier="2"
  PartitionName="atuador"
  SharedMemory="70020"
  Startup="/bin/sh ./samples/com/partitions/atuador/atuador.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/atuador/fifo_atuador"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>
  <Destination_Port
    Name="E1"
    Type="UDP"
    IP="127.0.0.1"
    Port="12341"/>

  <Destination_Port
    Name="E2"
    Type="UDP"
    IP="127.0.0.1"
    Port="12351"/>

  <Destination_Port
    Name="E3"
    Type="UDP"
    IP="127.0.0.1"
    Port="12361"/>
</Partition>
```

Figura 6.38 - Configuração no arquivo sima.xml da Partição 2

6.4.2.2.3. Partição 3

6.4.2.2.3.1. a653.xml

```
<Partition PartitionIdentifier="3"
  PartitionName="planta"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do planta-->
  <Queuing_Port Name="S7"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S8"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S9"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <!-- entradas do planta-->
  <Queuing_Port Name="E4"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E5"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E6"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

</Partition>
```

Figura 6.39 - Configuração no arquivo a653.xml da Partição 3

6.4.2.2.3.2. sima.xml

```
<Partition PartitionIdentifier="3"
  PartitionName="planta"
  SharedMemory="70030"
  Startup="/bin/sh ./samples/com/partitions/planta/planta.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/planta/fifo_planta"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>
  <Destination_Port
    Name="E4"
    Type="UDP"
    IP="127.0.0.1"
    Port="12401"/>

  <Destination_Port
    Name="E5"
    Type="UDP"
    IP="127.0.0.1"
    Port="12411"/>

  <Destination_Port
    Name="E6"
    Type="UDP"
    IP="127.0.0.1"
    Port="12421"/>
</Partition>
```

Figura 6.40 - Configuração no arquivo sima.xml da Partição 3

6.4.2.2.4. Partição 4

6.4.2.2.4.1. a653.xml

```
<Partition PartitionIdentifier="4"
  PartitionName="sensor"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do sensor-->
  <Queuing_Port Name="S10"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32" />
  <Queuing_Port Name="S11"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32" />
  <Queuing_Port Name="S12"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32" />

  <!-- entradas do sensor-->
  <Queuing_Port Name="E7"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32" />
  <Queuing_Port Name="E8"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32" />
  <Queuing_Port Name="E9"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32" />

</Partition>
```

Figura 6.41 - Configuração no arquivo a653.xml da Partição 4

6.4.2.2.4.2. sima.xml

```
<Partition PartitionIdentifier="4"
  PartitionName="sensor"
  SharedMemory="70040"
  Startup="/bin/sh ./samples/com/partitions/sensor/sensor.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/sensor/fifo_sensor"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>
  <Destination_Port
    Name="E7"
    Type="UDP"
    IP="127.0.0.1"
    Port="12371"/>
  <Destination_Port
    Name="E8"
    Type="UDP"
    IP="127.0.0.1"
    Port="12381"/>
  <Destination_Port
    Name="E9"
    Type="UDP"
    IP="127.0.0.1"
    Port="12391"/>
</Partition>
```

Figura 6.42 - Configuração no arquivo sima.xml da Partição 4

6.4.3. Agendando (Scheduling)

O agendamento das partições foi feito da seguinte forma: a janela de execução da partição 1 se inicia em 0,0 segundos e tem um tempo de 0,1 segundos de execução. Finalizando este intervalo de tempo, os processos desta partição sofrem preempção e a janela de execução da partição 2 ganha acesso ao processador. Esta, como a anterior, executa seus processos de 0,1 segundos.

Terminado esse tempo, a terceira partição ganha acesso ao processador, e assim sucessivamente. Quando a janela de execução da primeira partição for iniciada novamente, os processos que haviam sido interrompidos são retomados do ponto que estavam. A Figura 6.43 ilustra o agendamento das partições. Como pode ser visto o tempo total de execução de todas as janelas (1 ciclo) é 0.4 segundos.

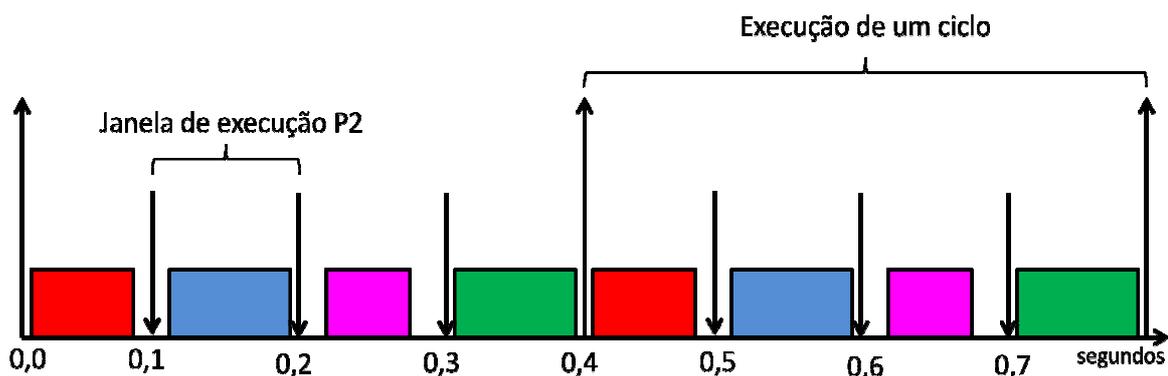


Figura 6.43 - Agendamento no tempo das partições do AOCS

As configurações de agendamento utilizadas nas simulações, também efetuadas no arquivo de configuração do SIMA (a653.xml) são ilustradas na Figura 6.44, seguir:

6.4.3.1. a653.xml

```
<Module_Schedule MajorFrameSeconds="0.4">
  <Partition_Schedule PartitionIdentifier="1"
    PartitionName="controlador"
    PeriodSeconds="0.4"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="101"
      WindowStartSeconds="0.0"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="2"
    PartitionName="atuador"
    PeriodSeconds="0.4"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="201"
      WindowStartSeconds="0.1"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="3"
    PartitionName="planta"
    PeriodSeconds="0.4"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="301"
      WindowStartSeconds="0.2"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="4"
    PartitionName="sensor"
    PeriodSeconds="0.4"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="401"
      WindowStartSeconds="0.3"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
</Module_Schedule>
```

Figura 6.44 – Configuração de agendamento no arquivo a653.xml das Partições 1 a 4

6.4.3.2. Funcionamento do programa no SIMA:

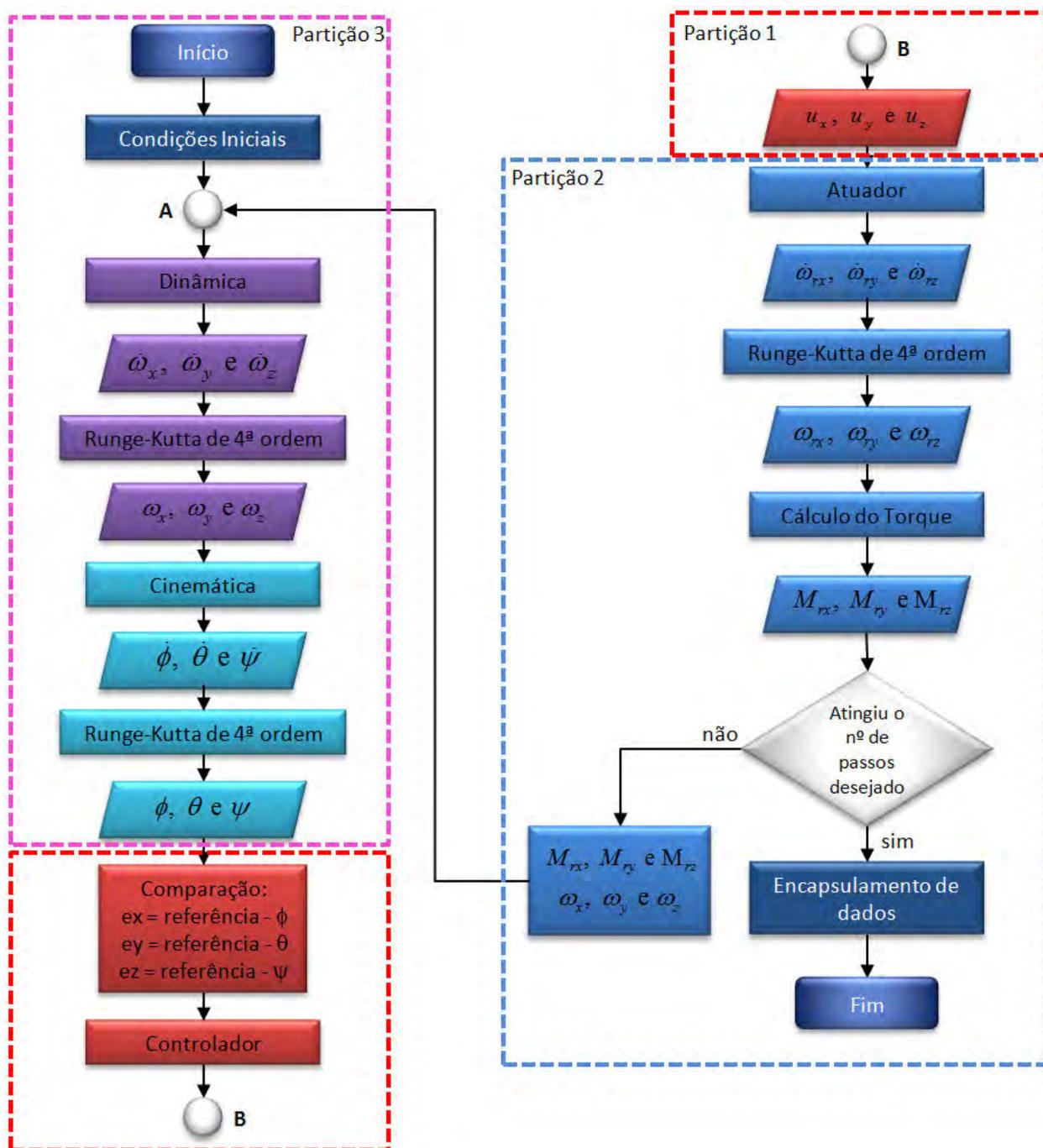


Figura 6.45 - Fluxograma do funcionamento do programa

A dinâmica dos sensores (partição 4) não é apresentada no fluxograma, pois, utilizou-se um sensor ideal. Desta forma, os valores são recebidos da partição 3 e reenviados para a partição 1.

6.5. Implementação do SCA e do MM no SIMA utilizando 6 partições

6.5.1. Visão geral

Para implementação do SCA da PMM e do Massa-Mola-Amortecedor no SIMA foram criadas 6 partições. Alocou-se o SCA nas 4 primeiras partições, do mesmo modo que no tópico anterior, e o MMA nas 2 partições restantes. A figura 6.46 ilustra a distribuição do SCA e do MMA no módulo:

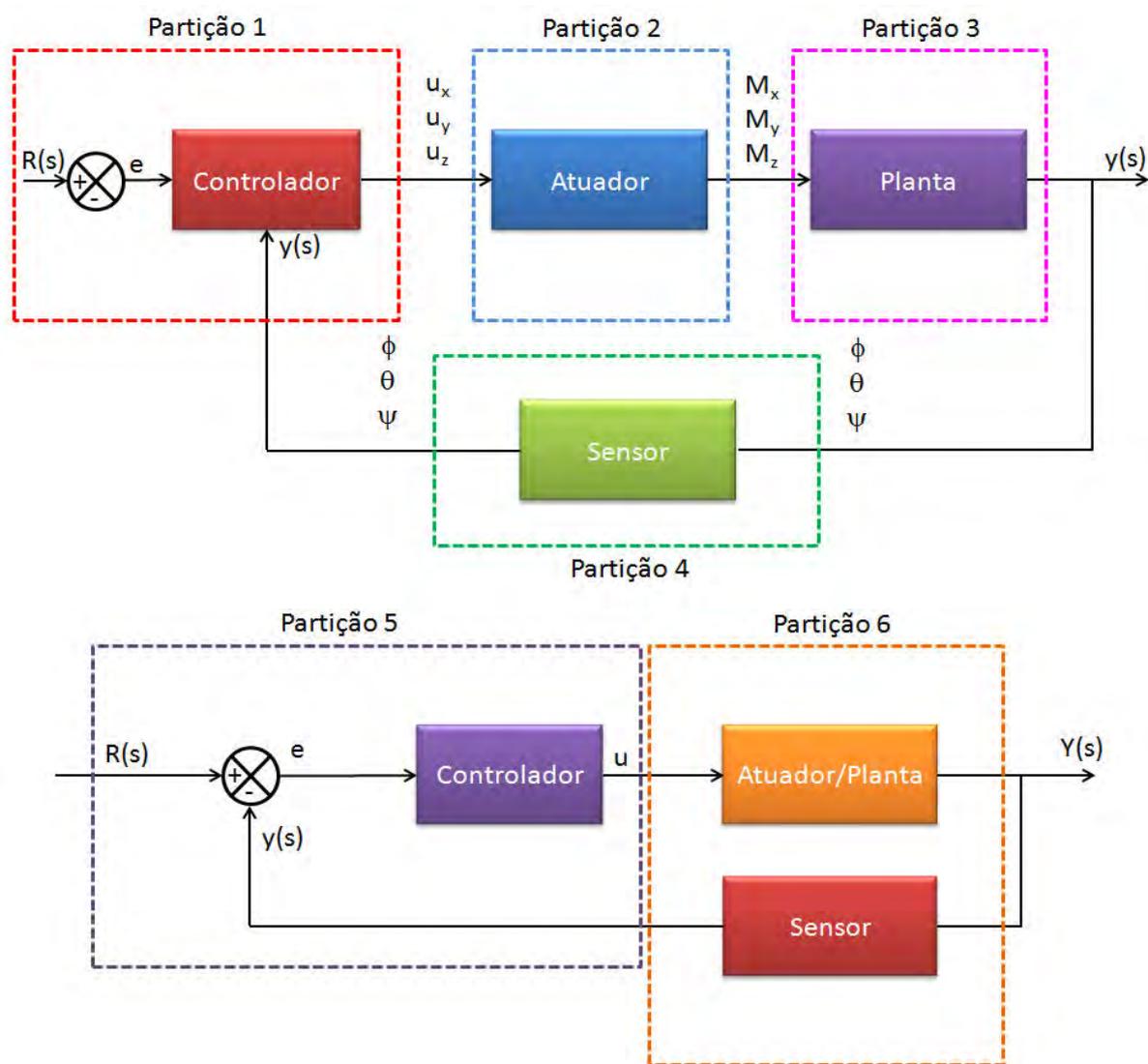


Figura 6.46 – SCA e MMA dividido em seis partições

É importante notar que não há comunicação entre o SCA e o MMA.

6.5.2. Portas e canais

Da mesma maneira que as implementações anteriores, é necessário se criar canais de comunicação para que as mensagens em uma porta de origem (*source port*) sejam copiadas para área de memória da porta de destino (*destination port*).

Deste modo, para a comunicação entre as partições foram criados 14 canais e 28 portas de comunicação (*queuing ports*), como ilustrado nas Figuras 6.47 e 6.48 abaixo:

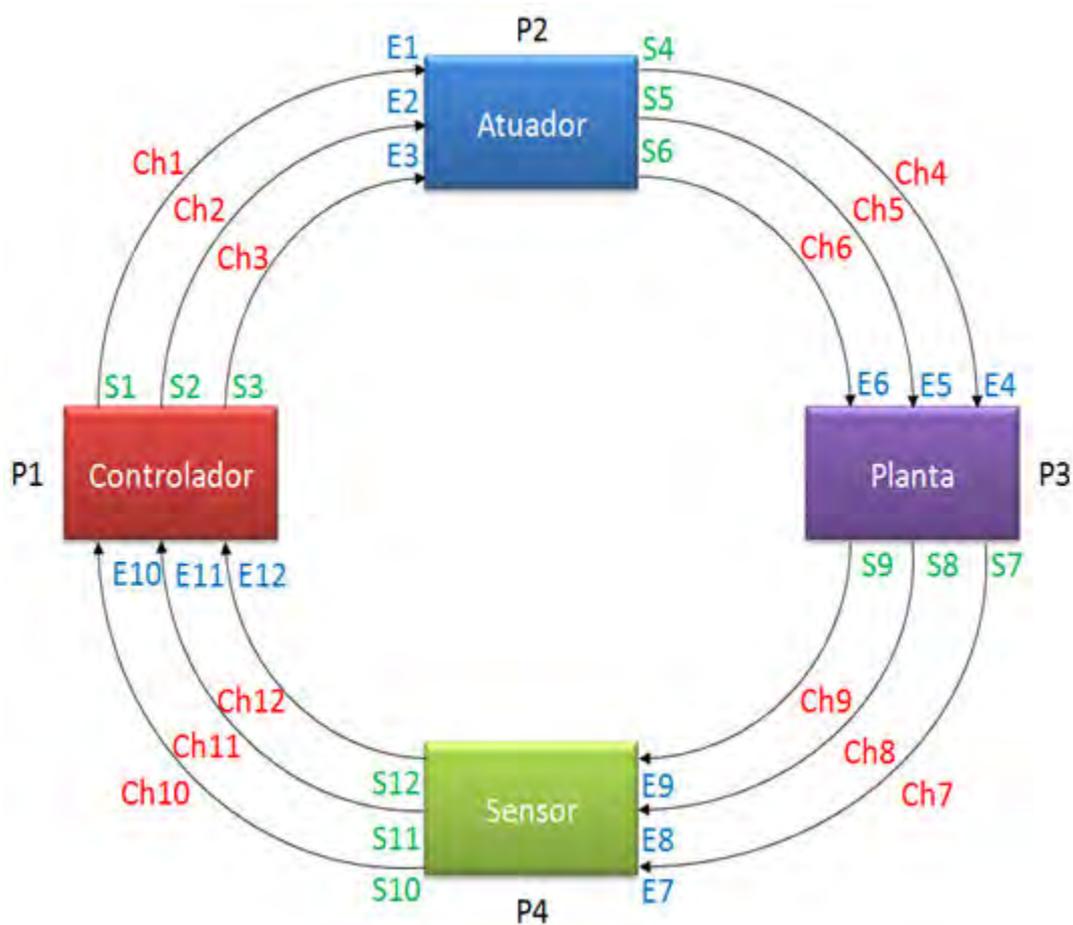


Figura 6.47 - Representação das portas e canais de comunicação do SCA

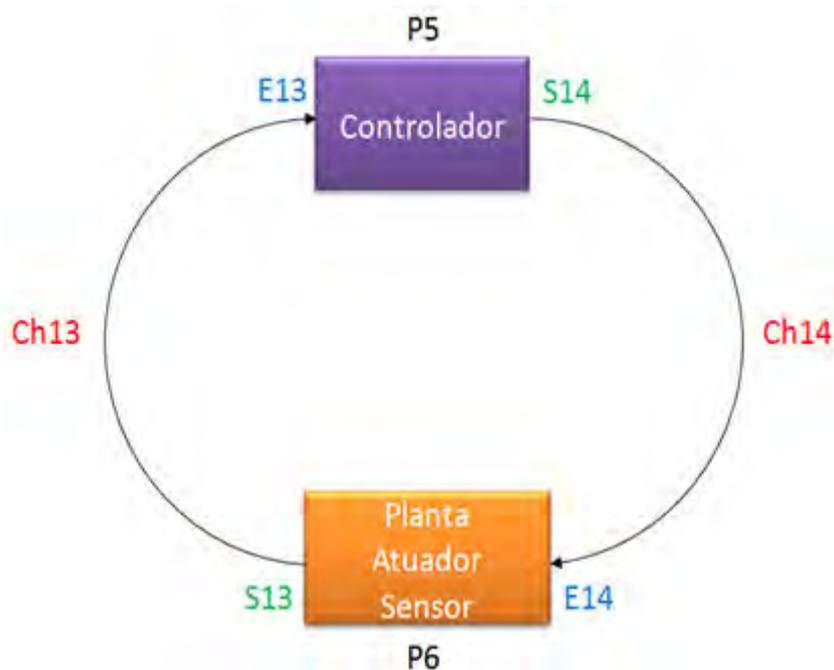


Figura 6.48 - Representação das portas e canais de comunicação do MM

Nelas, E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13 e E14 representam as portas de entrada de cada partição, e S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13 e S14 representam as portas de saída de cada partição.

Assim, o controlador envia os Controles U_x , U_y e U_z a serem executados pelo atuador através dos canais Ch1, Ch2 e Ch3. O atuador recebe os controles e envia os momentos M_x , M_y e M_z para a planta através dos canais Ch4, Ch5 e Ch6. A planta recebe os momentos e envia os ângulos ψ , θ e ϕ para os sensores através dos canais Ch7, Ch8 e Ch9. Fechando o ciclo, os sensores enviam ψ , θ e ϕ ao controlador através dos canais Ch10, Ch11, Ch12, como mostra a Figura 6.49:

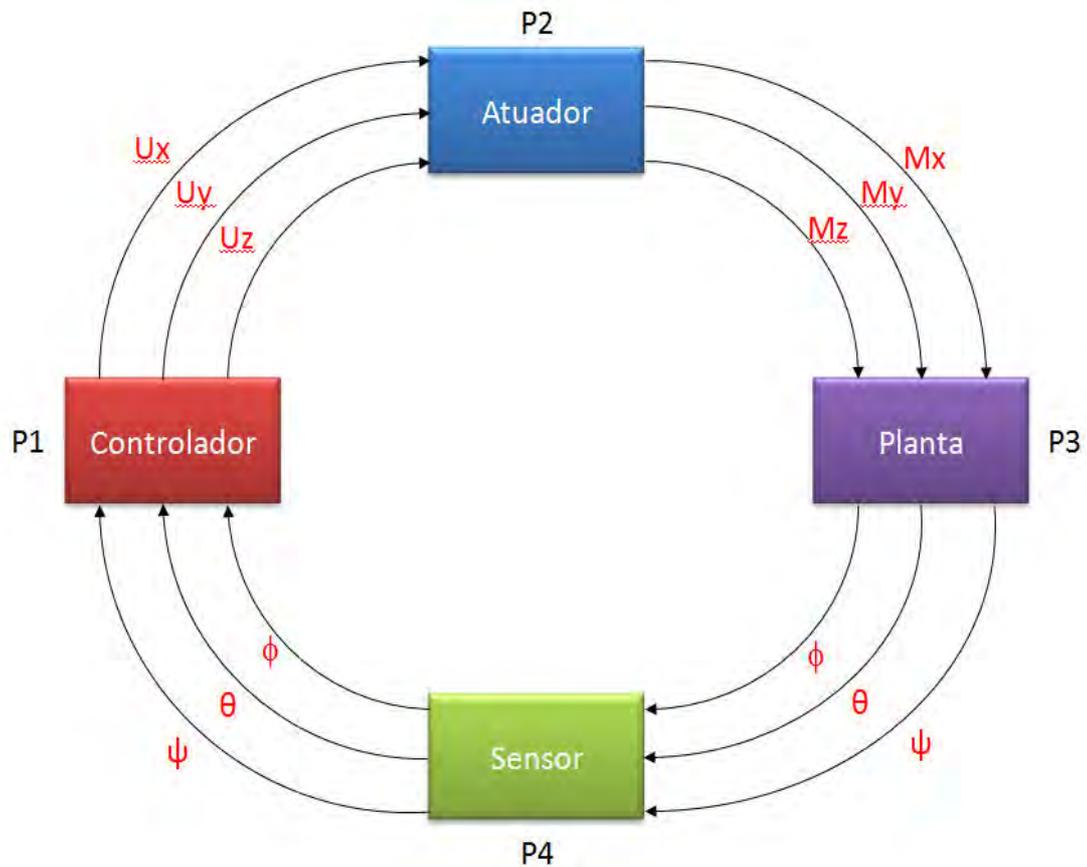


Figura 6.49 - Representação da comunicação entre as partições

Da mesma forma, a partição Atuador/Planta/Sensor envia, através do canal Ch13 o valor da saída, Y ; o valor é comparado com o valor de referência e o controlador envia um sinal de controle, U , para o atuador, através do canal 14, como mostrado na Figura 6.50:

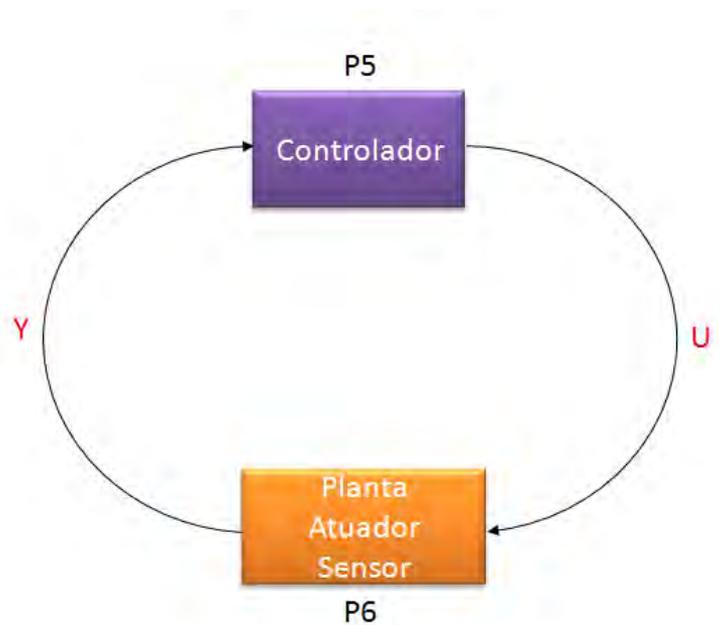


Figura 6.50 - Representação da comunicação entre as partições

6.5.2.1. Prioridade

Dentro de cada partição há diversos processos sendo executados. Para que haja uma sequência lógica, algumas prioridades foram definidas. Deste modo, em cada partição os processos de recebimento e leitura de dados tinham a maior prioridade, seguidos por cálculos executados a partir desses valores e, em seguida, os processos de envio de resultados. Além disso, foram definidas prioridades internas para os processos de leitura, cálculo e envio de dados. Portanto, no SCA, as leituras no eixo x tinham maior prioridade que as leituras no eixo y que, por sua vez, tinham maior prioridade que as leituras no eixo z. As prioridades para os cálculos e envio de dados foram definidas da mesma maneira, ou seja, primeiro os cálculos foram executados para, em seguida, serem enviados.

Como o controlador do SCA foi a primeira partição a ser executada, não havia resultado a ser lido, portanto, esta partição se inicia enviando os dados iniciais. Já as demais partições se iniciam lendo os resultados das partições anteriores

e, executando seus processos de cálculos e, em seguida, enviando os resultados.

6.5.2.2. Configuração

No SIMA, os arquivos que devem ser alterados para a configuração das portas e canais são: a653.xml e sima.xml. Nas Figuras 6.51 a 6.62 são ilustradas as configurações utilizadas nas simulações:

6.5.2.2.1. Partição 1

6.5.2.2.1.1. a653.xml

```
<Partition PartitionIdentifier="1"
  PartitionName="controlador"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do controlador-->
  <Queuing_Port Name="S1"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <Queuing_Port Name="S2"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <Queuing_Port Name="S3"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <!-- entradas do controlador-->
  <Queuing_Port Name="E10"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

  <Queuing_Port Name="E11"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

  <Queuing_Port Name="E12"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

</Partition>
```

Figura 6.51 – Configuração no arquivo a653.xml da Partição 1

6.5.2.2.1.2. sima.xml

```
<Partition PartitionIdentifier="1"
  PartitionName="controlador"
  SharedMemory="70010"
  Startup="/bin/sh ./samples/com/partitions/controlador/controlador.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/controlador/fifo_controlador"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>

  <Destination_Port
    Name="E10"
    Type="UDP"
    IP="127.0.0.1"
    Port="12311"/>

  <Destination_Port
    Name="E11"
    Type="UDP"
    IP="127.0.0.1"
    Port="12321"/>

  <Destination_Port
    Name="E12"
    Type="UDP"
    IP="127.0.0.1"
    Port="12331"/>
</Partition>
```

Figura 6.52 - Configuração no arquivo sima.xml da Partição 1

6.5.2.2. Partição 2

6.5.2.2.1. a653.xml

```
<Partition PartitionIdentifier="2"
  PartitionName="atuador"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do atuador-->
  <Queuing_Port Name="S4"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S5"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S6"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <!-- entradas do atuador-->
  <Queuing_Port Name="E1"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E2"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E3"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

</Partition>
```

Figura 6.53 - Configuração no arquivo a653.xml da Partição 2

6.5.2.2.2. sima.xml

```
<Partition PartitionIdentifier="2"
  PartitionName="atuador"
  SharedMemory="70020"
  Startup="/bin/sh ./samples/com/partitions/atuador/atuador.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/atuador/fifo_atuador"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>
  <Destination_Port
    Name="E1"
    Type="UDP"
    IP="127.0.0.1"
    Port="12341"/>

  <Destination_Port
    Name="E2"
    Type="UDP"
    IP="127.0.0.1"
    Port="12351"/>

  <Destination_Port
    Name="E3"
    Type="UDP"
    IP="127.0.0.1"
    Port="12361"/>
</Partition>
```

Figura 6.54 - Configuração no arquivo sima.xml da Partição 2

6.5.2.2.3. Partição 3

6.5.2.2.3.1. a653.xml

```
<Partition PartitionIdentifier="3"
    PartitionName="planta"
    Criticality="LEVEL_A"
    SystemPartition="true"
    EntryPoint="Initial">

    <!-- saídas do planta-->
    <Queuing_Port Name="S7"
        MaxMessageSize="1024"
        Direction="SOURCE"
        MaxNbMessages="32"/>
    <Queuing_Port Name="S8"
        MaxMessageSize="1024"
        Direction="SOURCE"
        MaxNbMessages="32"/>
    <Queuing_Port Name="S9"
        MaxMessageSize="1024"
        Direction="SOURCE"
        MaxNbMessages="32"/>

    <!-- entradas do planta-->
    <Queuing_Port Name="E4"
        MaxMessageSize="1024"
        Direction="DESTINATION"
        MaxNbMessages="32"/>
    <Queuing_Port Name="E5"
        MaxMessageSize="1024"
        Direction="DESTINATION"
        MaxNbMessages="32"/>
    <Queuing_Port Name="E6"
        MaxMessageSize="1024"
        Direction="DESTINATION"
        MaxNbMessages="32"/>

</Partition>
```

Figura 6.55 - Configuração no arquivo a653.xml da Partição 3

6.5.2.2.3.2. sima.xml

```
<Partition PartitionIdentifier="3"
  PartitionName="planta"
  SharedMemory="70030"
  Startup="/bin/sh ./samples/com/partitions/planta/planta.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/planta/fifo_planta"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>
  <Destination_Port
    Name="E4"
    Type="UDP"
    IP="127.0.0.1"
    Port="12401"/>

  <Destination_Port
    Name="E5"
    Type="UDP"
    IP="127.0.0.1"
    Port="12411"/>

  <Destination_Port
    Name="E6"
    Type="UDP"
    IP="127.0.0.1"
    Port="12421"/>
</Partition>
```

Figura 6.56 - Configuração no arquivo sima.xml da Partição 3

6.5.2.2.4. Partição 4

6.5.2.2.4.1. a653.xml

```
<Partition PartitionIdentifier="4"
  PartitionName="sensor"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do sensor-->
  <Queuing_Port Name="S10"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S11"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>
  <Queuing_Port Name="S12"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <!-- entradas do sensor-->
  <Queuing_Port Name="E7"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E8"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>
  <Queuing_Port Name="E9"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

</Partition>
```

Figura 6.57 - Configuração no arquivo a653.xml da Partição 4

6.5.2.2.4.2. sima.xml

```
<Partition PartitionIdentifier="4"
  PartitionName="sensor"
  SharedMemory="70040"
  Startup="/bin/sh ./samples/com/partitions/sensor/sensor.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/sensor/fifo_sensor"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>
  <Destination_Port
    Name="E7"
    Type="UDP"
    IP="127.0.0.1"
    Port="12371"/>

  <Destination_Port
    Name="E8"
    Type="UDP"
    IP="127.0.0.1"
    Port="12381"/>

  <Destination_Port
    Name="E9"
    Type="UDP"
    IP="127.0.0.1"
    Port="12391"/>
</Partition>
```

Figura 6.58 - Configuração no arquivo sima.xml da Partição 4

6.5.2.2.5. Partição 5

6.5.2.2.5.1. a653.xml

```
<Partition PartitionIdentifier="5"
  PartitionName="extra"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">
  <!-- saídas do MM-->
  <Queuing_Port Name="S14"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32"/>

  <!-- entradas do MM-->
  <Queuing_Port Name="E13"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32"/>

</Partition>
```

Figura 6.59 - Configuração no arquivo a653.xml da Partição 5

6.5.2.2.5.2. sima.xml

```
<Partition PartitionIdentifier="5"
  PartitionName="extra"
  SharedMemory="70050"
  Startup="/bin/sh ./samples/com/partitions/extra/extra.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/extra/fifo_extra"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>

  <Destination_Port
    Name="E13"
    Type="UDP"
    IP="127.0.0.1"
    Port="12431"/>

</Partition>
```

Figura 6.60 - Configuração no arquivo sima.xml da Partição 5

6.5.2.2.6. Partição 6

6.5.2.2.6.1. a653.xml

```
<Partition PartitionIdentifier="6"
  PartitionName="extra1"
  Criticality="LEVEL_A"
  SystemPartition="true"
  EntryPoint="Initial">

  <!-- saídas do MM-->
  <Queuing_Port Name="S13"
    MaxMessageSize="1024"
    Direction="SOURCE"
    MaxNbMessages="32" />

  <!-- entradas do MM-->
  <Queuing_Port Name="E14"
    MaxMessageSize="1024"
    Direction="DESTINATION"
    MaxNbMessages="32" />

</Partition>
```

Figura 6.61 - Configuração no arquivo a653.xml da Partição 6

6.5.2.2.6.2. sima.xml

```
<Partition PartitionIdentifier="6"
  PartitionName="extral"
  SharedMemory="70060"
  Startup="/bin/sh ./samples/com/partitions/extral/extral.sh"
  visible="true">
  <Output
    type="fifo"
    path="./samples/com/partitions/extral/fifo_extral"/>
  <Transport Start="all"
    End="all"
    StartDurationSeconds="0.01"
    EndDurationSeconds="0.01"/>

  <Destination_Port
    Name="E14"
    Type="UDP"
    IP="127.0.0.1"
    Port="12441"/>

</Partition>
```

Figura 6.62 - Configuração no arquivo sima.xml da Partição 6

6.5.3. Agendando (*Scheduling*) das partições

O agendamento das partições foi feito da seguinte forma: a janela de execução da partição 1 se inicia em 0,0 segundos e tem um tempo de 0,1 segundos de execução. Finalizando este intervalo de tempo, os processos desta partição sofrem preempção e a janela de execução da partição 2 ganha acesso ao processador e, como a anterior, executa seus processos de 0,1 segundos. Terminado esse tempo, a terceira partição ganha acesso ao processador, e assim sucessivamente. Quando a janela de execução da primeira partição for iniciada novamente, os processos que haviam sido interrompidos são retomados do ponto que estavam. A Figura 6.63 ilustra o agendamento das partições.

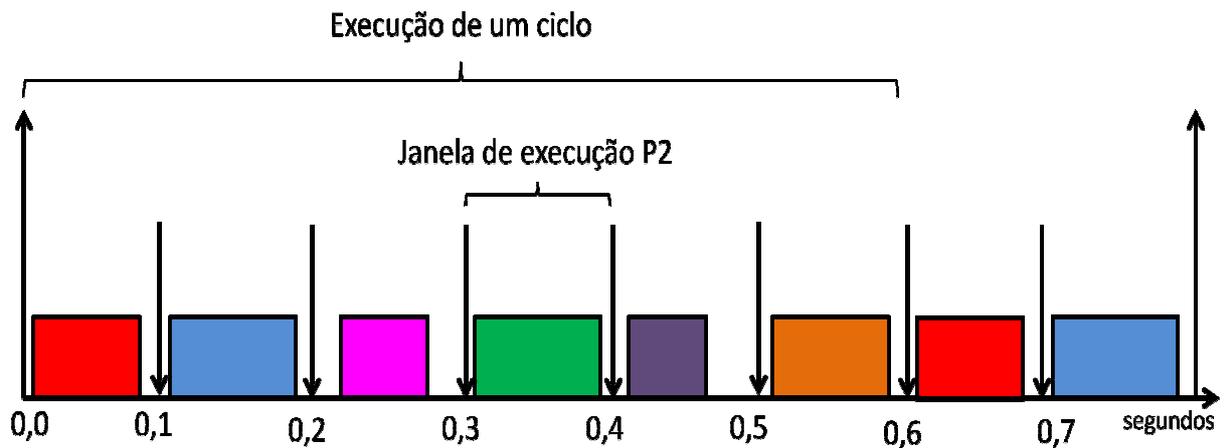


Figura 6.63 - Agendamento no tempo das partições do SCA e MM

As configurações de agendamento utilizadas nas simulações, também efetuadas no arquivo de configuração do SIMA (a653.xml) são ilustradas na Figura 6.64:

6.5.3.1. a653.xml

```

<Module_Schedule MajorFrameSeconds="0.6"> <!-- tempo total de todas as janelas-->
  <Partition_Schedule PartitionIdentifier="1"
    PartitionName="controlador"
    PeriodSeconds="0.6"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="101"
      WindowStartSeconds="0.0"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="2"
    PartitionName="atuador"
    PeriodSeconds="0.6"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="201"
      WindowStartSeconds="0.1"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="3"
    PartitionName="planta"
    PeriodSeconds="0.6"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="301"
      WindowStartSeconds="0.2"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="4"
    PartitionName="sensor"
    PeriodSeconds="0.6"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="401"
      WindowStartSeconds="0.3"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="5"
    PartitionName="extra"
    PeriodSeconds="0.6"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="501"
      WindowStartSeconds="0.4"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
  <Partition_Schedule PartitionIdentifier="6"
    PartitionName="extral"
    PeriodSeconds="0.6"
    PeriodDurationSeconds="0.1">
    <Window_Schedule WindowIdentifier="601"
      WindowStartSeconds="0.5"
      WindowDurationSeconds="0.1"
      PartitionPeriodStart="true"/>
  </Partition_Schedule>
</Module_Schedule>

```

Figura 6.64 – Configuração de agendamento no arquivo a653.xml (Partições 1 a 6)

7 RESULTADOS

Como dito anteriormente, com intuito de se aprender a utilizar o SIMA, julgou-se conveniente iniciar as simulações por um sistema mais simples e, em seguida, partir para um modelo mais complexo. Deste modo, a simulação do massa-mola-amortecedor tem uma importância secundária nesse trabalho. Apesar disso, também são apresentados os resultados dessa simulação.

7.1. Massa-Mola-Amortecedor

As curvas obtidas nas simulações em C, MatLab e SIMA foram muito próximas. No entanto, notou-se que quanto maior o número de cálculos envolvidos maior a divergência das curvas. Isso ocorre, pois cada programa tem uma maneira própria de tratar cálculos numéricos, arredondamentos, etc. Desta forma, simplificaram-se ao máximos os cálculos envolvidos. Onde possível, simplificaram-se equações acrescentando-se os valores calculados a priori.

O gráfico abaixo ilustra as curvas obtidas em linguagem C, MatLab e SIMA:

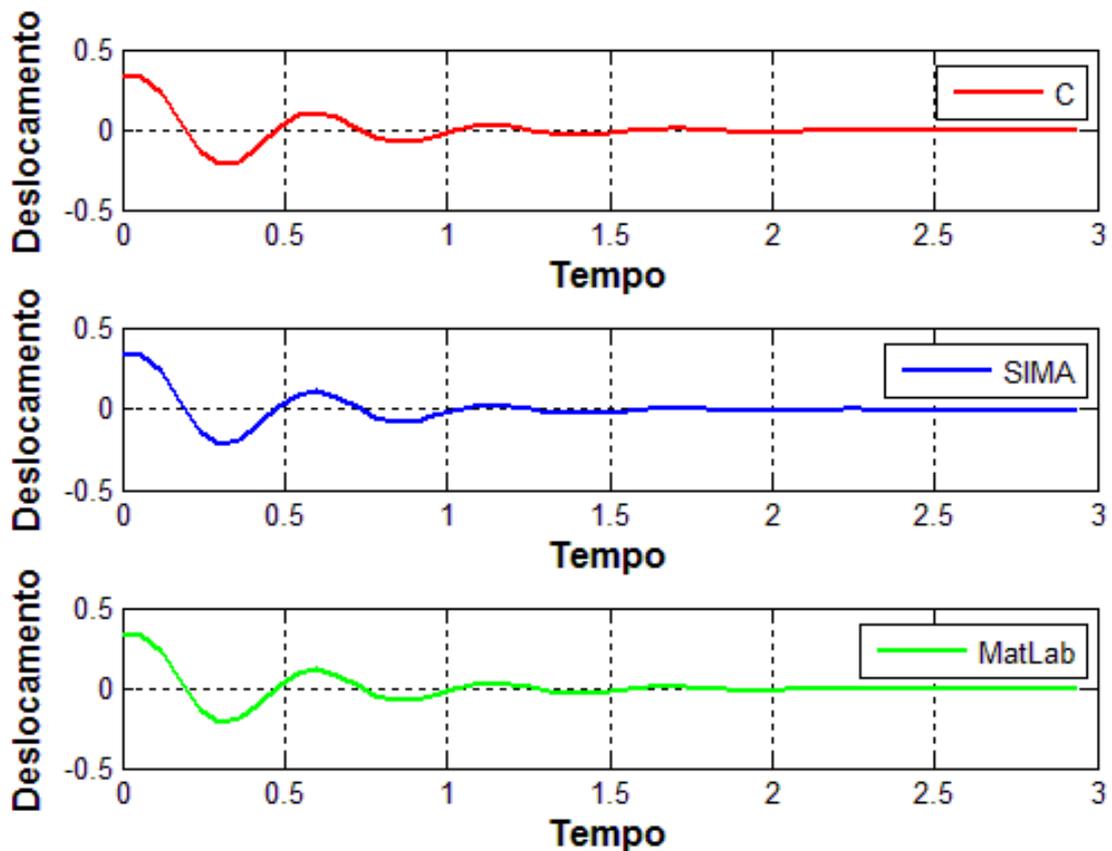


Figura 7.1 – Curvas dos resultados dos programas simulados em linguagem C, MatLab e SIMA

Como as curvas são muito próximas, julgou-se conveniente obter-se as diferenças entre as mesmas. Deste modo, como o foco deste trabalho é a simulação no SIMA, comparou-se a curva obtida em C com a curva obtida no SIMA; e a curva obtida MatLab também com a curva obtida no SIMA.

A seguir são apresentadas, no mesmo gráfico, as curvas obtidas em C e no SIMA. E, em seguida, a diferença entre as mesmas.

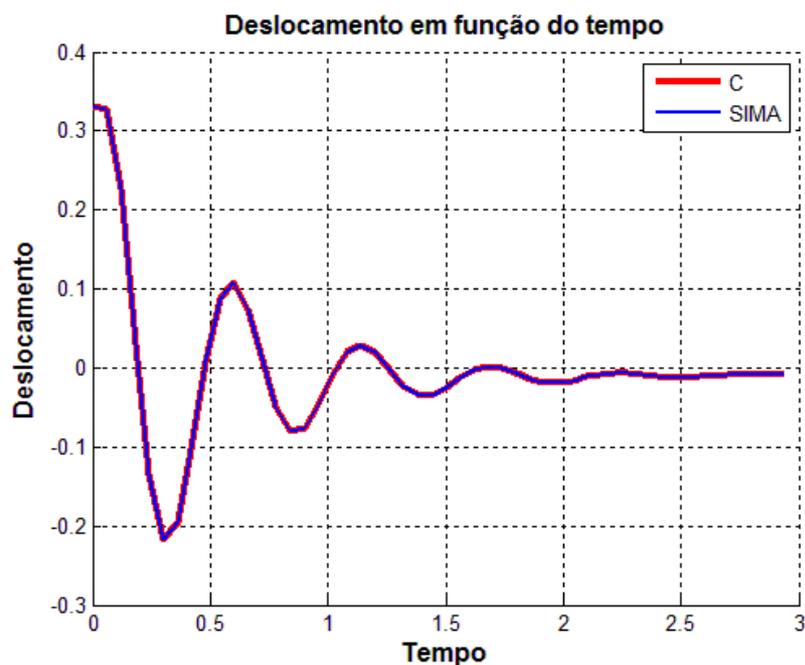


Figura 7.2 – Curvas dos resultados dos programas simulados em linguagem C e SIMA

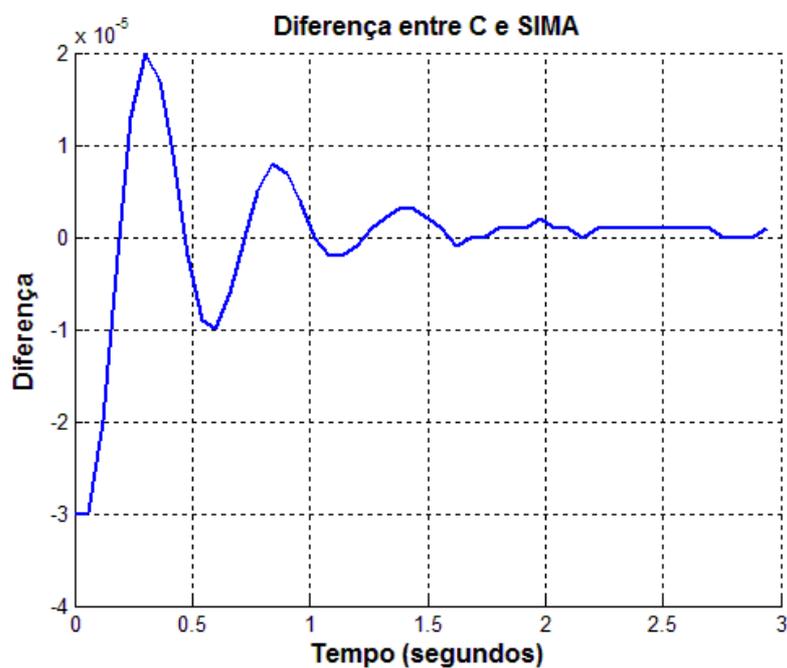


Figura 7.3 – Diferença das curvas dos programas simulados em linguagem C e SIMA

Abaixo são apresentadas, no mesmo gráfico, as curvas obtidas em MatLab e no SIMA. E, em seguida, a diferença entre as mesmas.

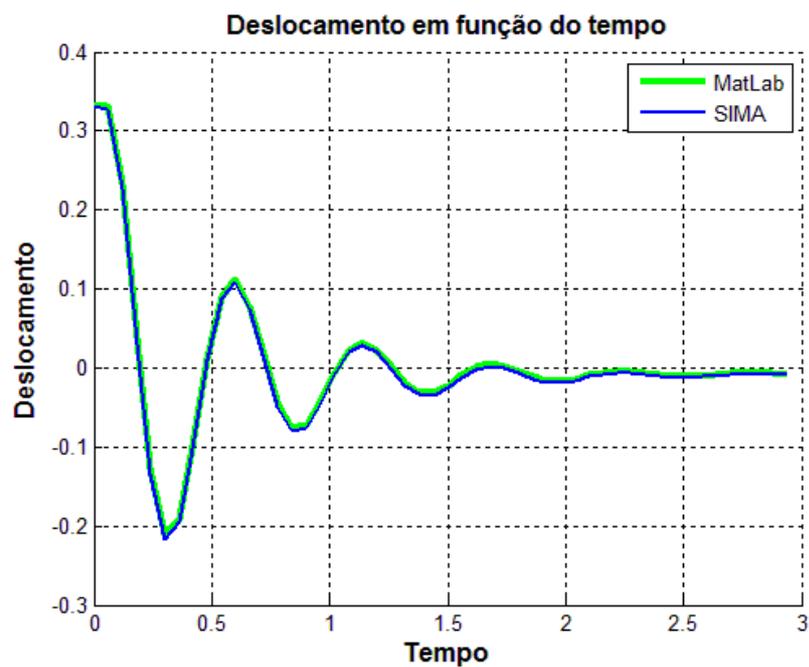


Figura 7.4 - Curvas dos resultados dos programas simulados em MatLab e SIMA

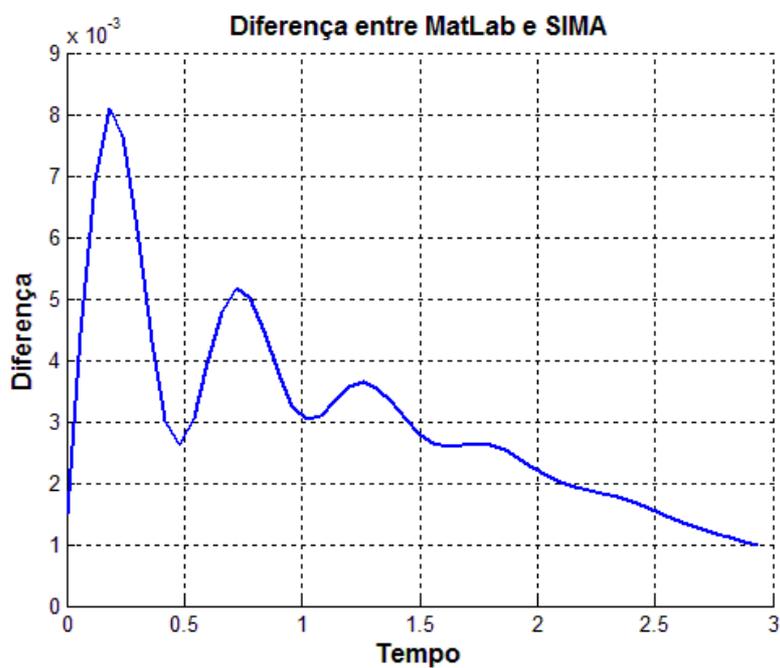


Figura 7.5 - Diferença das curvas dos programas simulados em MatLab e SIMA

7.2. Sistema de Controle de Atitude

7.2.1. Passo de integração

A escolha do incremento do passo de integração foi feita baseando-se na melhor aproximação da curva versus o tempo de processamento das simulações. Foi utilizado o mesmo passo de integração para todos os programas. Abaixo é apresentada a curva do ângulo phi (ϕ) em graus com diversos passos de integração em segundos:

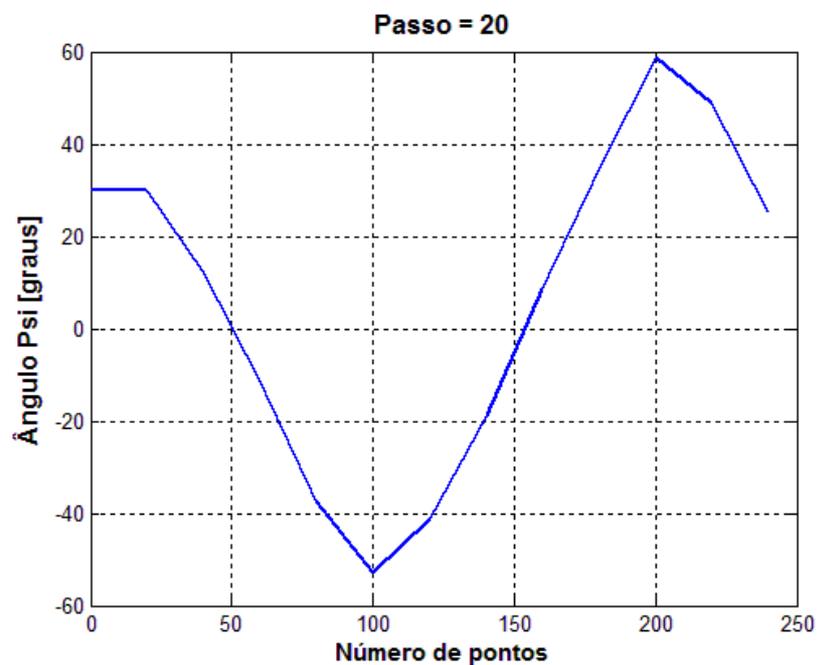


Figura 7.6 – Curva do ângulo psi utilizando passo de integração de 20 segundos

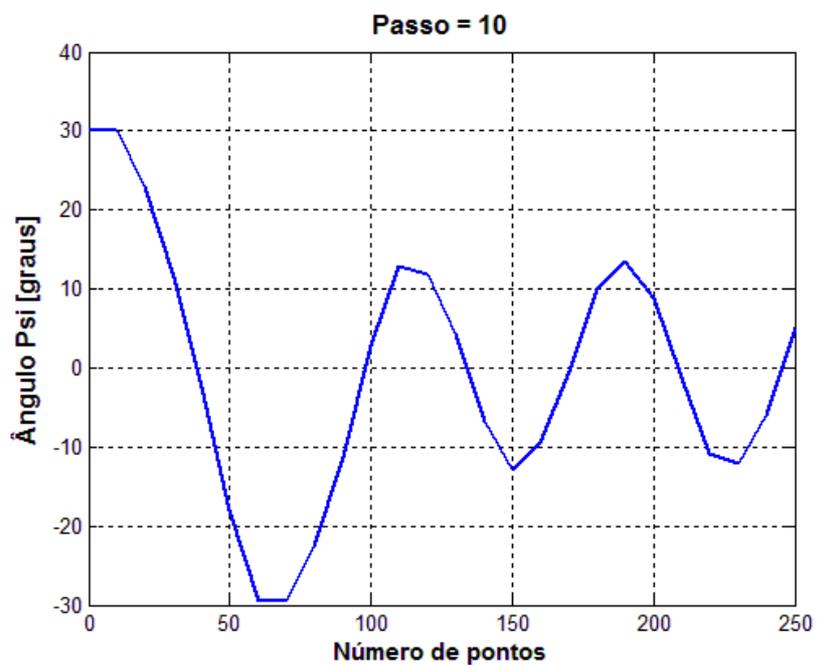


Figura 7.7 - Curva do ângulo psi utilizando passo de integração de 10 segundos

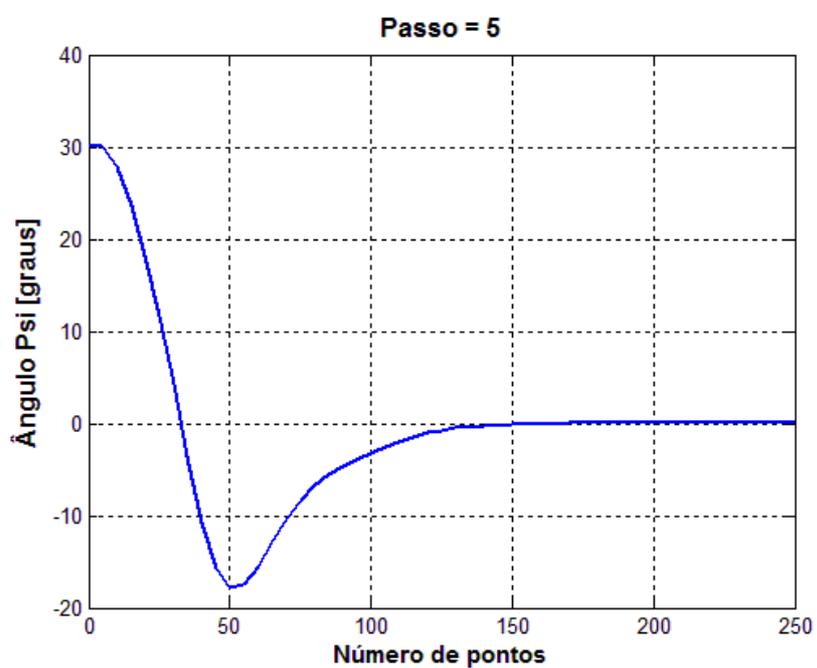


Figura 7.8 - Curva do ângulo psi utilizando passo de integração de 5 segundos

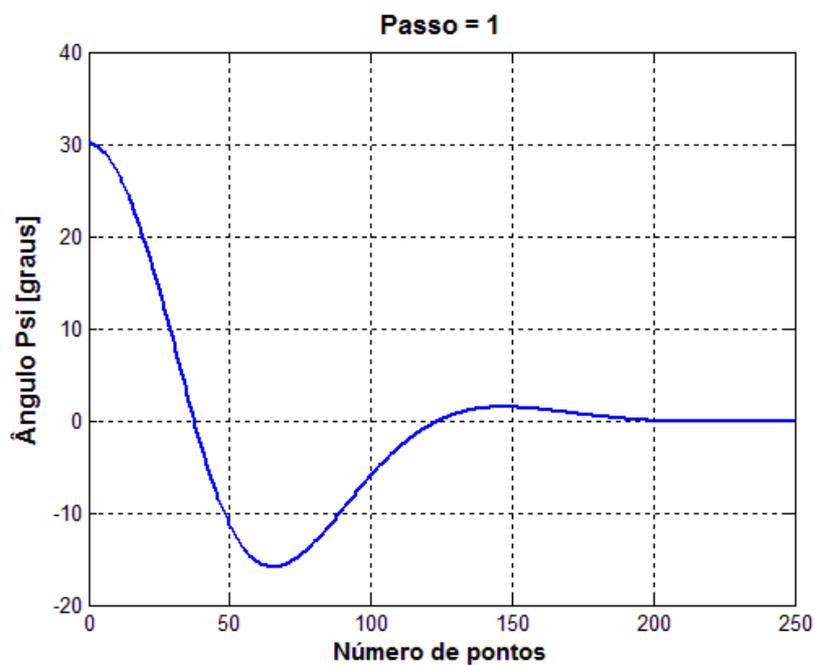


Figura 7.9 - Curva do ângulo psi utilizando passo de integração de 1 segundo

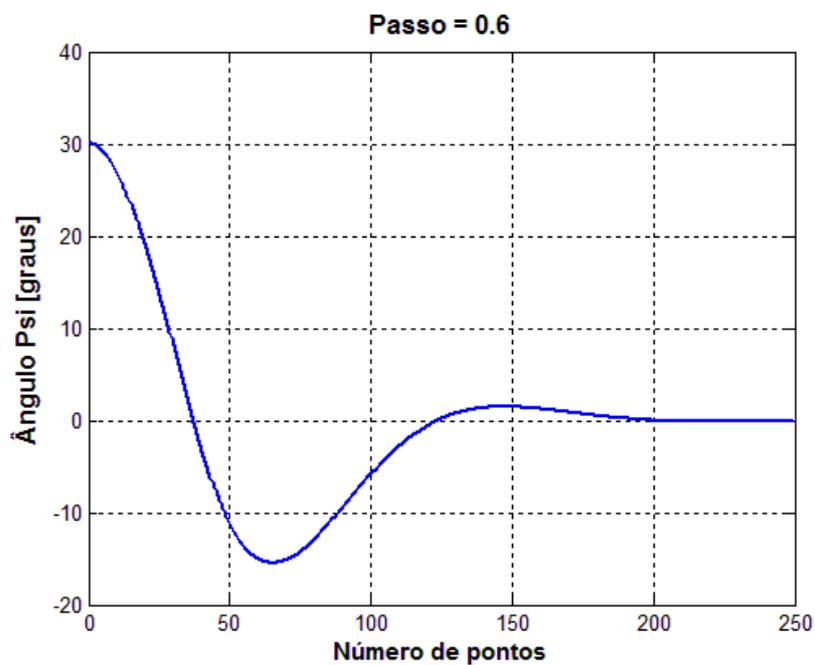


Figura 7.10 - Curva do ângulo psi utilizando passo de integração de 0.6 segundos

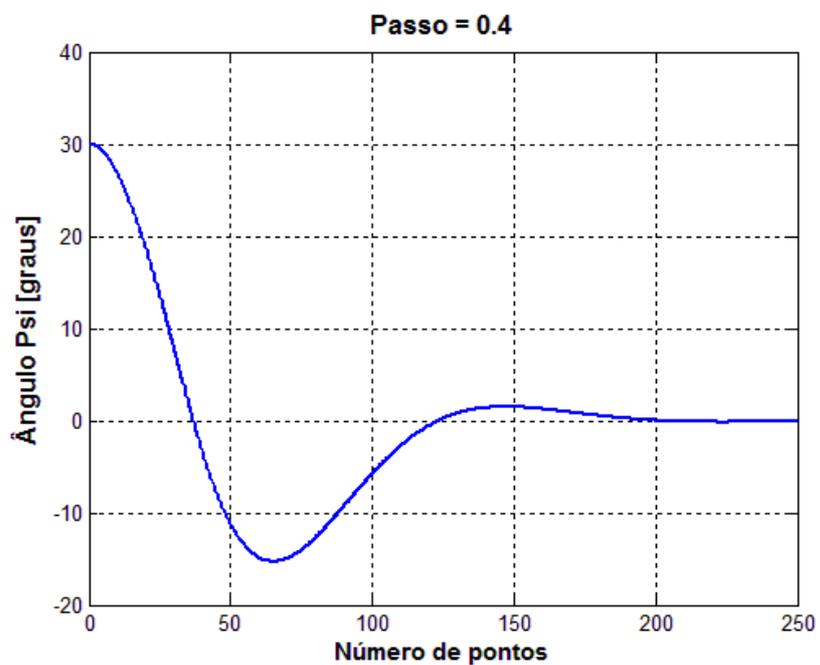


Figura 7.11 - Curva do ângulo psi utilizando passo de integração de 0.4 segundos

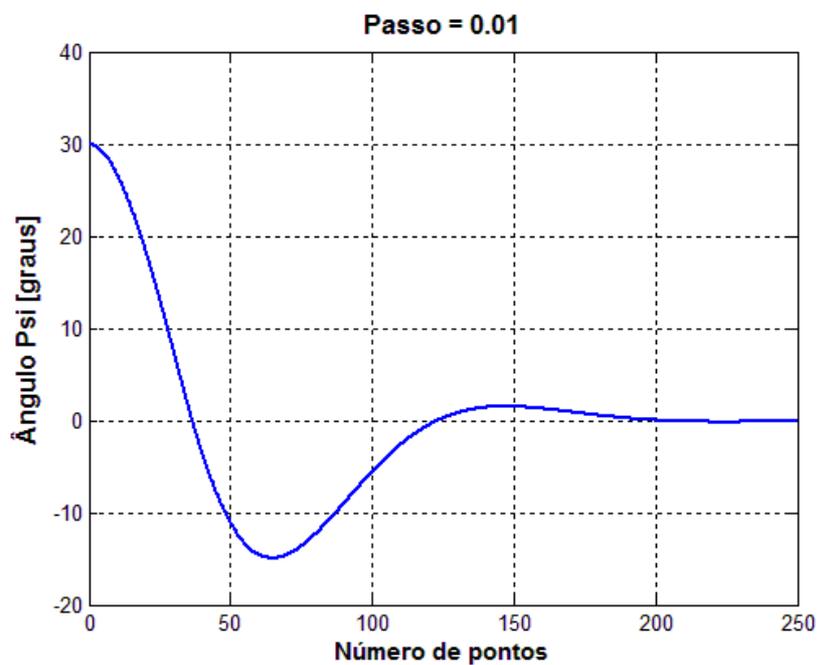


Figura 7.12 - Curva do ângulo psi utilizando passo de integração de 0.01 segundos

Como pode ser visto pelos gráficos apresentados, a partir do passo de integração 1 s a variação da curva não é significativa. Assim, utilizou-se passo de integração 0.4 s para as simulações realizadas no SIMA que utilizaram 4 partições e passo de integração 0.6 s para as simulações que utilizaram 6 partições. Com esta escolha foi possível coincidir o tempo de amostragem das partições com o passo de integração dos Runge-Kutta.

7.2.2. Tempo

De acordo com ARINC (2005), o gerenciamento de tempo é uma característica muito importante dos sistemas operacionais de tempo real. O tempo é único e independente da execução das partições num módulo, e tudo deve ser relacionado com esse tempo único.

Como neste trabalho foi feito utilizando apenas um módulo com quatro partições, foi possível se utilizar como tempo o incremento do passo utilizado nos integradores numéricos (Runge Kutta de 4ª ordem), o que em sistemas mais complexos se tornaria dificultoso e até mesmo inviável.

Com intuito de se validar a abordagem utilizada neste trabalho, são apresentados abaixo os resultados obtidos a partir da simulação do sistema de controle de atitude da PMM no SIMA utilizando o tempo do SIMA (t_s) e o tempo incrementado a partir do passo dos integradores (t). O passo de integração do Runge-Kutta utilizado nas simulações foi definido de acordo com o tempo total das janelas de execução (tempo de amostragem - 1 ciclo). Desta forma, para as simulações com 4 partições utiliza-se passo 0.4 s, pois este é o tempo total de um ciclo com 4 partições. Para as simulações com 6 partições definiu-se um passo de integração de 0.6 s.

O gráfico abaixo ilustra os tempos t e t_s :

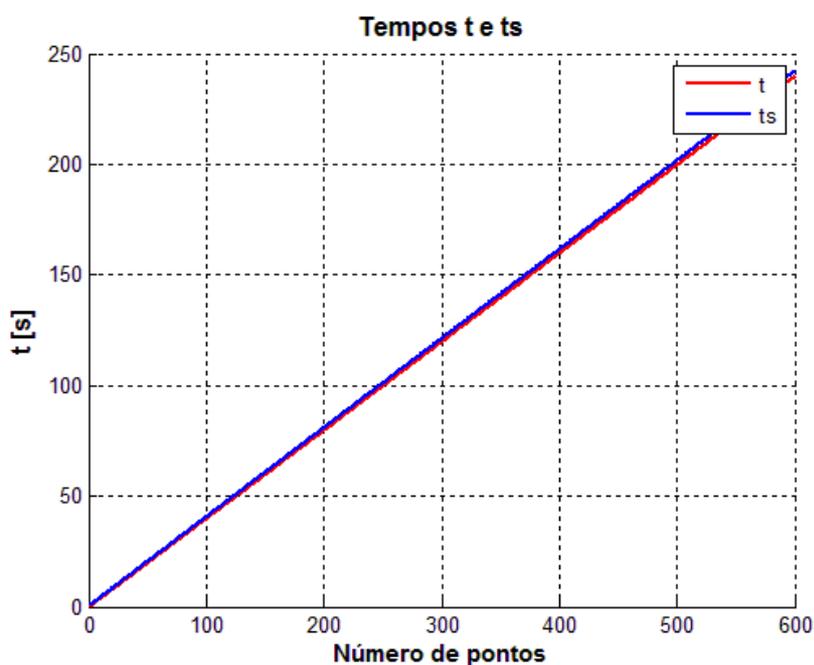


Figura 7.13 – Tempo do SIMA (t_s) e tempo do incremento do passo de integração (t)

A figura 7.13 ilustra as curvas do tempo em relação ao número de pontos plotados do tempo do SIMA (t_s) e do tempo de incremento do passo de integração. É possível notar que as curvas são muito próximas, portanto, para o nosso problema é possível utilizar qualquer uma das duas.

7.2.2.1. Simulação do Modo Nominal do SCA da PMM utilizando t e t_s

Abaixo são apresentados os gráficos com as simulações do sistema de controle de atitude da PMM utilizando t e t_s . São apresentados os ângulos de atitude em cada eixo, a velocidade angular da PMM, a velocidade angular das rodas, a tensão enviada a cada roda de reação e torque fornecido por roda.

7.2.2.1.1. Ângulos de atitude

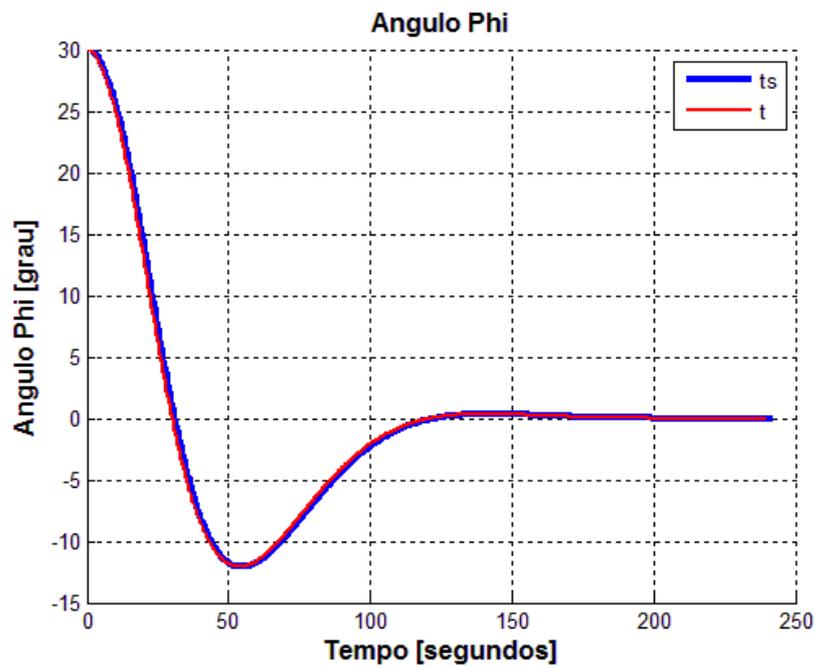


Figura 7.14 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

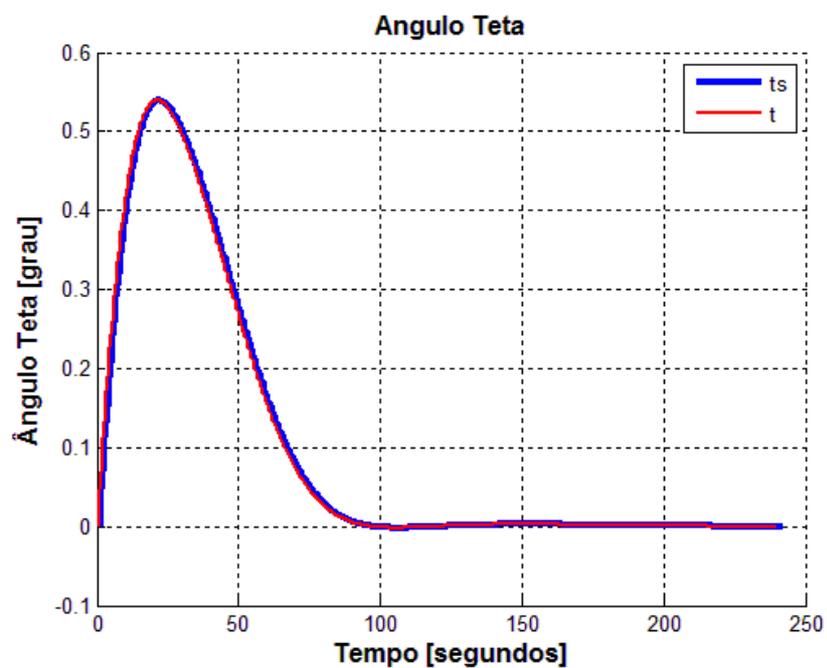


Figura 7.15 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

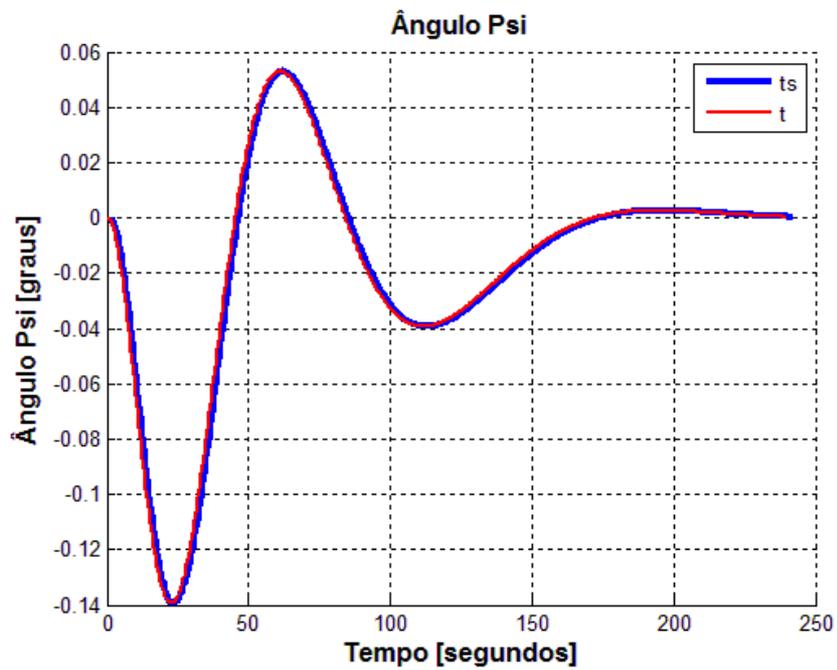


Figura 7.16 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

7.2.2.1.2. Velocidade angular da PMM

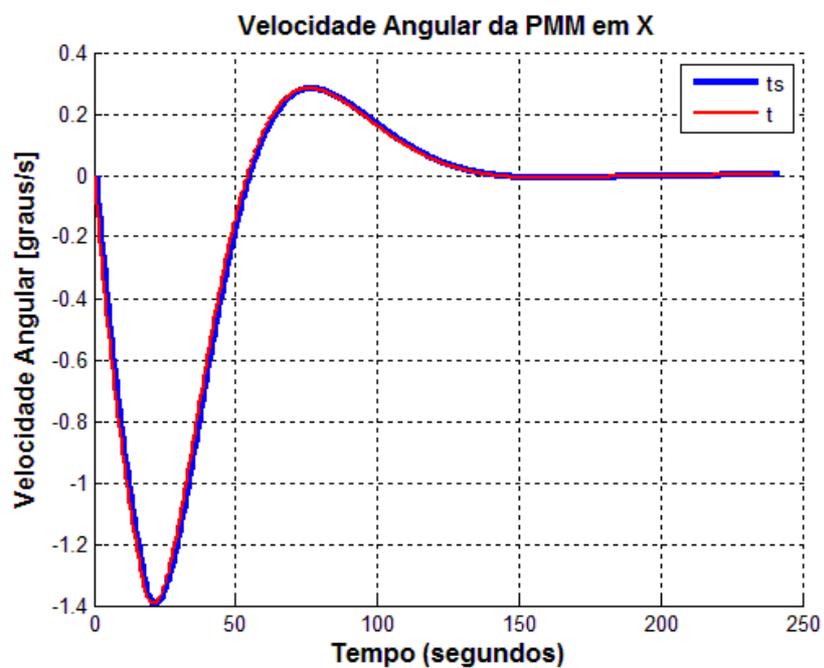


Figura 7.17 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

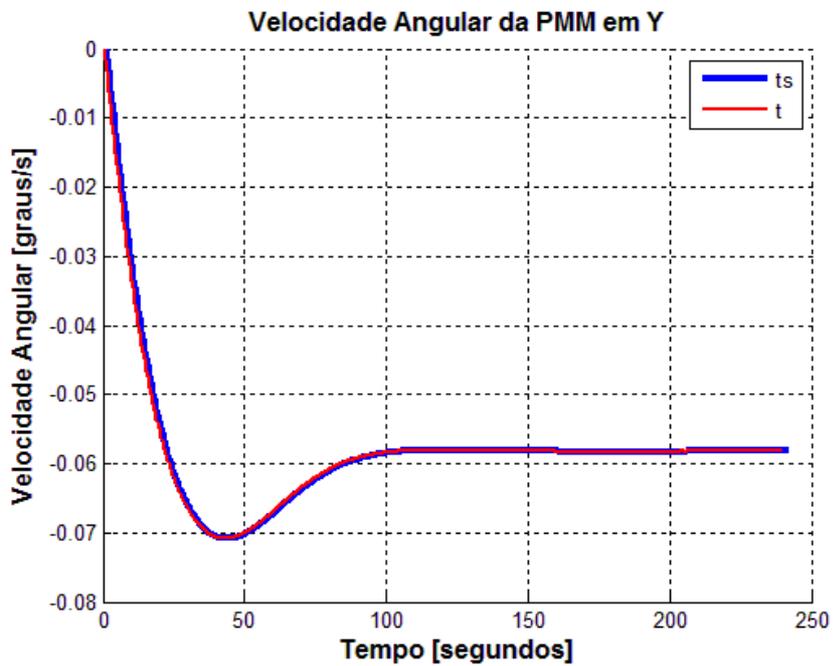


Figura 7.18 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

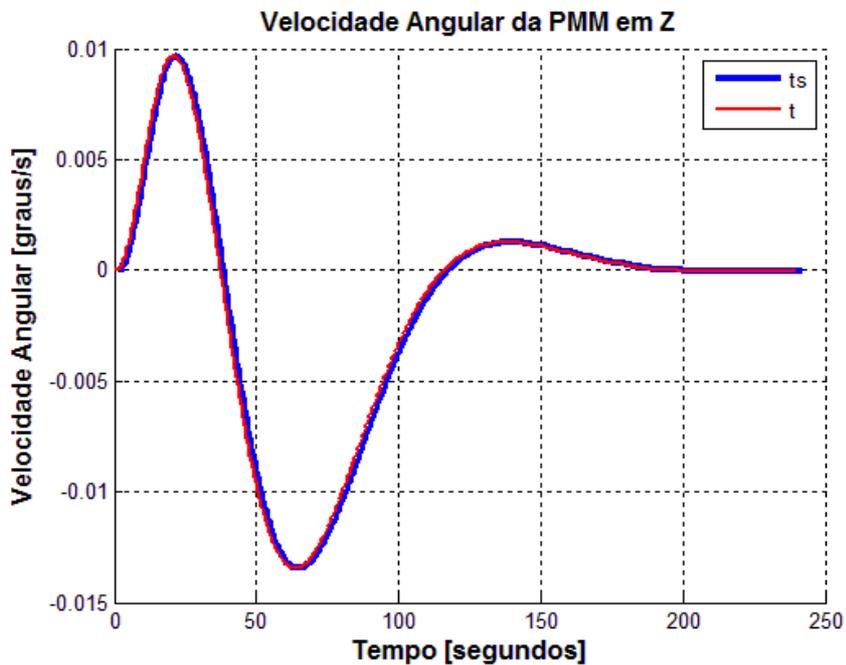


Figura 7.19 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

7.2.2.1.3. Velocidade angular da Roda

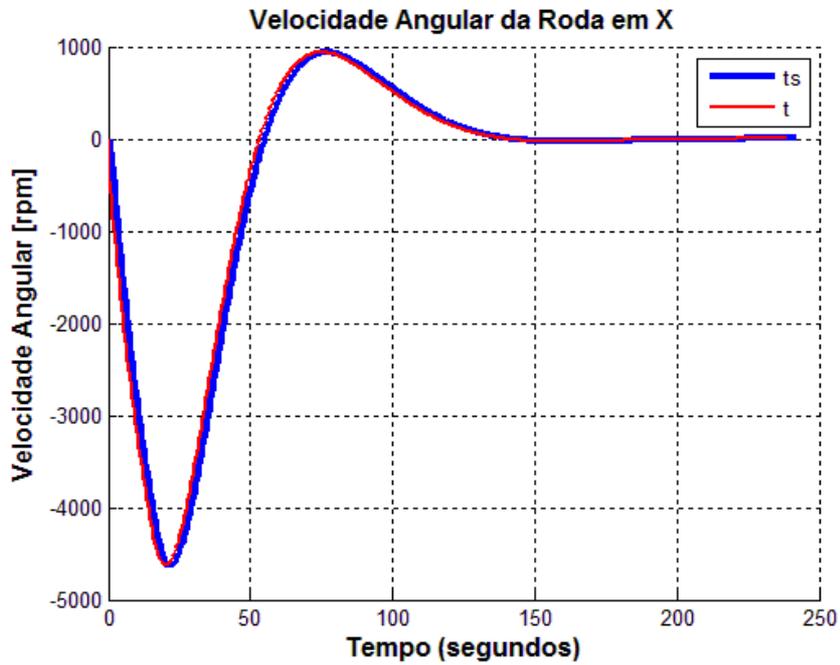


Figura 7.20 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

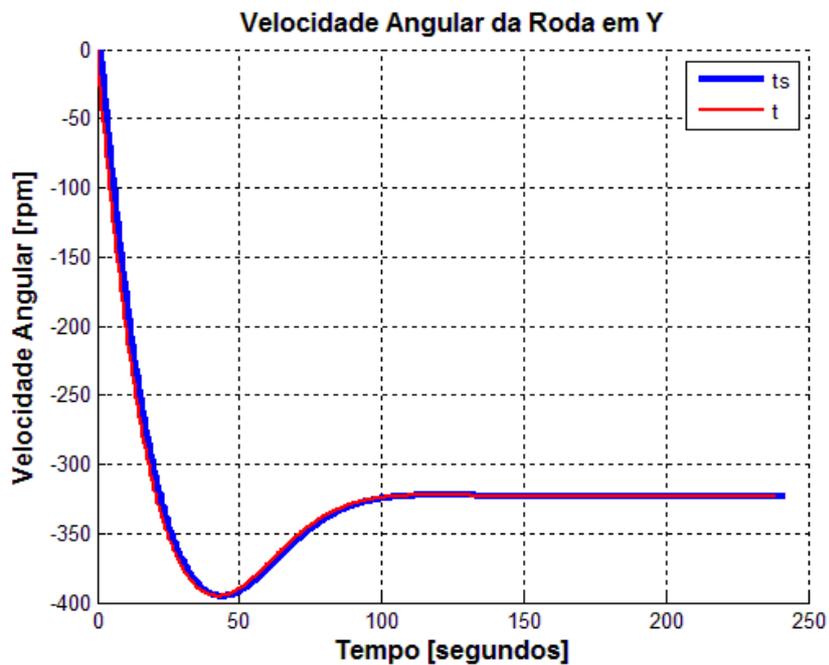


Figura 7.21 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

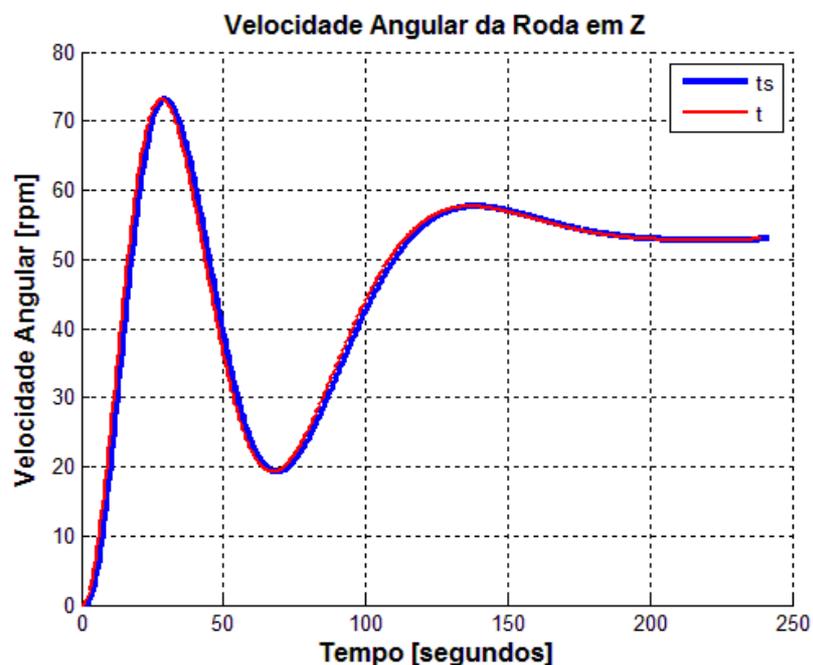


Figura 7.22 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

7.2.2.1.4. Tensão

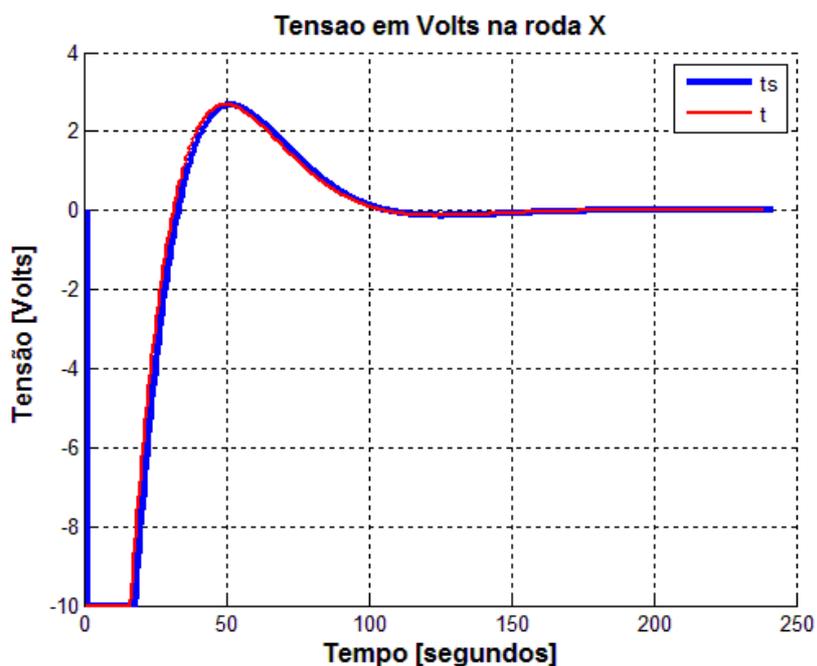


Figura 7.23 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

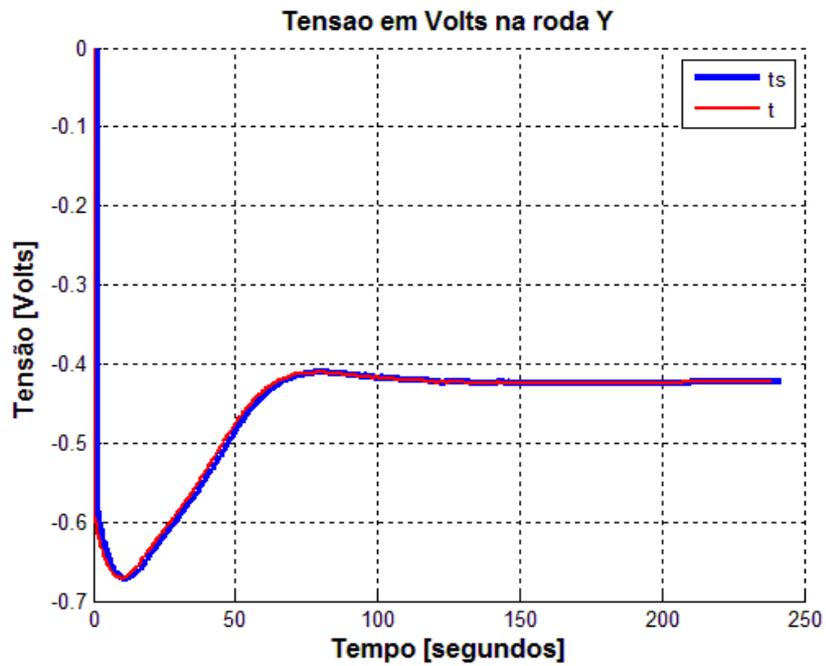


Figura 7.24 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

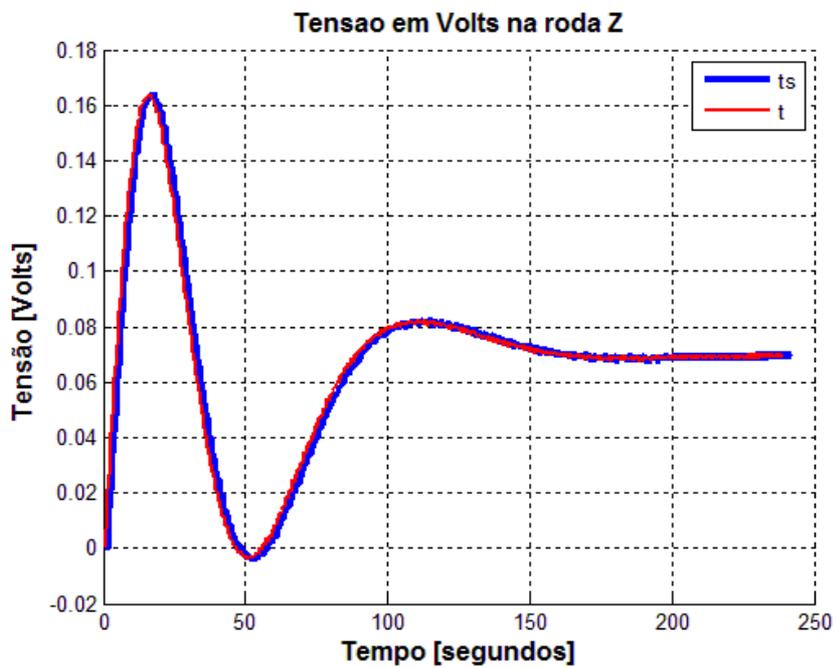


Figura 7.25 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

7.2.2.1.5. Torque

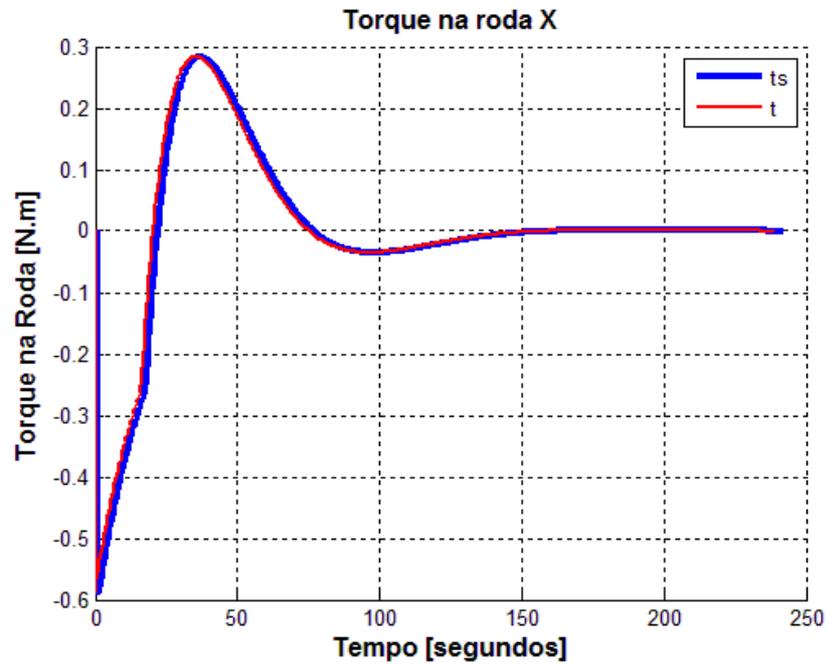


Figura 7.26 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

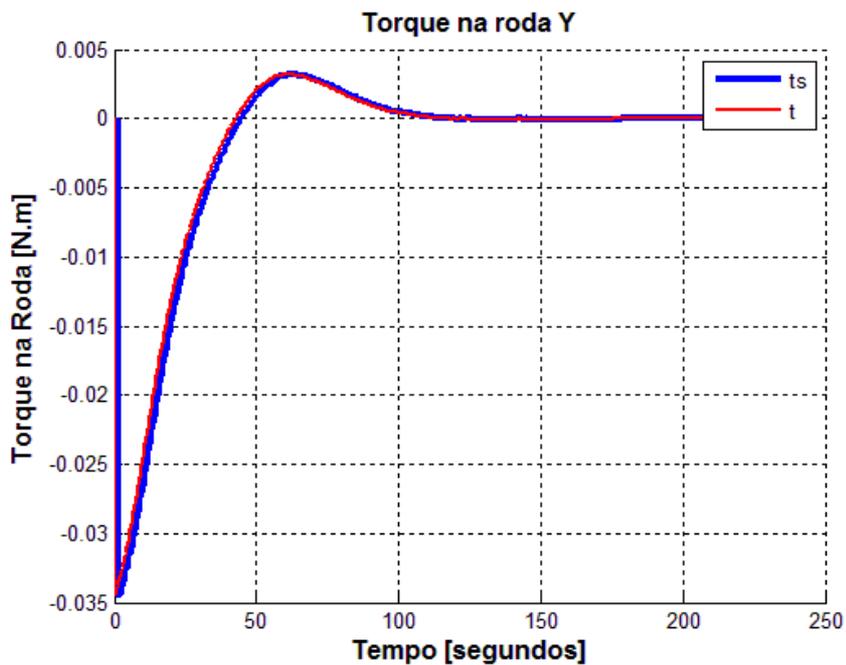


Figura 7.27 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

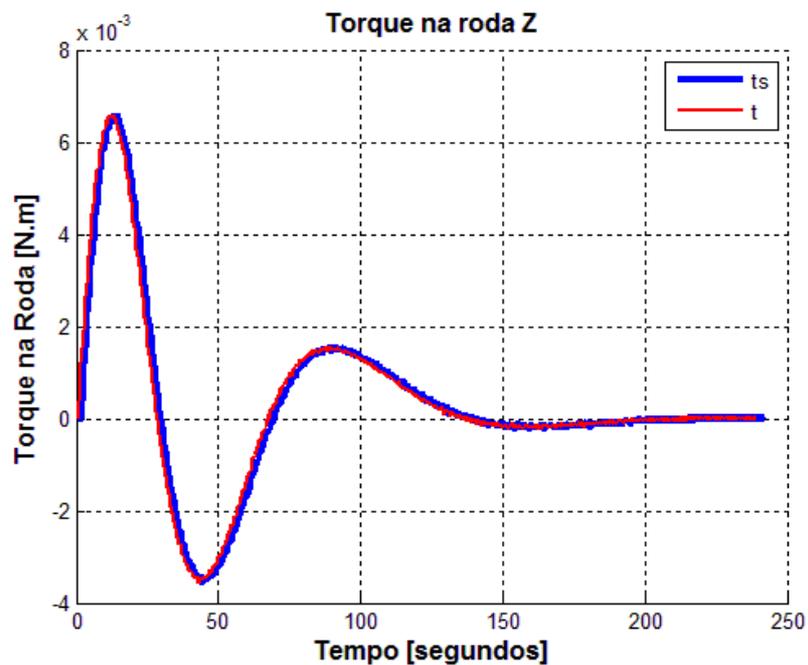


Figura 7.28 - Curvas dos resultados do programa simulado utilizando o tempo do SIMA (ts) e o tempo de incremento do passo de integração (t)

Os gráficos demonstram que a abordagem utilizada nesse trabalho é válida, pois a diferença na utilização do tempo do SIMA (ts) ou do tempo incrementado a partir do passo dos integradores (t) é desprezível para o objetivo deste trabalho.

7.2.3. Casos

Neste trabalho foram efetuadas diversas simulações do sistema de controle de atitude da PMM. Os programas utilizados para este fim foram MatLab, linguagem C e SIMA.

Para um melhor entendimento, as configurações de simulação são identificadas na Tabela 7:

Tabela 6.7 – Casos das simulações do Modo Nominal do SCA da PMM

Casos	Configurações
1	Simulação do Modo Nominal do SCA da PMM em linguagem C e em MatLab
2	Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e MatLab
3	Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e linguagem C
4	Simulação do Modo Nominal do SCA da PMM e do MMA em SIMA divididos em 6 partições
5	Simulação do Modo Nominal do SCA da PMM e do MMA em SIMA divididos em 6 partições com falha inserida

Para uma melhor avaliação, as configurações foram comparadas entre si.

7.2.3.1. Caso 1: Simulação do Modo Nominal do SCA da PMM em linguagem C e em MatLab

A seguir são apresentados os gráficos obtidos a partir da simulação do sistema de controle de atitude da PMM no MatLab e em linguagem C. Os gráficos são ordenados da seguinte maneira, ângulos de atitude em cada eixo, velocidade angular da PMM, velocidade angular das rodas, tensão enviada a cada roda de reação e torque fornecido por roda.

7.2.3.1.1. Ângulos de atitude

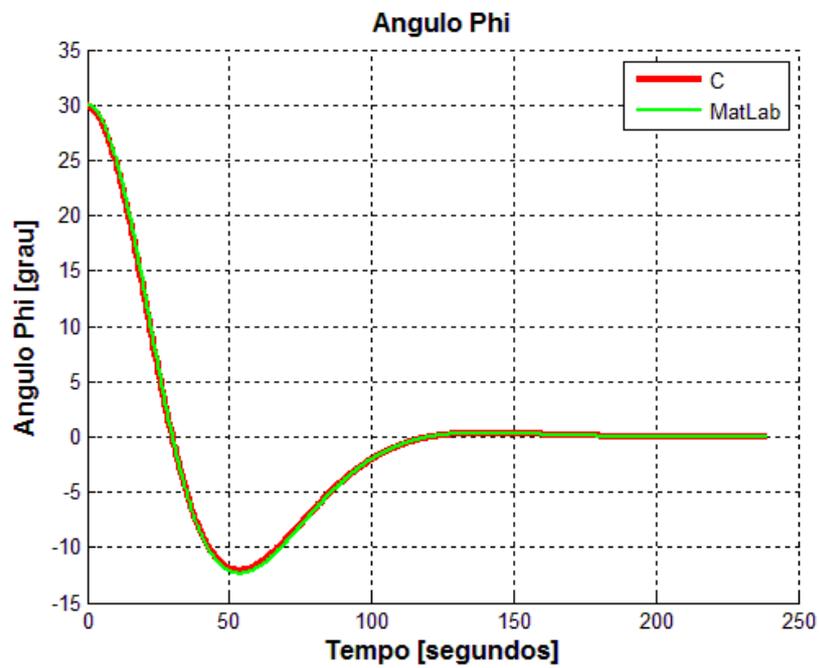


Figura 7.29 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

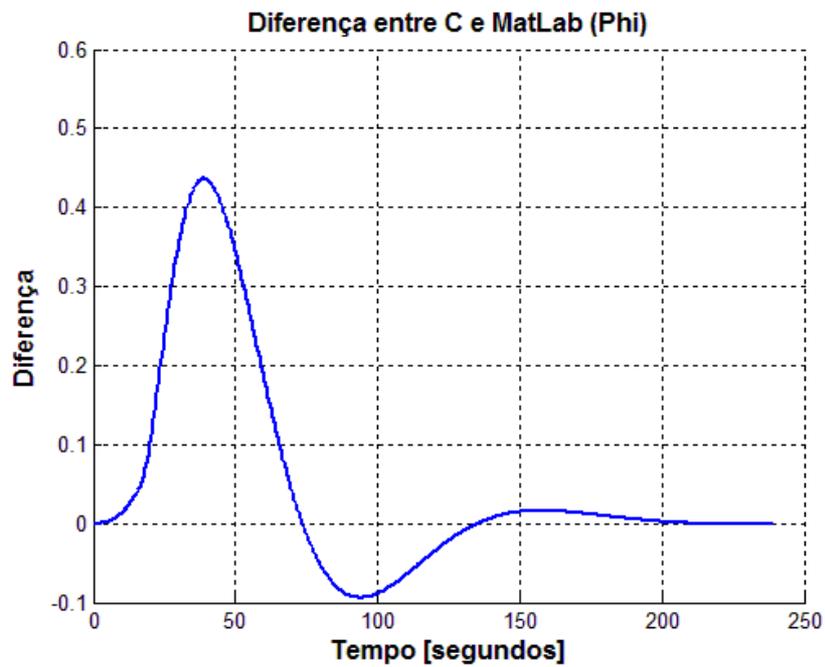


Figura 7.30 - Diferença das curvas dos programas simulados em linguagem C e MatLab

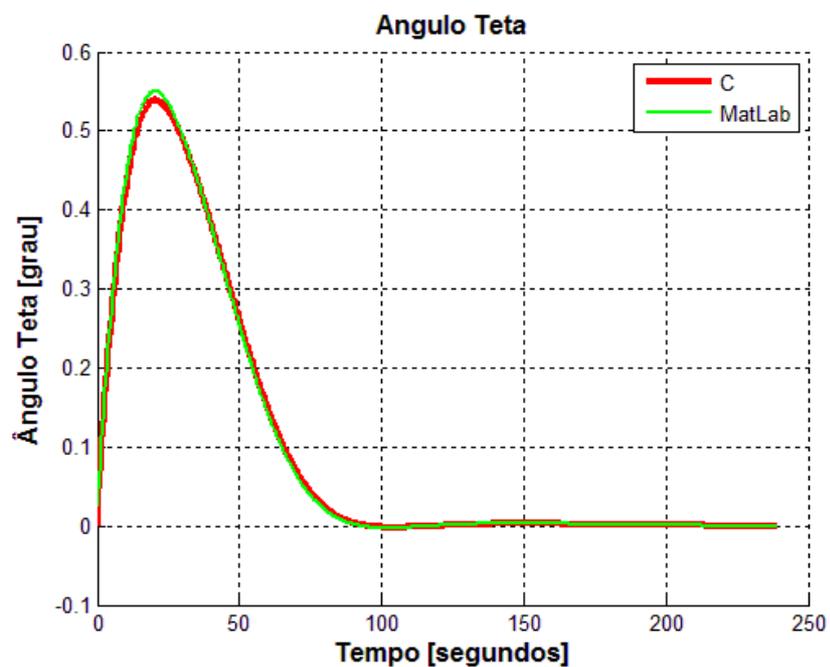


Figura 7.31 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

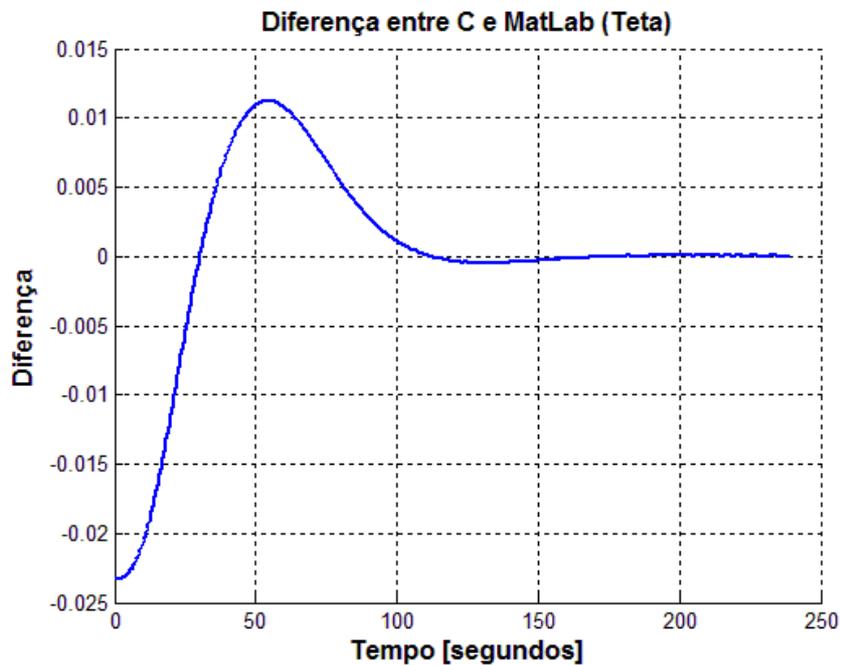


Figura 7.32 - Diferença das curvas dos programas simulados em linguagem C e MatLab

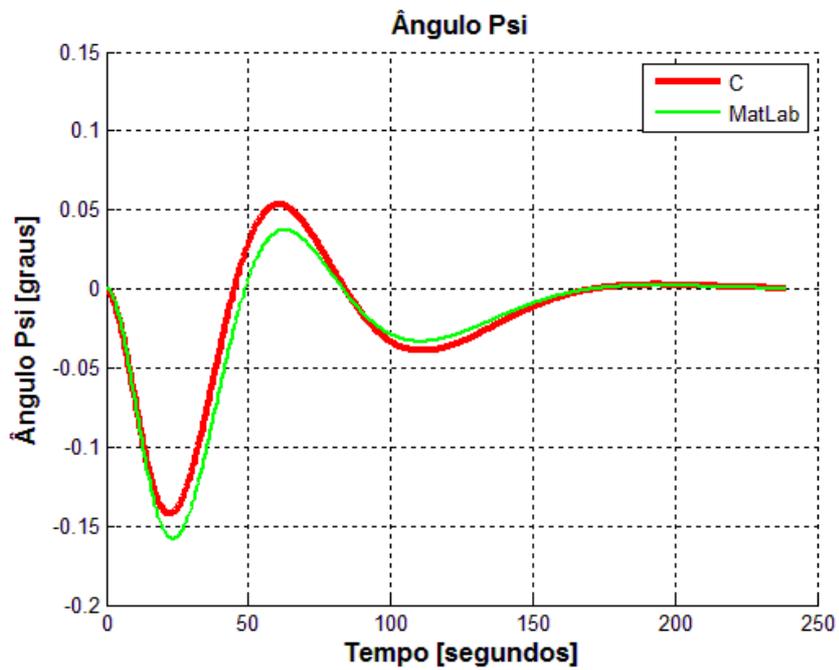


Figura 7.33 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

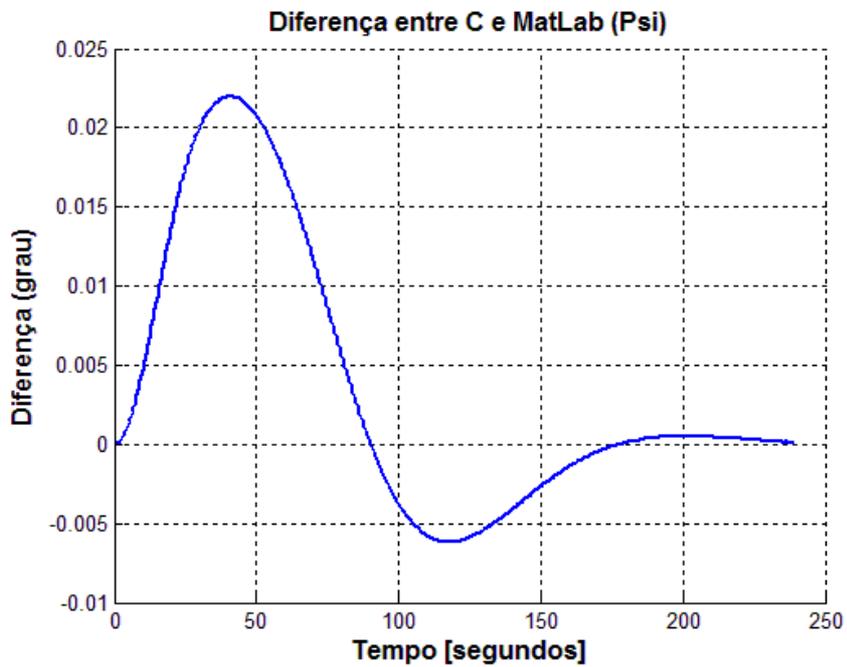


Figura 7.34 - Diferença das curvas dos programas simulados em linguagem C e MatLab

7.2.3.1.2. Velocidade angular da PMM

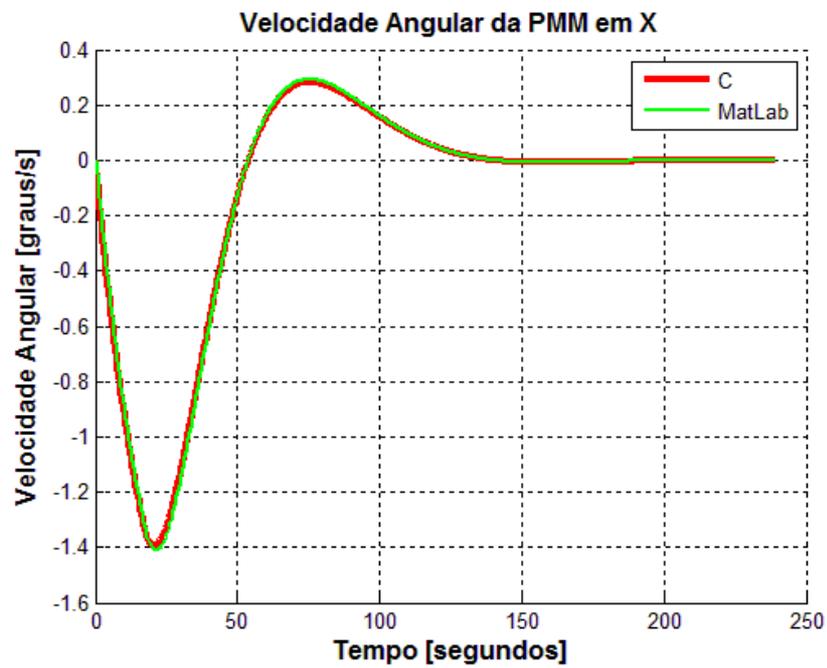


Figura 7.35 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

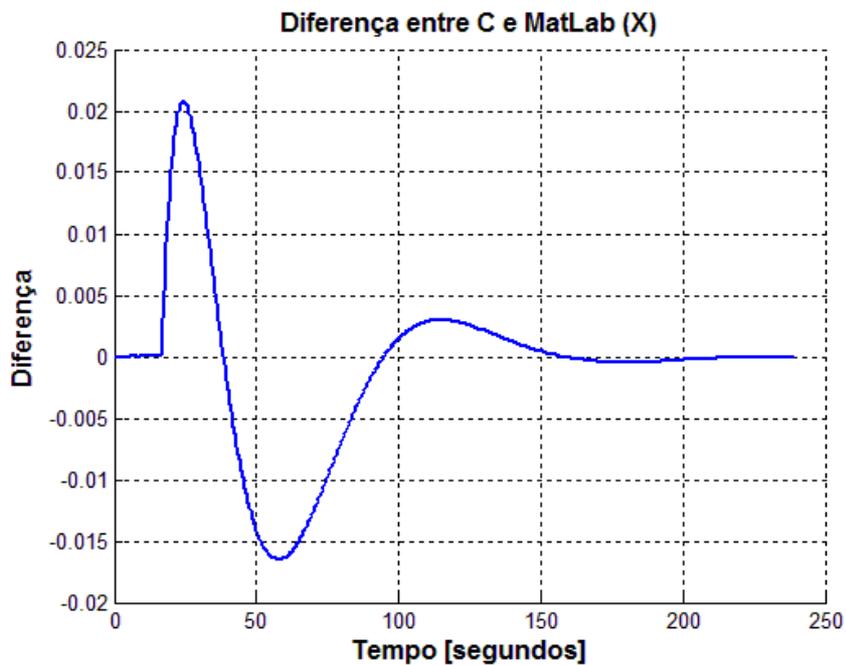


Figura 7.36 - Diferença das curvas dos programas simulados em linguagem C e MatLab

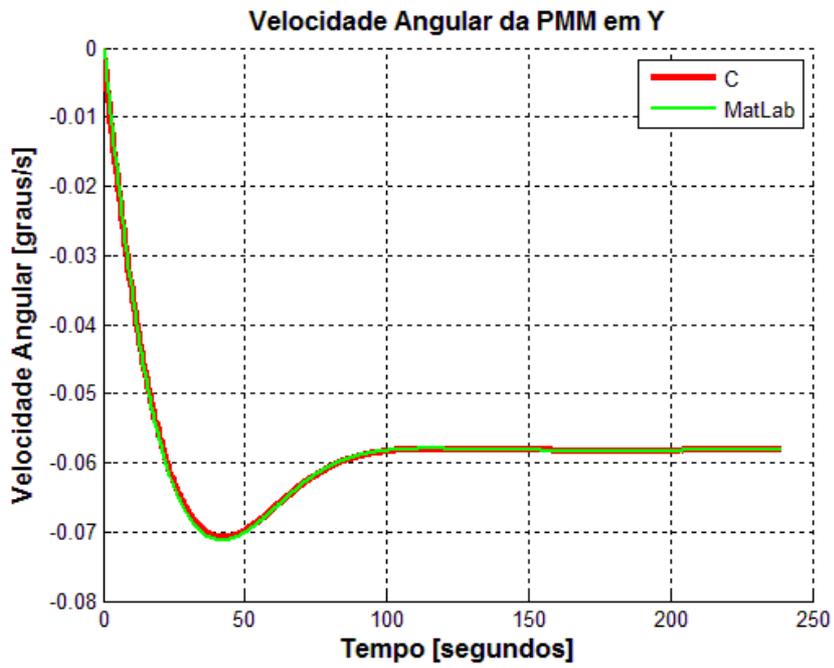


Figura 7.37 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

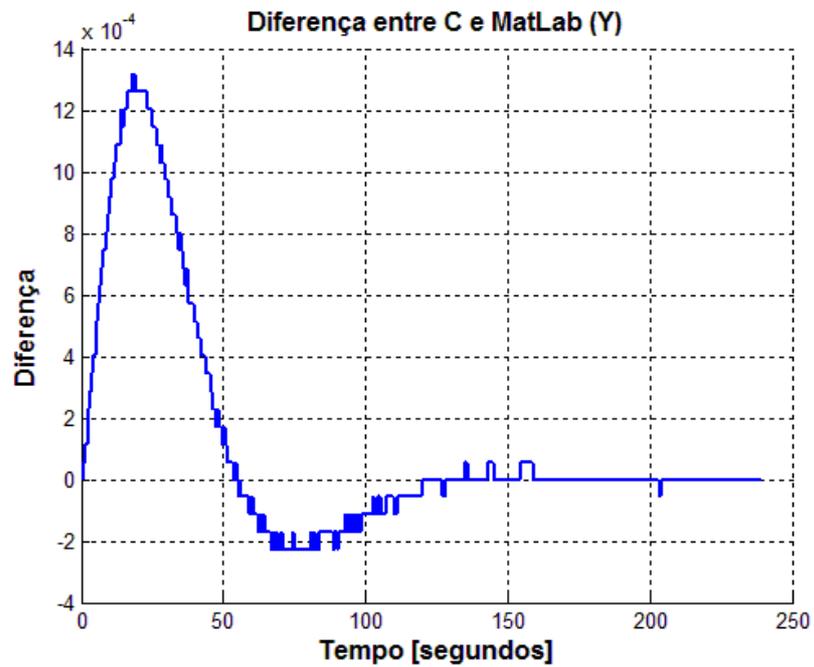


Figura 7.38 - Diferença das curvas dos programas simulados em linguagem C e MatLab

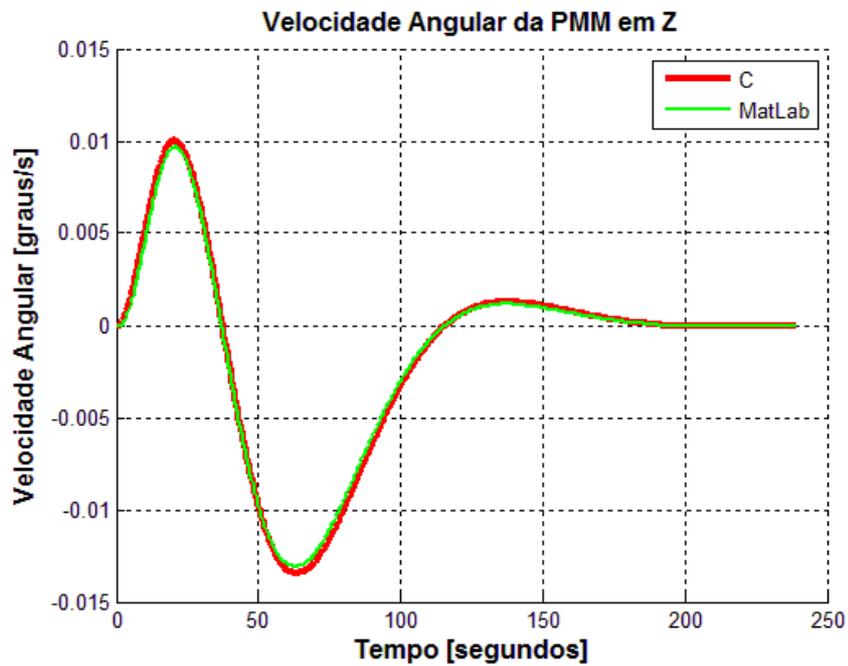


Figura 7.39 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

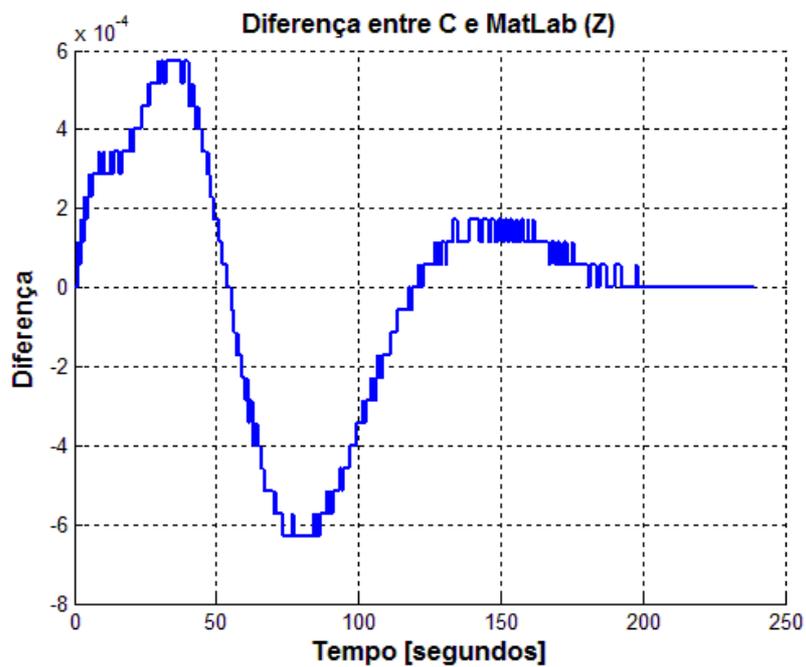


Figura 7.40 - Diferença das curvas dos programas simulados em linguagem C e MatLab

7.2.3.1.3. Velocidade angular das rodas

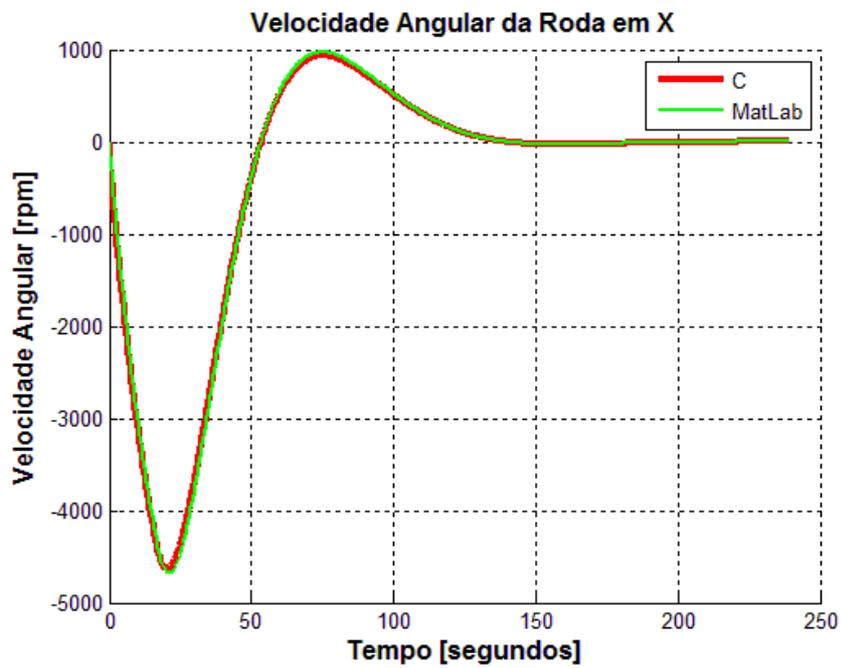


Figura 7.41 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

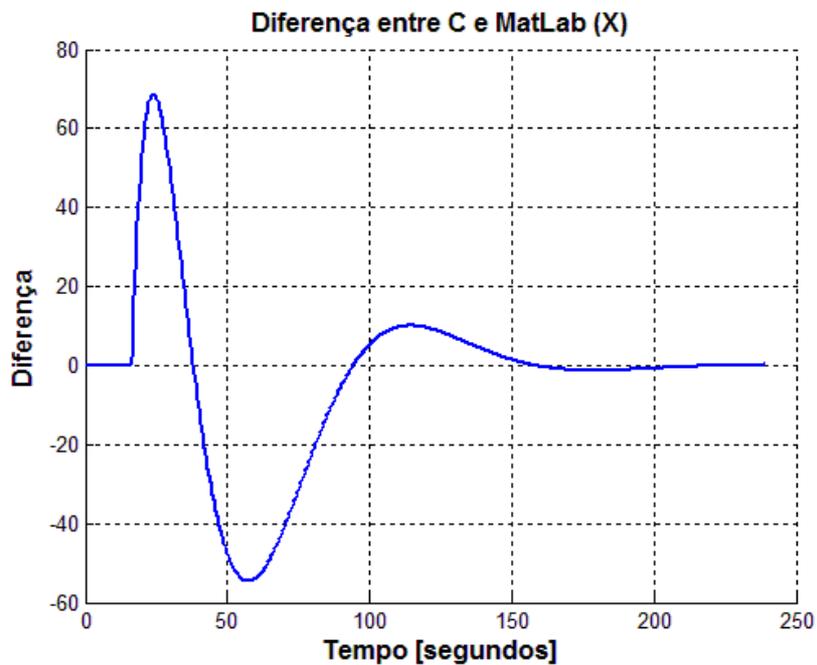


Figura 7.42 - Diferença das curvas dos programas simulados em linguagem C e MatLab

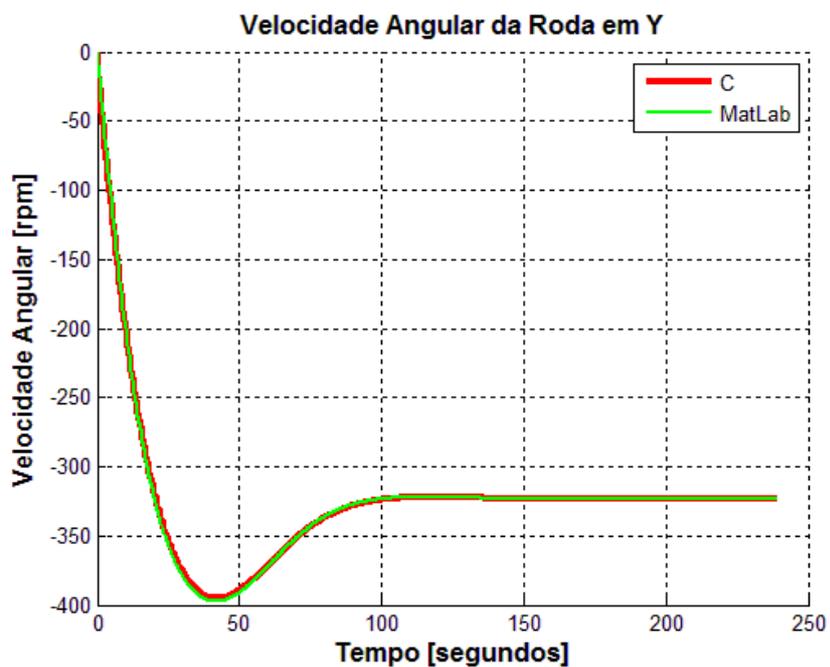


Figura 7.43 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

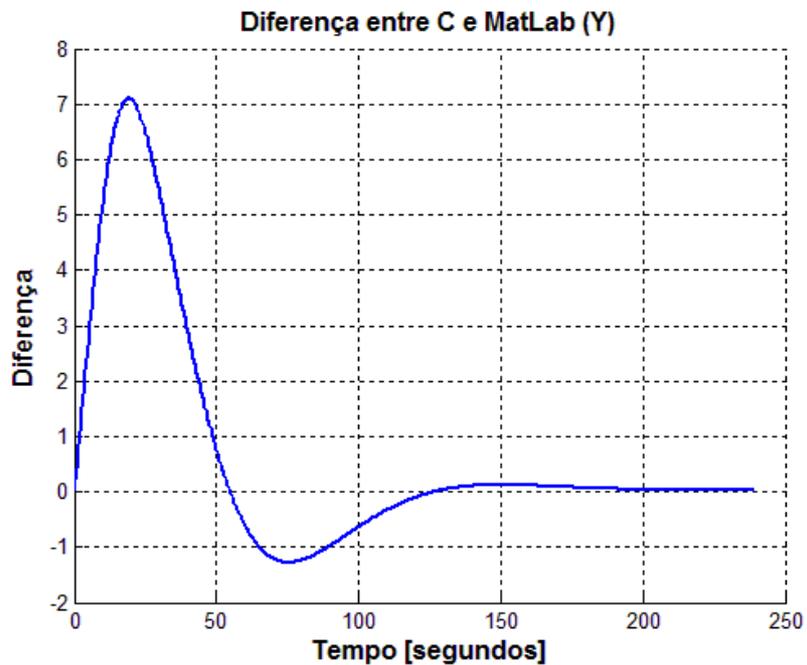


Figura 7.44 - Diferença das curvas dos programas simulados em linguagem C e MatLab

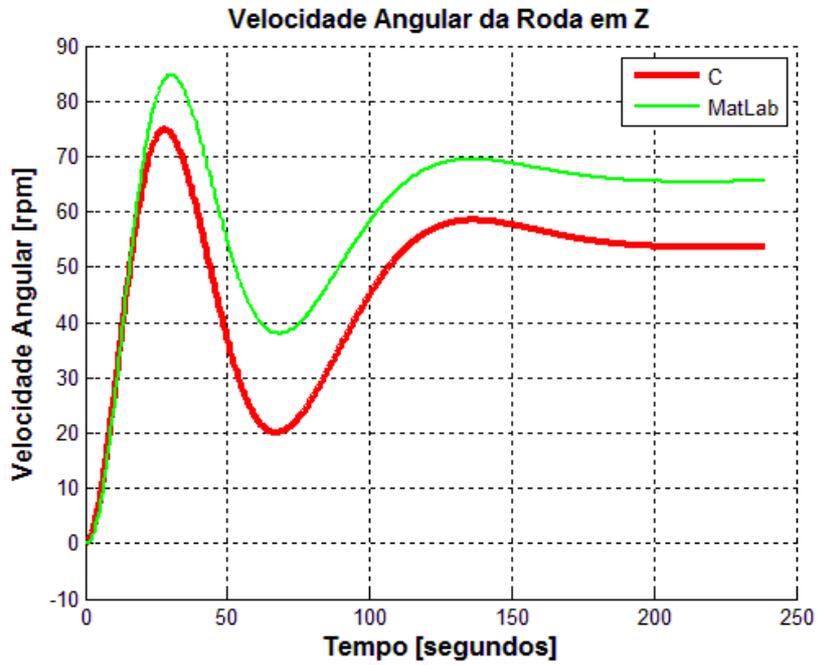


Figura 7.45 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

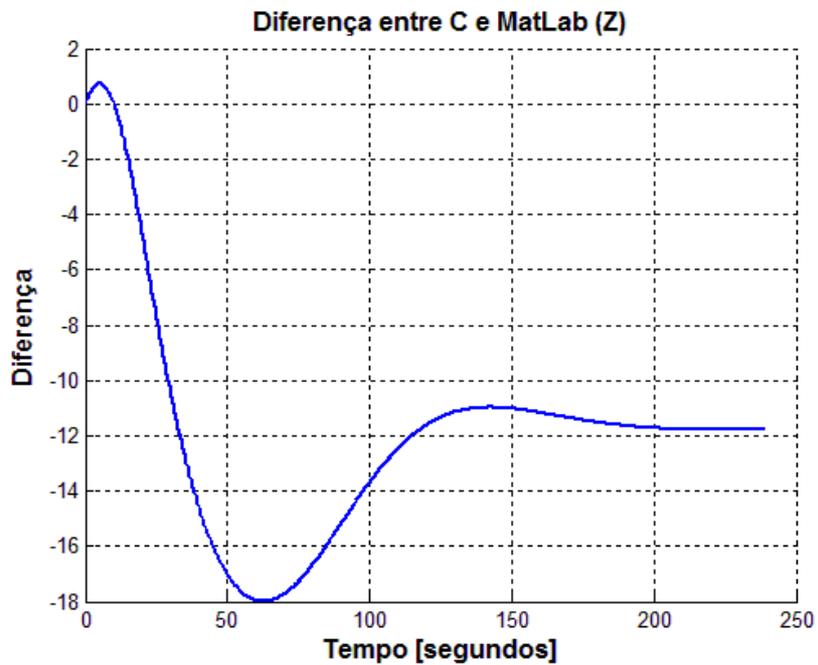


Figura 7.46 - Diferença das curvas dos programas simulados em linguagem C e MatLab

7.2.3.1.4. Tensão

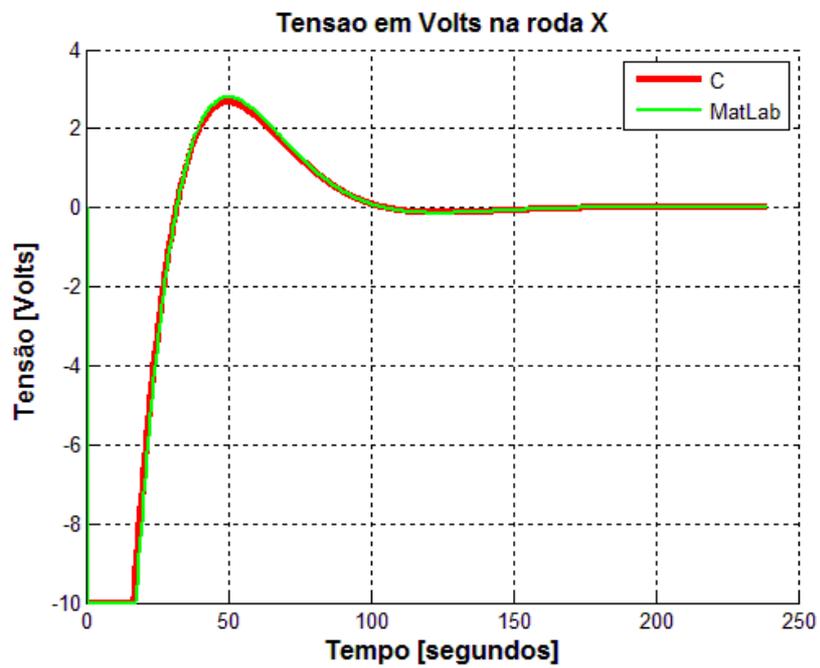


Figura 7.47 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

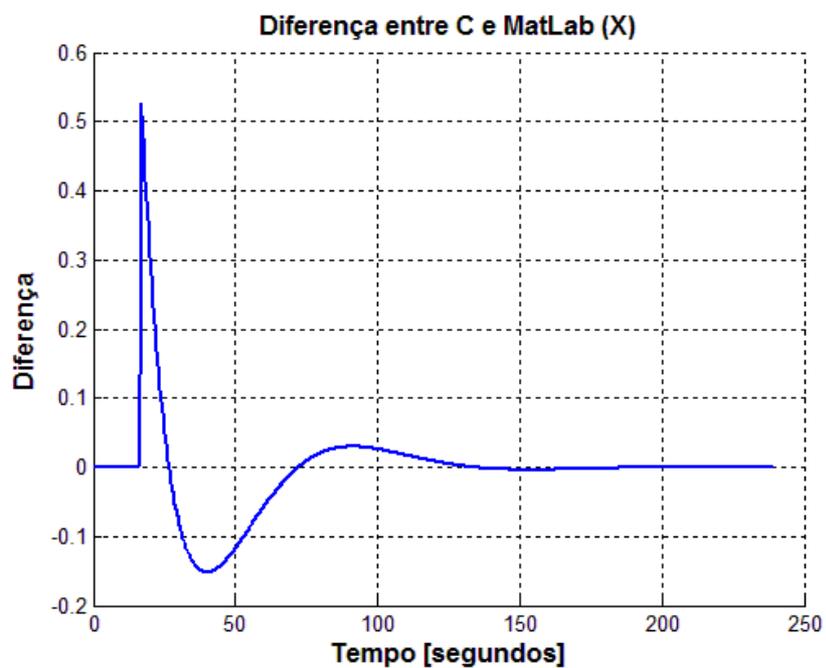


Figura 7.48 - Diferença das curvas dos programas simulados em linguagem C e MatLab

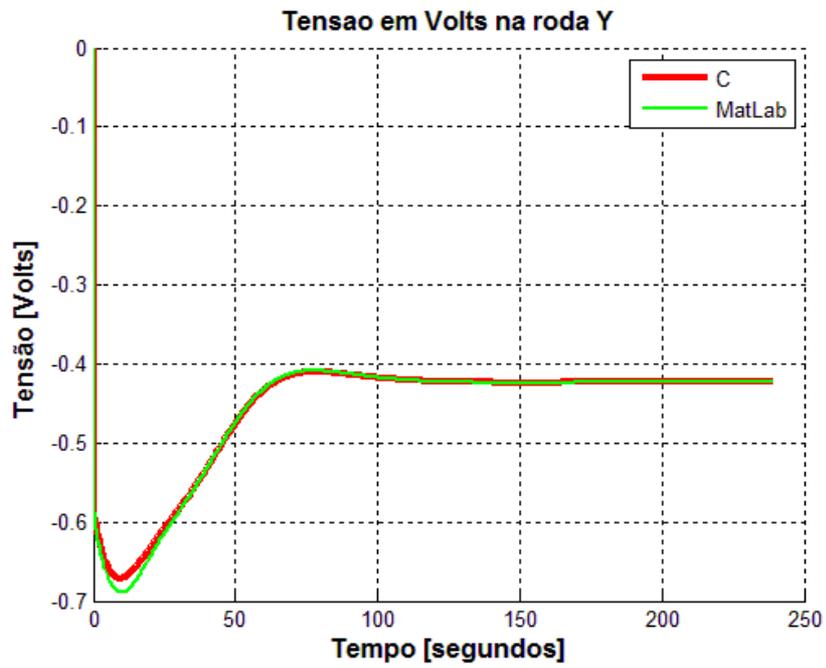


Figura 7.49 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

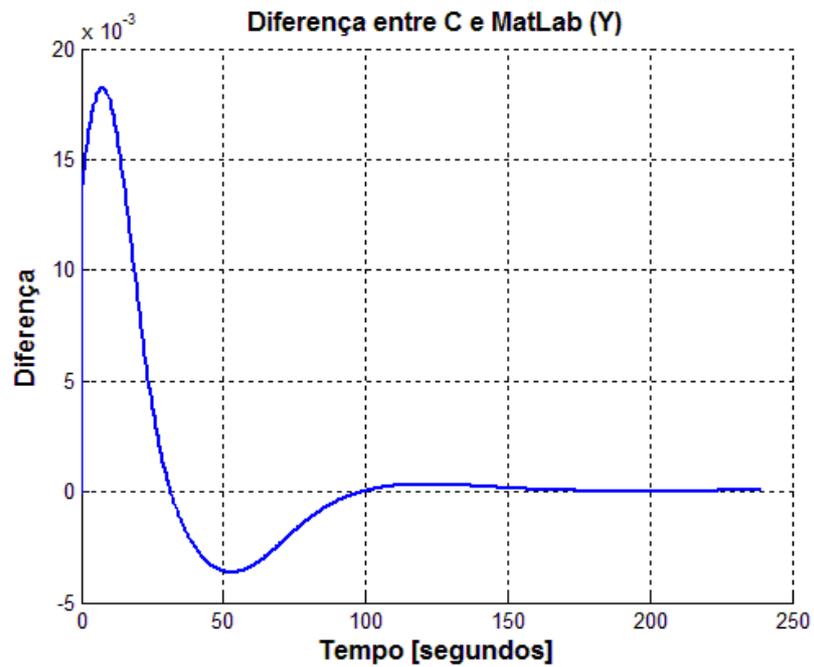


Figura 7.50 - Diferença das curvas dos programas simulados em linguagem C e MatLab

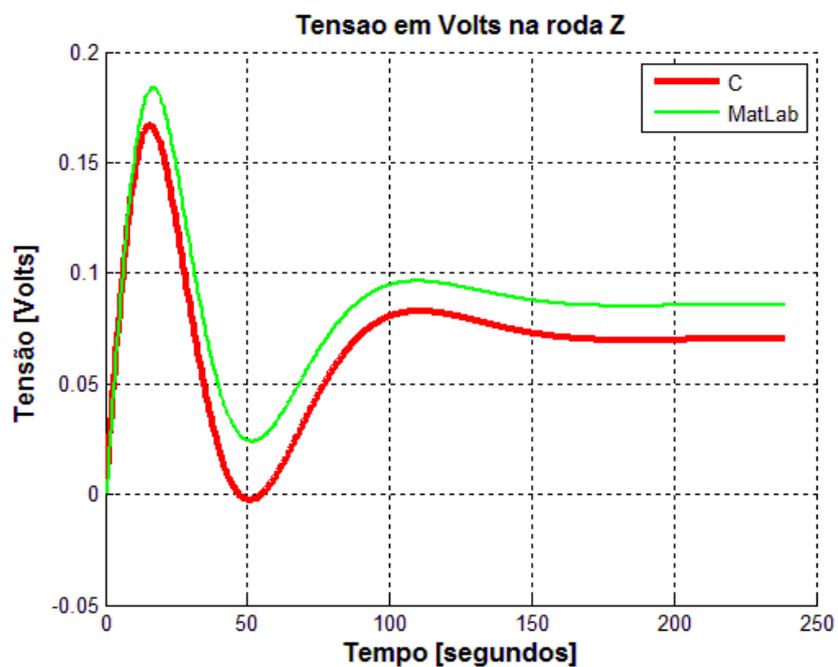


Figura 7.51 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

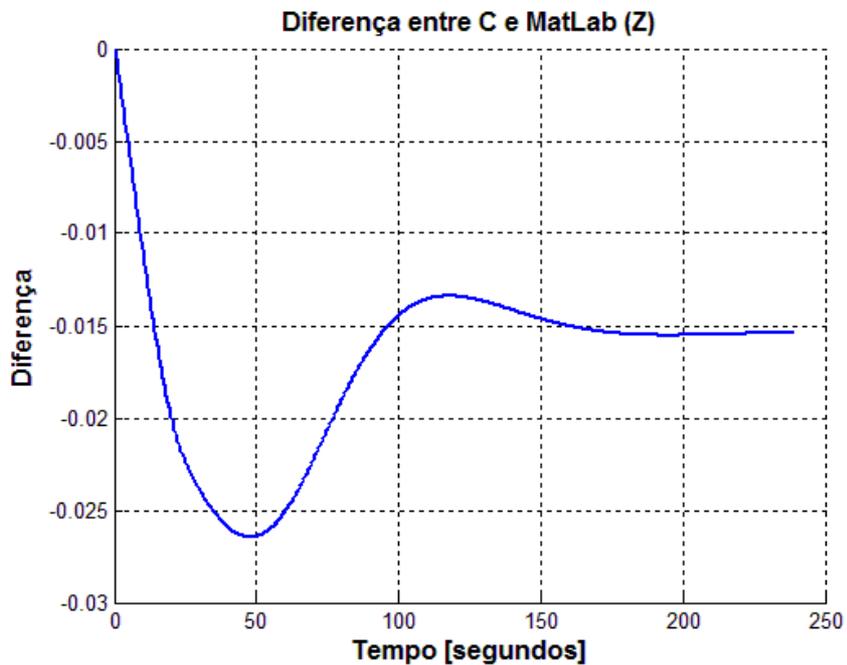


Figura 7.52 - Diferença das curvas dos programas simulados em linguagem C e MatLab

7.2.3.1.5. Torque da roda

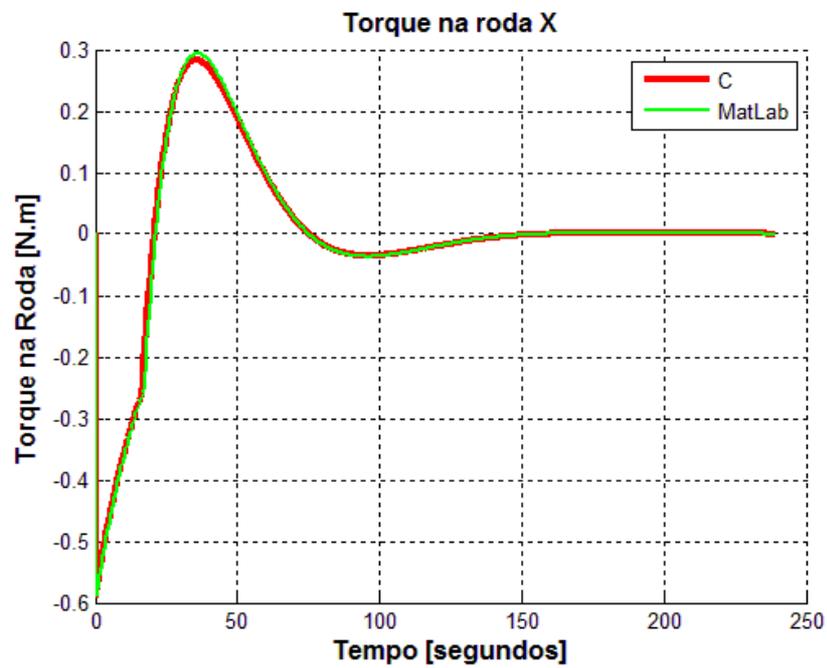


Figura 7.53 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

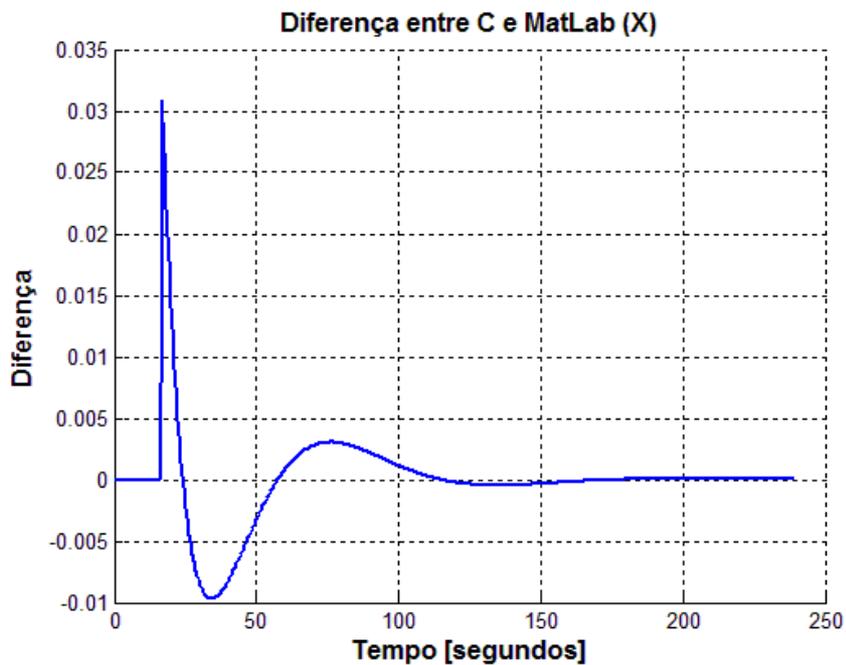


Figura 7.54 - Diferença das curvas dos programas simulados em linguagem C e MatLab

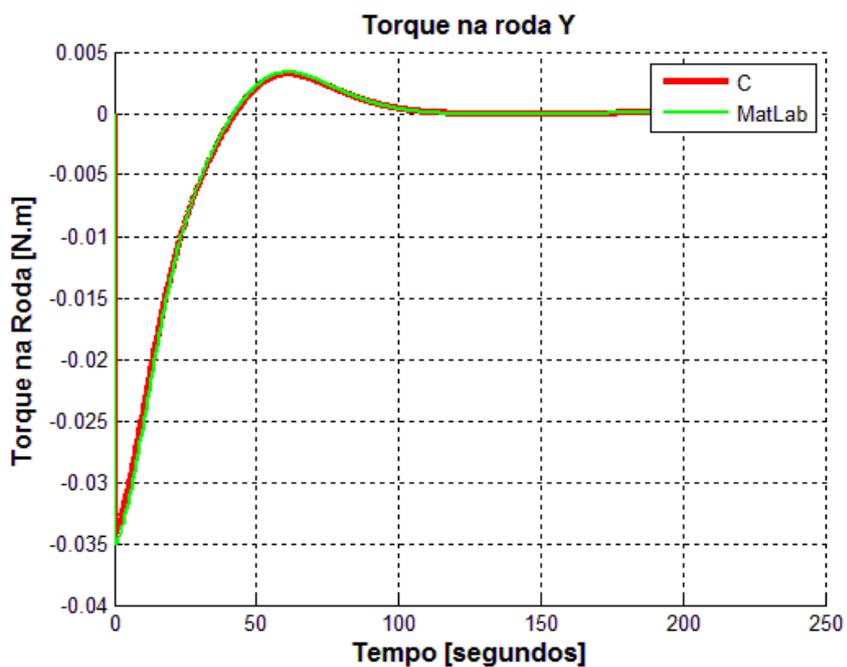


Figura 7.55 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

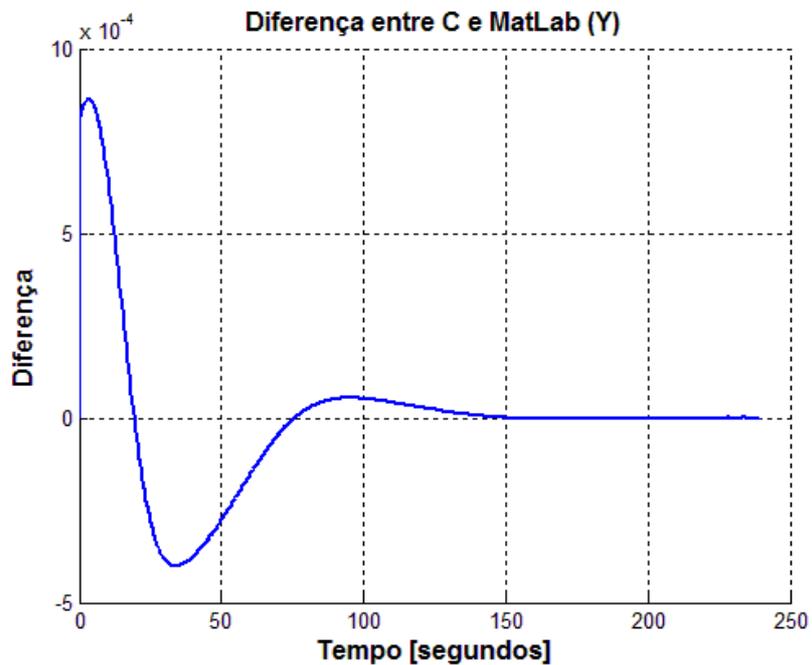


Figura 7.56 - Diferença das curvas dos programas simulados em linguagem C e MatLab

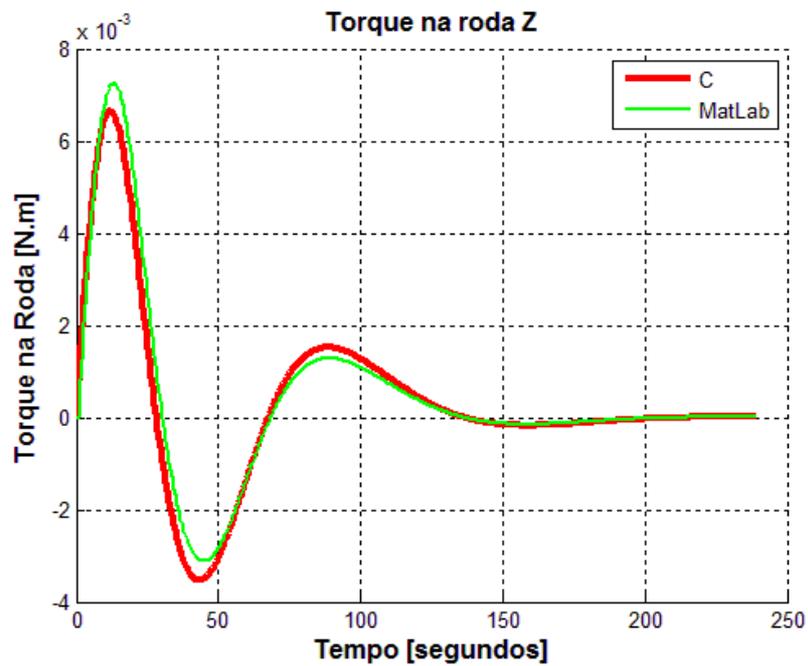


Figura 7.57 - Curvas dos resultados dos programas simulados em linguagem C e MatLab

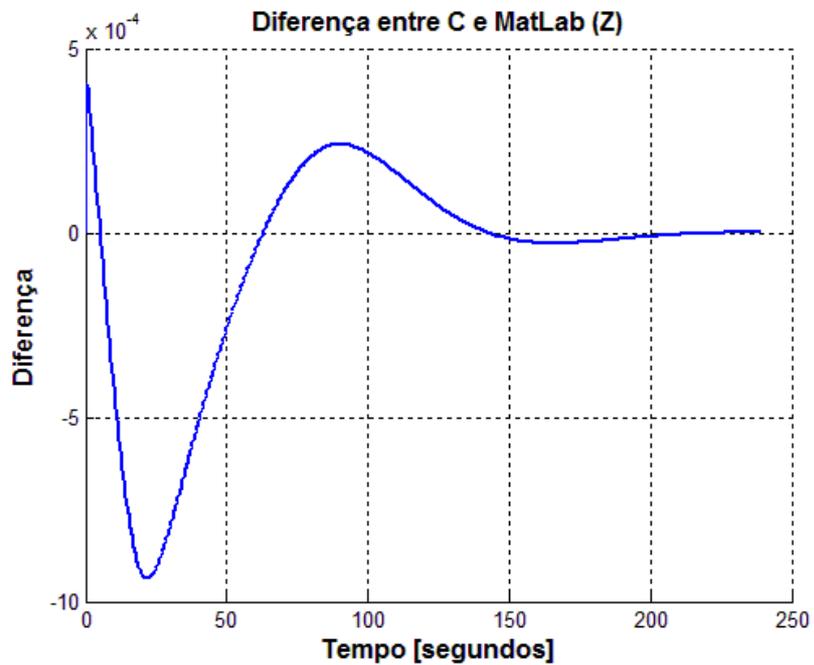


Figura 7.58- Diferença das curvas dos programas simulados em linguagem C e MatLab

7.2.3.2. Caso 2: Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e MatLab

A seguir são apresentados os gráficos obtidos a partir da simulação do sistema de controle de atitude da PMM no SIMA e em MatLab. Os gráficos são ordenados da seguinte maneira, ângulos de atitude em cada eixo, velocidade angular da PMM, velocidade angular das rodas, tensão enviada a cada roda de reação e torque fornecido por roda.

7.2.3.2.1. Ângulos de atitude

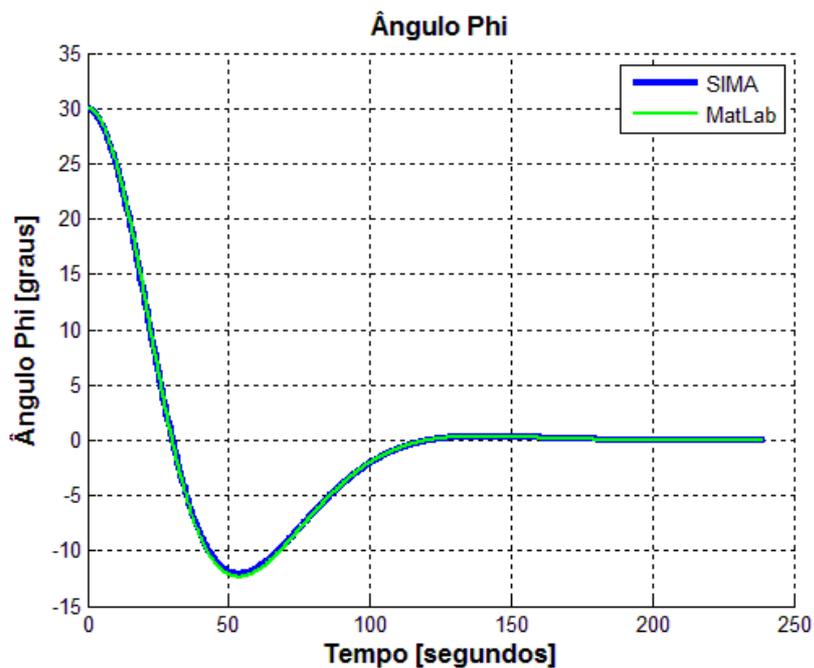


Figura 7.59 - Curvas dos resultados dos programas simulados em SIMA e MatLab

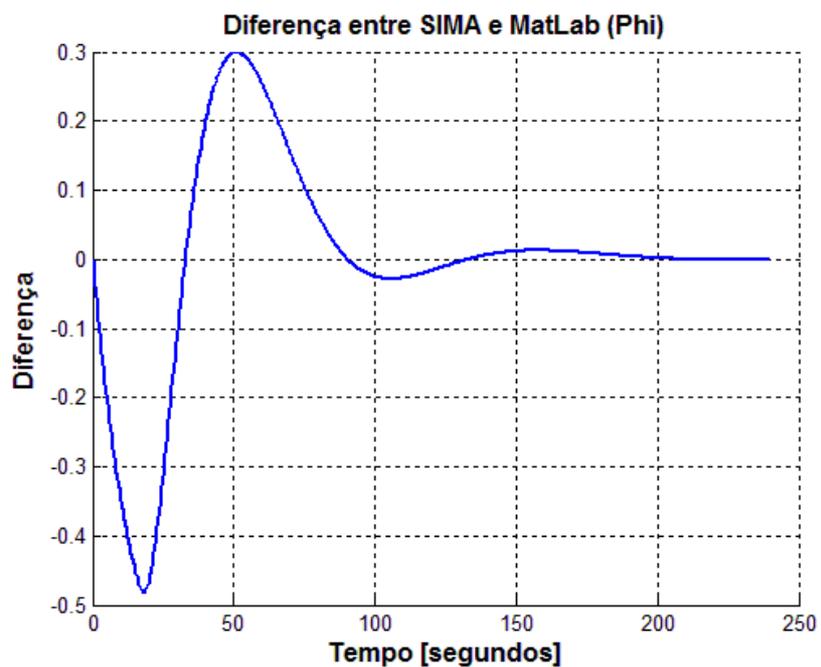


Figura 7.60 - Diferença das curvas dos programas simulados em SIMA e MatLab

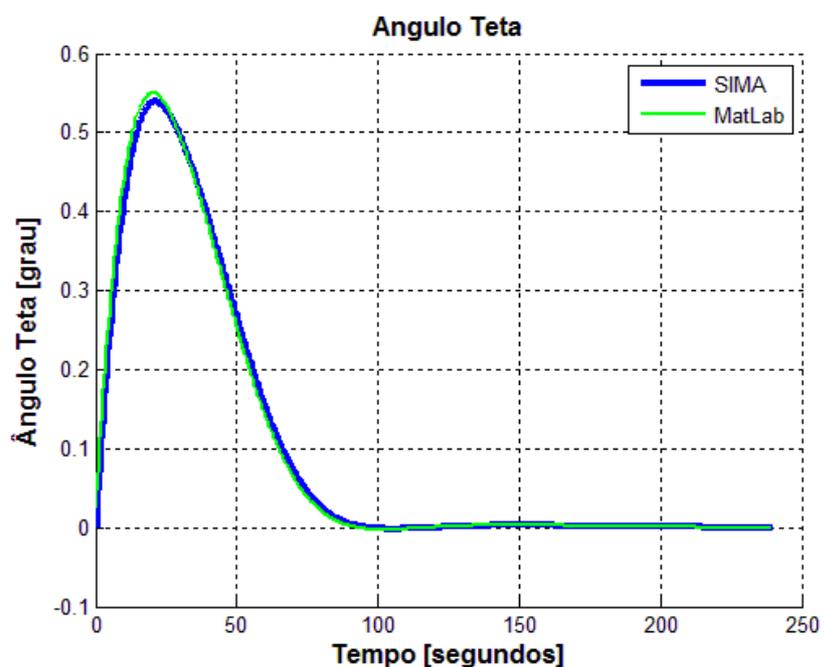


Figura 7.61 - Curvas dos resultados dos programas simulados em SIMA e MatLab

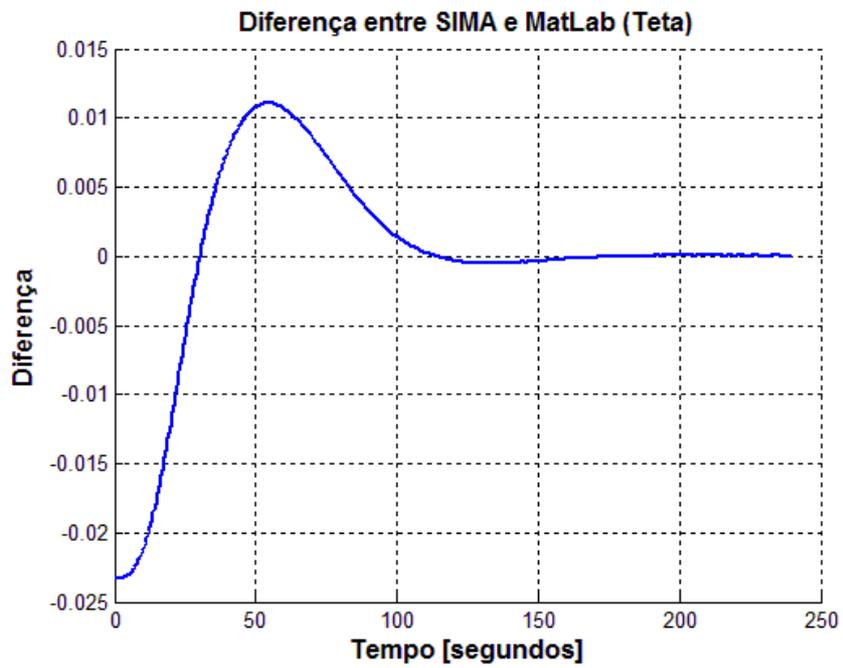


Figura 7.62 - Diferença das curvas dos programas simulados em SIMA e MatLab

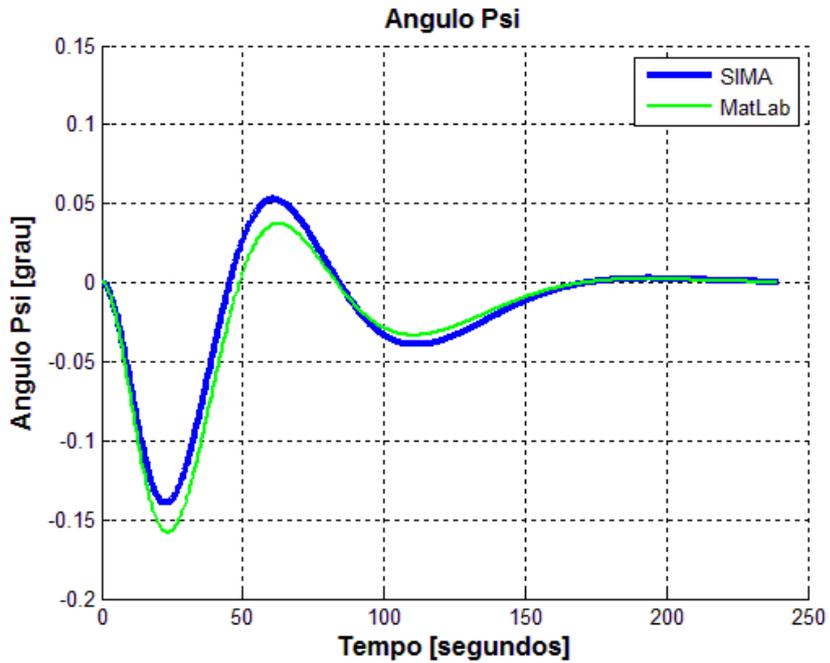


Figura 7.63 - Curvas dos resultados dos programas simulados em SIMA e MatLab

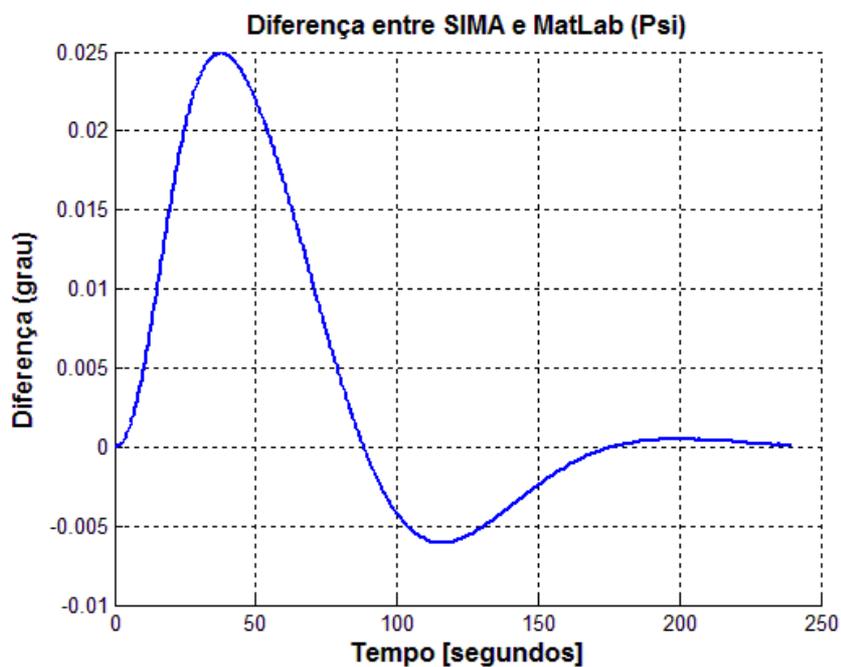


Figura 7.64 - Diferença das curvas dos programas simulados em SIMA e MatLab

7.2.3.2.2. Velocidade angular da PMM

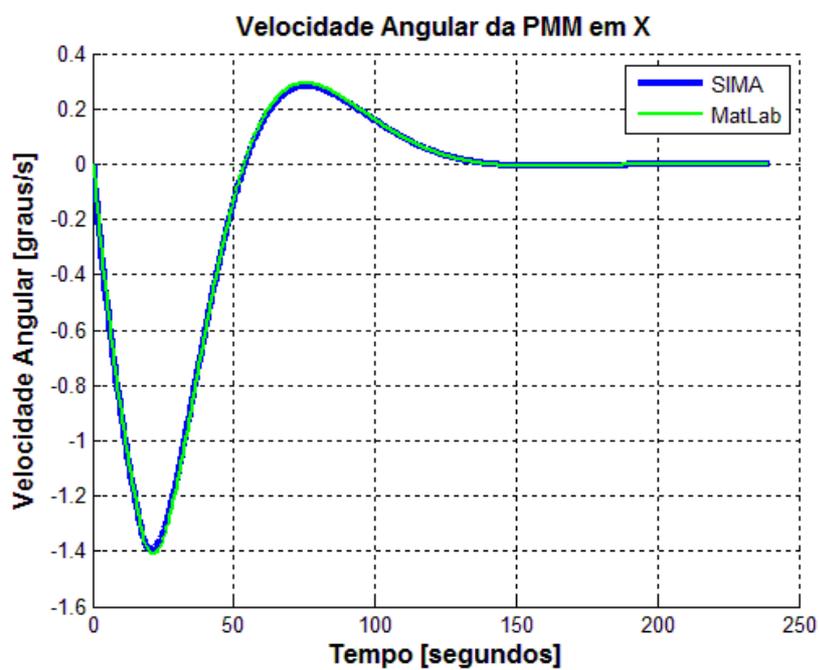


Figura 7.65 - Curvas dos resultados dos programas simulados em SIMA e MatLab

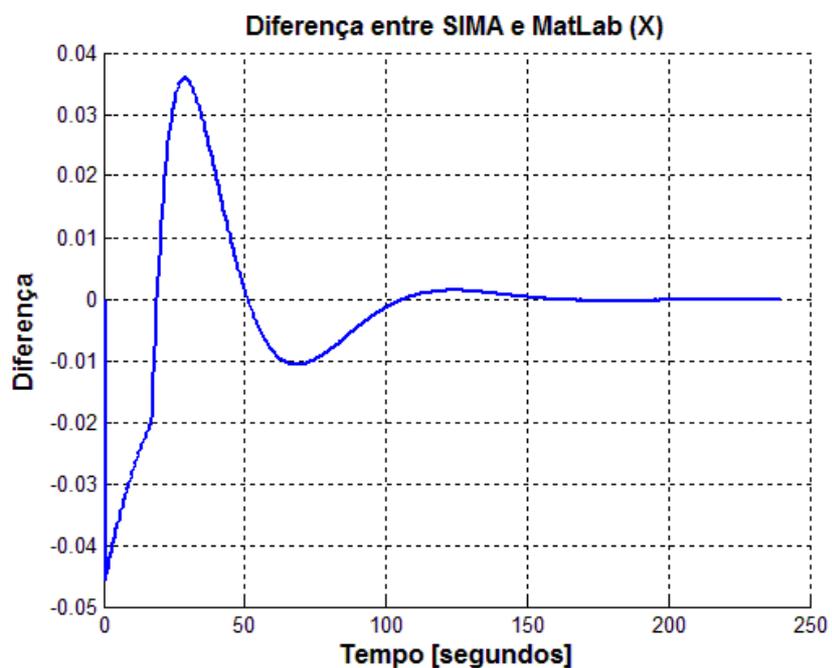


Figura 7.66 - Diferença das curvas dos programas simulados em SIMA e MatLab

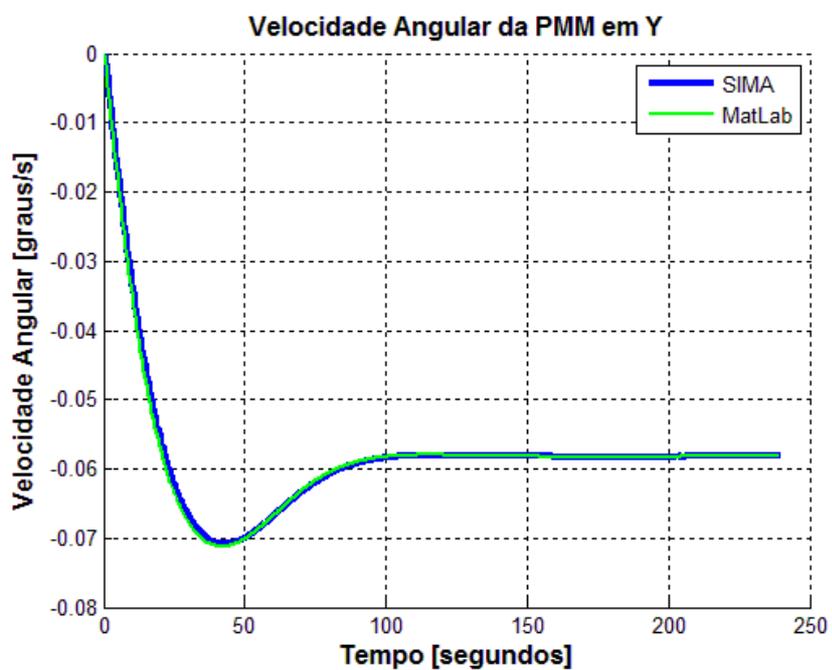


Figura 7.67 - Curvas dos resultados dos programas simulados em SIMA e MatLab

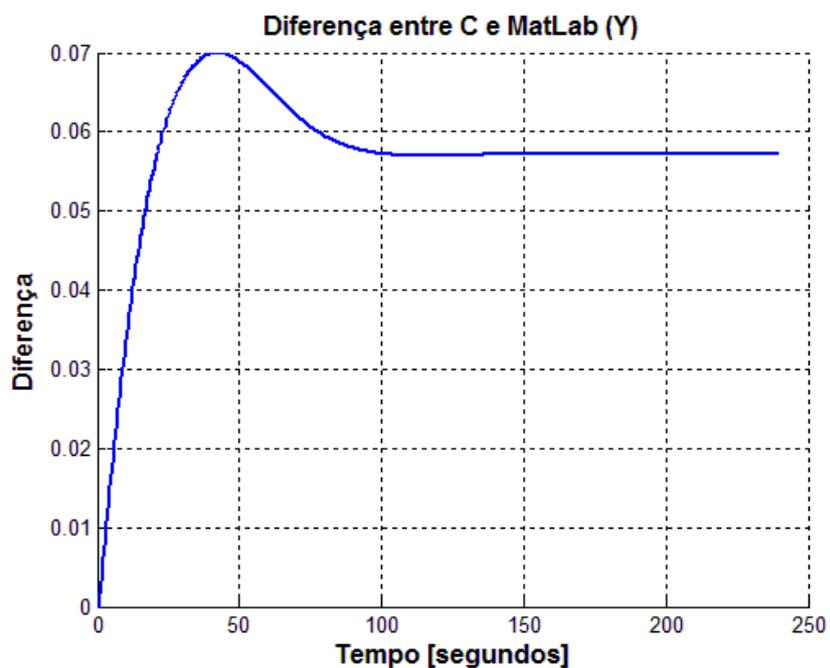


Figura 7.68 - Diferença das curvas dos programas simulados em SIMA e MatLab

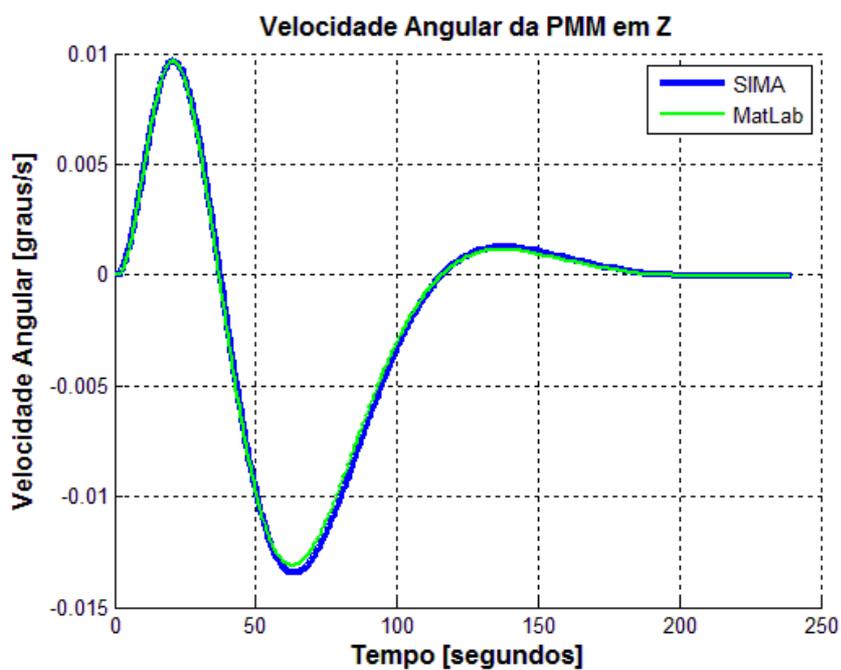


Figura 7.69 - Curvas dos resultados dos programas simulados em SIMA e MatLab

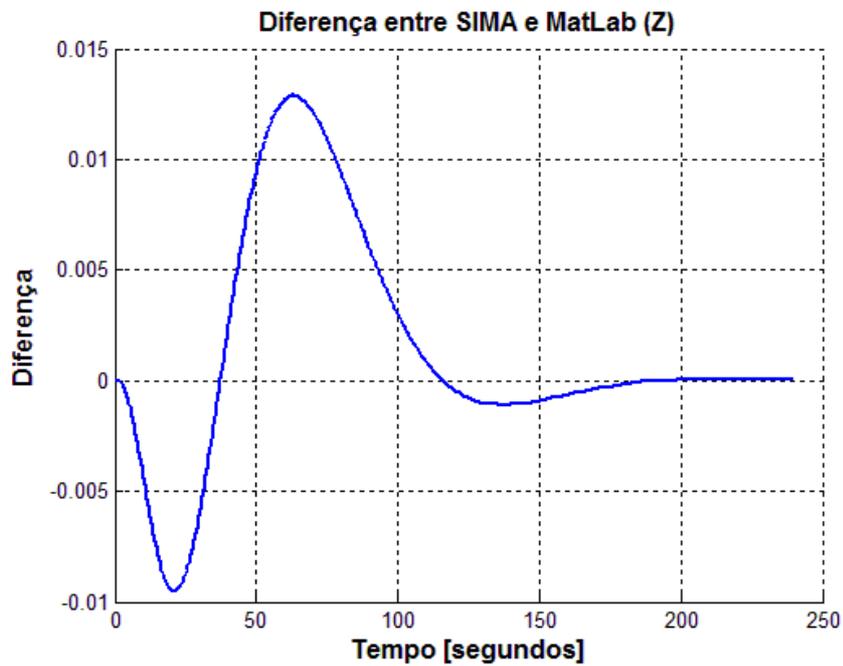


Figura 7.70 - Diferença das curvas dos programas simulados em SIMA e MatLab

7.2.3.2.3. Velocidade angular das rodas

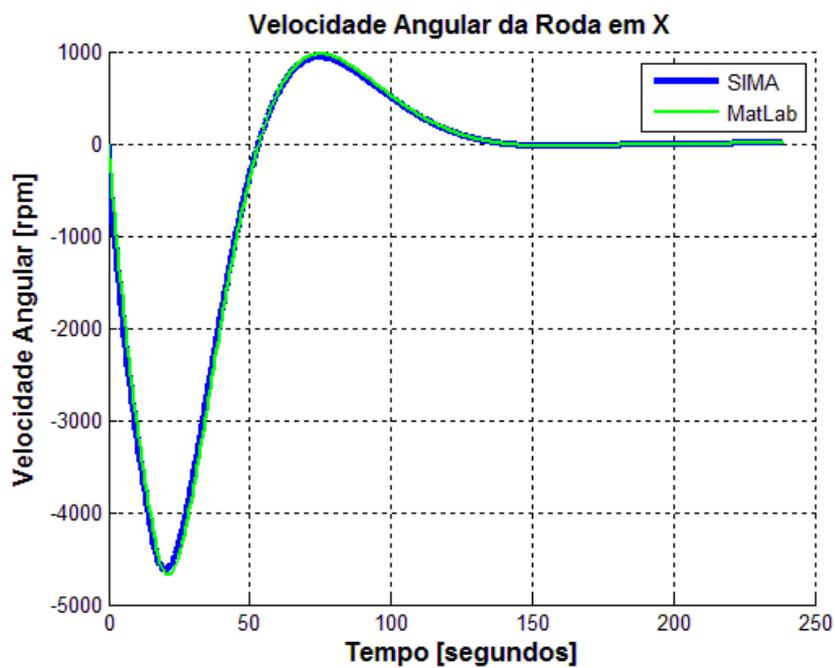


Figura 7.71 - Curvas dos resultados dos programas simulados em SIMA e MatLab

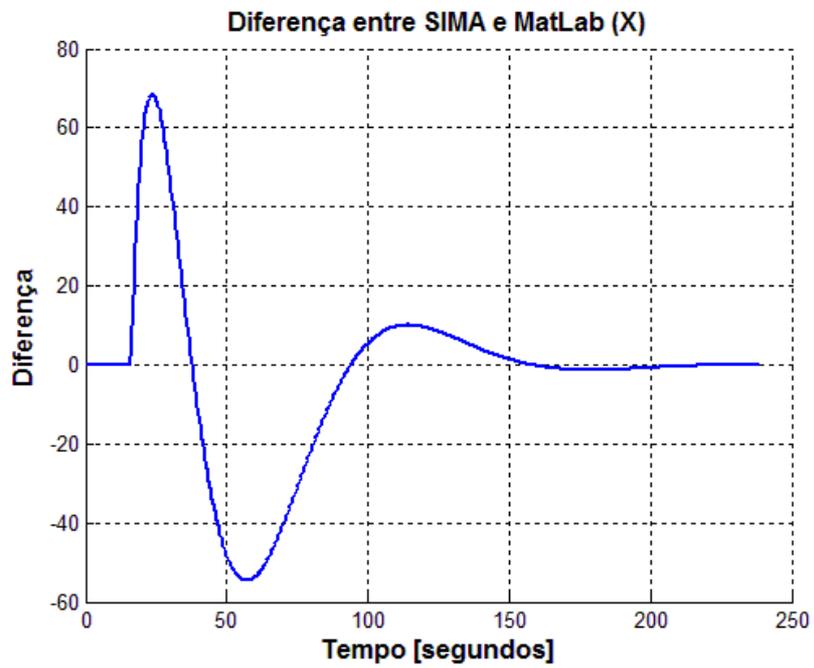


Figura 7.72 - Diferença das curvas dos programas simulados em SIMA e MatLab

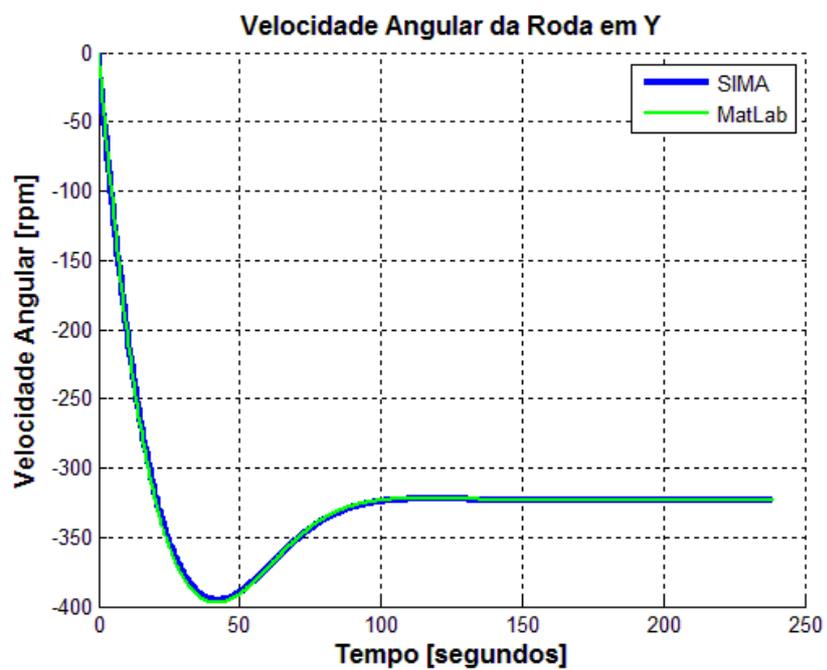


Figura 7.73 - Curvas dos resultados dos programas simulados em SIMA e MatLab

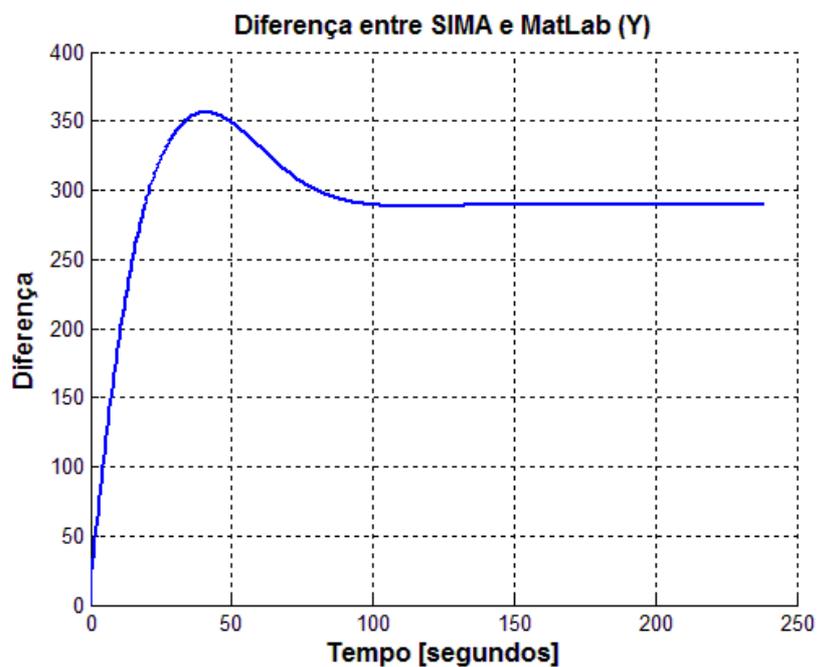


Figura 7.74 - Diferença das curvas dos programas simulados em SIMA e MatLab

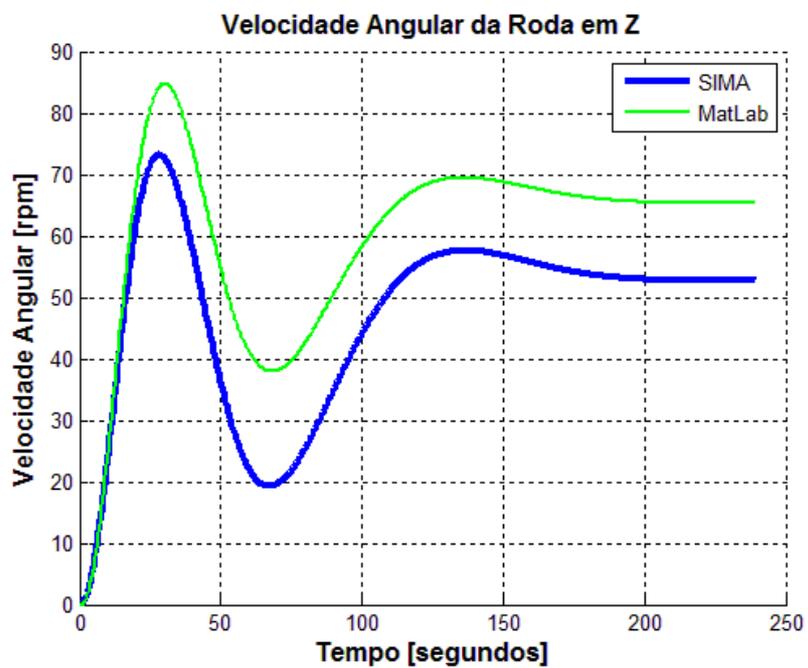


Figura 7.75 - Curvas dos resultados dos programas simulados em SIMA e MatLab

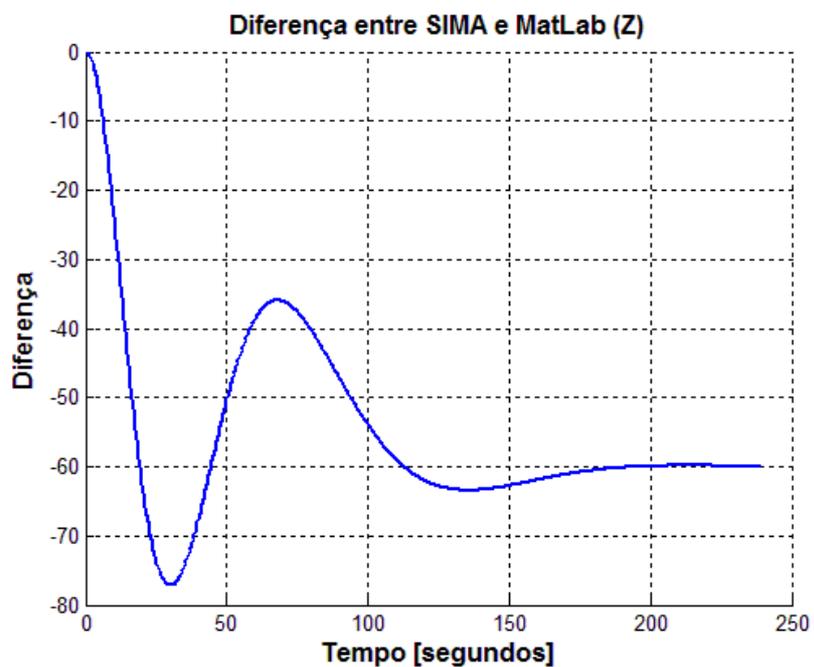


Figura 7.76 - Diferença das curvas dos programas simulados em SIMA e MatLab

7.2.3.2.4. Tensão

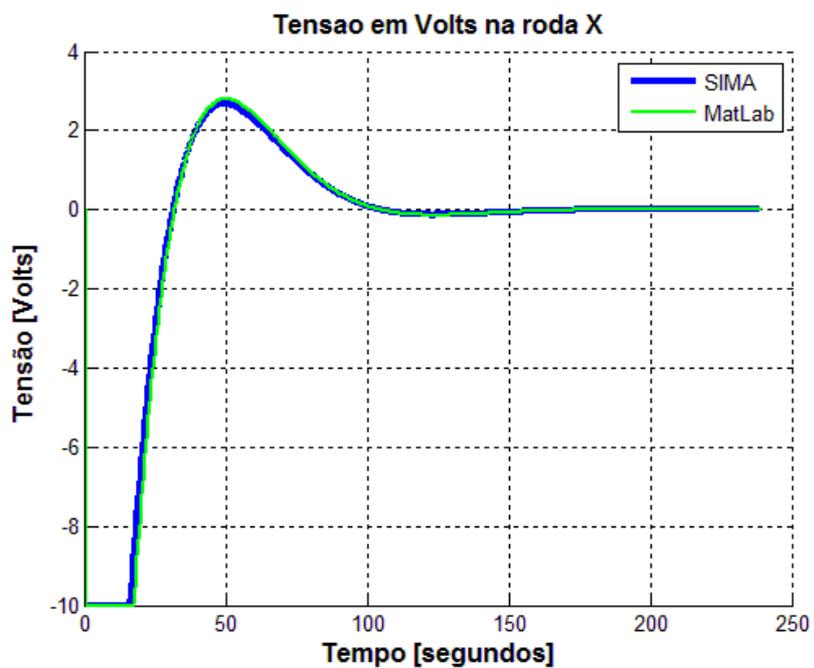


Figura 7.77 - Curvas dos resultados dos programas simulados em SIMA e MatLab

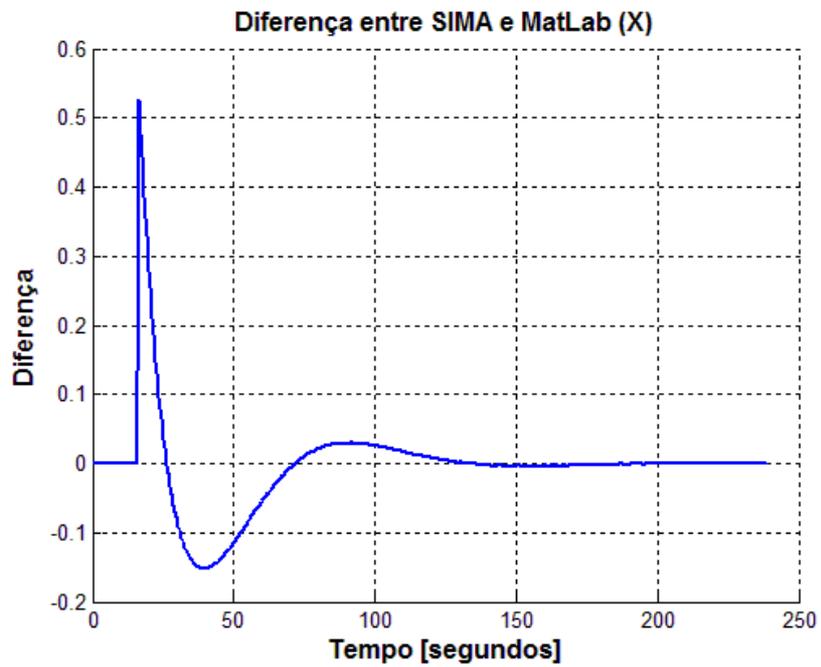


Figura 7.78 - Diferença das curvas dos programas simulados em SIMA e MatLab

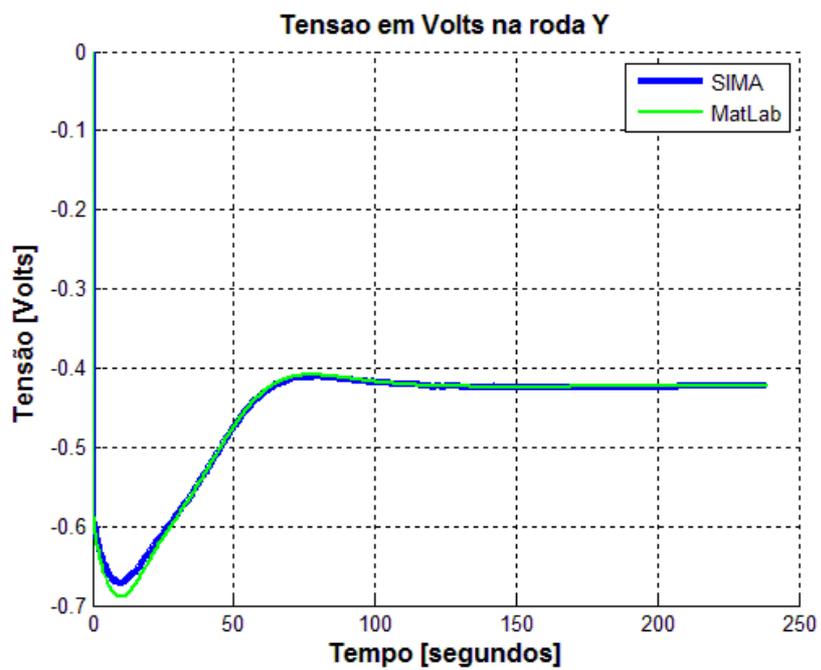


Figura 7.79 - Curvas dos resultados dos programas simulados em SIMA e MatLab

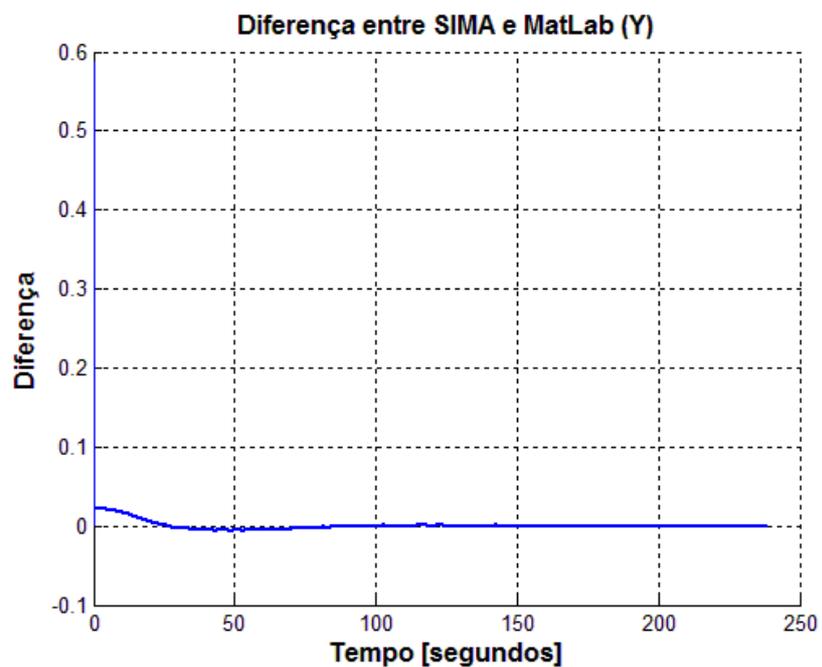


Figura 7.80 - Diferença das curvas dos programas simulados em SIMA e MatLab

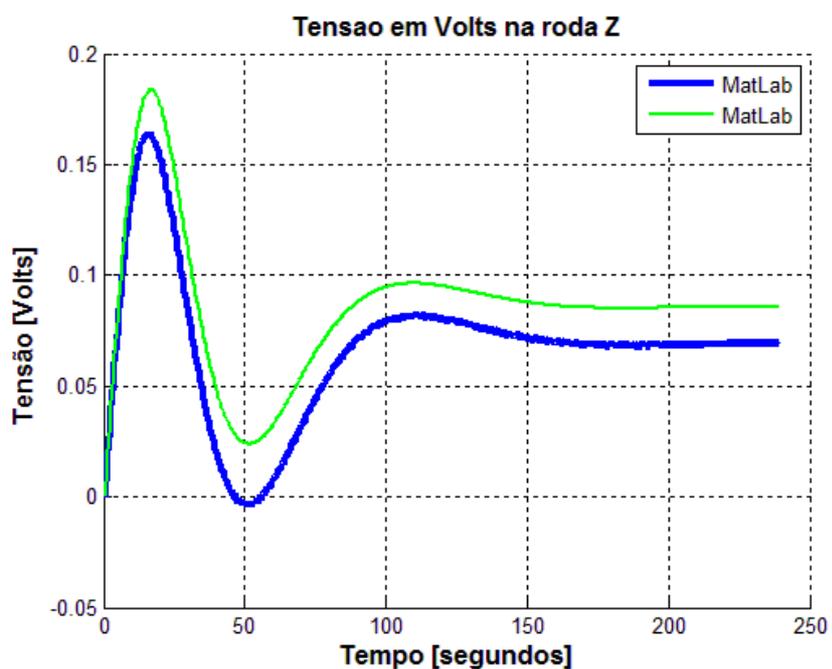


Figura 7.81 - Curvas dos resultados dos programas simulados em SIMA e MatLab

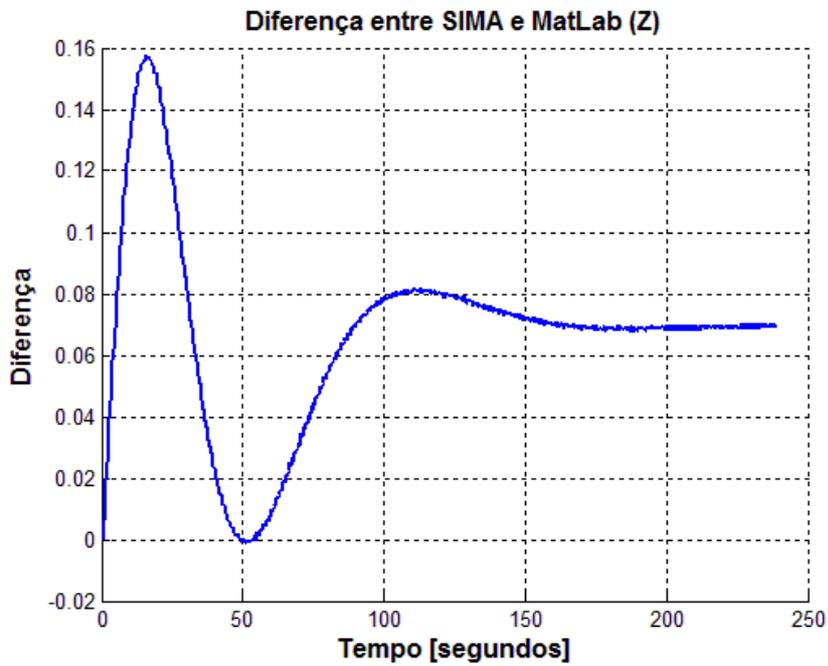


Figura 7.82 - Diferença das curvas dos programas simulados em SIMA e MatLab

7.2.3.2.5. Torque da roda

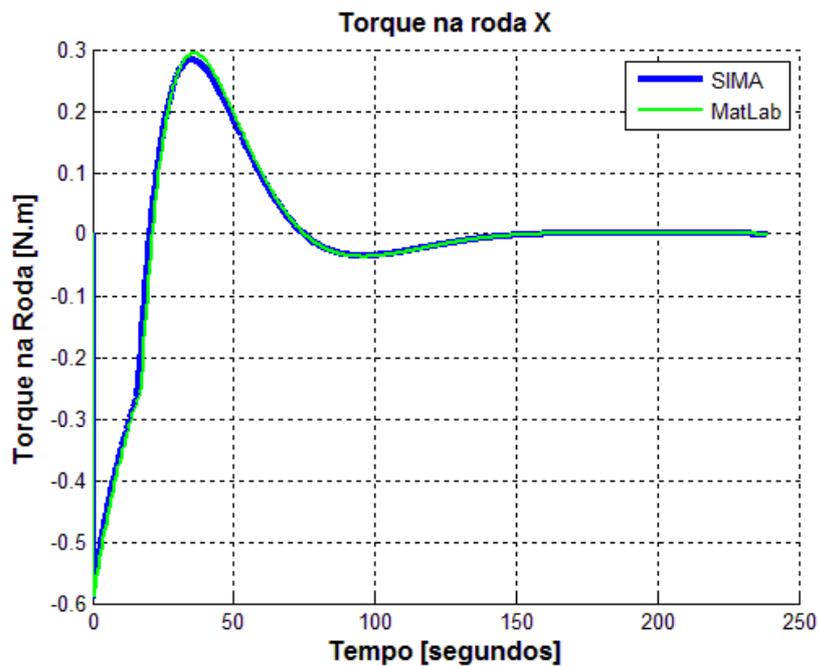


Figura 7.83 - Curvas dos resultados dos programas simulados em SIMA e MatLab

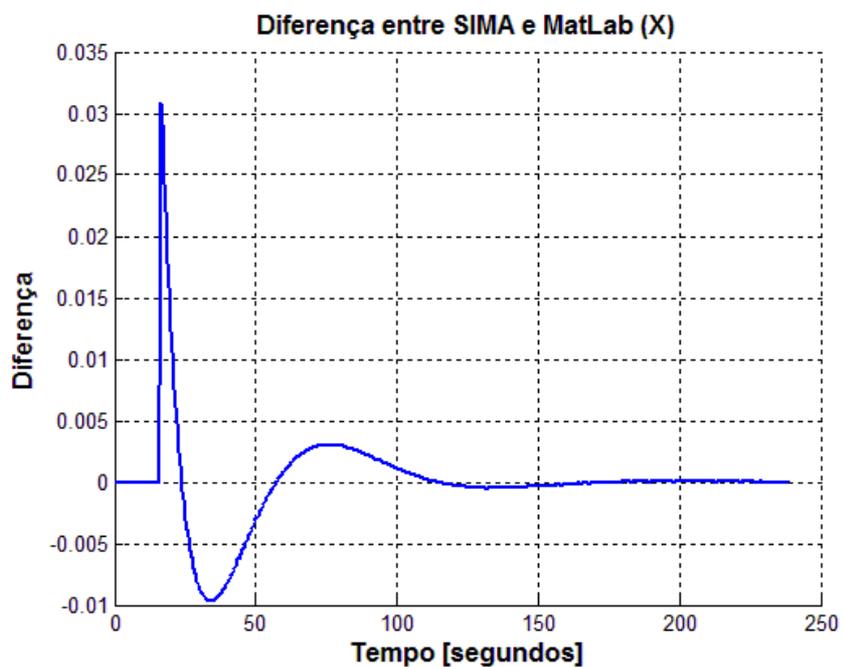


Figura 7.84 - Diferença das curvas dos programas simulados em SIMA e MatLab

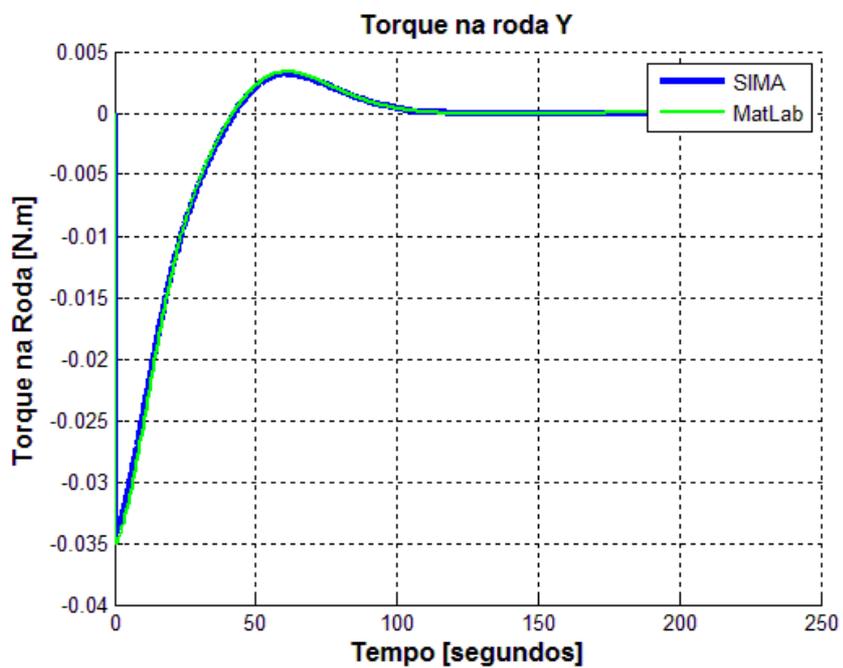


Figura 7.85 - Curvas dos resultados dos programas simulados em SIMA e MatLab

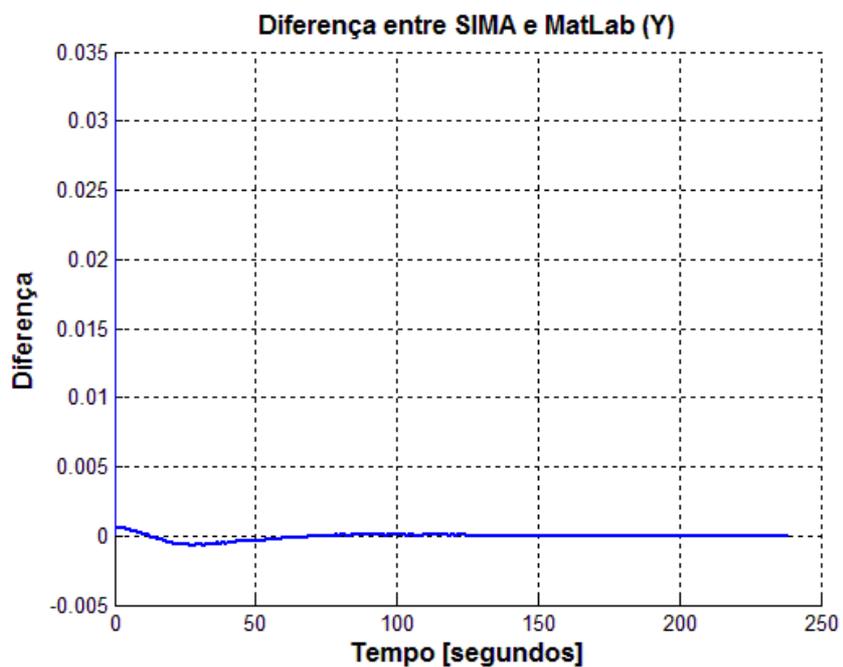


Figura 7.86 - Diferença das curvas dos programas simulados em SIMA e MatLab

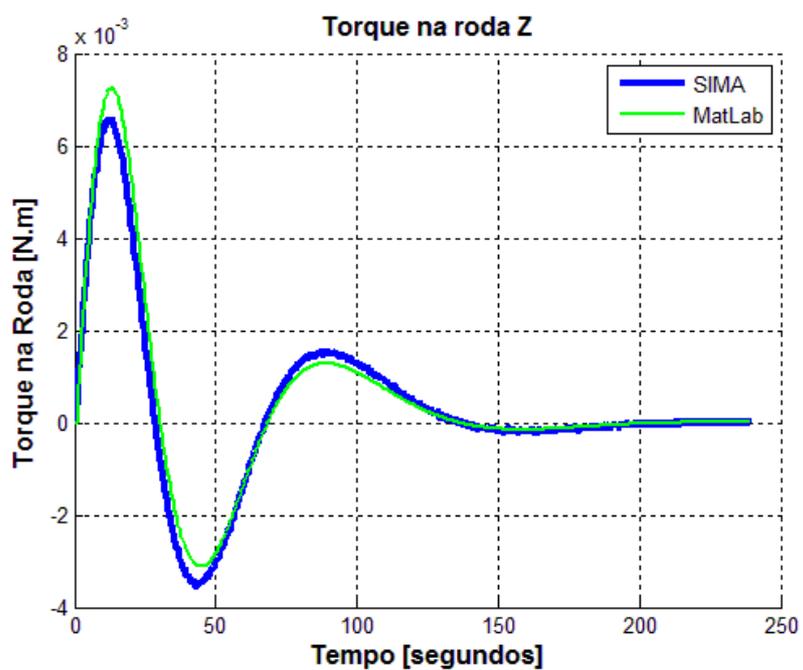


Figura 7.87 - Curvas dos resultados dos programas simulados em SIMA e MatLab

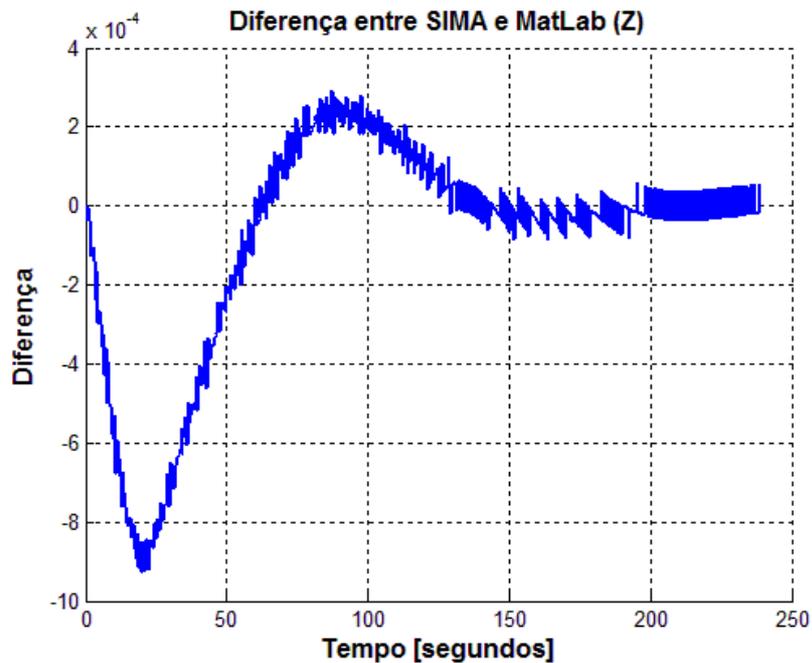


Figura 7.88 - Diferença das curvas dos programas simulados em SIMA e MatLab

7.2.3.3. Caso3: Simulação do Modo Nominal do SCA da PMM em SIMA (4 partições) e linguagem C

A seguir são apresentados os gráficos obtidos a partir da simulação do sistema de controle de atitude da PMM no SIMA e em linguagem C. Os gráficos são ordenados da seguinte maneira, ângulos de atitude em cada eixo, velocidade angular da PMM, velocidade angular das rodas, tensão enviada a cada roda de reação e torque fornecido por roda.

7.2.3.3.1. Ângulos de atitude

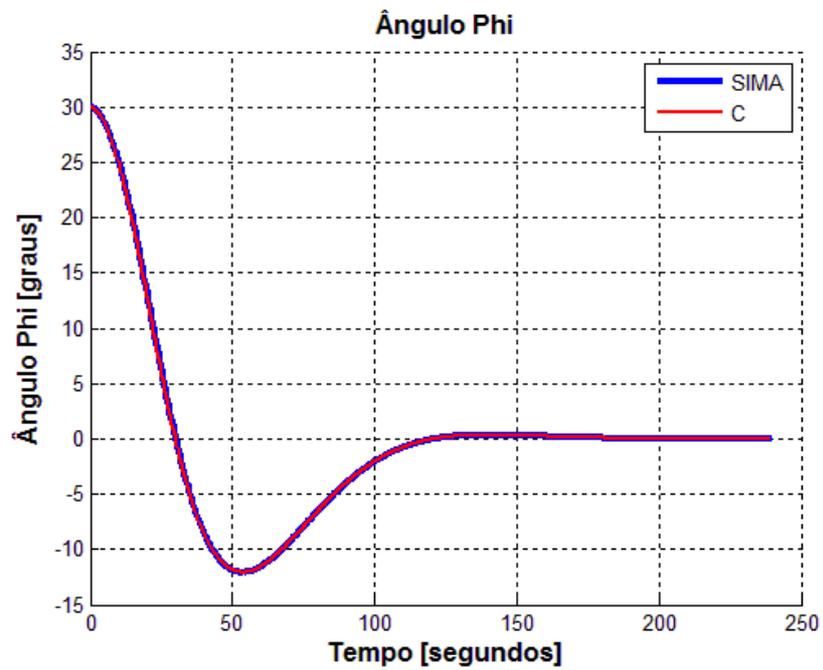


Figura 7.89 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

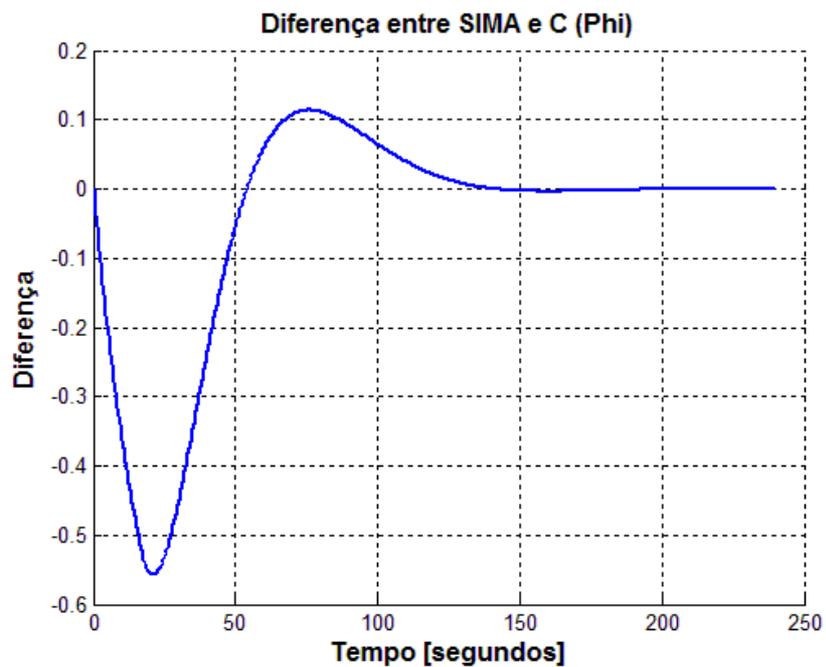


Figura 7.90- Diferença das curvas dos programas simulados em SIMA e linguagem C

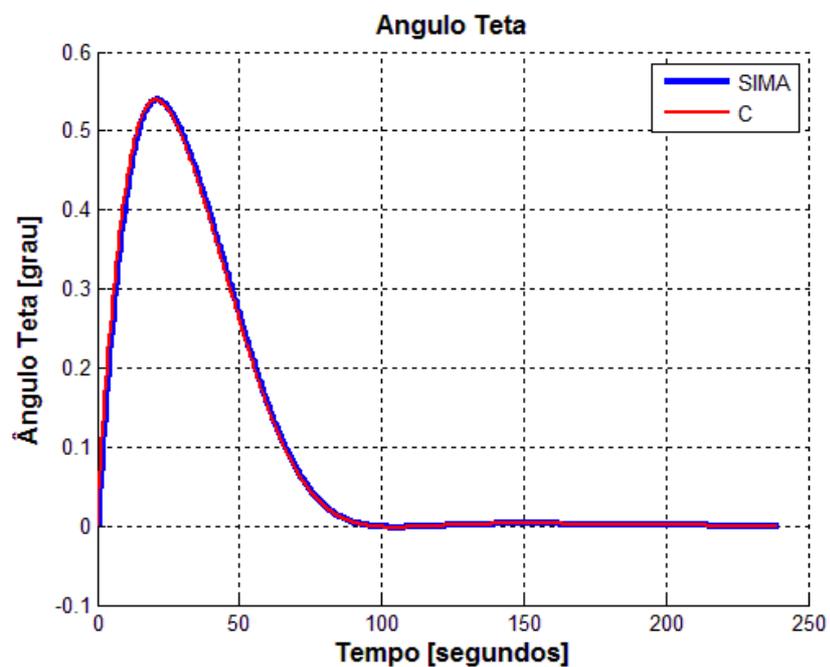


Figura 7.91 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

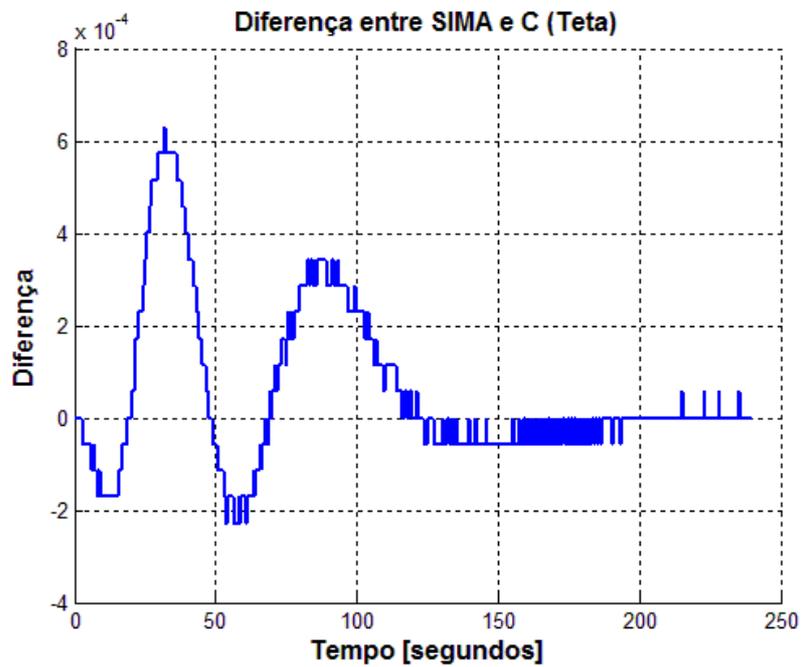


Figura 7.92- Diferença das curvas dos programas simulados em SIMA e linguagem C

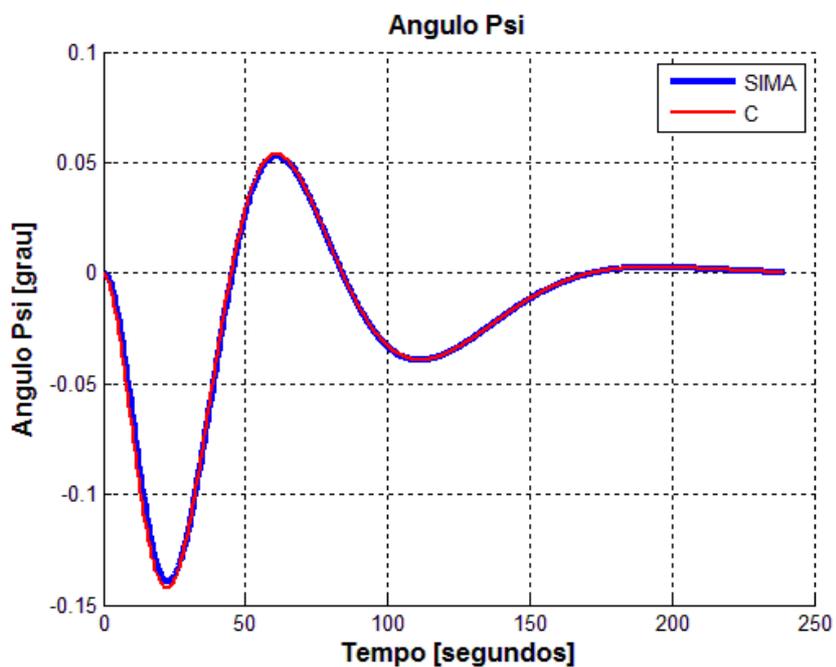


Figura 7.93 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

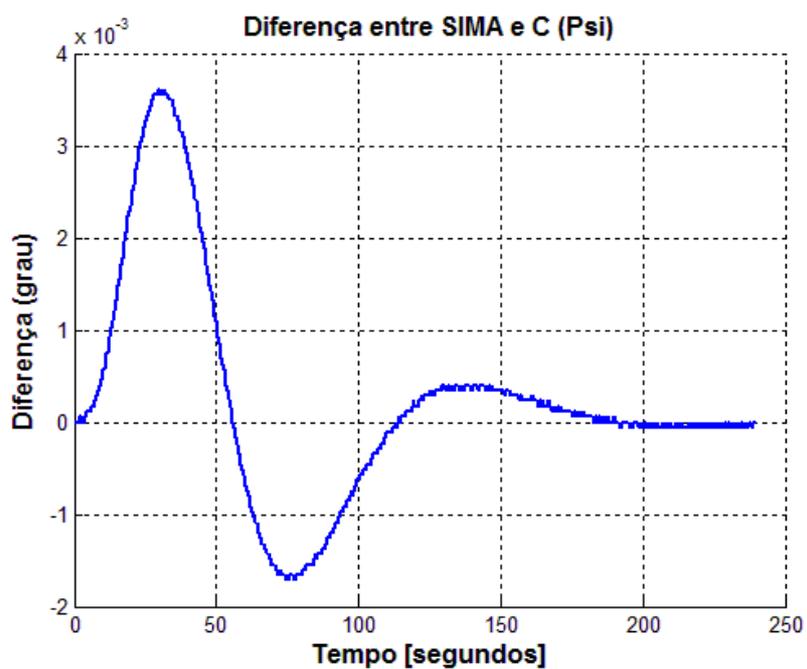


Figura 7.94- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.3.2. Velocidade angular da PMM

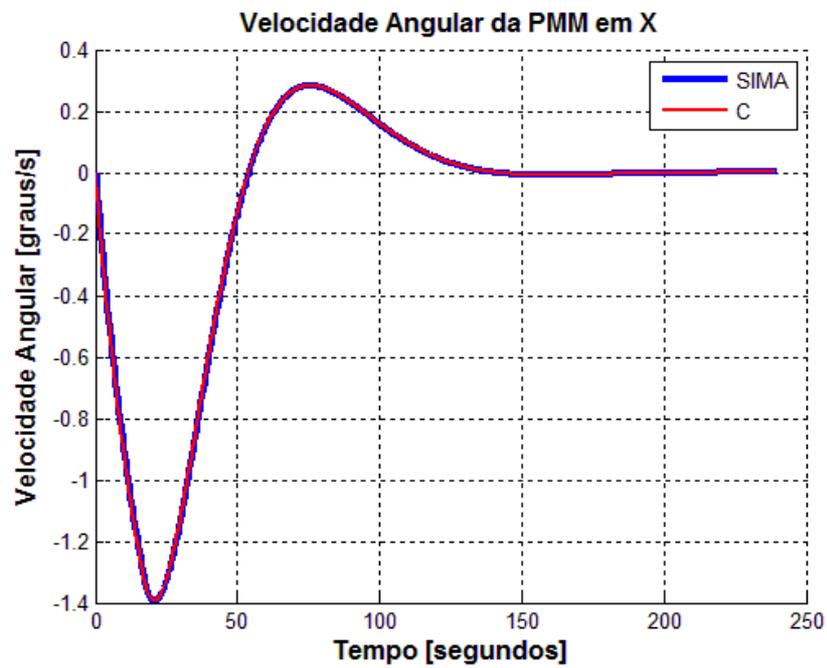


Figura 7.95 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

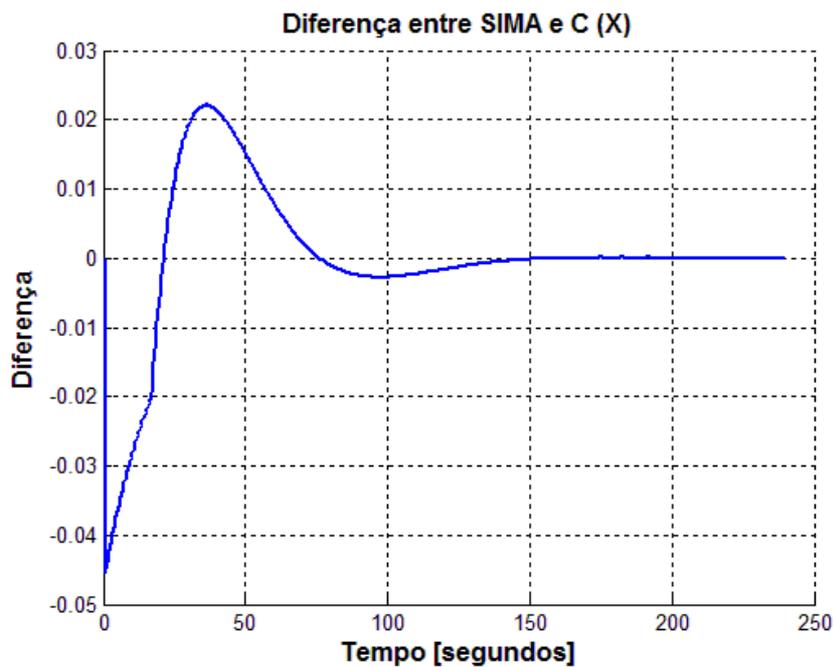


Figura 7.96- Diferença das curvas dos programas simulados em SIMA e linguagem C

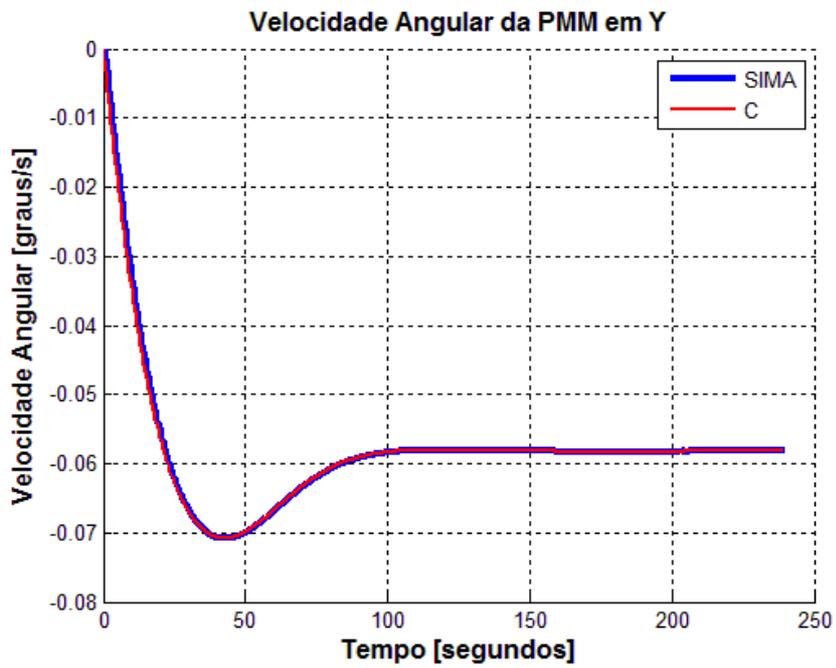


Figura 7.97 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

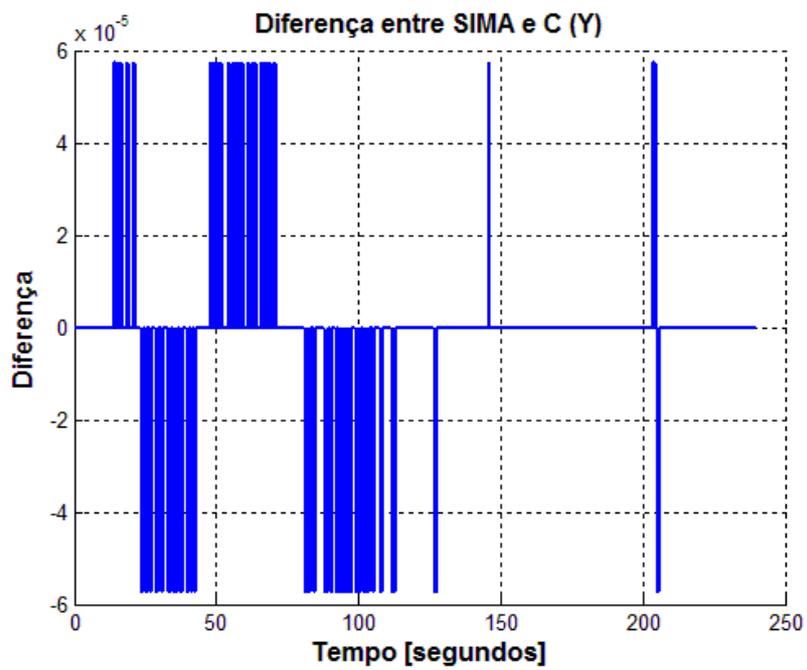


Figura 7.98 - Diferença das curvas dos programas simulados em SIMA e linguagem C

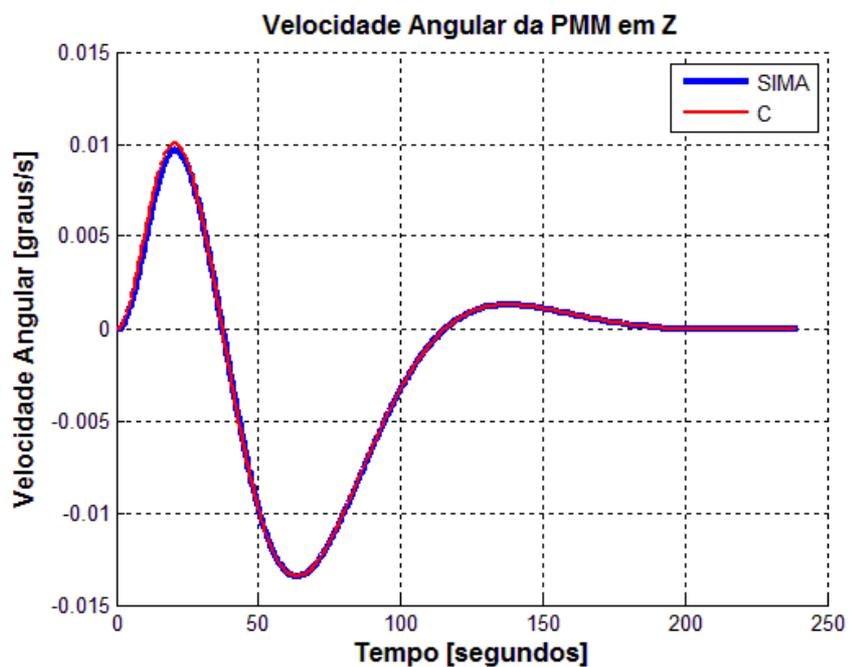


Figura 7.99 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

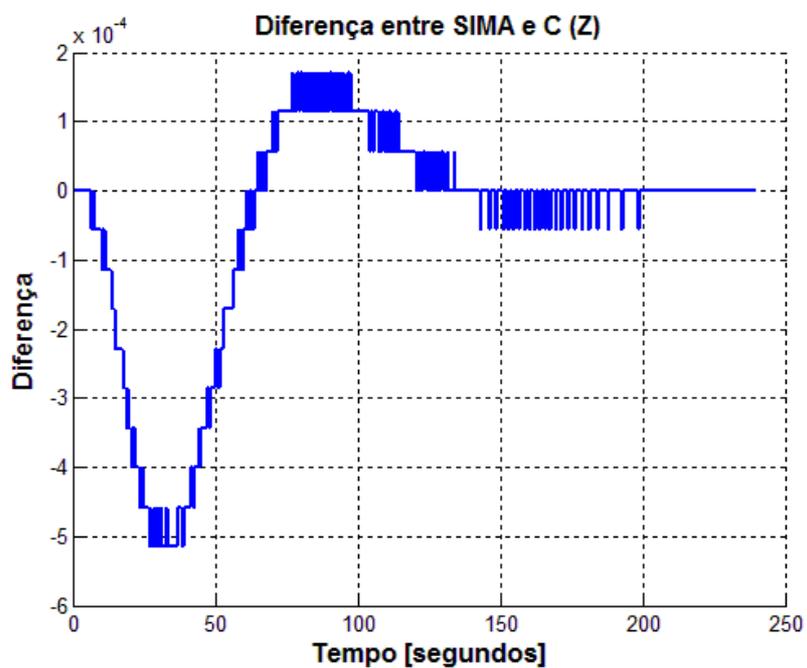


Figura 7.100- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.3.3. Velocidade angular das rodas

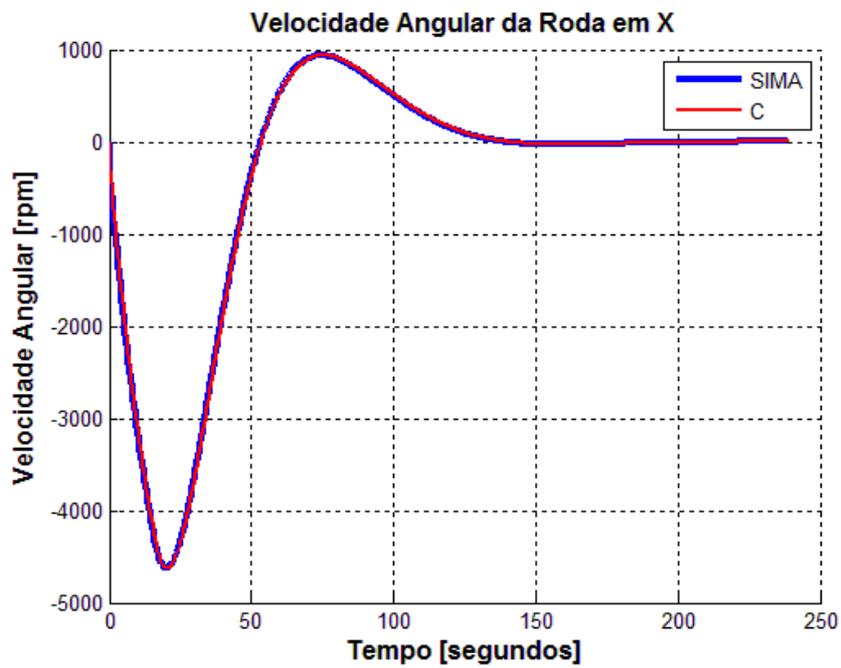


Figura 7.101 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

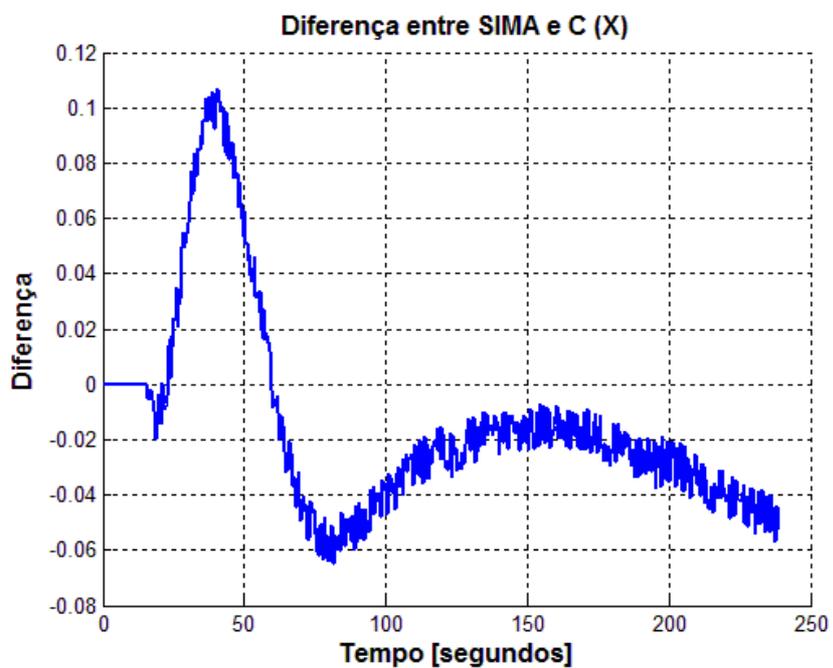


Figura 7.102 - Diferença das curvas dos programas simulados em SIMA e linguagem C

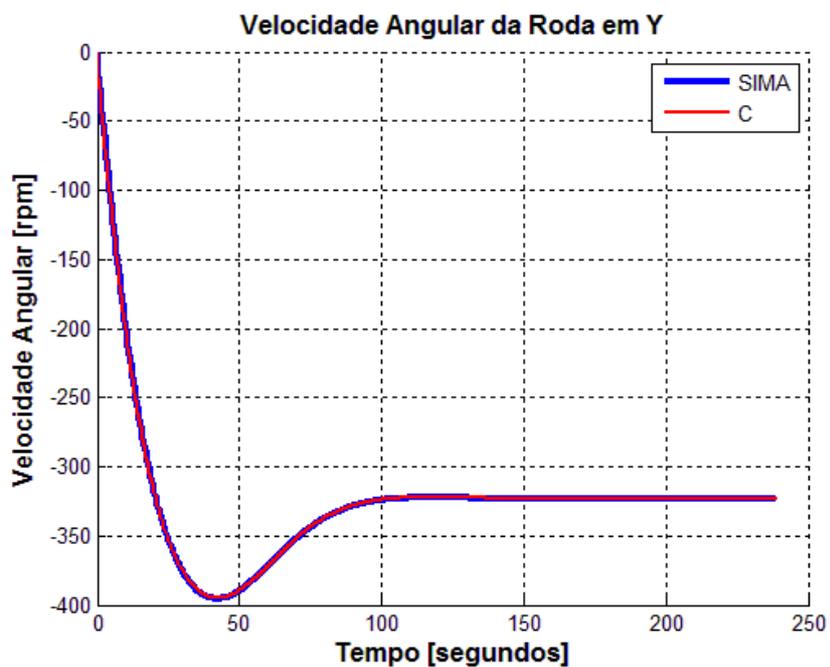


Figura 7.103 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

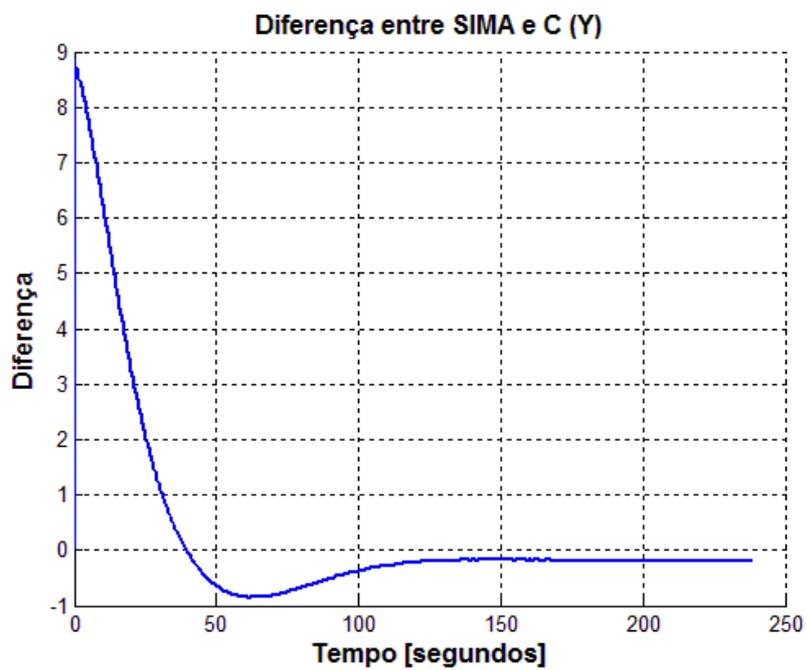


Figura 7.104 - Diferença das curvas dos programas simulados em SIMA e linguagem C

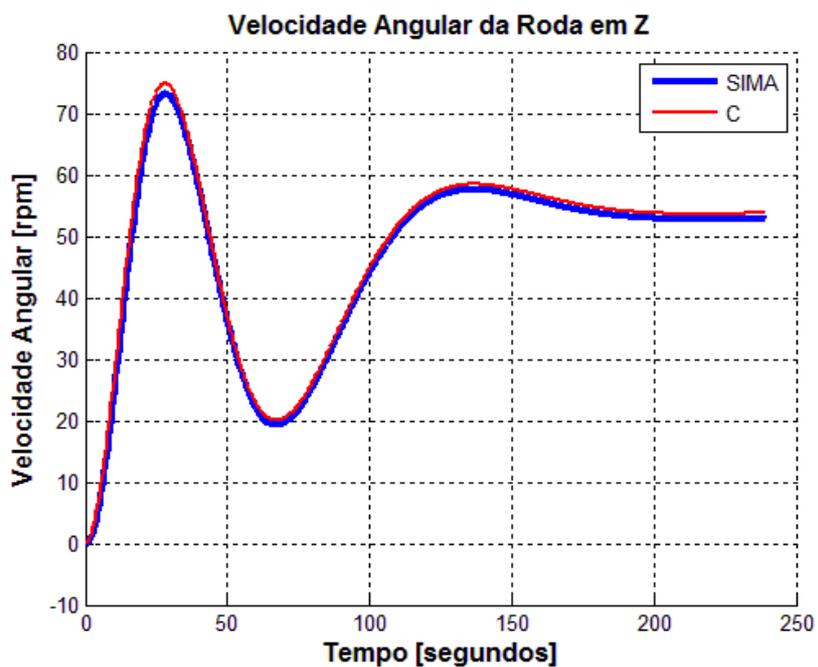


Figura 7.105 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

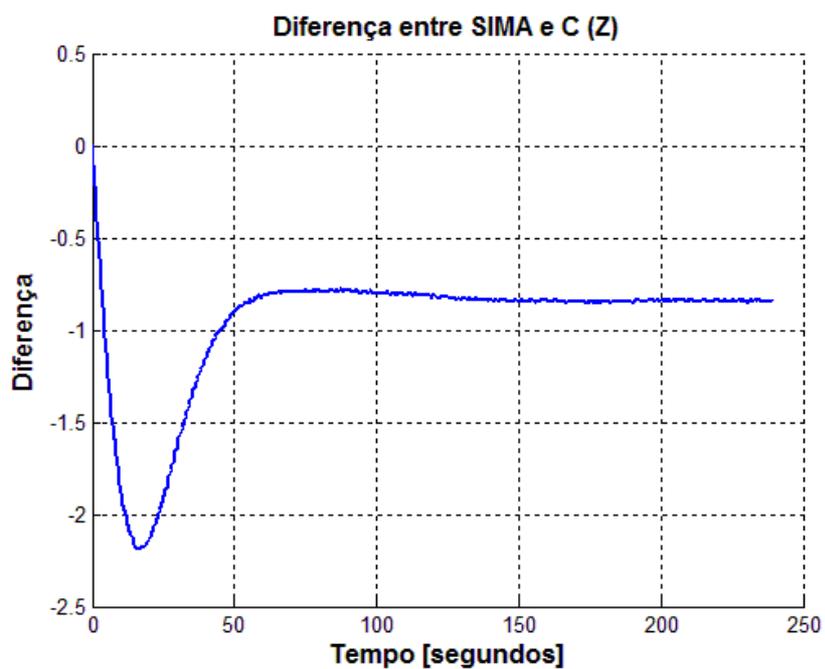


Figura 7.106- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.3.4. Tensão

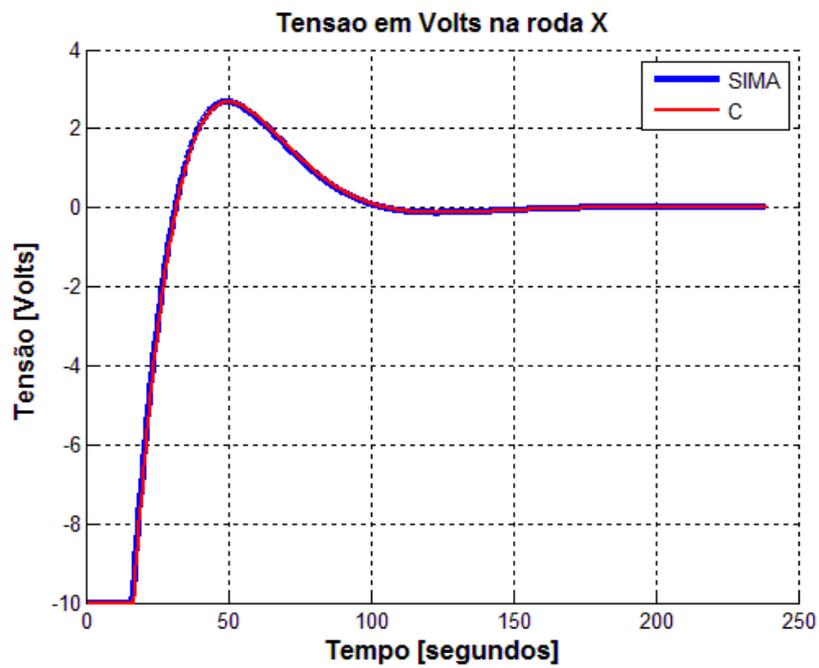


Figura 7.107 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

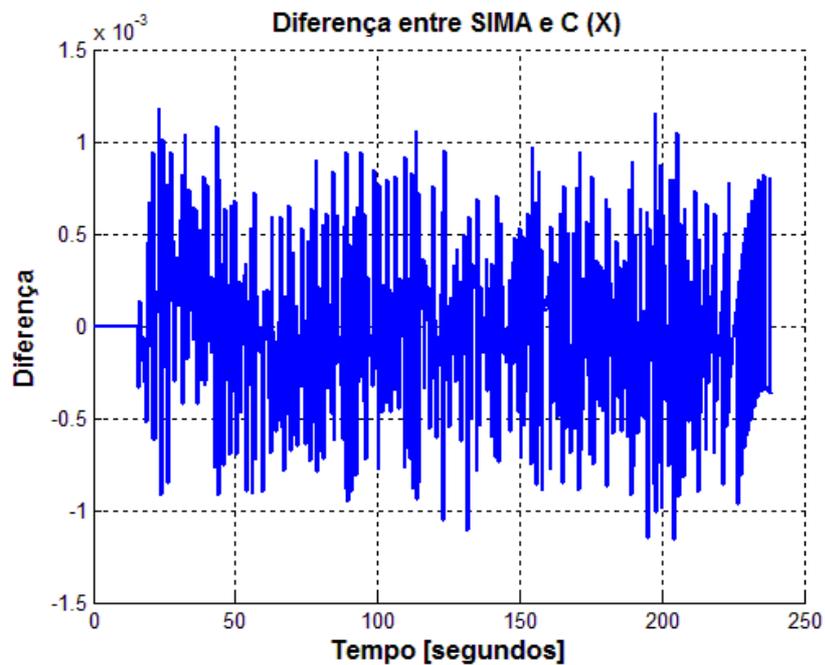


Figura 7.108- Diferença das curvas dos programas simulados em SIMA e linguagem C

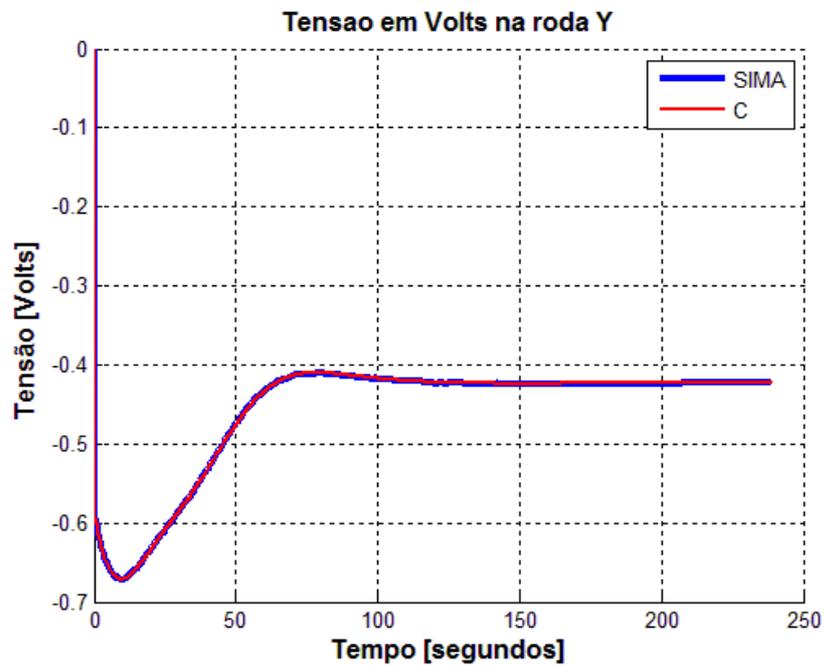


Figura 7.109 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

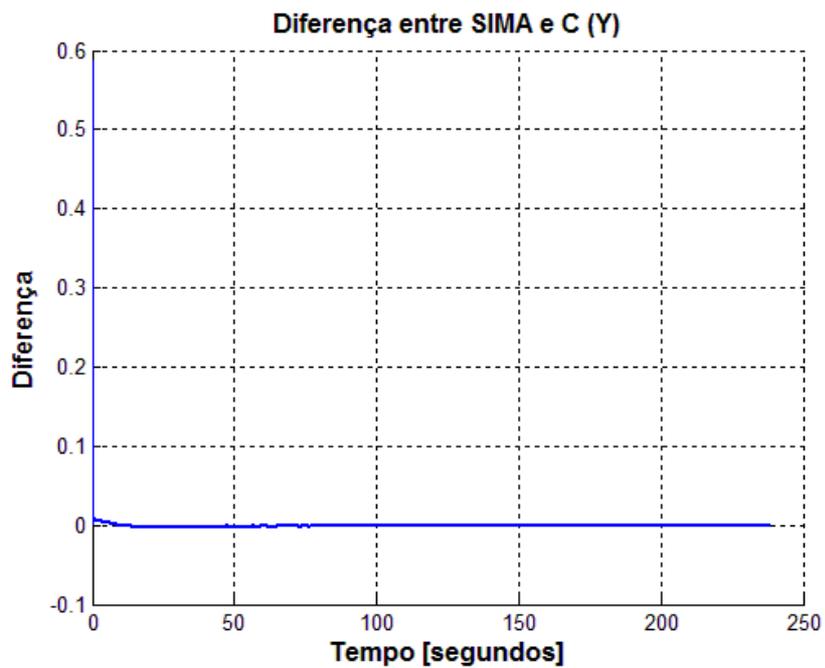


Figura 7.110- Diferença das curvas dos programas simulados em SIMA e linguagem C

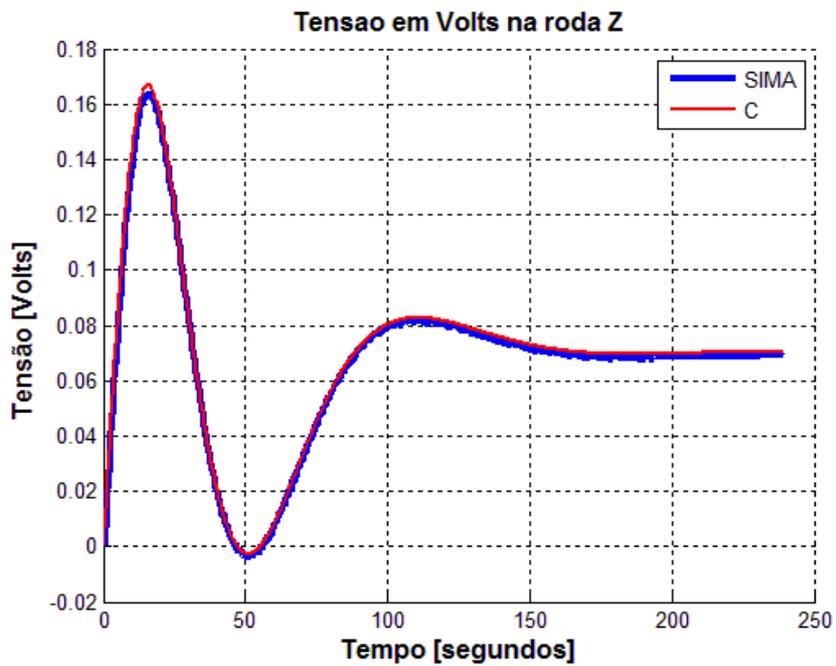


Figura 7.111 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

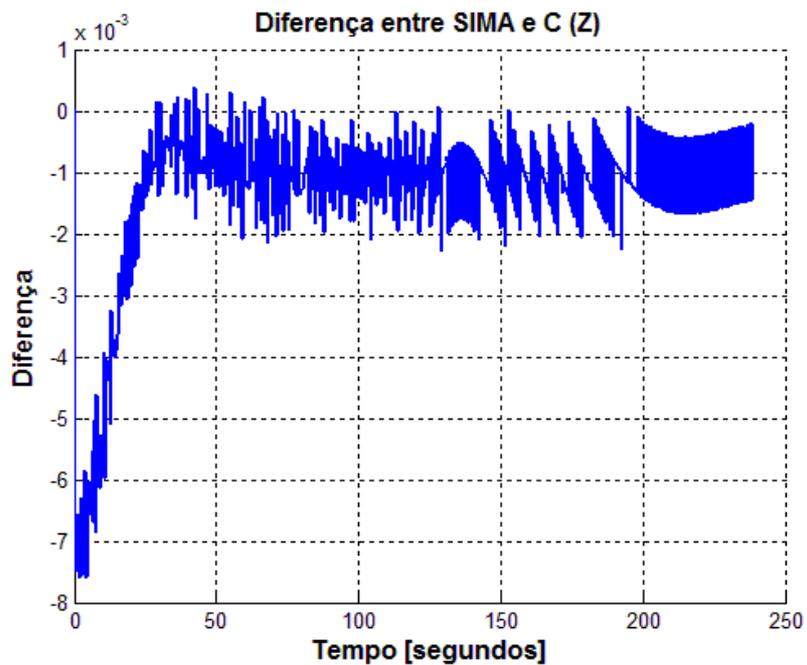


Figura 7.112 - Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.3.5. Torque da roda

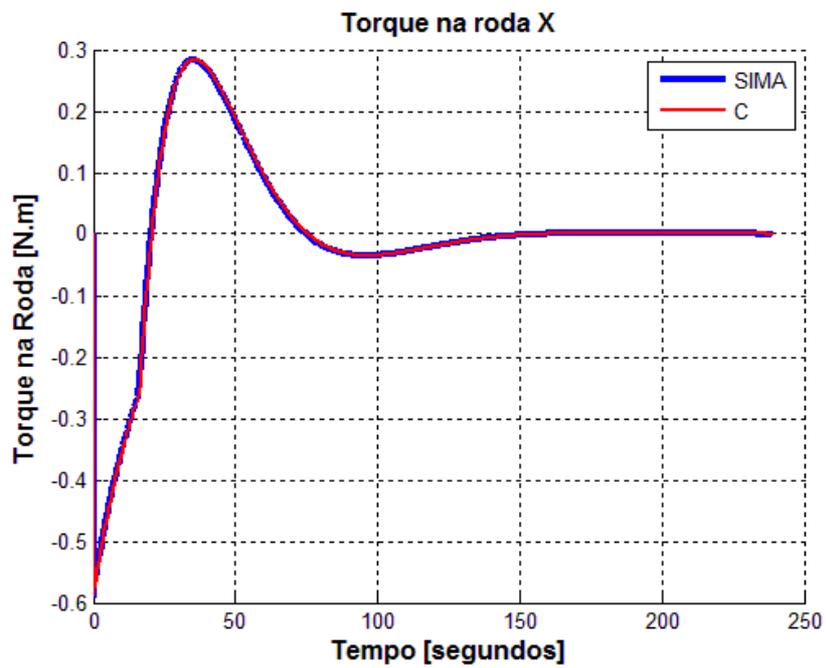


Figura 7.113 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

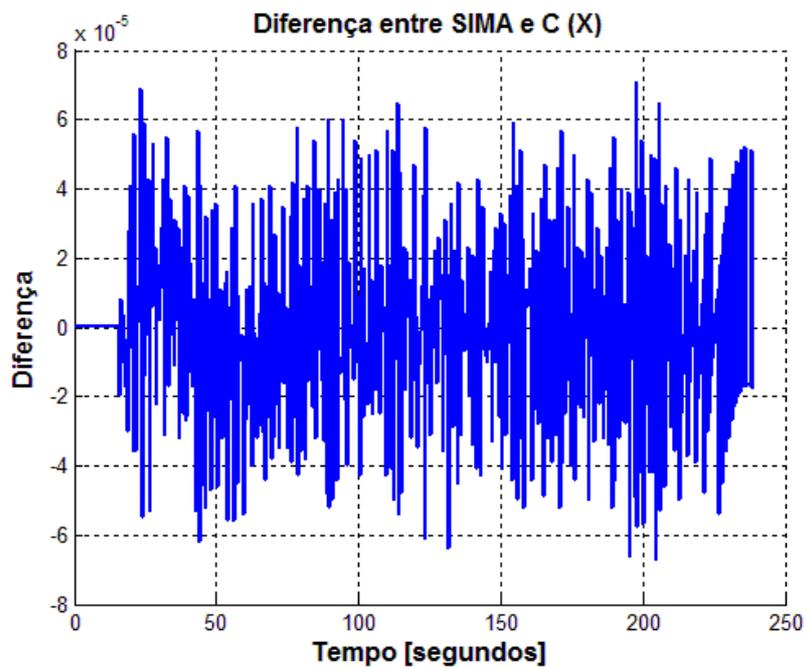


Figura 7.114- Diferença das curvas dos programas simulados em SIMA e linguagem C

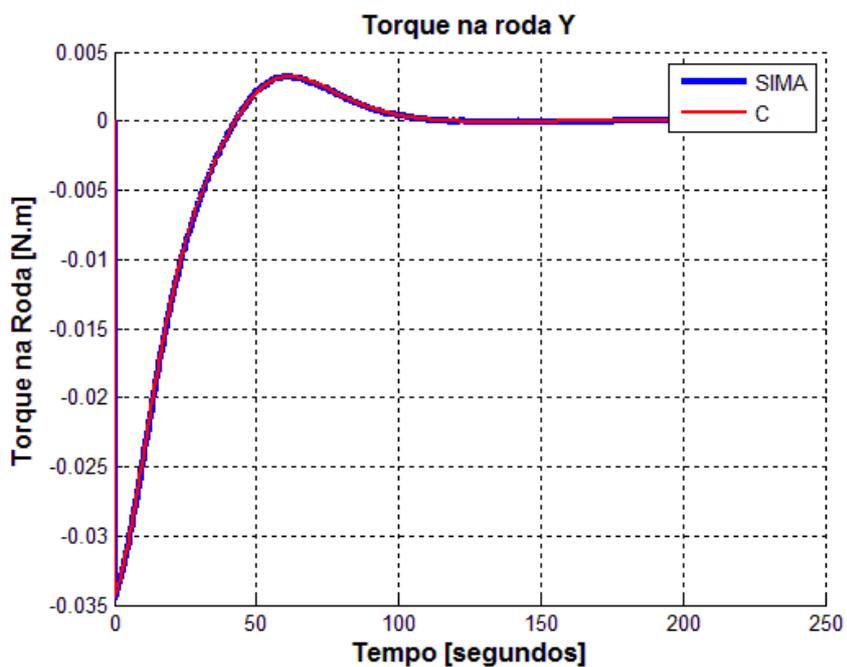


Figura 7.115 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

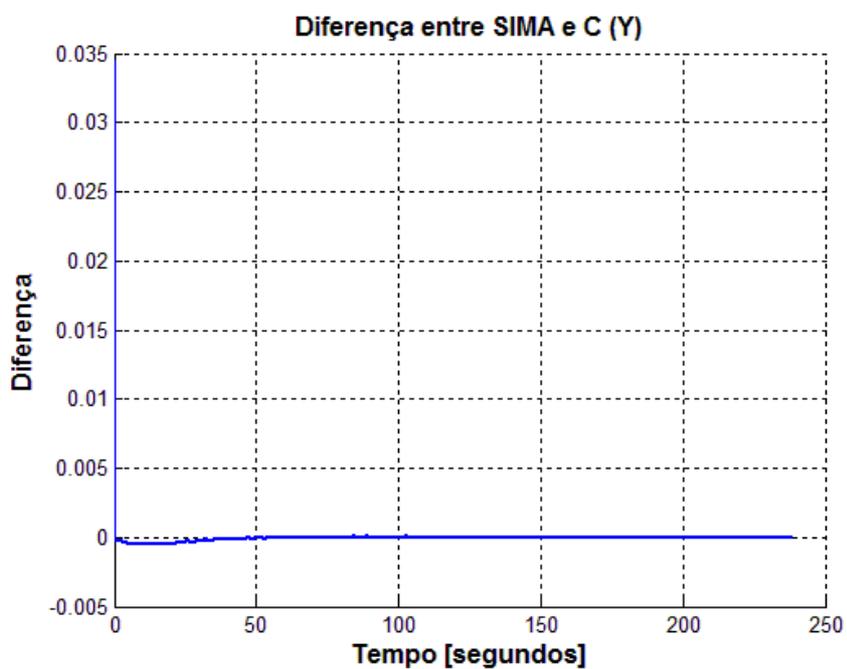


Figura 7.116- Diferença das curvas dos programas simulados em SIMA e linguagem C

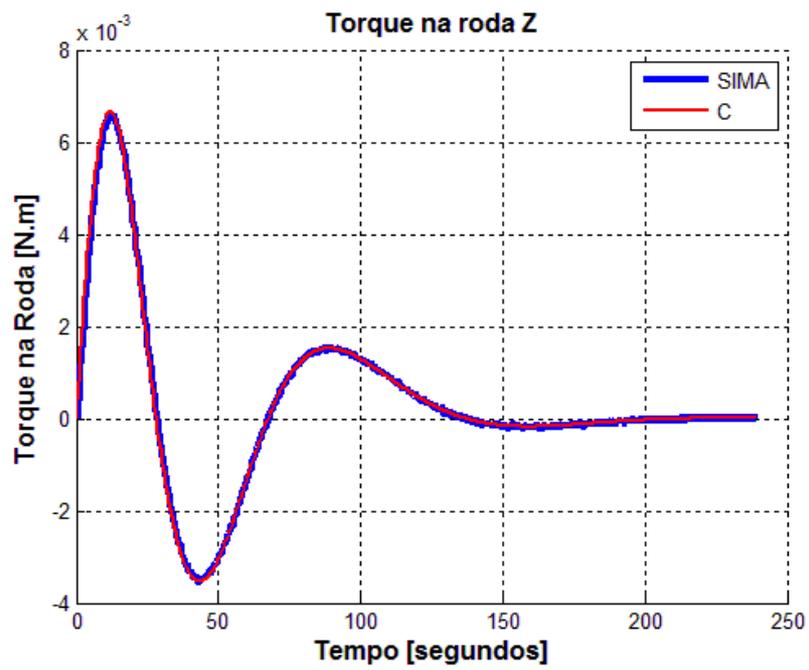


Figura 7.117 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

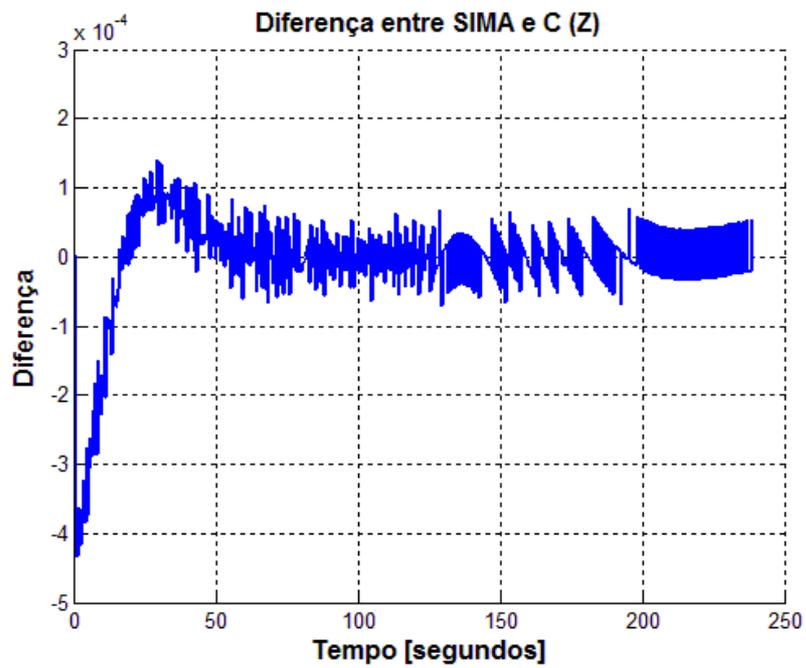


Figura 7.118- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.4. Caso 4: Simulação do Modo Nominal do SCA da PMM e do MMA divididos em 6 partições

A seguir são apresentados os gráficos obtidos a partir da simulação do SCA e do MMA no SIMA utilizando 6 partições. As partições foram organizadas da seguinte maneira, as partições de 1 a 4 alocaram o SCA e as partições 5 e 6 o MMA.

Os gráficos são ordenados da seguinte maneira, ângulos de atitude em cada eixo, velocidade angular da PMM, velocidade angular das rodas, tensão enviada a cada roda de reação e torque fornecido por roda.

Como já mencionado (7.2.1), por utilizarmos 6 partições o passo de integração (no SIMA e em linguagem C) foi alterado de 0.4 s (1 ciclo - com 4 partições), para 0.6 s para coincidir com o tempo de amostragem (1 ciclo – com 6 partições) de cada partição. Apesar desta alteração, nota-se que as curvas são muito próximas às curvas com passo de integração de 0.4 s.

7.2.3.4.1. Ângulos de atitude

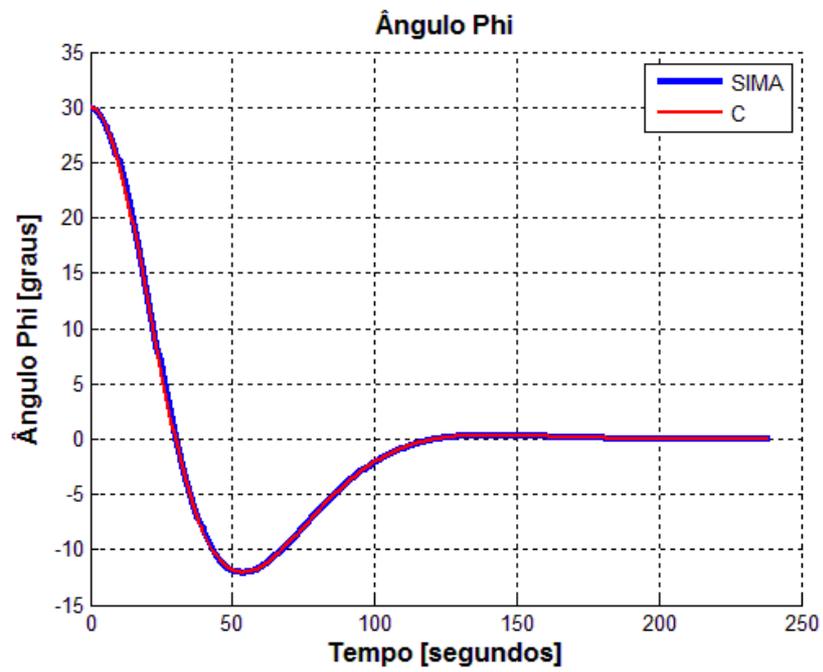


Figura 7.119 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

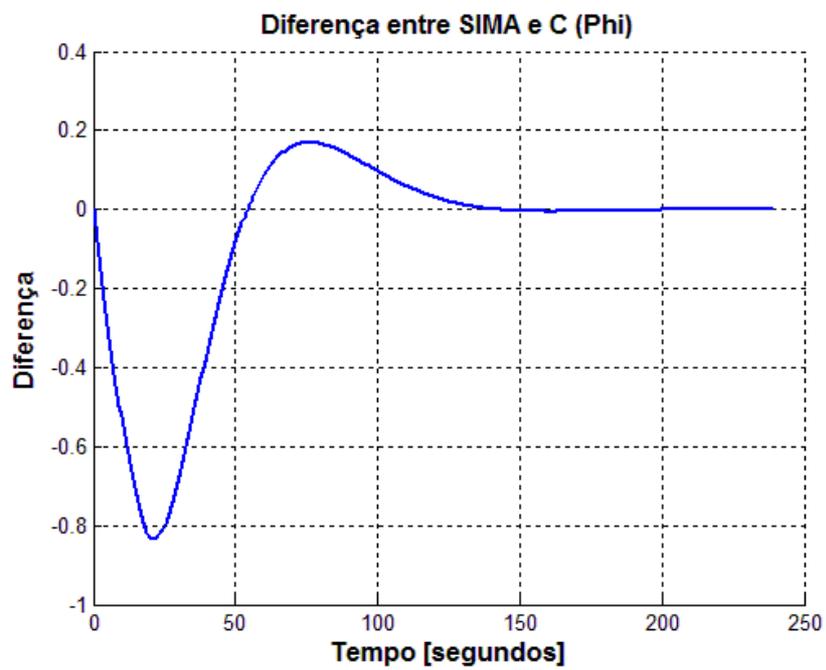


Figura 7.200- Diferença das curvas dos programas simulados em SIMA e linguagem C

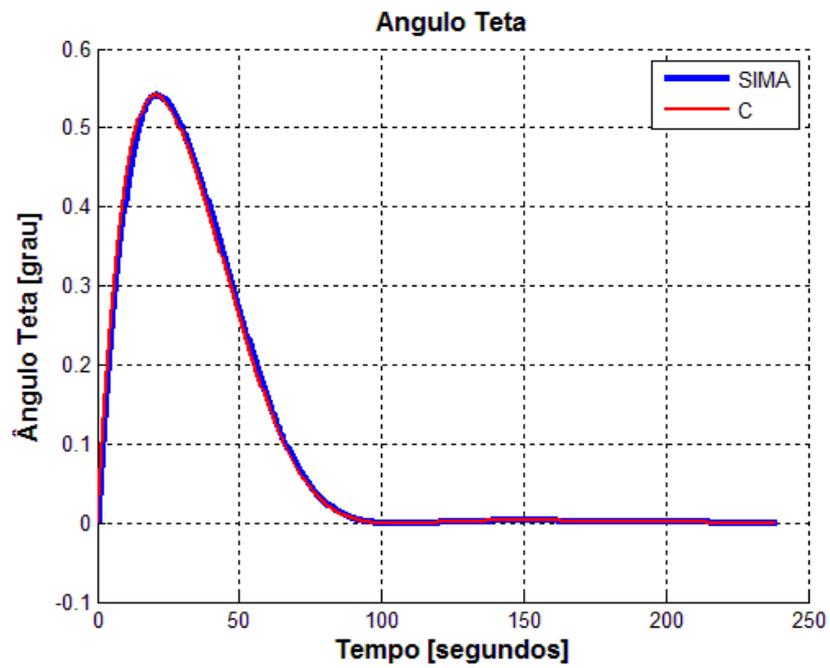


Figura 7.201 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

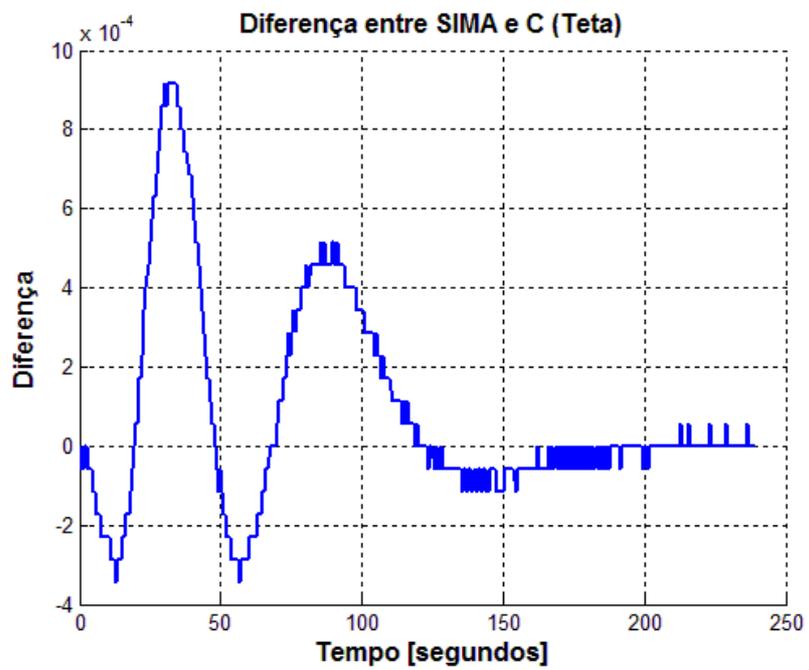


Figura 7.202- Diferença das curvas dos programas simulados em SIMA e linguagem C

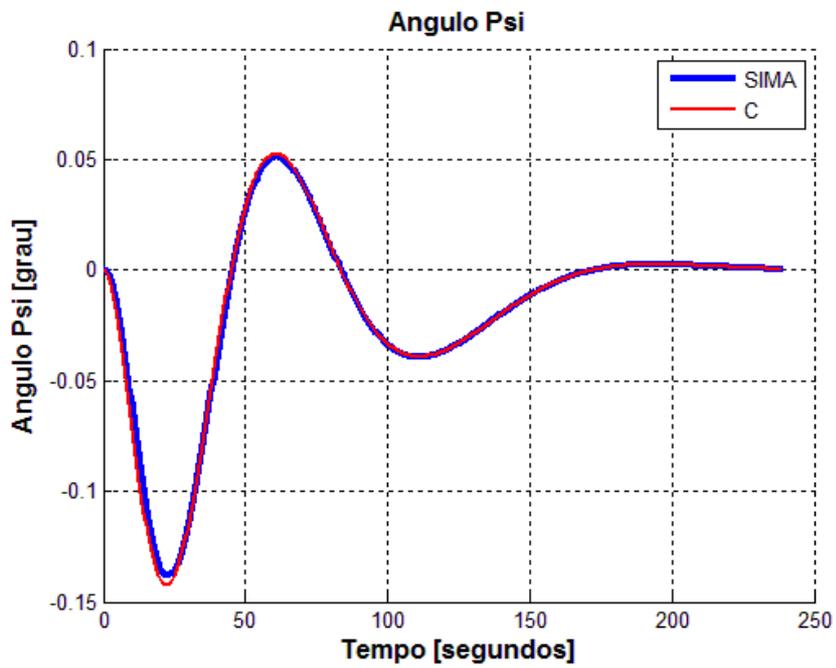


Figura 7.203 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

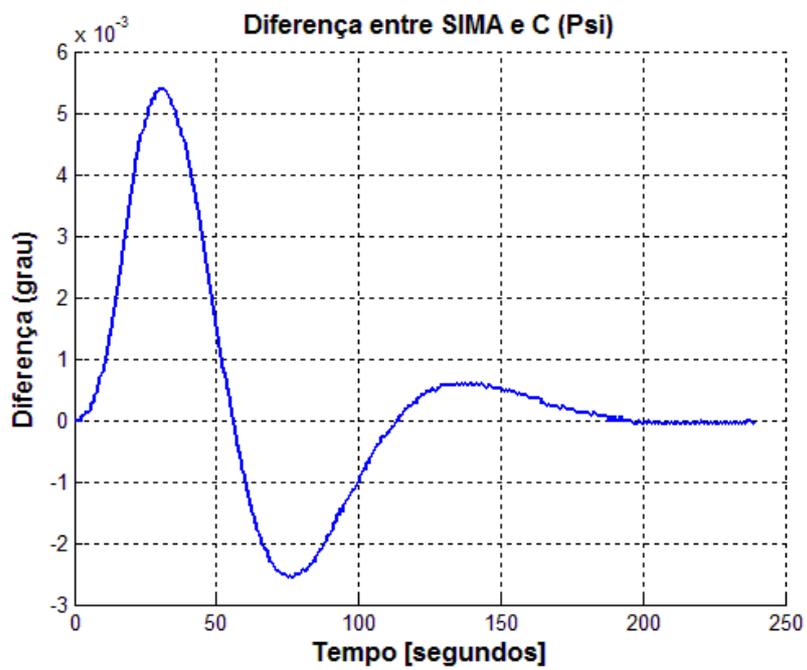


Figura 7.204- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.4.2. Velocidade angular da PMM

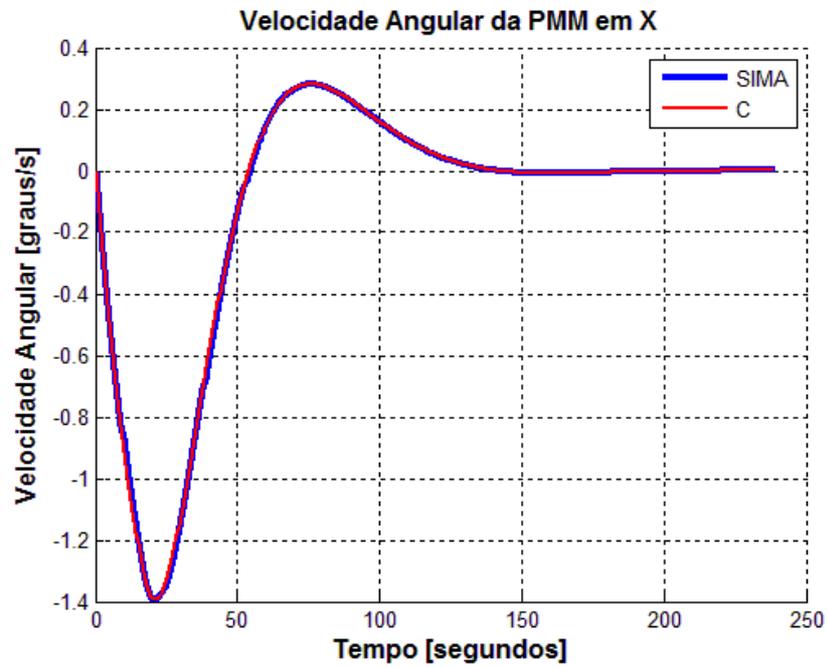


Figura 7.205 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

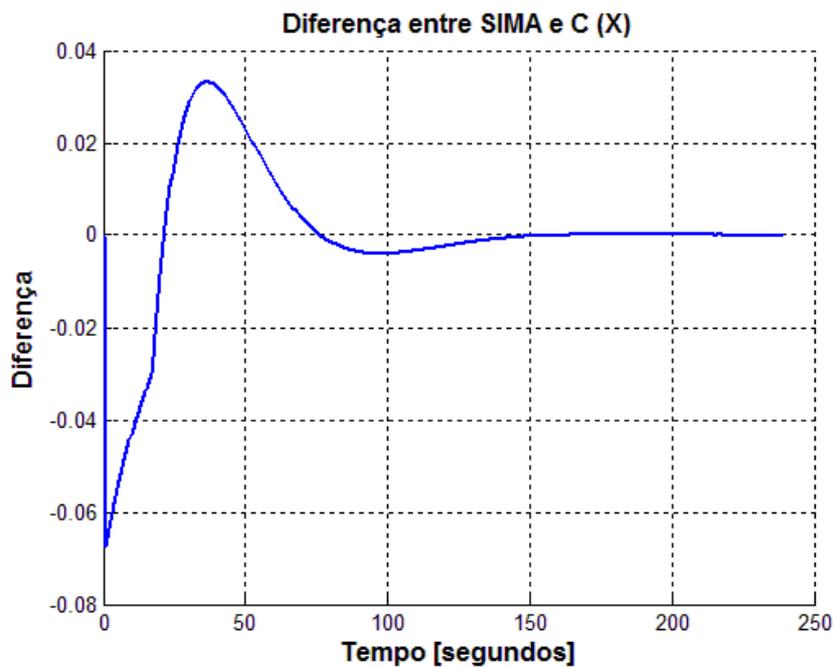


Figura 7.206- Diferença das curvas dos programas simulados em SIMA e linguagem C

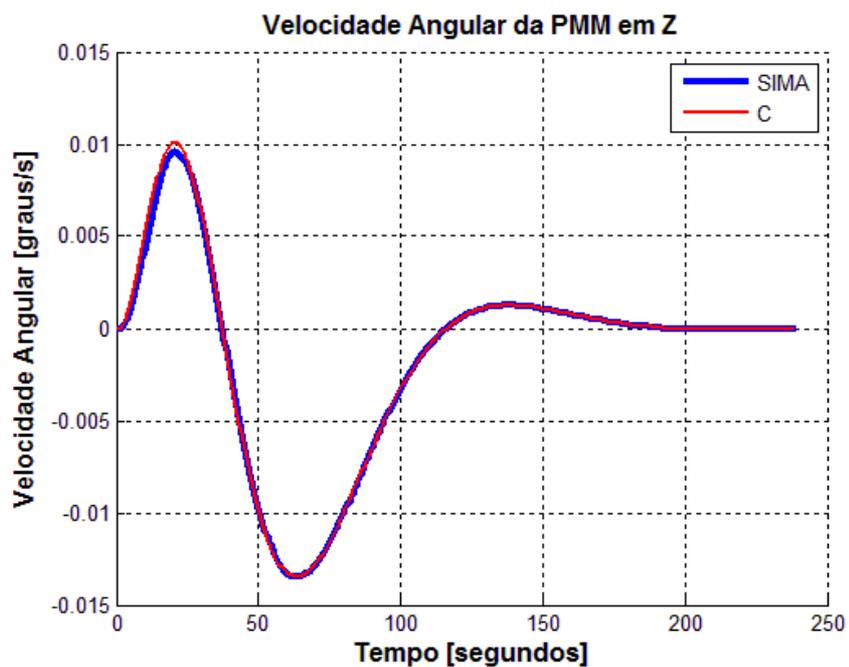


Figura 7.207 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

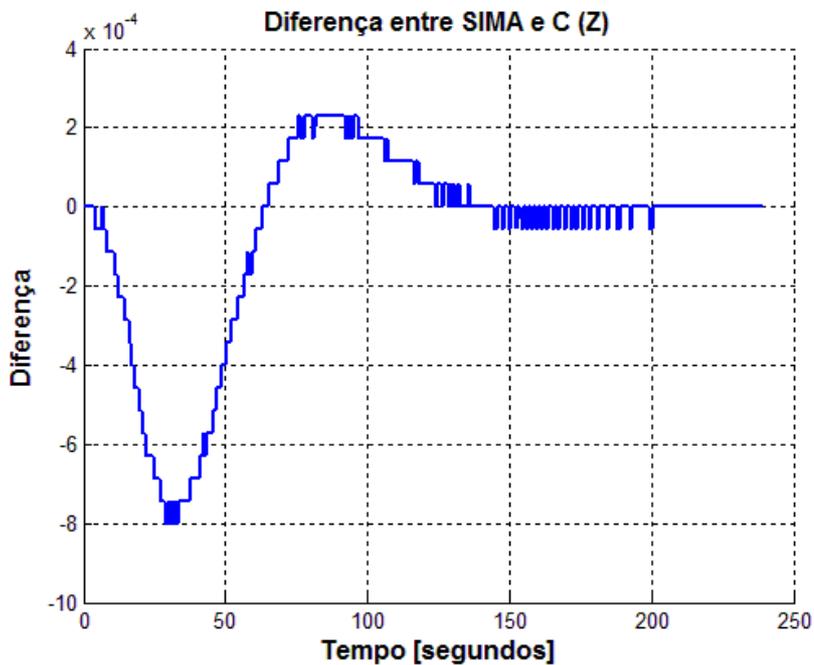


Figura 7.208- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.4.3. Velocidade angular das rodas

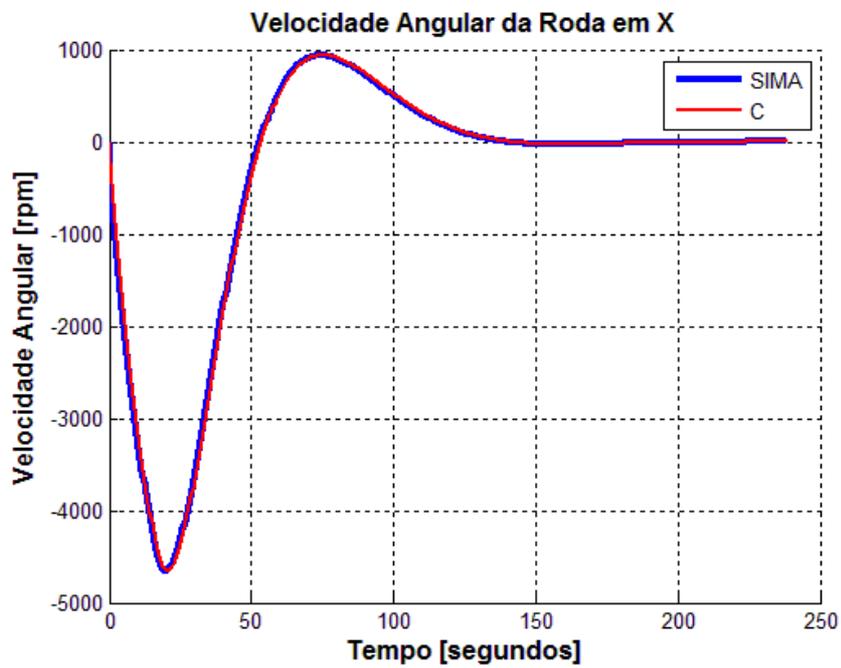


Figura 7.209 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

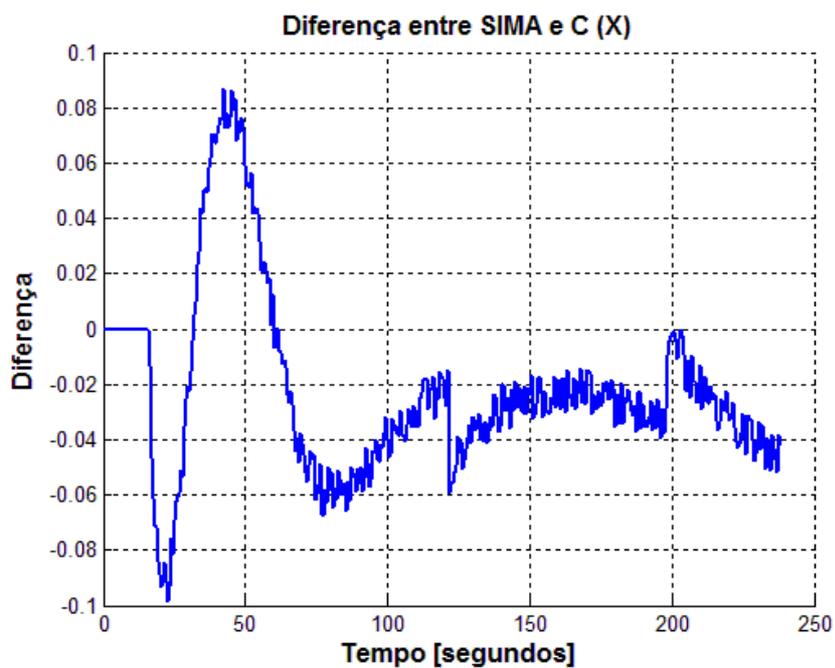


Figura 7.210- Diferença das curvas dos programas simulados em SIMA e linguagem C

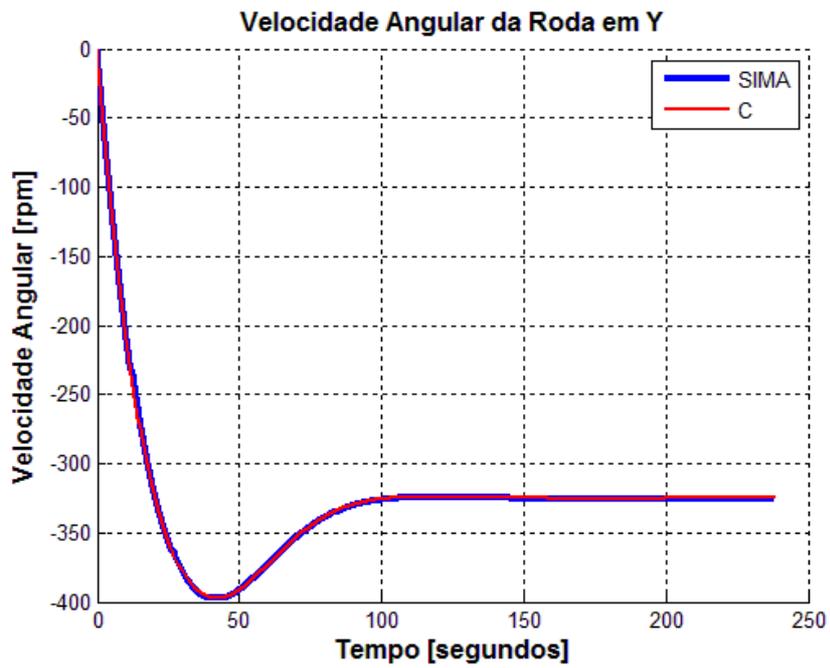


Figura 7.211 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

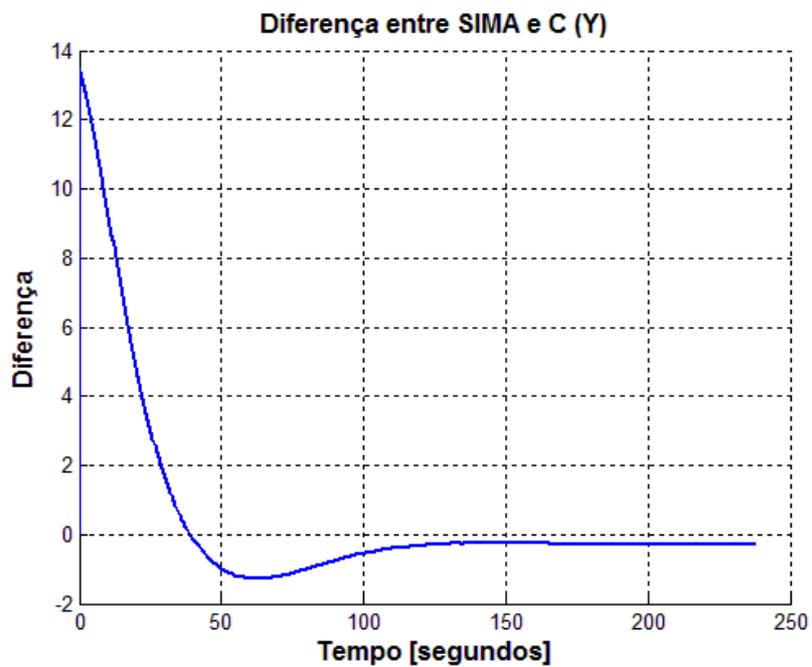


Figura 7.212 - Diferença das curvas dos programas simulados em SIMA e linguagem C

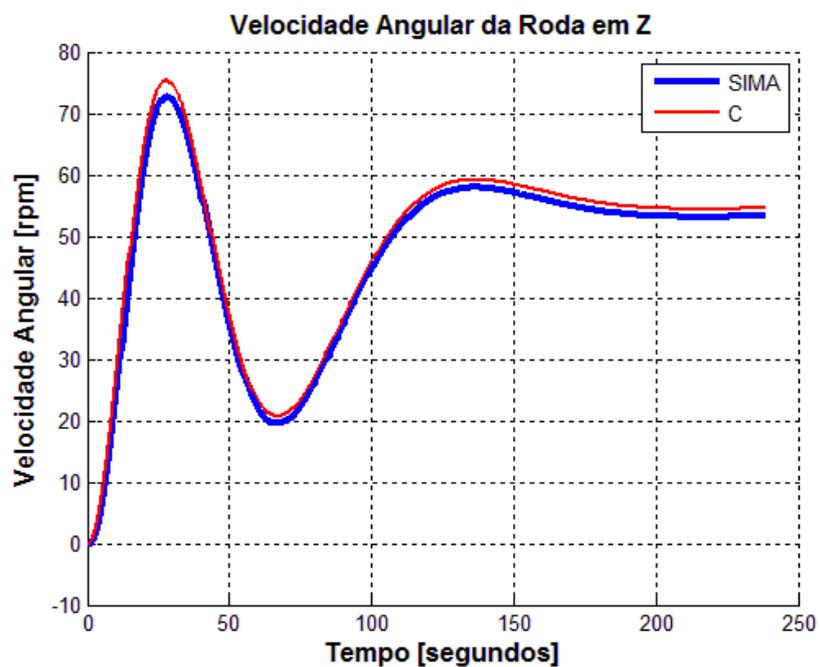


Figura 7.213 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

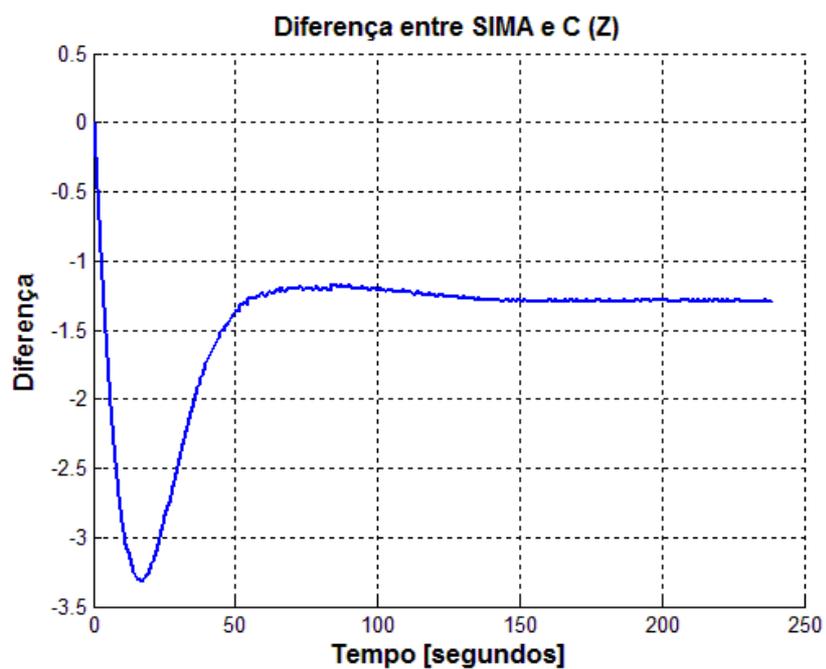


Figura 7.214- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.4.4. Tensão

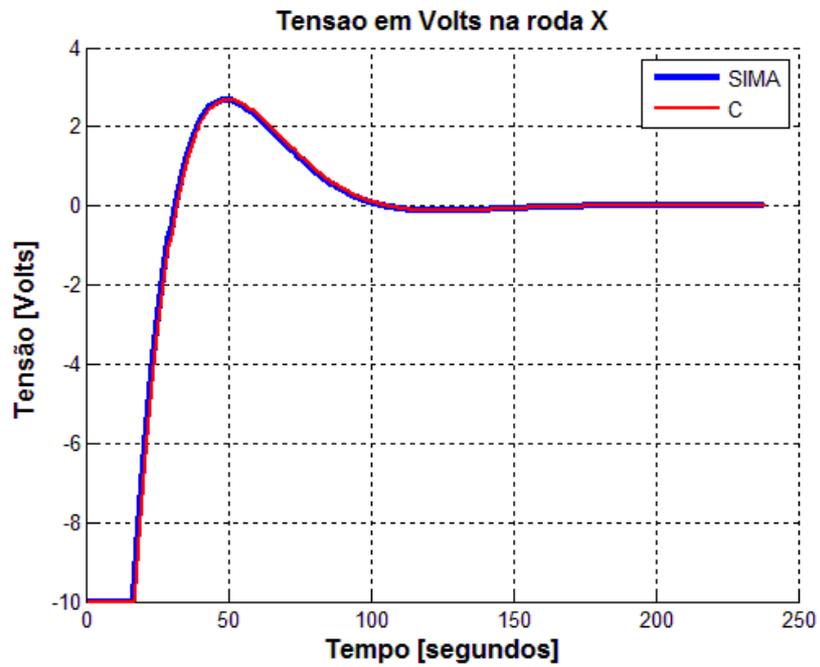


Figura 7.215 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

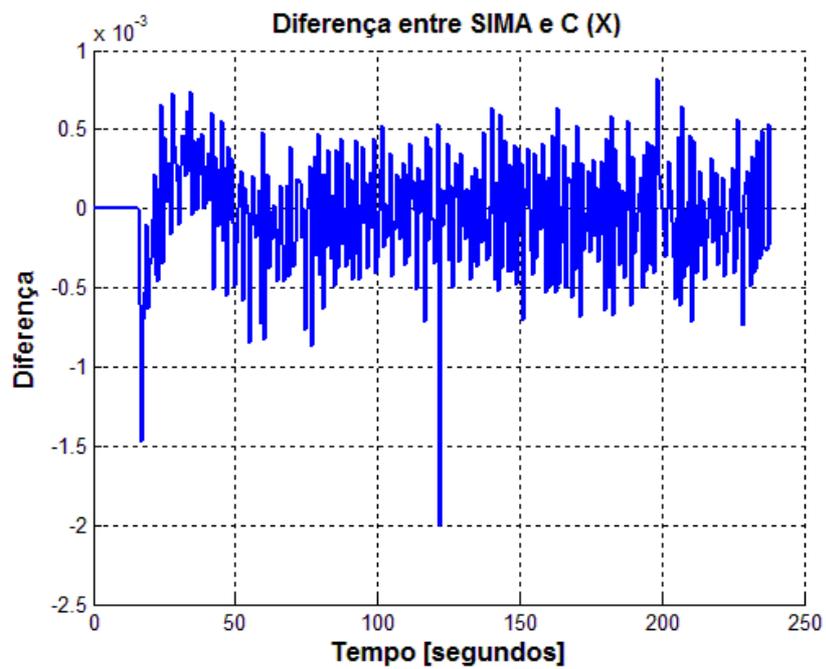


Figura 7.216- Diferença das curvas dos programas simulados em SIMA e linguagem C

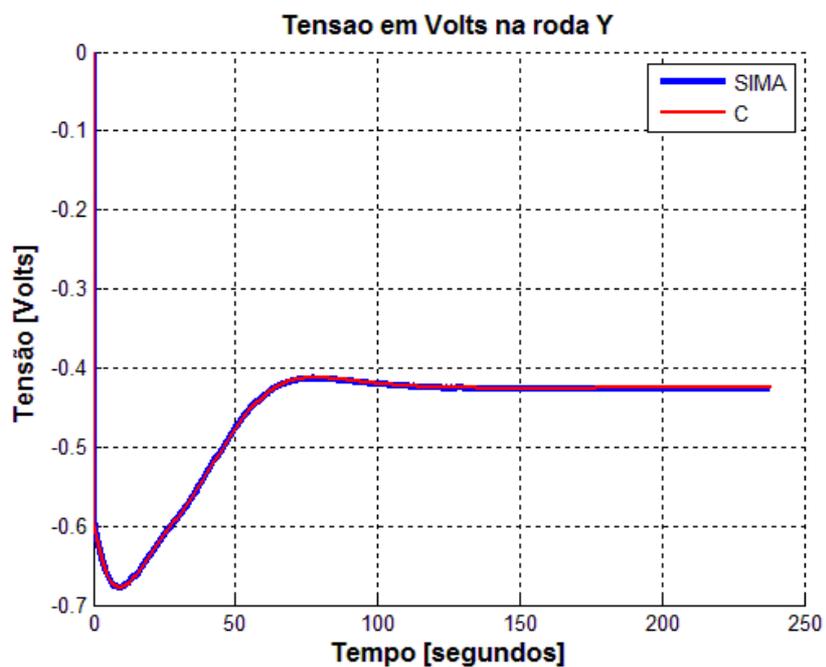


Figura 7.217 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

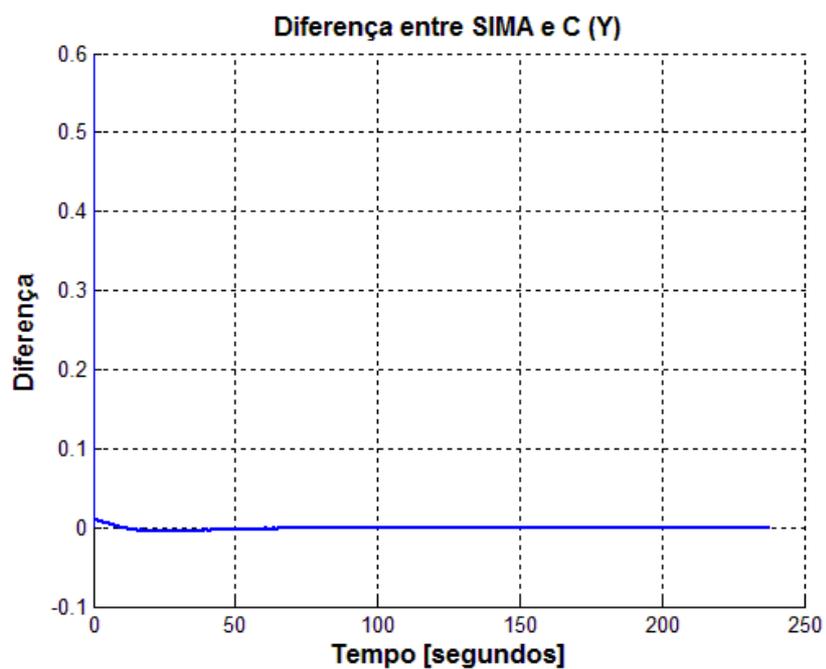


Figura 7.218- Diferença das curvas dos programas simulados em SIMA e linguagem C

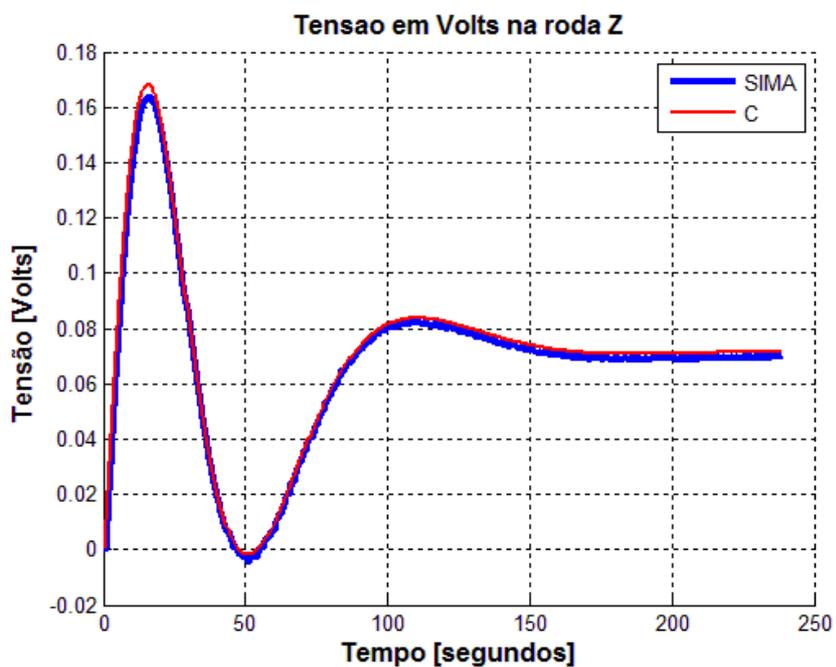


Figura 7.219 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

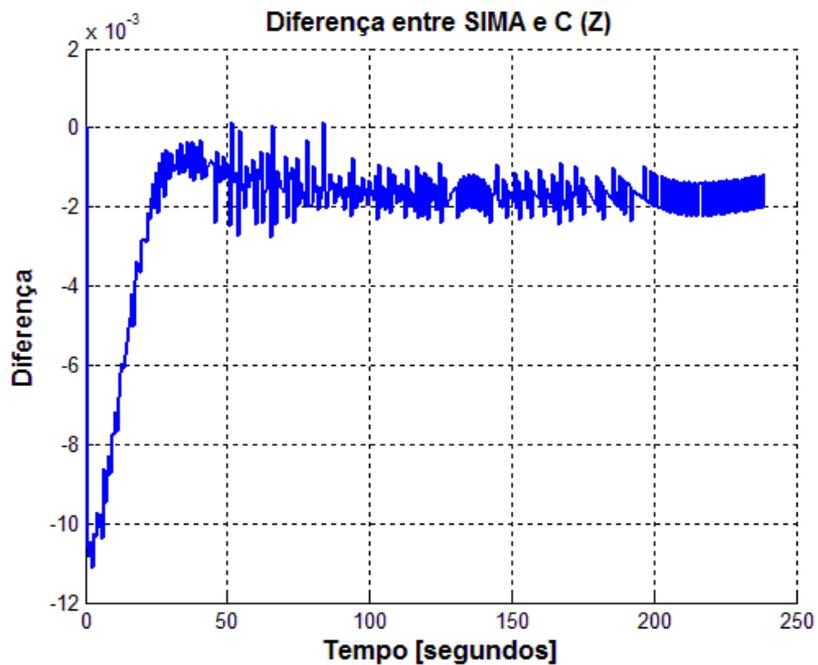


Figura 7.220- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.4.5. Torque da roda

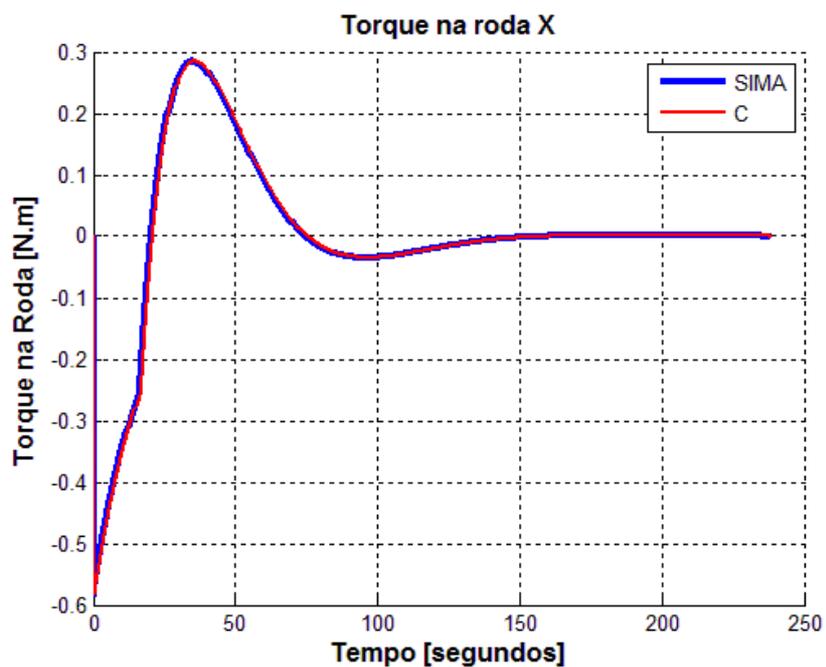


Figura 7.221 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

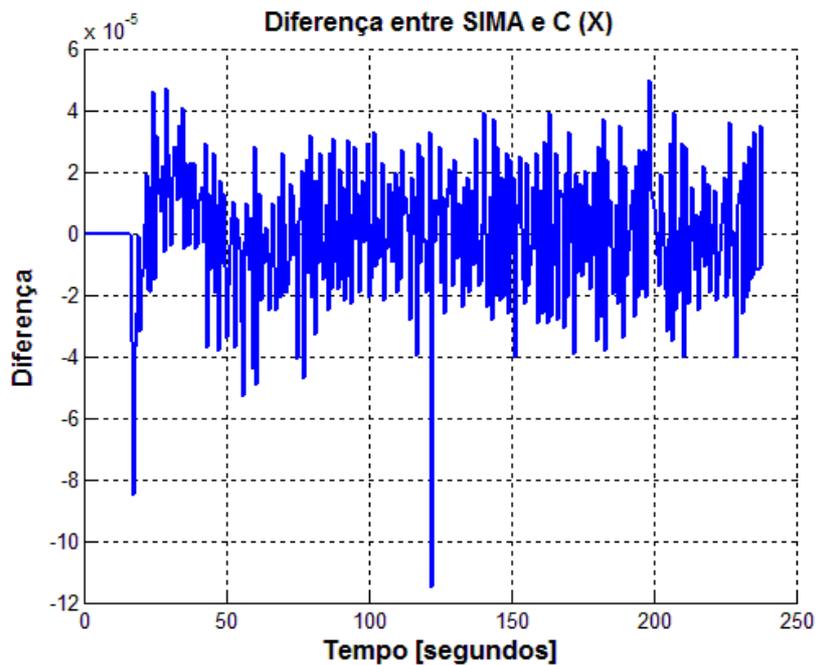


Figura 7.222- Diferença das curvas dos programas simulados em SIMA e linguagem C

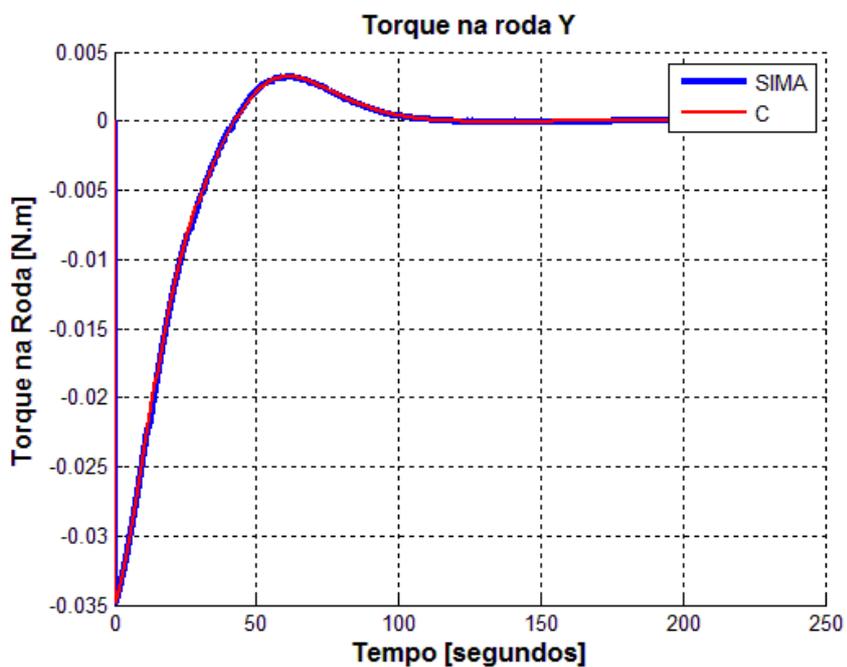


Figura 7.223 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

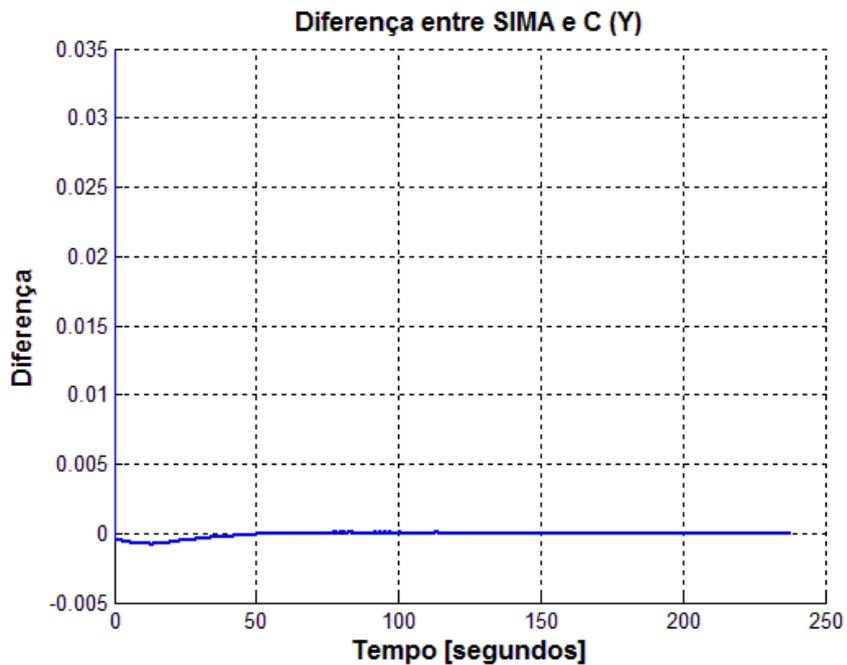


Figura 7.224- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.5. Caso 5: Simulação do Modo Nominal do SCA da PMM e do MMA divididos em 6 partições com falha inserida

Como já apresentado, uma das grandes vantagens do IMA é o seu particionamento robusto. Portanto, com intuito de testar e demonstrar essa robustez utilizou-se a simulação anterior com seis partições e inseriu-se uma falha em uma das partições do MMA. Esta falha ocasionou a interrupção dessa partição e, conseqüentemente, seu desligamento. Apesar disso, as outras cinco partições continuaram em seu funcionamento normalmente.

Como já foi dito anteriormente, o SIMA é um ambiente para desenvolvedores, portanto, seu ambiente não é gráfico. Desta forma, a única maneira de se apresentar os resultados é plotando-se gráficos. Apesar disso, julgou-se conveniente apresentar aqui o *print-screen* de algumas telas do programa em execução para ilustrar o resultado da falha inserida.

A figura 7.225 apresenta um *print-screen* da tela de execução antes da falha ocorrer:

```

Aplicativos Locais Sistema
root@gitsuzo-desktop: /usr/sima
Arquivo Editar Ver Terminal Ajuda

1 - controlador
-----
Recebeu Fix = -0.042395
Recebeu Fiy = 0.000005
Recebeu Fiz = -0.000539
Sending ux'0.162723'
Sending uy'0.418872'
Sending uz'0.079038'
Recebeu Fix = -0.040606
Recebeu Fiy = 0.000001
Recebeu Fiz = -0.000553
Sending ux'0.147012'
Sending uy'0.419594'
Sending uz'0.079177'

2 - atuador
-----
Recebeu ux = 0.162723
Recebeu uy = -0.418872
Recebeu uz = 0.079039
Sending Mx'0.034905'
Sending My'0.000541'
Sending Mz'0.001366'
Recebeu ux = 0.147012
Recebeu uy = -0.419594
Recebeu uz = 0.079178
Sending Mx'0.034788'
Sending My'0.000483'
Sending Mz'0.001333'

3 - planta
-----
Recebeu Mx = -0.034905
Recebeu My = 0.000541
Recebeu Mz = 0.001367
Sending Fix'0.040606'
Sending Fiy'0.000001'
Sending Fiz'0.000553'
Recebeu Mx = -0.034788
Recebeu My = 0.000483
Recebeu Mz = 0.001334
Sending Fix'0.038859'
Sending Fiy'0.000002'
Sending Fiz'0.000566'

4 - sensor
-----
Recebeu Fix = -0.040606
Recebeu Fiy = 0.000001
Recebeu Fiz = -0.000553
Sending fix'0.040606'
Sending fiy'0.000001'
Sending fiz'0.000553'
Recebeu Fix = -0.038859
Recebeu Fiy = -0.000002
Recebeu Fiz = -0.000566
Sending fix'0.038859'
Sending fiy'0.000002'
Sending fiz'0.000566'

5 - extra
-----
Sending u '0.506877'
Sending u '0.511747'
Sending u '0.516618'
Sending u '0.521489'
Sending u '0.526359'
Sending u '0.531230'
Sending u '0.536101'
Sending u '0.540971'
Sending u '0.545842'
Sending u '0.550713'
Sending u '0.555583'
Sending u '0.560454'

6 - extra1
-----
Sending y'1.008011'
Recebeu u = 0.132754
Sending y'0.936185'
Recebeu u = 0.203323
Sending y'0.979130'
Recebeu u = -0.075439
Sending y'1.061237'
Recebeu u = -0.195285
Sending y'1.047245'
Recebeu u = 0.028466
Sending y'0.977240'
Recebeu u = 0.178906
ERROR!

0 - Test
-----
ESRCH
[0000099841600000] Switching to 0101 for 0.100000000 seconds; assigned to partition 'controlador' in operating mode NORMAL
[0000099941900000] Switching to 0201 for 0.100000000 seconds; assigned to partition 'atuador' in operating mode NORMAL
[0000100042100000] Switching to 0301 for 0.100000000 seconds; assigned to partition 'planta' in operating mode NORMAL
[0000100143600000] Switching to 0401 for 0.100000000 seconds; assigned to partition 'sensor' in operating mode NORMAL
[0000100243800000] Switching to 0501 for 0.100000000 seconds; assigned to partition 'extra' in operating mode NORMAL

```

Figura 7.225 - Tela com todas as partições funcionando

Nas Figuras 7.226 a 7.231, são apresentadas imagens mais aproximadas de cada partição:

```
1 - controlador
-----
Recebeu fix = 0.481158
Recebeu fiy = 0.005499
Recebeu fiz = -0.000550
Sending ux' -10.000000'
Sending uy' -0.668945'
Sending uz' 0.098275'
Recebeu fix = 0.473917
Recebeu fiy = 0.005869
Recebeu fiz = -0.000642
Sending ux' -10.000000'
Sending uy' -0.672134'
Sending uz' 0.105184'
```

Figura 7.226 - Partição 1

```
2 - atuador
-----
Recebeu ux = -10.000000
Recebeu uy = -0.668945
Recebeu uz = 0.098275
Sending Mx' -0.418606'
Sending My' -0.029319'
Sending Mz' 0.004995'
Recebeu ux = -10.000000
Recebeu uy = -0.672134
Recebeu uz = 0.105184
Sending Mx' -0.406234'
Sending My' -0.028638'
Sending Mz' 0.005250'
```

Figura 7.227 – Partição 2

```
3 - planta
-----
Recebeu Mx = -0.418606
Recebeu My = -0.029319
Recebeu Mz = 0.004995
Sending Fix'0.473917'
Sending Fiy'0.005869'
Sending Fiz'-0.000642'
Recebeu Mx = -0.406234
Recebeu My = -0.028638
Recebeu Mz = 0.005250
Sending Fix'0.466183'
Sending Fiy'0.006217'
Sending Fiz'-0.000738'
```

Figura 7.228 - Partição 3

```
4 - sensor
-----
Recebeu Fix = 0.473917
Recebeu Fiy = 0.005869
Recebeu Fiz = -0.000642
Sending fix'0.473917'
Sending fiy'0.005869'
Sending fiz'-0.000642'
Recebeu Fix = 0.466183
Recebeu Fiy = 0.006217
Recebeu Fiz = -0.000738
Sending fix'0.466183'
Sending fiy'0.006217'
Sending fiz'-0.000738'
```

Figura 7.229 – Partição 4

```
5 - extra
-----
Recebeu y = 1.651015
Sending u '-2.312723'
Recebeu y = 1.596194
Sending u '-0.050183'
Recebeu y = 0.753001
Sending u '2.030257'
Recebeu y = 0.618563
Sending u '0.422670'
Recebeu y = 1.315841
Sending u '-1.755194'
Recebeu y = 1.535875
Sending u '-0.751817'
```

Figura 7.230 – Partição 5

```
6 - extral
-----
Recebeu u = -2.312723
Sending y'1.596194'
Recebeu u = -0.050183
Sending y'0.753001'
Recebeu u = 2.030257
Sending y'0.618563'
Recebeu u = 0.422670
Sending y'1.315841'
Recebeu u = -1.755194
Sending y'1.535875'
Recebeu u = -0.751817
Sending y'0.911753'
```

Figura 7.231 – Partição 6

Como já foi dito, as partições de 1 a 4 alocam o SCA e as partições 5 e 6 alocam o MMA. Para não criar dois nomes iguais dentro do programa, optou-se por chamar o controlador do MMA de extra e a planta de extra1.

Como esperado, ao ocorrer a falha na partição 6 as demais partições continuaram seu funcionamento normalmente. Mais adiante apresentaremos os gráficos que mostram que os valores obtidos são os mesmos que os obtidos sem inserção de falhas. Desta forma, neste momento, daremos mais atenção às partições 5 e 6, pois a falha é inserida na partição 6 e a partição 5 tem comunicação com a mesma.

As figuras 7.232 e 7.233 apresentam as telas logo que a falha ocorre.

```
6 - extral
-----
Sending y '1.008011'
Recebeu u = 0.132754
Sending y '0.936185'
Recebeu u = 0.203323
Sending y '0.979130'
Recebeu u = -0.075439
Sending y '1.061237'
Recebeu u = -0.195285
Sending y '1.047245'
Recebeu u = 0.028466
Sending y '0.977240'
Recebeu u = 0.178906
ERROR!
```

Figura 7.232 – Partição com a falha inserida

```
5 - extra
-----
Recebeu y = 0.936185
Sending u '0.203322'
Recebeu y = 0.979130
Sending u '-0.075439'
Recebeu y = 1.061237
Sending u '-0.195285'
Recebeu y = 1.047245
Sending u '0.028466'
Recebeu y = 0.977240
Sending u '0.178906'
Sending u '0.014942'
Sending u '0.019813'
```

Figura 7.233 – Partição 5 logo que ocorre a falha da partição 6

Como pode ser visto na figura 7.233, em um determinado momento a partição 5 somente envia os resultados de seus cálculos. Isso ocorre, pois a partição 6 parou de enviar valores devido à falha. Assim, como a partição 5 não tem valores a serem lidos, ela mantém os cálculos utilizando o último valor lido.

As figuras 7.234 e 7.235 mostram a divergência dos valores do controle do MMA devido ao encerramento dos valores antes enviados pela partição 6. A figura 7.234 apresenta a tela dos primeiros segundos de simulação. Já a figura 7.235 apresenta a tela depois de um tempo bem mais longo.

```
5 - extra
-----
Sending u '-0.195285'
Recebeu y = 1.047245
Sending u '0.028466'
Recebeu y = 0.977240
Sending u '0.178906'
Sending u '0.014942'
Sending u '0.019813'
Sending u '0.024683'
Sending u '0.029554'
Sending u '0.034425'
Sending u '0.039295'
Sending u '0.044166'
```

Figura 7.234 - Partição 5 pouco tempo após falha da partição 6

```
5 - extra
-----
Sending u '197.457700'
Sending u '197.671700'
Sending u '197.885700'
Sending u '198.099700'
Sending u '198.313700'
Sending u '198.527700'
Sending u '198.741700'
Sending u '198.955700'
Sending u '199.169700'
Sending u '199.383700'
Sending u '199.597700'
Sending u '199.811700'
```

Figura 7.235 - Partição 5 num tempo considerável após a falha da partição 6

A seguir são apresentados os gráficos obtidos a partir desta simulação, utilizando as 6 partições com uma falha inserida na partição 6 do MMA.

7.2.3.5.1. Ângulos de atitude

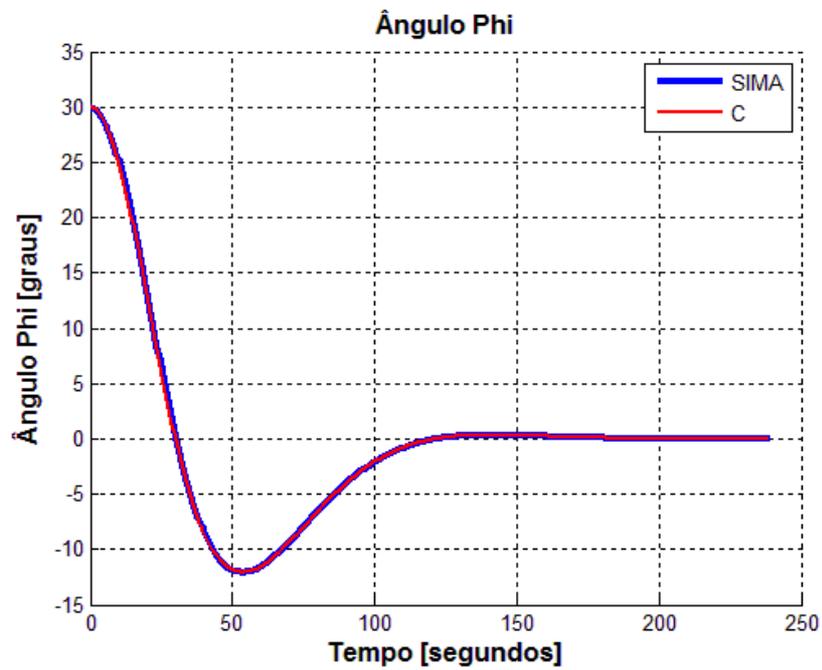


Figura 7.236 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

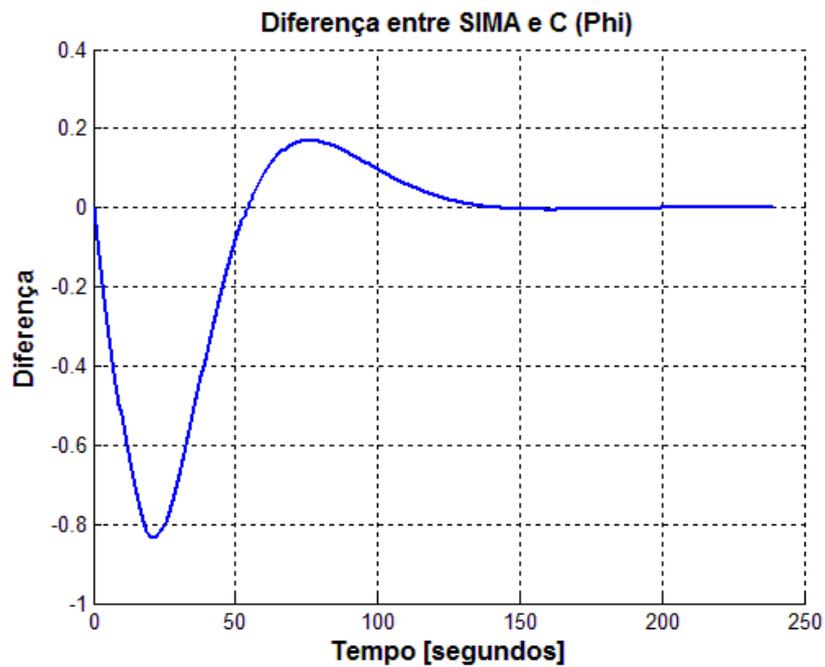


Figura 7.237- Diferença das curvas dos programas simulados em SIMA e linguagem C

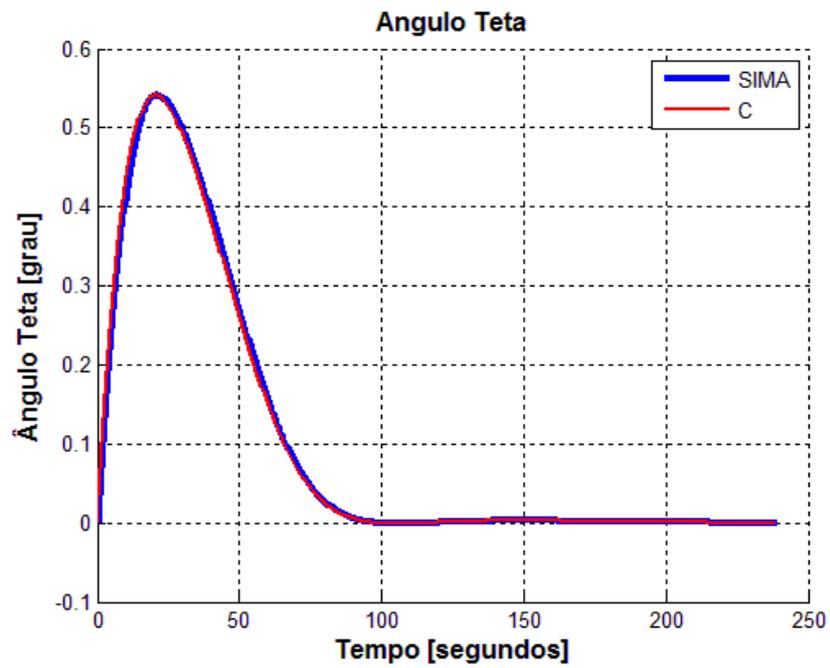


Figura 7.238 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

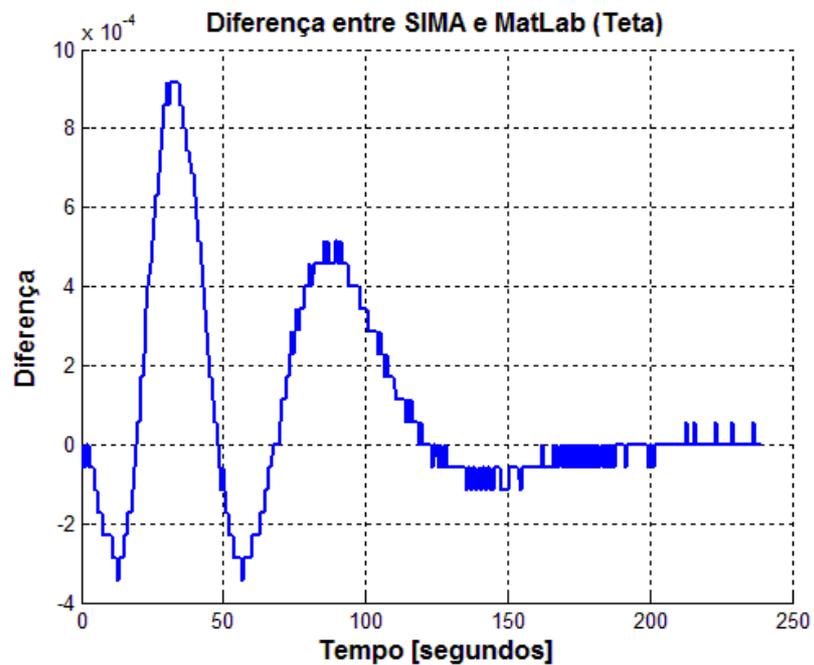


Figura 7.239- Diferença das curvas dos programas simulados em SIMA e linguagem C

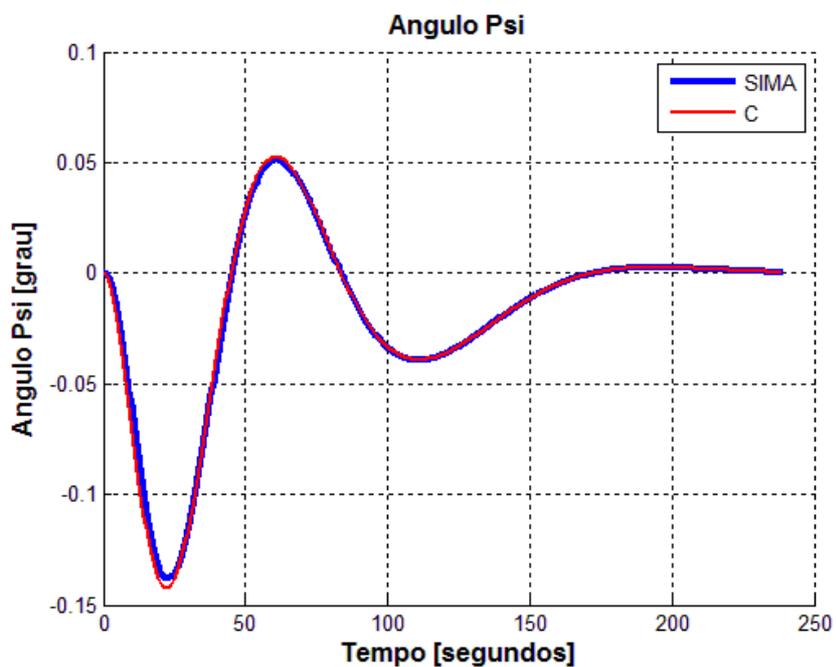


Figura 7.240 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

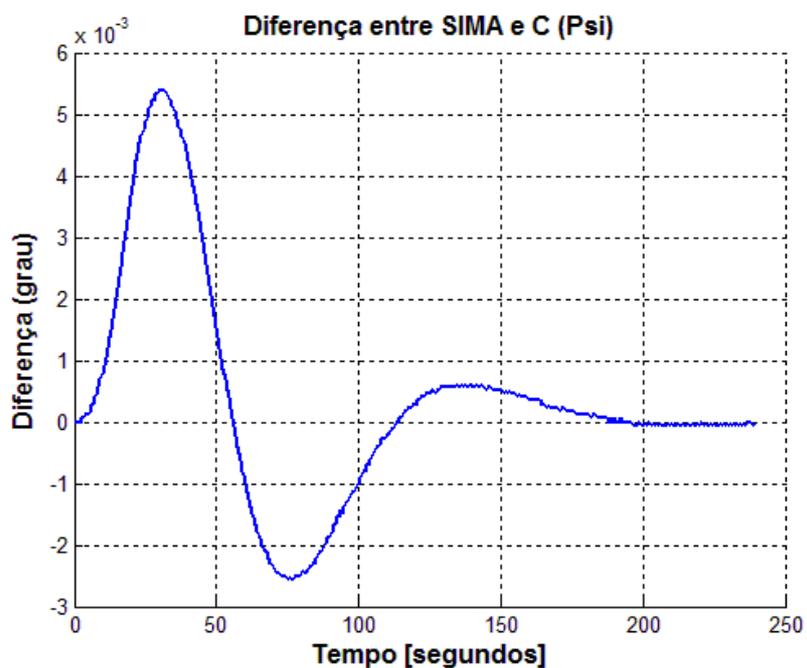


Figura 7.241- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.5.2. Velocidade angular da PMM

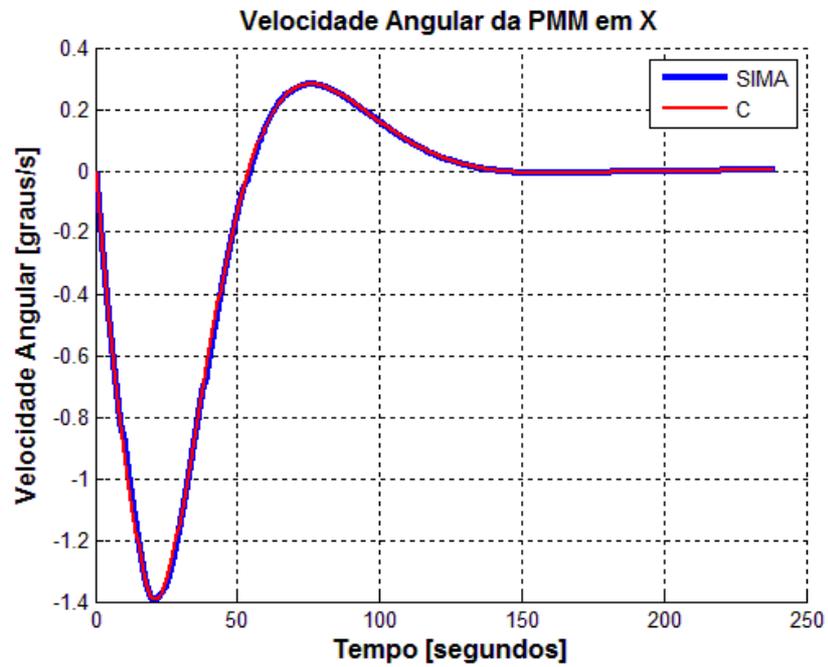


Figura 7.242 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

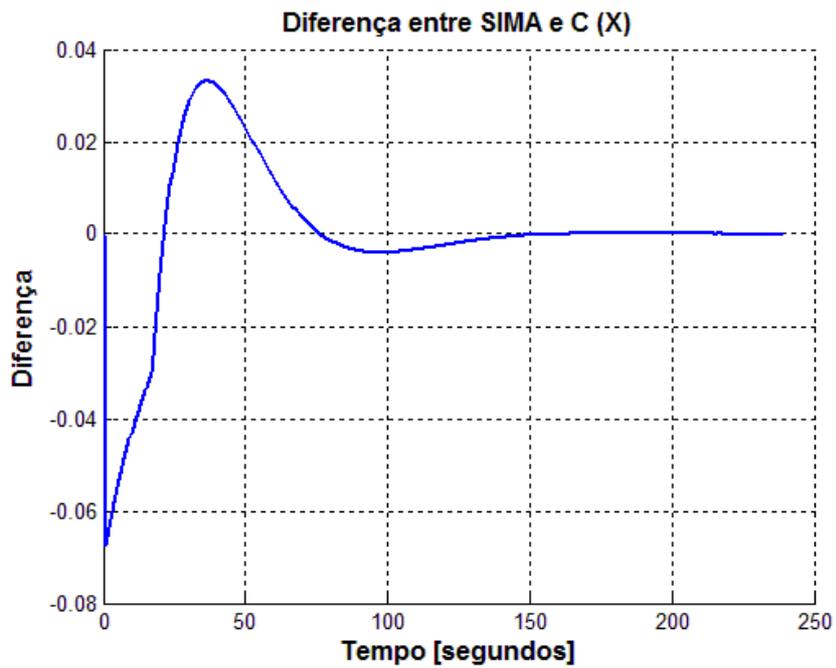


Figura 7.243- Diferença das curvas dos programas simulados em SIMA e linguagem C

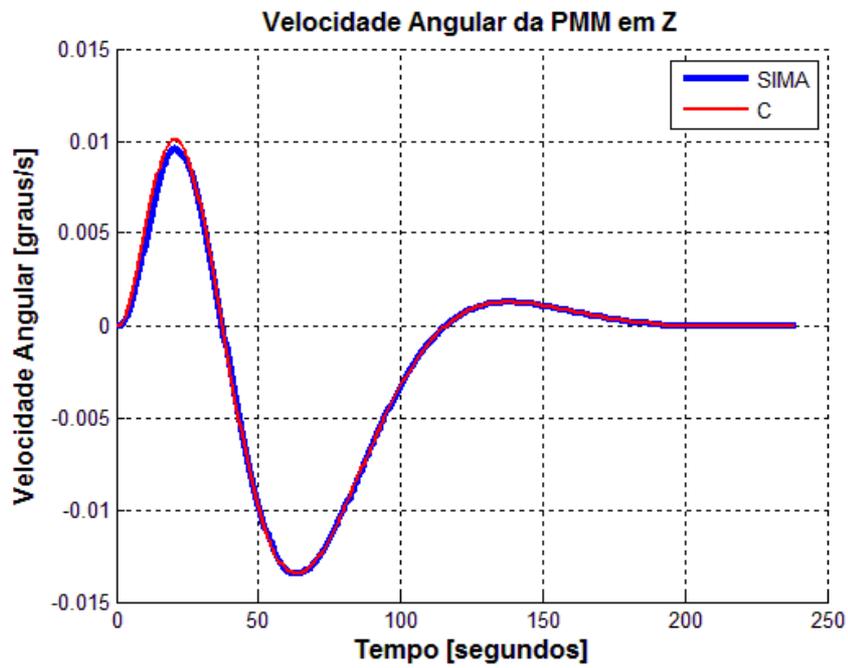


Figura 7.244 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

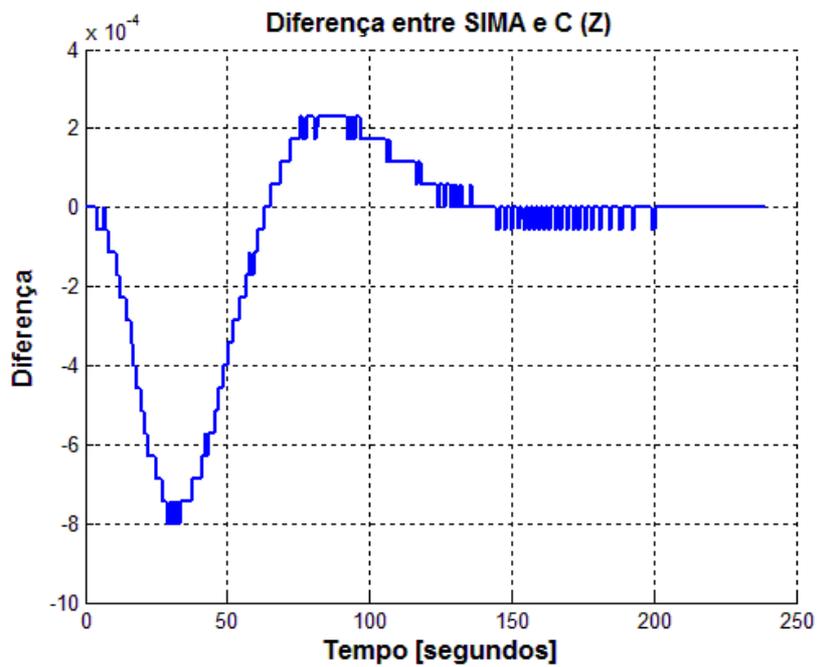


Figura 7.245- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.5.3. Velocidade angular das rodas

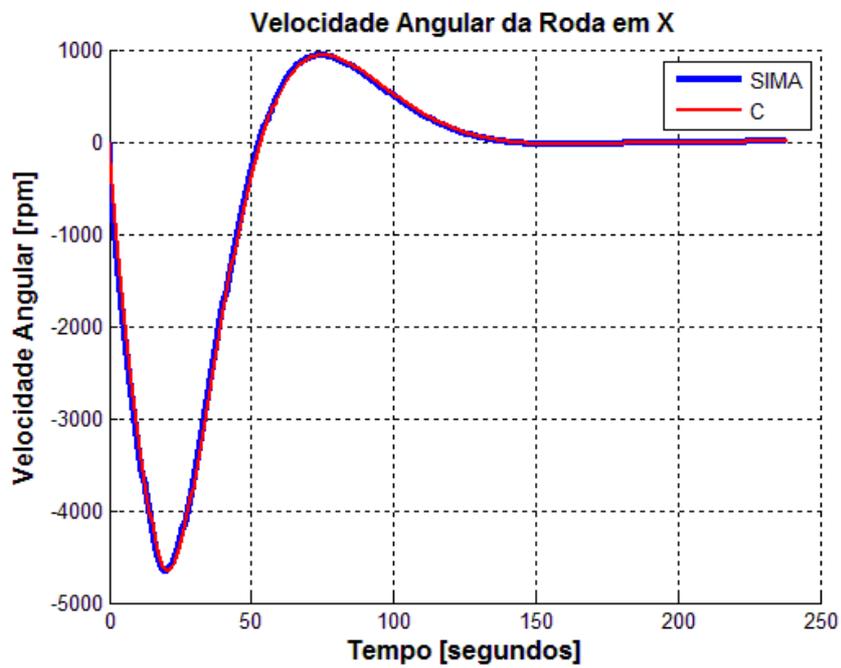


Figura 7.246 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

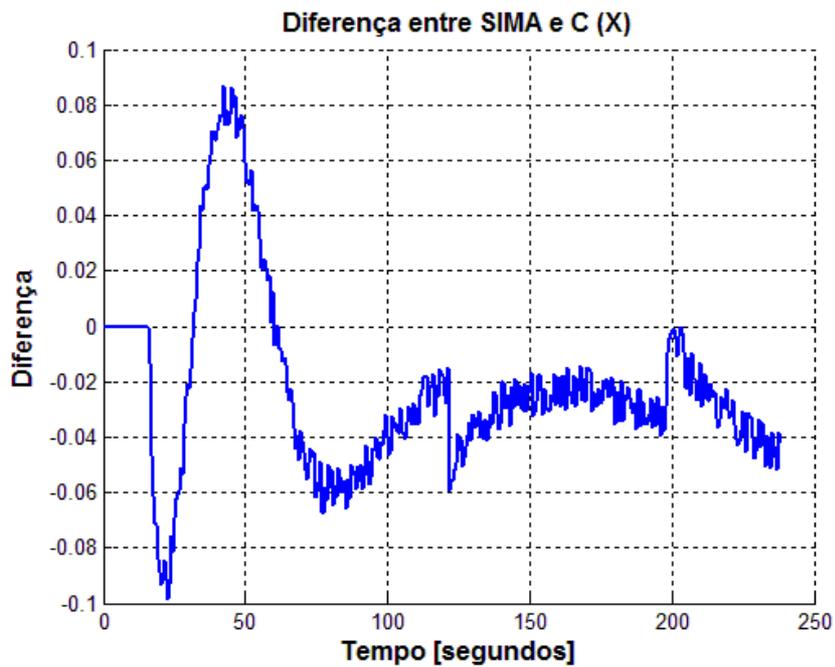


Figura 7.247- Diferença das curvas dos programas simulados em SIMA e linguagem C

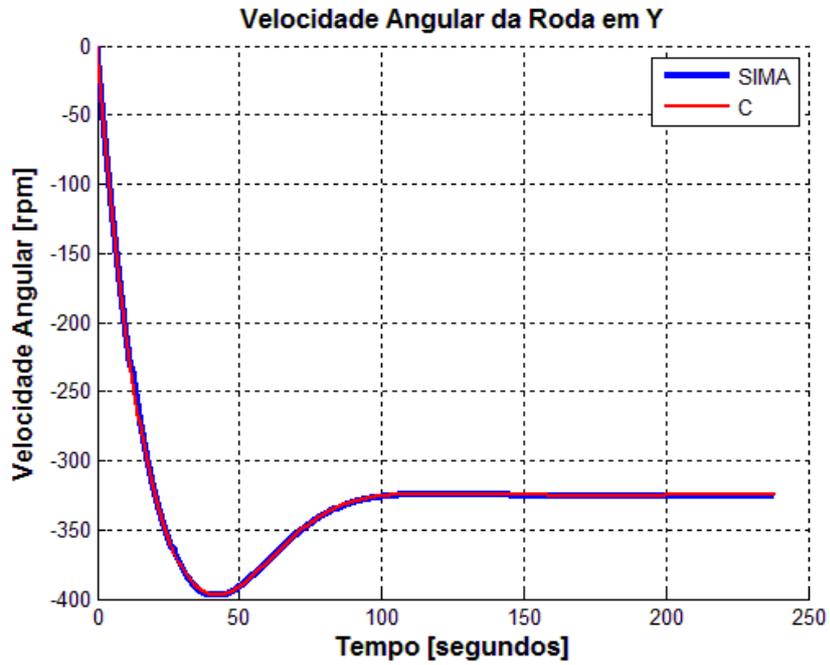


Figura 7.248 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

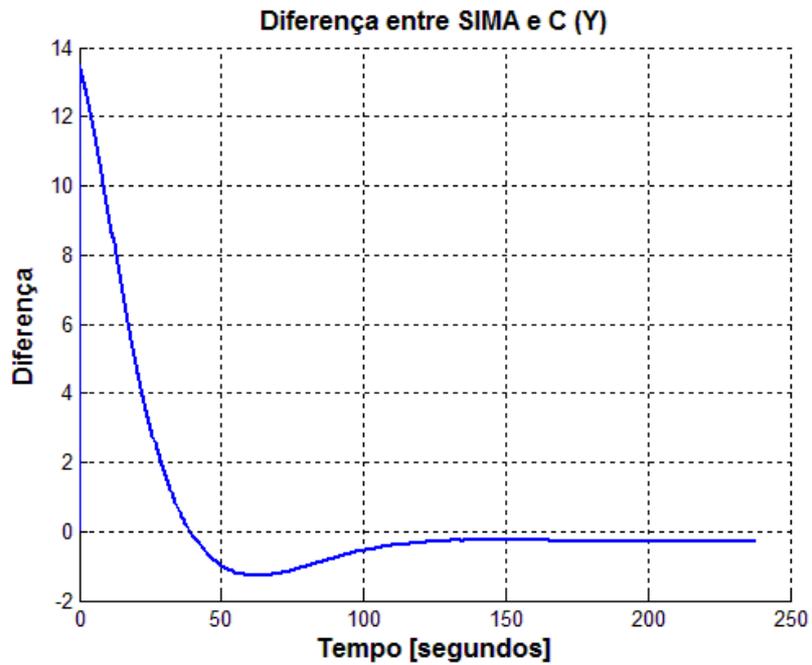


Figura 7.249- Diferença das curvas dos programas simulados em SIMA e linguagem C

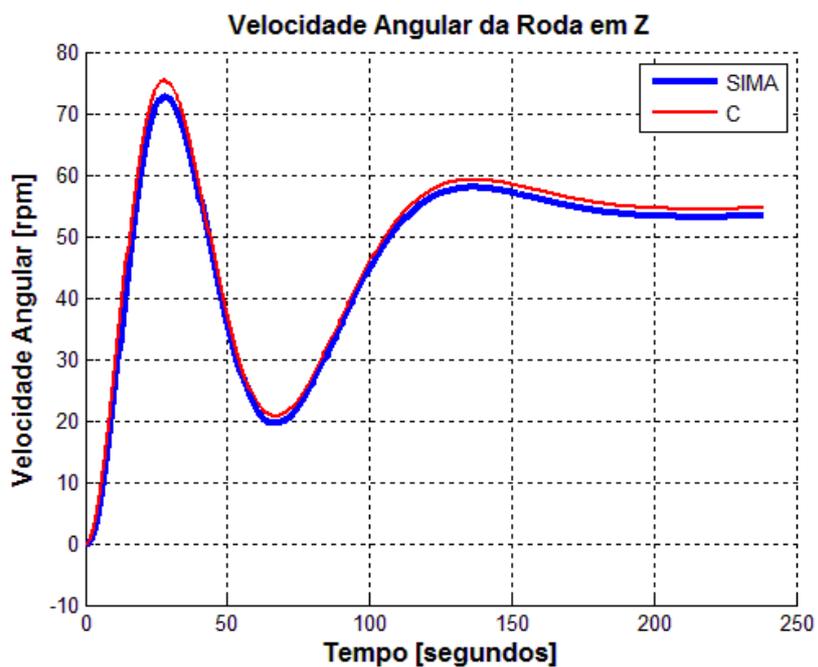


Figura 7.250 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

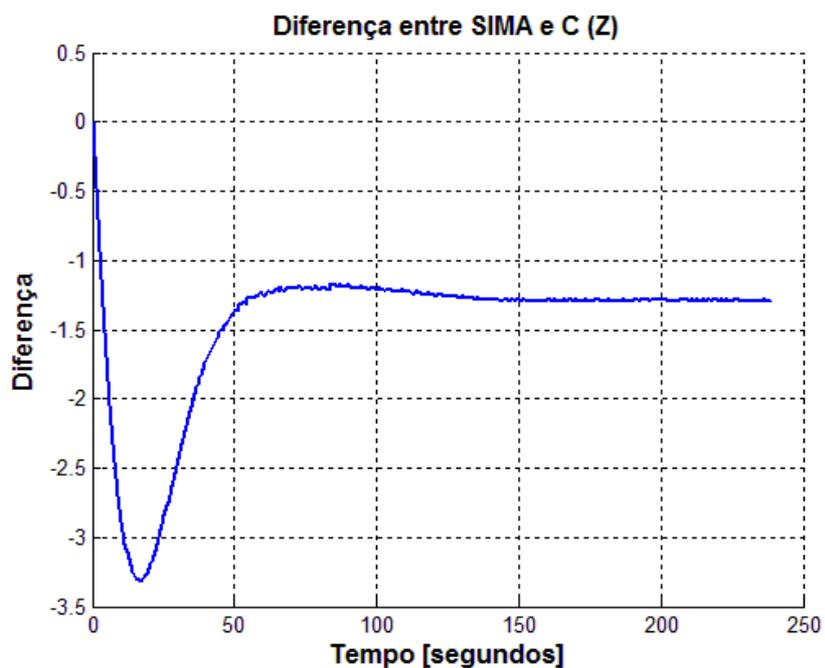


Figura 7.251- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.5.4. Tensão

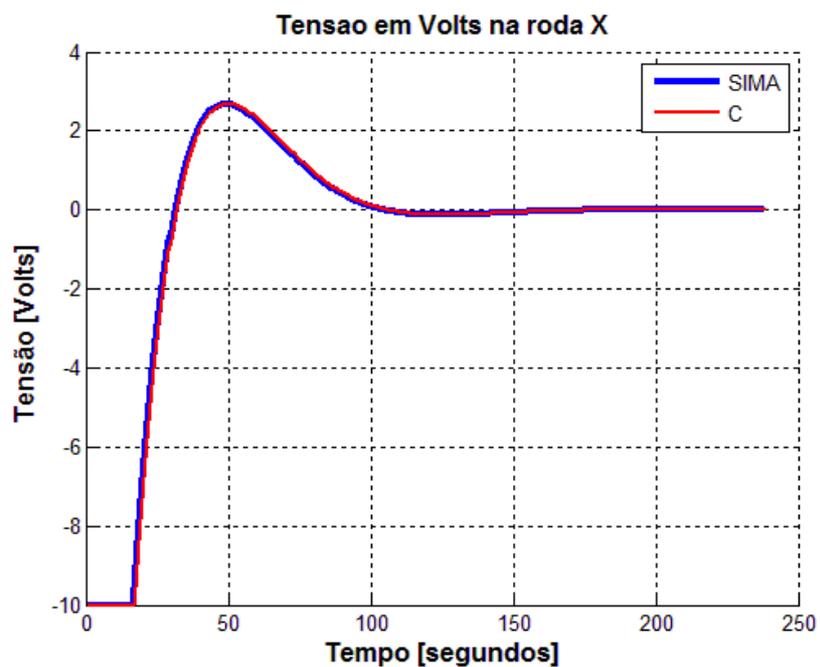


Figura 7.252 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

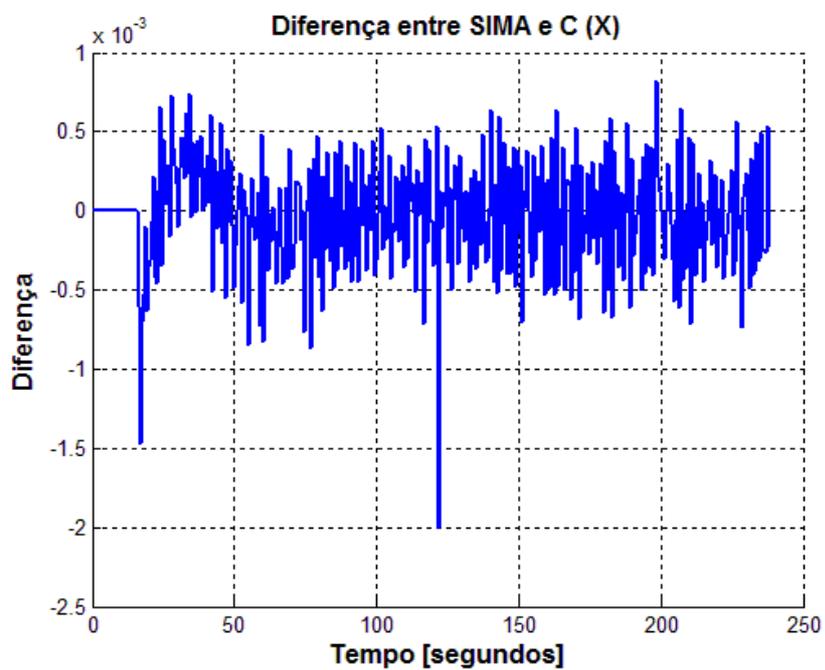


Figura 7.253- Diferença das curvas dos programas simulados em SIMA e linguagem C

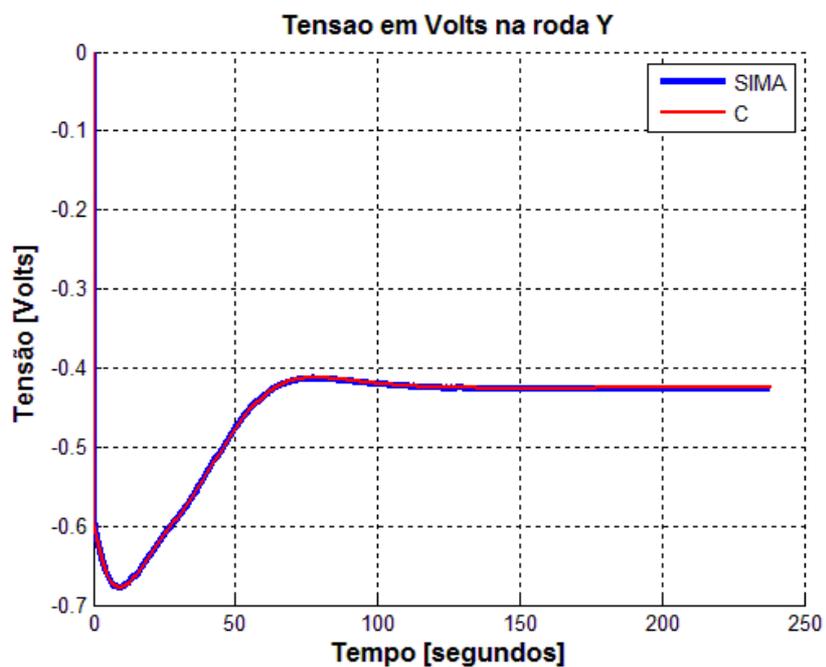


Figura 7.254 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

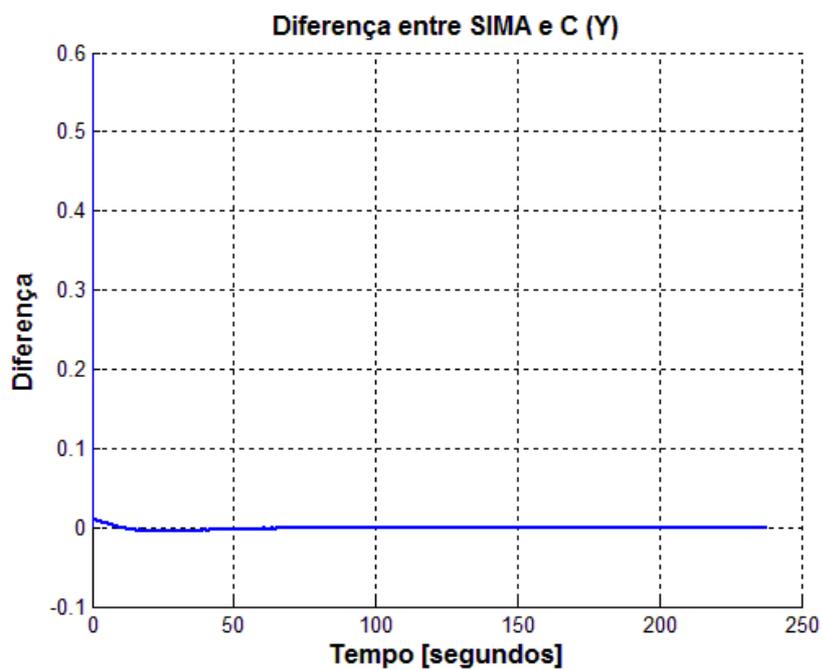


Figura 7.255- Diferença das curvas dos programas simulados em SIMA e linguagem C

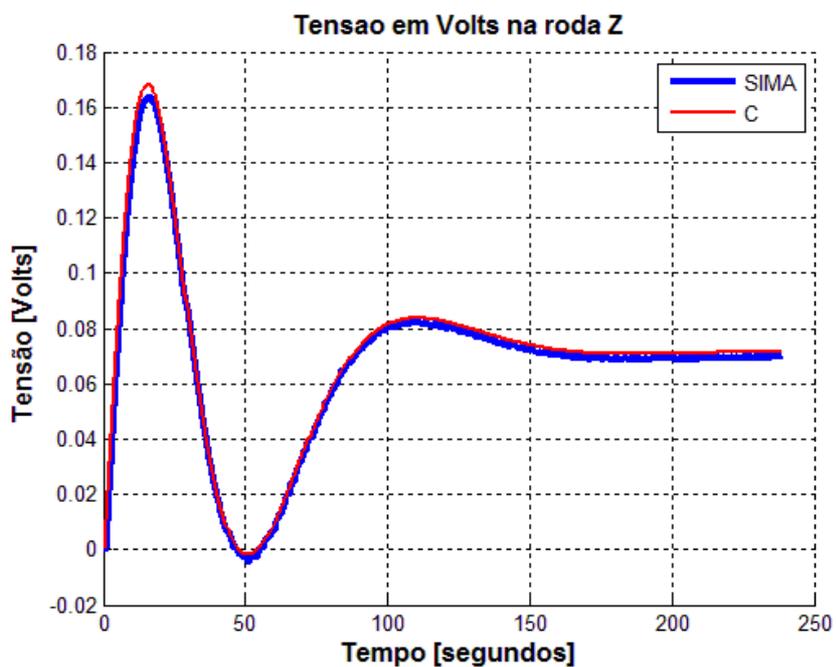


Figura 7.256 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

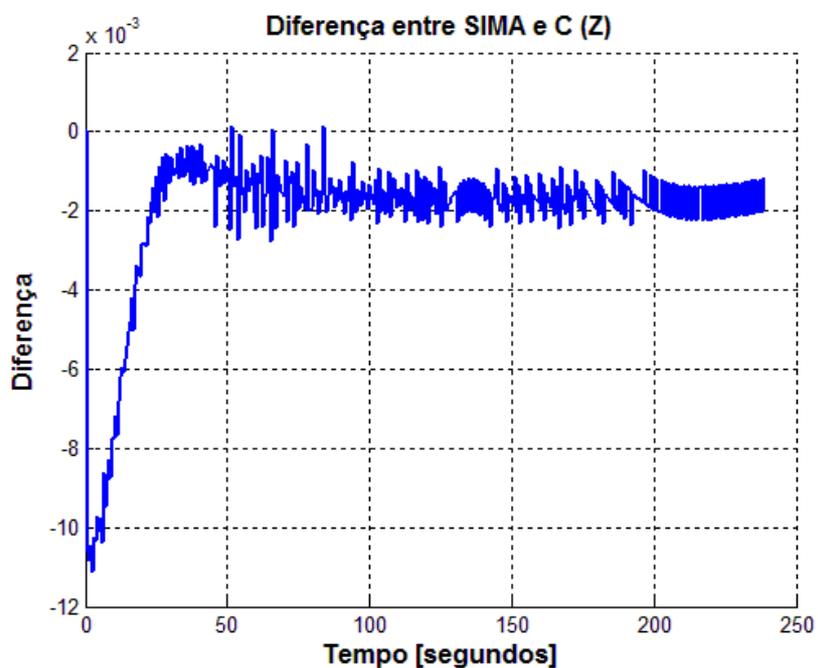


Figura 7.257- Diferença das curvas dos programas simulados em SIMA e linguagem C

7.2.3.5.5. Torque da roda

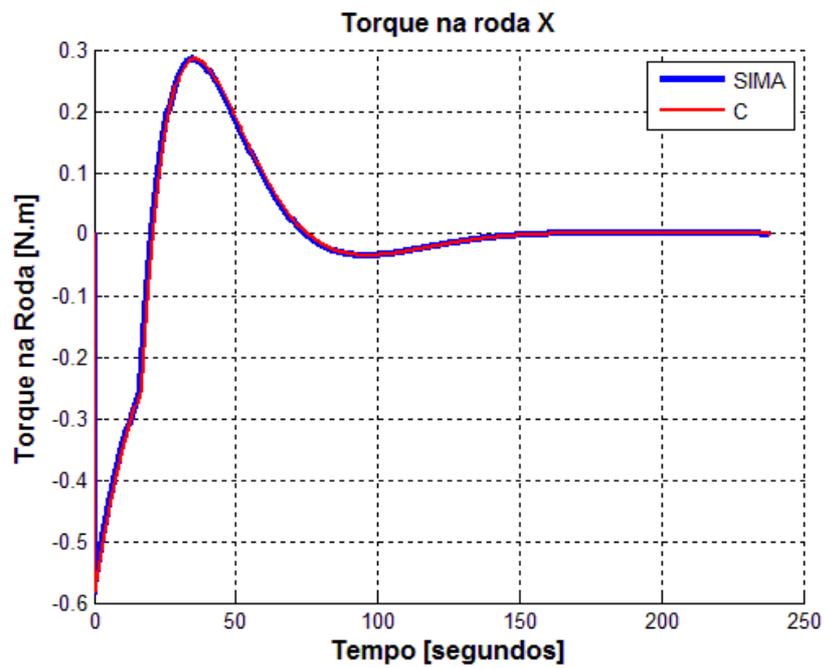


Figura 7.258 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

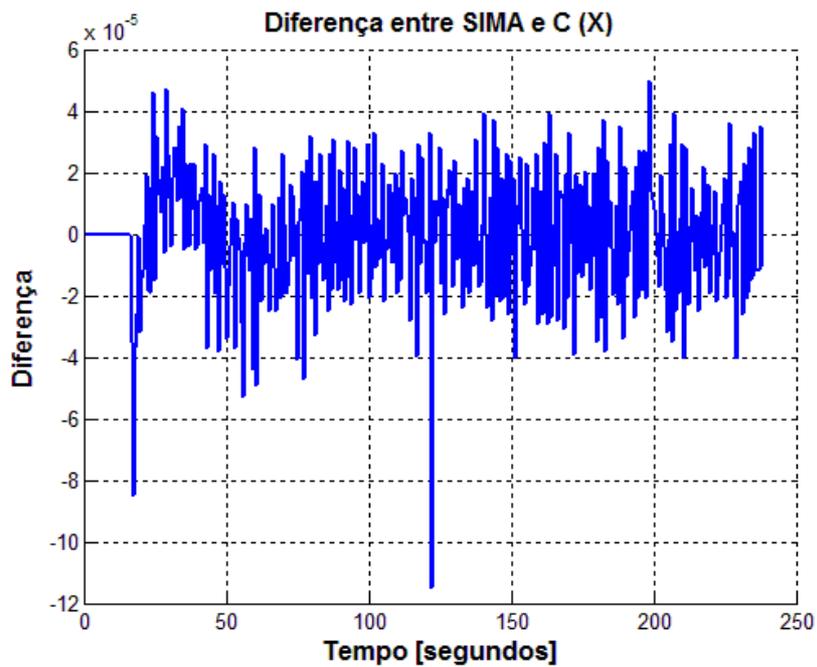


Figura 7.259- Diferença das curvas dos programas simulados em SIMA e linguagem C

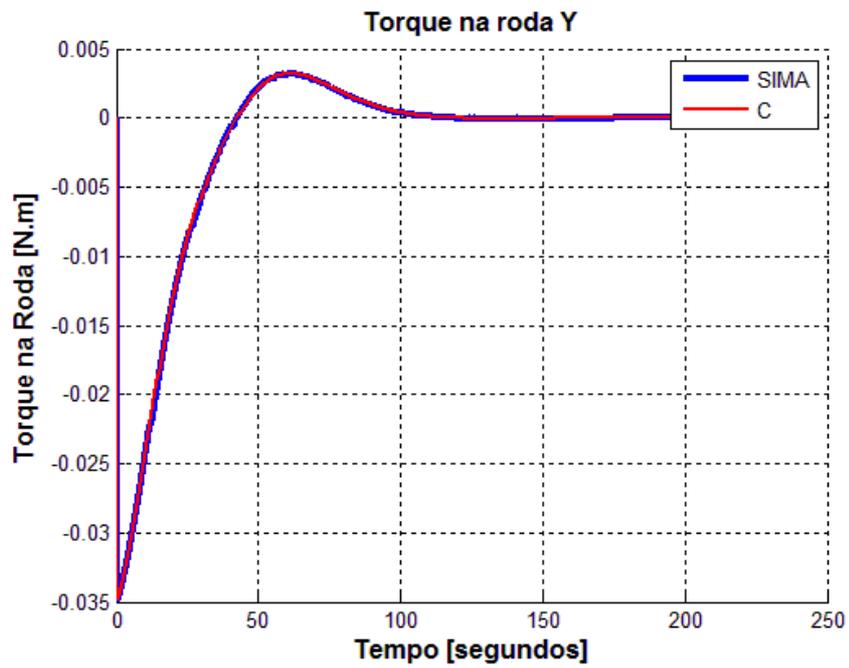


Figura 7.260 - Curvas dos resultados dos programas simulados em SIMA e linguagem C

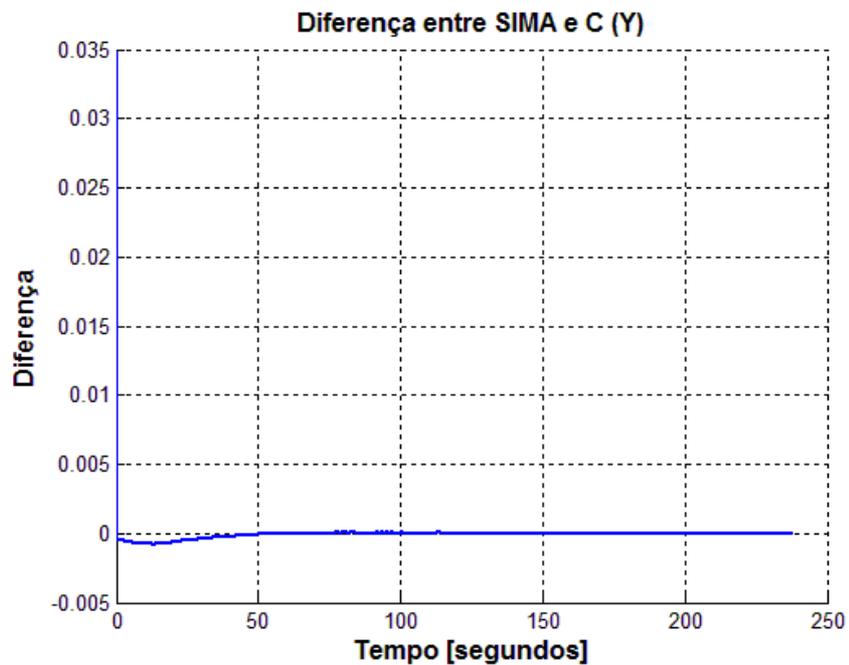


Figura 7.261- Diferença das curvas dos programas simulados em SIMA e linguagem C

8 COMENTÁRIOS E CONCLUSÕES

8.1. Geral

Neste trabalho foram modelados, implementados e simulados um sistema Massa-Mola-Amortecedor (MMA) e um Sistema de Controle de Atitude (SCA) compatível com o modo nominal do Satélite Amazônia-1 em desenvolvimento no Instituto Nacional de Pesquisas Espaciais (INPE) em MatLab, linguagem C, e em um Simulador de Aviônica Modular Integrada (SIMA).

Assim, conseguiu-se explorar diversas propriedades do Integrated Modular Avionics (IMA) como o particionamento, o agendamento das partições, comunicação intrapartição e interpartição, e prioridade entre os processos. Além disso, explorou-se a ocorrência de uma falha (intencional) em uma das partições e o resultado que isso causou nas demais partições.

Como dito anteriormente, optou-se por iniciar as modelagens e simulações com um sistema mais simples e conhecido, pois o ambiente Simulated Integrated Modular Avionics (SIMA) era, até então, pouco conhecido. Assim, iniciou-se pelo MMA. E, em seguida, partiu-se para o SCA.

A modelagem, implementação e simulação dos dois sistemas foi feita em MatLab, linguagem C e, finalmente, no SIMA. O MatLab oferece muitos recursos (trabalha com matrizes, toolboxes, etc), no entanto, optou-se por não utilizar esses recursos. Procurou-se implementar todas as funções de maneira semelhante. Deste modo, apesar das diferentes sintaxes, as funcionalidades desenvolvidas pelos programas foram as mesmas. No SIMA, além das diferenças de sintaxes, foi feita a divisão do programa em diversas partições, o que envolveu o envio e recebimento de dados através das portas e dos canais.

Apesar do particionamento no SIMA, os resultados dos programas executados nesse ambiente, foram semelhantes aos resultados dos programas executados em linguagem C e MatLab.

8.1.1. Massa-Mola Amortecedor

Foi feita a comparação dos resultados obtidos a partir do programa do MMA em linguagem C e do programa do MMA no SIMA chegando-se a diferenças entre as curvas na ordem de 10^{-5} metros. Já a comparação dos resultados obtidos em MatLab e linguagem C, resultou em diferenças um pouco maiores (10^{-3} metros), porém ainda muito pequenas. Essas diferenças são geradas devido a maneira própria de cada programa tratar os dados. Em geral, os resultados mostraram-se muito próximos no caso do SIMA e linguagem C, pois, os dois programas são compilados e, a diferença mais significativa entre os dois foi apenas o particionamento do SIMA. Já o programas executados em MatLab apresentaram diferenças um pouco mais significativas, pois além da maneira própria de tratar os dados, o MatLab é um programa interpretado, ou seja, sua compilação se dá durante a execução do programa.

8.1.2. Sistema de controle de atitude

A simulação do SCA foi mais explorada, pois é o foco deste trabalho. Deste modo, cinco casos foram simulados:

No caso (1), foi feita a simulação do SCA em MatLab e linguagem C, e os resultados foram comparados entre si. Estes resultados foram importantes para demonstrar as diferenças geradas entre programas diferentes (MatLab e Linguagem C) devido à forma como tratam os dados, realizam cálculos numéricos etc.

No caso (2) foi feita a simulação do SCA em SIMA e MatLab, e os resultados foram comparados entre si. Neste caso, as simulações foram feitas utilizando o particionamento do SCA (4 partições) no SIMA, e seu funcionamento em um

único bloco, no MatLab. Apesar do particionamento utilizado no SIMA, os resultados também foram próximos. Notou-se um pequeno atraso dos resultados no SIMA em relação ao MatLab. Este atraso foi percebido em todos os casos em que se utilizou o SIMA.

No caso (3) foi feita a simulação do SCA em SIMA e em linguagem C. Neste caso, as simulações foram feitas utilizando o particionamento do SCA (4 partições) no SIMA, e seu funcionamento em um único bloco em linguagem C. Neste caso os resultados foram mais próximos, mas também houve um atraso do SIMA em relação à linguagem C.

No caso (4) criou-se 6 partições no SIMA e inseriu-se o MMA e o SCA em uma mesma simulação. As 4 primeiras partições embarcaram o SCA e as 2 seguintes o MMA. Nessa simulação utilizou-se o passo de 0.6 segundos para coincidir o passo de integração com o tempo de amostragem das partições. Os resultados foram muito próximos dos obtidos nas simulações anteriores.

No caso (5) fez-se a mesma simulação anterior (caso 4), porém inseriu-se uma falha na partição 6 para explorar a confiabilidade do sistema de particionamento do IMA. Apesar da falha ocorrida na partição 6, as demais partições continuaram seu funcionamento normalmente. As partições de 1 a 4 apresentaram exatamente os mesmos resultados apresentados no caso 4. Já a partição 5 teve seus valores alterados (valores divergiram devido a abertura da malha que era inicialmente fechada), pois dependia de valores enviados pela partição 6, no entanto seus cálculos continuaram a ser executados normalmente, o que demonstra o robusto isolamento entre as partições.

Com já dito, os casos (1) e (2) foram importantes para demonstrar as diferenças entre os resultados dos programas executados em diferentes ambientes devido à maneira de tratamento de dados, cálculos numéricos, arredondamentos e truncamentos, etc.

Os casos (3), (4) e (5) tiveram maior importância, pois provaram que o IMA pode ser utilizado na aplicação espacial em questão, um Sistema de Controle de Atitude compatível com o modo nominal da Plataforma MultiMissão. Além disso, a comparação feita no caso (5) demonstra a robustez do sistema de particionamento do IMA, pois, mesmo após a parada de uma das partições as demais continuaram funcionando normalmente sem sofrer influência alguma.

É importante enfatizar, que os resultados obtidos dos programas executados em linguagem C e SIMA apresentam uma proximidade muito maior que o programa executado no MatLab. Na comparação entre os programas simulados em linguagem C e MatLab, as figuras 7.45 e 7.51 (Velocidade angular da roda no eixo z e tensão na roda do eixo z, respectivamente) apresentam as curvas com as maiores diferenças obtidas. Já as curvas com as maiores diferenças obtidas a partir da comparação das simulações em SIMA e MatLab são apresentadas nas figuras 7.75 e 7.81 (Velocidade angular da roda no eixo z e tensão na roda do eixo z, respectivamente). A comparação das curvas obtidas em linguagem C e MatLab com as curvas obtidas em SIMA e MatLab são muito próximas, o que demonstra que a maior diferença foi gerada pelo MatLab. Os resultados obtidos a partir das simulações entre SIMA e linguagem C apresentam erros bem menores. As figuras 7.105 e 7.111 (Velocidade angular da roda no eixo z e tensão na roda do eixo z, respectivamente) ilustram essa diferença.

Durante as simulações foi desconsiderado o tempo de inicialização das partições e dos processos apesar de ter sua execução uma única vez.

O tempo de execução das partições definido no SIMA sofre uma pequena variação, ou seja, não é exato. Desta forma, foi possível notar um pequeno dessincronismo entre o passo de integração dos Runge-Kutta e o tempo de amostragem das partições. Para nossas simulações, esse dessincronismo não influenciou os resultados, no entanto, o mesmo pode não acontecer em outros sistemas.

8.2. Sugestões para trabalhos futuros

Baseado no trabalho feito, conclui-se ser de suma importância o estudo do dessincronismo entre os tempos do SIMA e do incremento do passo dos integradores (Runge-Kutta).

Também se mostra interessante o estudo dos outros modos de operação do sistema de controle de atitude da PMM no simulador SIMA e em uma plataforma IMA (hardware).

Como um próximo passo, sugere-se a implementação dos sistemas simulados neste trabalho em uma plataforma IMA (hardware) e a comparação do funcionamento da mesma com os resultados obtidos neste trabalho.

Outra sugestão de estudo muito interessante é o uso do IMA para se obter redundância com dissimilaridade, ou seja, redundância de um sistema com hardware e/ou software totalmente diferente do original.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, P. C. G.; SANTOS, C. C., **GPS para iniciantes**. São José dos Campos: INPE, 2003. (INPE-9602-PUD/124). Disponível em: <<http://urlib.net/sid.inpe.br/jeferson/2003/06.02.09.16>>. Acesso em: 17 jan. 2013.

ALEKSA, B.D.; CARTER, J.P. Boeing 777 Airplane Information Management System Operational Experience. In: DIGITAL AVIONICS SYSTEMS CONFERENCE (16th DASC), 16., 1997, Phoenix. **Proceedings...** Phoenix, AZ, USA: AIAA/IEEE, 1997. v. 1, p. 3.1 - 21-7.

ALENA, R.L.; GOFORTH, A.; FIGUEROA, F., OSSENFORT, J.; LAWS, K.I. Communication for Integrated Modular Avionics. In: IEEE AEROSPACE CONFERENCE, 2007, Big Sky, Montana. **Proceedings...** Montana: IEEE/IAAA, 2007. DOI: 10.1109/AERO.2007.352635.

ARANTES JÚNIOR, G. **Estudo comparativo de técnicas de controle de atitude em três eixos para satélites artificiais**. 2005. 201 p. (INPE-12970-TDI/1018). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2005. Disponível em: <<http://urlib.net/sid.inpe.br/jeferson/2005/03.09.14.25>>. Acesso em: 28 jan. 2013.

ASTROM, K.J. **Control system design**, 2002. Disponível em: <<http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf>>. Acesso em 28 de jan. de 2013.

AVIONICS ELECTRONIC ENGINEERING COMMITTEE - ARINC. **Avionics application software standard interface: Part 1 - required services** (ARINC specification 653-2). Technical report. Annapolis, Maryland, USA, Maio, 2006.

AVIONICS MAGAZINE. **Integrated modular avionics: less is more**. 2007. Disponível em: <<http://www.aviationtoday.com/av/issue/feature/8420.html>>. Acesso em: 25 de abril 2011.

BARBOUR, N.M. **Inertial navigation sensors**. Cambridge, USA: Charles Stark Draper Laboratory, , 2011. Disponível em: <<ftp://ftp.rta.nato.int/pubfulltext/RTO/EN/RTO-EN-SET-116-2011/EN-SET-116%282011%29-02.pdf>>. Acesso em: 04 de abril de 2013.

BLACK, R.; FLETCHER, M. Simplified robotics avionics system: an integrated modular architecture applied across a group of robotic elements. In: DIGITAL AVIONICS SYSTEMS CONFERENCE, 25., 2006, Portland, OR. **Proceedings...** Portland: IEEE/AIAA, 2006.

DIAZ, K.D. **Performance analysis of a fixed point star tracker algorithm for use onboard a picosatellite**. Thesis (Master Degree, Aerospace Engineering Science) – Polytechnic State University, California, 2006.

DI NATALE, M.; SANGIOVANNI-VINCENTELLI, A.L. **Moving from federated to integrated architectures in automotive: the role of standards, methods and tools**. Proceedings of the IEEE, v.98, n.4, pp. 603-620, abr. 2010.

DI VITO, B.L. A Model of cooperative noninterference for integrated modular avionics. In: IFPI INT. WORKING CONF. ON DEPENDABLE COMPUTING FOR CRITICAL APPLICATIONS (DCCA-7), 7.,1999, San Jose. **Proceedings...** San Jose: IEEE, 1999.

DINIZ, N.; RUFINO, J. ARINC 653 in Space. In: DASIA – DATA SYSTEM IN AEROSPACE CONFERENCE, 2005, Edinburg, UK. **Proceedings...** Noordwijk, The Netherlands: ESA, 2005. (ESA SP - 602, August 2005).

ESA SCIENCE & TECHNOLOGY . **Final testing of digital sun sensor**. Frascati, Italy, 2007. Disponível em: <<http://sci.esa.int/science-e/www/object/index.cfm?fobjectid=40619>>. Acesso em: 09 jan. 2013.

Experiment ARCADE. **ARCADE reaction Wheel**. Padova, Italy, 2011. Disponível em: <<http://www.experimentarcade.org/wordpress/?p=511> >. Acesso em: 09 jan. 2013.

FONSECA, I.M.; BAINUM, P. M. The Brazilian satellite program – a survey. In: THE MALCOLM D. SHUSTER ASTRONAUTICS SYMPOSIUM, 2005, Grand Island, New York. **Proceedings...** New York: American Astronautical Society, 2005 . This is volume 122 in the Advances in the Astronautical Sciences series.

GANGKOFER, M.; KADER, H.; KLOCKNER, W.; WHITE, C.G. Transitioning to integrated modular avionics with a mission management system. In: RTO SCI SYMPOSIUM, 2000, Budapest, Hungary. **Proceedings...**Budapest, 2000.

GEORGIA STATE UNIVERSITY - GST. **The ecliptic.**, 1997. Disponível em: <<http://hyperphysics.phy-astr.gsu.edu/hbase/eclip.html>>. Acesso em 12 de abril 2013.

GMV IMA Research Group. **SIMA command line tools application development and configuration guide**, 2011. Disponível em: <http://www.gmv.com/export/sites/gmv/DocumentosPDF/sima/User_Manual.pdf>. Acesso em: 10 de abril de 2013.

GOBATO, M. F. **Controles monovariáveis e multivariáveis aplicados a sistemas aeroespaciais fracamente ou fortemente acoplados**. 2006. 388 p. (INPE-14494-TDI/1175). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006. Disponível em: <<http://urlib.net/sid.inpe.br/MTC-m13@80/2006/06.30.18.04>>. Acesso em: 07 de fev. 2013.

HONEYWELL INTERNATIONAL INC. **Smart digital magnetometer HMR2300**. Plymouth, Minnesota, United States of America, 2012. Disponível em: <<http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Missiles-Munitions/HMR2300.pdf>>. Acesso em: 09 jan. 2013.

JENA-OPTRONIK GMBH. **Autonomous star sensor ASTRO 15**. Jena, Germany, 2011. Disponível em <http://www.electronicnote.com/Jena/Astro15_0901.pdf>. Acesso em: 09 jan. 2013.

HIBBELER, R.C., **Engineering mechanics: dynamics**. 20 ed. Prentice Hall, 2010.

KAPLAN, M.H., **Modern spacecraft dynamics & control**. New York: John Wiley & Sons, 1976.

KOPETZ, H.; OBERMAISSER, R. **GENESYS: a candidate for an artemis cross-domain reference architecture for embedded systems**. Saarbrücken, Germany: Südwestdeutscher Verlag für Hochschulschriften (SVH) Aktiengesellschaft & Co., 2009. ISBN 978-3-8381-1040-0.

LEE, J.; NG A.; JOBANPUTRA, R. On determining dipole moments of a magnetic torquer rod – experiments and discussions. **Canadian Aeronautics and Space Journal**, v. 48, Mar. 2002. Disponível em: <<http://mac6.ma.psu.edu/lionsat/q02-014.pdf>>. Acesso em: 17 jan. 2013.

MANGIERI, M.L.; VICE, J. Orion flight software v&v and kedalion engineering lab insight. In: AIAA SPACE 2010 CONFERENCE & EXPOSITION (Space 2010), 2010, Anaheim, CA. **Proceedings...** Anaheim: AIAA, 2010.

MCCABE, M.; BAGGERMAN, C. Avionics architecture interface considerations between constellation vehicles. In: DIGITAL AVIONICS SYSTEM CONFERENCE (DASC '09), 28., 2009 Vienna, Austria. **Proceedings...** Viena: IEEE/AIAA, 2009.

MICROCOSM INC. **Magnetic torquers for spacecraft**. Hawthorne, CA, USA, 2008. Disponível em : <<http://www.smad.com/analysis/torquers.html>>. Acesso em: 09 jan. 2013.

OBERMAISSER, R.; PETI, P.; TAGLIABO, F. An integrated architecture for future car generations. **Real-Time Systems Journal**, v. 36, p. 101-133, 2007.

OGATA, K. **Modern control engineering**. 3 ed. Prentice-Hall, 1997.

PASCOAL E.; RUFINO, J.; SCHOOF, T.; WINDSOR, J. AMOBA – ARINC 653 simulator for modular space based applications. In: DATA SYSTEMS IN AEROSPACE CONFERENCE, 2008. Disponível em: <<http://dx.doi.org/10.1109/DASC.2008.4702767>>. Acesso em 10 de abril de 2013.

ROCKWELL COLLINS. **RSI 45 Momentum and reaction wheels**. Cedar Rapids, Iowa, USA, 2012. Disponível em: <<http://www.rockwellcollins.com>>

/sitecore/content/Data/Products/Space_Components/Satellite_Stabilization_Wheels/RSI_45_Momentum_and_Reaction_Wheels.aspx>. Acesso em: 09 jan. 2013.

RTC GROUP, Inc. Modular approach eases avionics certification challenges. **COTS Journal**. Julho de 2010. Disponível em: <<http://www.cotsjournalonline.com/articles/view/101451>>. Acesso em 09 de julho de 2013.

SAMAAN, M. A. **toward faster and more accurate star sensors using recursive centroiding and star identification**. Thesis (Doctor of Philosophy Degree, Aerospace Engineering Science) – Texas A&M University, 2003.

SANTOS, S.; RUFINO, J.; SCHOOF, T.; TATIBANA, C.; WINDSOR, J. **A portable ARINC 653 standard interface**. In: IEEE SECTIONS CONGRESS, 2008, Quebec. **Proceedings...** Quebec: IEEE, 2008.

SCHOOF, T.; SANTOS, S.; TATIBANA, C.; ANJOS, J. **An integrated modular avionics development environment**. In: DIGITAL AVIONICS SYSTEMS CONFERENCE, DASC '09, 28., 2009, Orlando FL. **Proceedings...** Orlando: IEEE/AIAA, 2009.

SELEX ES S.p.A. **IRES Infrared Earth Sensor**. Campi Bisenzio, Italy, 2013. Disponível em: <http://www.selex-es.com/~media/Files/S/Selex-Galileo/product-directory/space/IRES_NE.pdf>. Acesso em: 09 jan. 2013.

SIDI, M. J. **Spacecraft dynamics and control**. 1.ed. Cambridge: University Press, 1997.

SOUZA, M. L. O. **Estudo e desenvolvimento de um sistema de controle de atitude ativo em três eixos para satélites artificiais usando atuadores pneumáticos a gás frio e volantes de reação**. 1980. (INPE-2000-TDL/042). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1980.

SPACEREF INTERACTIVE INC. **The engines that make LRO and LCROSS Go**. Reston, Virginia, United States of America, 2012. Disponível em: <<http://onorbit.com/node/1364>>. Acesso em: 09 jan. 2013.

SSBV AEROSPACE & TECHNOLOGY GROUP. **Earth horizon sensor**. United Kingdom, 2012. Disponível em: <<http://www.ssbv.com/ProductDatashets/page39/page28/index.html>>. Acesso em: 09 jan. 2013.

SSBV AEROSPACE & TECHNOLOGY GROUP. **Fine sun sensor**. United Kingdom, 2013. Disponível em: <<http://www.ssbv.com/ProductDatashets/page39/page25/index.html>>. Acesso em: 09 jan. 2013.

TEWARI, A. **Atmospheric and space flight dynamics**. Boston: Birkhäuser, 2006.

UNIVERSITY OF SURREY. **Propulsion**. Guildford, Surrey, United Kingdom, 2012. Disponível em: <<http://www.surrey.ac.uk/ssc/research/propulsion/index.htm>>. Acesso em: 09 jan. 2013.

VECTRONIC AEROSPACE GmbH. **Star Tracker VST-41M**. Berlin, Germany, 2010a. Disponível em: <<http://www.vectronic-aerospace.com/space.php?p=starsensor>>. Acesso em: 09 jan. 2013.

VECTRONIC Aerospace GmbH. **Reaction wheel VRW-1**. Berlin, Germany, 2010b. Disponível em: <<http://www.vectronic-aerospace.com/space.php?p=reactionwheel>>. Acesso em: 09 jan. 2013.

VECTRONIC AEROSPACE GmbH. **Magnetic Torquer VMT-35**. Berlin, Germany, 2010c. Disponível em: <<http://www.vectronic-aerospace.com/space.php?p=torquer>>. Acesso em: 09 jan. 2013.

WATKINS, C.B.; WALTER, R. Transitioning from federated avionics architectures to integrated modular avionics. In: DIGITAL AVIONICS SYSTEMS CONFERENCE, DASC '07, 26., 2007, Dallas TX. **Proceedings...** Dallas: IEEE/AIAA, 2007.

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

WIKIPÉDIA. **C-Linguagem de programação**. Flórida: Wikimedia Foundation, 2013. Disponível em: <[http://pt.wikipedia.org/w/index.php?title=C_\(linguagem_de_programa%C3%A7%C3%A3o\)&oldid=35345655](http://pt.wikipedia.org/w/index.php?title=C_(linguagem_de_programa%C3%A7%C3%A3o)&oldid=35345655)>. Acesso em: 18 abr. 2013.

WIKIPÉDIA. **GNU compiler collection**. Flórida: Wikimedia Foundation, 2013. Disponível em: <http://pt.wikipedia.org/w/index.php?title=GNU_Compiler_Collection&oldid=34468789>. Acesso em: 18

WIKIPÉDIA. **MATLAB**. Flórida: Wikimedia Foundation, 2013. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=MATLAB&oldid=34608842>>. Acesso em: 18 abr. 2013.

WIKIPÉDIA. **UNICAST**. Flórida: Wikimedia Foundation, 2013. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Unicast&oldid=34521111>>. Acesso em: 11 abr. 2013.

WINETRAUB, Y.; BITAN, S.; HELLER, A.B. **Attitude determination – advanced sun sensor for pico-satellites**. Tel-Aviv, Israel: Handasaim School, 2010.

WOODMAN, O. J. **An introduction to inertial navigation**. Cambridge: University of Cambridge, 2007. Technical Report, Number 696.