

Ministério da
Ciência e Tecnologia



INPE-15732-TDI/1478

ANÁLISE DA ESTABILIDADE DE SISTEMAS DE CONTROLE ANALÓGICOS-DIGITAIS EM FUNÇÃO DO PERÍODO DE AMOSTRAGEM

Marcelo Ricardo Alves da Costa Tredinnick

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 27 de fevereiro de 2009.

Registro do documento original:
<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/02.09.14.45>

INPE
São José dos Campos
2009

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) 3945-6911/6923
Fax: (012) 3945-6919
E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO:
Presidente:
Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)
Membros:
Drª Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação
Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)
Drª Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)
Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)
Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)
Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)
BIBLIOTECA DIGITAL:
Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)
Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)
Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)
Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)
REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:
Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)
Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)
Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)
EDITORAÇÃO ELETRÔNICA:
Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
Ciência e Tecnologia



INPE-15732-TDI/1478

ANÁLISE DA ESTABILIDADE DE SISTEMAS DE CONTROLE ANALÓGICOS-DIGITAIS EM FUNÇÃO DO PERÍODO DE AMOSTRAGEM

Marcelo Ricardo Alves da Costa Tredinnick

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 27 de fevereiro de 2009.

Registro do documento original:
<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/02.09.14.45>

INPE
São José dos Campos
2009

Dados Internacionais de Catalogação na Publicação (CIP)

Tredinnick, Marcelo Ricardo Alves da Costa.
T714a Análise da estabilidade de sistemas de controle analógicos-digitais em função do período de amostragem / Marcelo Ricardo Alves da Costa Tredinnick. – São José dos Campos : INPE, 2009.
583p. ; (INPE-15732-TDI/1478)

Tese (Doutorado em Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009.
Orientador : Dr. Marcelo Lopes de Oliveira e Souza.

1. Controle de atitudes de satélites. 2. Estabilidade. 3. Sistemas de controles analógicos digitais. 4. Período de amostragem. 5. Falseamento. 6. Funções de Lyapunov. 7. Teoria das diferenças finitas. I.Título.

CDU 681.511.22

Copyright © 2009 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecánico, fotográfico, microfílmico, reprográfico ou outros, sem a permissão escrita da Editora, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2009 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, eletronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

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

Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle

Dr. Paulo Giácomo Milani

Presidente / INPE / SJCampos - SP

Dr. Marcelo Lopes de Oliveira e Souza

Orientador(a) / INPE / SJCampos - SP

Dr. Mario Cesar Ricci

Membro da Banca / INPE / SJCampos - SP

Dr. Gilberto da Cunha Trivelato

Convidado(a) / MECTRON / São José dos Campos - SP

Dr. Fernando José de Oliveira Moreira

Convidado(a) / EMBRAER / SJCampos - SP

Aluno (a): Marcelo Ricardo Alves da Costa Tredinnick

São José dos Campos, 27 de fevereiro de 2009

"*A base de uma ciência está em sua capacidade para predizer. Predizer significa dizer o que ocorrerá num experimento que nunca tenha sido feito antes*."

Richard Phillips Feynman
(1918-1988)

Dedico esta Tese a meus filhos Gabriel Vítor e Larissa,

a minha esposa Cláudia Helena

e a meu pai Dr. Felipe Edwin Tredinnick-Abasto (In Memoriam).

AGRADECIMENTOS

Ao Professor Marcelo Lopes de Oliveira e Souza, pelos seus extensos conhecimentos multidisciplinares a mim ensinados, pela sua excepcional orientação, por ter me iluminado o caminho correto para a obtenção dos resultados desta Tese e por ter sido uma forte e constante fonte de motivação e inspiração no decorrer do meu trabalho de Doutorado.

Ao Professor Gilberto da Cunha Trivelato, pelos seus esforços no sentido de conectar os avanços proporcionados nesta Tese ao cenário industrial contemporâneo, pelas valiosas observações no sentido de organizar os resultados desta Tese e pelo seu louvável espírito hospitaleiro.

Ao Professor Paulo Giácomo Milani pelas suas valiosas e precisas observações e contribuições que tanto ajudaram a valorizar esta Tese.

Ao Professor Mário Cezar Ricci pelas suas valiosas sugestões, recomendações e críticas que ajudaram a valorizar os resultados deste trabalho de Tese.

Ao Dr. Fernando José de Oliveira Moreira pelas suas valiosas observações, sugestões e críticas que ajudaram a aumentar a qualidade desta Tese.

À minha família por tudo o que representam para mim e o que fizeram para me apoiar nessa difícil, insegura e longa jornada de se fazer uma Tese de Doutorado. Agradeço especialmente à minha esposa Cláudia Helena da Costa. Agradeço também ao meu saudoso Pai, Embaixador Dr. Felipe Edwin Tredinnick-Abasto pelos seus valiosos ensinamentos e apoio ao longo de toda sua gloriosa e iluminada vida.

À Sra. Yolanda Ribeiro da Silva Souza, do Serviço de Informação e Documentação, pela dedicação quanto a adequação do texto desta Tese aos padrões internacionais de qualidade para publicação pelo INPE.

A Deus, por ter me dado força, iluminação e sorte para atingir os resultados desta Tese.

RESUMO

Neste trabalho é analisada a estabilidade de sistemas de controle analógicos-digitais em função do período de amostragem. Uma nova-regra de mapeamento s-z bilinear a dois parâmetros é proposta, testada, analisada e comparada com algumas regras clássicas. Propõe-se um modelo matemático para o falseamento ("aliasing"). Apresentaram-se Investigações quanto às fronteiras de estabilidade/instabilidade de sistemas de controle analógicos-digitais. Finalmente, por meio da teoria da estabilidade de Liapunov são propostos teoremas sobre a estabilidade do controle digital de uma planta analógica em função do período de amostragem.

STABILITY ANALYSIS OF ANALOG-DIGITAL CONTROL SYSTEMS as a FUNCTION OF THE SAMPLING PERIOD

ABSTRACT

In this work the stability of analog-digital control systems as a function of the sampling period is analyzed. A new rule of bilinear s-z mapping with two parameters is proposed, tested, analyzed and compared with some classical rules. A mathematical model for the aliasing is also proposed. Investigations on the stability/instability boundaries of analog-digital control systems is presented. Finally, by means of the Liapunov stability theory, theorems are proposed on the stability of a digital control of an analog plant as a function of the sampling period.

SUMÁRIO

Pág.

CAPÍTULO 1 - INTRODUÇÃO, OBJETIVOS, JUSTIFICATIVAS E ORGANIZAÇÃO DA TESE

1.1.Introdução 39
1.2. Objetivos deste trabalho 39
1.3. Justificativa deste trabalho 40
1.4. Organização deste trabalho 41
1.5. Originalidade, generalidade e utilidade deste trabalho 42

CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA

2.1 - Revisão bibliográfica principal 45
2.1.1. Controle digital de plantas analógicas 45
2.1.2. Regras clássicas de mapeamento analógico-digital 67
2.1.3. Equações diferenciais com lado direito descontínuo 68
2.1.4. Teoria do "lifting" contínuo 70
2.2 - Revisão bibliográfica segundária 78
2.2.1. Rasband (1990) 78

CAPÍTULO 3 - CONTROLE DISCRETO DE PLANTAS ANALÓGICAS COM APÊNDICES FLEXÍVEIS

3.1 - Introdução 83
3.2 - A Transformada-Z e equivalências discretas: filtros discretos 93
3.2.1. Integração numérica: mapeamento de equações a diferenças.....94
3.3 - Regras de mapeamento analógico-digitais com parâmetros de sintonia 103
3.3.1 - Regra de "ST1" a um parâmetro de sintonia …….....103
3.3.2 - Regra de mapeamento analógico-digital com dois parâmetros de sintonia..... 106

CAPÍTULO 4 - PROPOSTA DE MODELO MATEMÁTICO PARA O FALSEAMENTO

4.1 - Um modelo matemático para o falseamento …….....109
4.2 - Representação do modelo discreto da planta por multicaminhos em paralelo …..…113

CAPÍTULO 5 - DIAGRAMAS DE BODE E GRÁFICOS FASORIAIS

5.1 - Introdução à análise de sistemas de controle discreto de plantas analógicas 117
5.2 - Resultados numéricos obtidos com plantas de primeira e Segunda ordens utilizando controladores e mapeamentos diversos ..…….....….....125
5.2.1. Diagrama de Bode e gráfico fasorial para o caso da planta de primeira ordem e controle integral mapeado pelo método forward …….....128

5.2.2. Diagrama de Bode e gráfico fasorial para o caso da planta de primeira ordem e controle integral mapeado pelo método de Tustin ….....139
5.2.3. Diagrama de Bode e gráfico fasorial para o caso da planta de primeira ordem e controle usando rede de atraso de fase (rede "lag") mapeado pelo método forward …….....142
5.2.4. Diagrama de Bode para o caso da planta de primeira ordem e controle usando PD mapeado pelo método backward …….....145
5.2.5. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método backward …….....147
5.2.6. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método forward …….....147
5.2.7. Diagrama de Bode para o caso da planta de Segunda ordem e controle integral mapeado pelo método ST2 …….....150
5.2.8. Diagrama de Bode para o caso da planta de Segunda ordem e controle usando rede de atraso de fase ("lag") mapeado pelo método backward…….....151
5.2.9. Diagrama de Bode para o caso da planta de Segunda ordem e controle usando rede de atraso de fase (("lag") mapeado pelo método ST1 …….....152
5.2.10. Diagrama de Bode para o caso da planta de segunda ordem e controle usando rede de atraso de fase (“lag”) mapeado pelo método ST2 …….....153
5.2.11. Diagrama de Bode para o caso da planta de segunda ordem e controle PID mapeado pelo método de Schneider …….....154
5.2.12. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PID mapeado pelo método de Schneider.....154
5.2.13. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PID mapeado pelo método ST2 155
5.2.14. Diagrama de Bode para o caso da planta de primeira ordem e controle usando controle PD mapeado pelo método ST1 …….....156
5.2.15. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método schneider …….....157
5.2.16. Diagrama de bode para o caso da planta de segunda ordem e controle “lead-lag” mapeado pelo método backward ..…….....160
5.2.17. Diagrama de Bode para o caso da planta de segunda ordem e controle “lead-lag” mapeado pelo método forward …….....162
5.2.18. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle “lead-lag” mapeado pelo método ST1 …….....164
5.2.19. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle “lead-lag” mapeado pelo método ST2 166
5.2.20. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle “lead-lag” mapeado pelo método Tustin 168
5.2.21. Diagrama de Bode e gráfico fasorial para o caso da planta de segunda ordem e controle usando controle PD mapeado pelo método backward …….....…170
5.2.22. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PD mapeado pelo método ST2 …….....….....172
5.2.23. Diagrama de Bode para o caso da planta de primeira ordem e controle usando rede “lag” mapeado pelo método forward …….....….....173
5.2.24. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método ST2 ..…….....175

5.2.25. Diagrama de Bode para o caso da planta de primeira ordem e controle usando controle PID mapeado pelo método Tustin ….....177
5.2.26. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle derivativo mapeado pelo método ST2 …….....….....179
5.2.27. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle "lag" mapeado pelo método Tustin……...…….....180
5.2.28. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle "lead" mapeado pelo método Tustin ……..….....….....181
5.2.29. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PD mapeado pelo método ST1 …….....182
5.2.30. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle proporcional …….....183
5.2.31. Diagrama de Bode e gráfico fasorial para o caso da planta de primeira ordem e controle integral mapeado pelo método backward ….....184
5.2.32. Diagrama de Bode para o caso da planta de primeira ordem e controle usando controle integral mapeado pelo método ST1 ….....185
5.2.33. Diagrama de Bode para o caso da planta de primeira ordem e controle usando rede lag mapeado pelo método backward …….....186
5.2.34. Diagrama de Bode para o caso da planta de primeira ordem e controle usando rede lag mapeado pelo método Tustin …….....187
5.2.35. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método ST1 .....…….....188
5.2.36. Diagrama de Bode para o caso da planta de segunda ordem e controle com rede de avanço de fase ("lead") mapeado pelo método ST1 189
5.2.37. Diagrama de Bode para o caso da planta de segunda ordem e controle com rede de avanço de fase ("lead") mapeado pelo método ST2 ….....190
5.2.38. Diagrama de Bode para o caso da planta de segunda ordem e controle PD mapeado pelo método forward …….....191

CAPÍTULO 6 - FRONTEIRAS DA ESTABILIDADE/INSTABILIDADE PARA SISTEMAS DE CONTROLE ANALÓGICOS-DIGITAIS SISO

6.1 - Introdução …….....….....193
6.2 - Abordagem analítica de majorantes reais …….....….....194

CAPÍTULO 7 - TEOREMA DA ESTABILIDADE DO SISTEMA DE CONTROLE ANALÓGICO-DIGITAL

7.1 - Introdução …….....209
7.2 - Resultados numéricos …….....222

CAPÍTULO 8 - CONCLUSÕES, SUGESTÕES E RECOMENDAÇÕES ……....259

REFERÊNCIAS BIBLIOGRÁFICAS ……....…261

REFERÊNCIAS BIBLIOGRÁFICAS COMPLEMENTARES ……...….269

APÊNDICE A - CONTRIBUIÇÃO CLÁSSICA PARA O ESTUDO DO PROJETO DE CONTROLADORES DISCRETOS.....275

APÊNDICE B - APLICAÇÃO DO CRITÉRIO DE ESTABILIDADE DE JURY AO OSCILADOR HARMÔNICO SIMPLES COM CONTROLE PD PROJETADO POR TUSTIN …….....303

APÊNDICE C - SIMULAÇÕES NUMÉRICAS COM UM MODELO DO SATÉLITE CBERS-1 …….....…….....….....313

APÊNDICE D - COMPARAÇÃO DE DESEMPENHO ENTRE CONTROLADOR PID CLÁSSICO E PID ASTRÖM DA POSIÇÃO DE UM OSCILADOR HARMÔNICO MAPEADOS POR TUSTIN …….....…….....….343

APÊNDICE E - SIMULAÇÕES NUMÉRICAS COM AS REGRAS DE MAPEAMENTO – CONTROLE PD DA POSIÇÃO DE OSCILADOR HARMÔNICO E COMPARAÇÕES COM A REGRA DE TUSTIN, REGRA ST1 E REGRA ST2 …….....359

APÊNDICE F - APROXIMAÇÕES DA EXPONENCIAL DE MATRIZ POR McLAURIN-TAYLOR (MT) E PADÉ …….....….367

APÊNDICE G - DEFINIÇÕES BÁSICAS DE ANÁLISE FUNCIONAL …….....375

APÊNDICE H - PROGRAMAS EM MATLAB …….....….....387

LISTA DE FIGURAS

2.1 – Mapeamento Tustin……..…....…58
2.2 – "Lifting" de Yamamoto: nessa figura original do trabalho de Yamamoto "h" corresponde ao período de amostragem T.....……..…….....….....…71
2.3 - Operador *SH* interpretado geometricamente.....74
2.4 - Sistema $\sum_{CL}$ .híbrido em malha-fechada.....74
2.5 - Sinais do sistema híbrido 75
2.6 – Seção de Poincaré $(x, \dot{x})$ para o oscilador harmônico excitado por impactos periódicos independentes de ……..…….....80
2.7 – Seção de Poincaré $(x, \dot{x})$ para o oscilador harmônico excitado por impactos periódicos onde $f(x) = x^4$.....81

3.1 - Diagrama de blocos de um sistema com controle digital…….....…….....…...83
3.2 - Microprocessador Intel 4004…….....…….....85
3.3 - Interconexão do sistema MCS-4…….....…….....….....85
3.4 - Microcontrolador de Hyatt…….....……..….....86
3.5 - Amostragem de um sinal…….....…….....87
3.6 - Espectro normalizado de Fourier de um sinal de entrada com largura de banda finita…….....88
3.7 - Sinal amostrado no domínio da frequência…….....…..….....89
3.8 - Representação analógica em malha-fechada…….....…..90
3.9 - Equivalência digital da Figura 3.8…….....90
3.10 - Sistema digital manipulado…….....….....90
3.11 - A/D generalizado…….....…..….....91
3.12 - D/A generalizado…….....….91
3.13 - Simplificação do modelo digital generalizado……..…....….....…92
3.14 - Equivalente discreto do sistema analógico……..….....….92
3.15 - Região de convergência para a equação 3.1…….....….93
3.16 - Região de convergência da equação 3.2……...….....94
3.17 - Integrador "backward" no domínio do tempo…….....….95
3.18 - Mapeamento do plano s para o plano z pela a equivalência "backward".....96
3.19 - Mapeamento "backward" atuando como integrador em z…….…….....96
3.20 - Equivalência "backward" atuando como integrador em k.T…….....….....…97
3.21 - Integrador "forward" no domínio do tempo……...…….....97
3.22 - Mapeamento do plano s para o plano z pela a aproximação "forward"….....98
3.23 - Aproximação "forward" atuando como integrador…….....…..….....….99
3.24 - Aproximação "forward" atuando como integrador……..….....….99
3.25 - Integrador "Tustin" no domínio do tempo…….....….....100
3.26 - Mapeamento entre os planos s e z pelo integrador "Tustin"…….....….....101
3.27 - Diagrama em blocos básico do integrador por "Tustin"…….....101
3.28 - "Tustin" como integrador.....102

3.29 - Mapeamentos das regras ST1, Tustin e “Backward”……..….....105
3.30 - Mapeamentos das regras ST2 (região hachurada) e Tustin…….....108

4.1 - Sistema de controle digital de uma planta analógica…….....….....109
4.2 - Processo de amostragem sem falseamento……...…….....…110
4.3 - Gráfico ω_d *x* ω_a …….....111
4.4 - Modelo de multicaminhos em forma fechada…….....113
4.5 - Identificação do falseamento no modelo de multicaminhos (blocos contidos dentro da linha tracejada)…….....114
4.6 - Identificação do falseamento no modelo de multicaminhos de forma aberta (blocos contidos dentro da linha tracejada)…….....115

5.1 - Modelo do sistema de controle amostrado……...….....118
5.2 - Margens de ganho e fase de um diagrama de Bode……...….....126
5.3 - Margens de ganho e fase de um diagrama de Bode em função do período de amostragem.....…….....127
5.4 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$. As linhas tracejadas horizontais (cor verde) indicam múltiplos ímpares de -180^o….....128
5.5 - Detalhe do diagrama de Bode para |H*(s)|=1 e fase = -180°, T=1 segundo.....129
5.6 - Detalhe dos diagramas de Bode para T=0,5; 1 e 1,5 segundos; $0.5 \le \omega \le 1$.....130
5.7 - Diagrama fasorial para T=1 segundo…….....131
5.8 - Diagrama fasorial para T=1 segundo…….....131
5.9 - Diagrama fasorial para T=1 segundo…….....132
5.10 - Diagrama fasorial para T=1 segundo…….....132
5.11 - Diagrama fasorial para T=1 segundo…….....133
5.12 - Diagrama fasorial para T=1 segundo…….....133
5.13 - Diagrama fasorial para T=1 segundo…….....134
5.14 - Diagrama fasorial para T=1 segundo…….....134
5.15 - Diagrama fasorial para T=1 segundo…….....135
5.16 - Diagrama fasorial para T=1 segundo…….....135
5.17 - Diagrama fasorial para T=1 segundo…….....136
5.18 - Diagrama fasorial para T=1 segundo…….....136
5.19 - Diagrama fasorial para T=1 segundo…….....…137
5.20 - Diagrama fasorial para T=1 segundo…….....…137
5.21 - Diagrama fasorial para T=1 segundo…….....…138
5.22 - Diagrama fasorial para T=1 segundo…….....….....…138
5.23 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$; a fase de D* fixou-se em 90°…….....139
5.24 - Detalhe do diagrama de Bode para |H*(s)|=1 e fase = -180°, T=2 segundos.....140
5.25 - Diagrama fasorial para T=2 segundos…….....….....140
5.26 - Diagrama fasorial para T=2 segundos…….....141
5.27 - Diagrama fasorial para T=2 segundos…….....141

5.28 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ ……..…....142
5.29 - Detalhe do diagrama de Bode para |H*(s)|=1 e fase = -180^{o}, T=2,22 segundos….....143
5.30 - Diagrama fasorial para T=2,22 segundos…….....….143
5.31 - Diagrama fasorial para T=2,22 segundos…….....….144
5.32 - Diagrama fasorial para T=2,22 segundos…….....….144
5.33 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=1 segundo…….....145
5.34 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$,com T=1,5 segundos……..….....….....146
5.35 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$,com T=0,5 segundo…….....146
5.36 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=1,89 segundos……...…….....147
5.37 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2 segundos……..…….....148
5.38 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2 segundos…….....149
5.39 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=1,48 segundos…….....150
5.40 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=0,35 segundos…….....151
5.41 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=0,77 segundos…….....152
5.42 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=1,23 segundos…….....…..153
5.43 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=1,15 segundos…….....154

5.44 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=1,65 segundos…….....155
5.45 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=1,07 segundos…….....156
5.46 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,86 segundos…….....157
5.47 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,6 segundos…….....158
5.48 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,6 segundos…….....159
5.49 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,63 segundos…….....160
5.50 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,63 segundos…….....161
5.51 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,01 segundos…….....162
5.52 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,01 segundos…….....163
5.53 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,11 segundos…….....164
5.54 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,11 segundos…….....165
5.55 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,97 segundos…….....166

5.56 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,97 segundos…….....…….....……167
5.57 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,97 segundos…….....…168
5.58 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,86 segundos…….....……169
5.59 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,01 segundos…….....170
5.60 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,01 segundos…….....….171
5.61 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,23 segundos…….....172
5.62 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,22 segundos…….....173
5.63 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,22 segundos…….....174
5.64 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,25 segundos…….....….....175
5.65 - Detalhe do diagrama de Bode para H*(s) para |H*(s)|=1 e fase = -180^o, em T=3,25 segundos…….....….....176
5.66 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=4,82 segundos…….....177
5.67 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=4,82 segundos…….....….178
5.68 - Detalhe do diagrama de Bode para H*(s) para |H*(s)|=1 e fase = -180^o, em T=4,82 segundos…….....178

5.69 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=4,71 segundos…….....…..179
5.70 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3,92 segundos…….....…180
5.71 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3,92 segundos…….....…181
5.72 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2,1 segundos…….....182
5.73 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3.92 segundos…….....….....183
5.74 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=6 segundos…….....184
5.75 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=6,67 segundos…….....185
5.76- Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T= 5,71 segundos…….....186
5.77 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T= 5,72 segundos…….....…187
5.78 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T= 5,39 segundos…….....…188
5.79 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T= 5,21 segundos…….....….....…189
5.80 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T= 6,09 segundos…….....190

5.81 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T= 5,22 segundos…….....191

6.1 - Prova da equivalência da norma “sup” do segurador-amostrador (“*sampler-hold*”) ao período de amostragem T, no caso igual a 2 segundos 197
6.2 - Teste de convergência pelo método da razão, em função de k e ω, para qualquer T: obtivemos L < 1.....200
6.3 - Teste de convergência pelo método da razão para T = 1 segundo obtivemos L<1.....201
6.4 - Teste de convergência pelo método da razão para T = 7.5 segundos obtivemos L > 1, isto é, a série do falseamento diverge.....202
6.5- Teste de convergência com T = 4 segundos.....203
6.6 - Teste de convergência com T = 5 segundos.....204
6.7 - Teste de convergência do falseamento função de do período de amostragem T e da planta $G(s)=\frac{1}{s^2+as+b}$ 205
6.8 - Teste de convergência com $G(s)=\frac{1}{s^2+as+b}$ mostrando que para T=6,5 segundos L > 1.....206

7.1 - Representação geométrica da estabilidade usando o conceito de raio espectral de $(e^{A_c.T})' P e^{A_c.T}$ onde T é o período de amostragem do sistema de controle digital, A_c é a matriz de dinâmica do sistema e P é uma matriz positiva-definida.……...….....217
7.2 - Resultados numéricos do Teorema 7.1 apresentando regiões aproximadas de estabilidade/instabilidade como função do período de amostragem.....223
7.3 - Resultados numéricos do Teorema 7.2 apresentando regiões aproximadas de estabilidade/instabilidade como função do período de amostragem T e do parâmetro ξ 224
7.4 - Mesmo resultado numérico da figura anterior visualizado por ângulo diferente.....225
7.5 - Resultados numéricos do Teorema 7.2 apresentando regiões aproximadas de estabilidade/instabilidade como função do período de amostragem T e do parâmetro ξ (vista superior da figura anterior).....226
7.6 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 0.5 segundo em função dos parâmetros ξ_1 e ξ_2 227
7.7 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 0.5 segundo em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....228
7.8 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1 segundo em função dos parâmetros ξ_1 e ξ_2 229

7.9 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1 segundo em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....230
7.10 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1,5 segundos em função dos parâmetros ξ_1 e ξ_2 231
7.11 - Resultados numéricos do Teorema 6.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1,5 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....232
7.12 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 3 segundos em função dos parâmetros ξ_1 e ξ_2 233
7.13 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 3 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....234
7.14 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 4 segundos em função dos parâmetros ξ_1 e ξ_2 235
7.15 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 4 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....236
7.16 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 5 segundos em função dos parâmetros ξ_1 e ξ_2 237
7.17 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 5 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....238
7.18 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10 segundos em função dos parâmetros ξ_1 e ξ_2 239
7.19 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....240
7.20 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 20 segundos em função dos parâmetros ξ_1 e ξ_2 241
7.21 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 20 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....242

7.22 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 30 segundos em função dos parâmetros ξ_1 e ξ_2 243
7.23 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 30 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....244
7.24 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 60 segundos em função dos parâmetros ξ_1 e ξ_2 245
7.25 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 60 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....246
7.26 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 100 segundos em função dos parâmetros ξ_1 e ξ_2 247
7.27 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 100 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....248
7.28 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 140 segundos em função dos parâmetros ξ_1 e ξ_2 249
7.29 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 140 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....250
7.30 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 500 segundos em função dos parâmetros ξ_1 e ξ_2 251
7.31 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 500 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....252
7.32 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1000 segundos em função dos parâmetros ξ_1 e ξ_2 253
7.33 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1000 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....254
7.34 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10000 segundos em função dos parâmetros ξ_1 e ξ_2 255

7.35 - Resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10000 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.....256
7.36 - resultados numéricos do Teorema 7.3. Regiões claras correspondem ás regiões em que o sistema em malha-fechada apresentou estabilidade assintótica e nas regiões escuras, a instabilidade, para ξ_1 no eixo das absissas e ξ_2 no eixo das ordenadas.....257

A.1 - Diagrama de blocos do sistema analógico em malha fechada…….....….....279
A.2 - Diagrama de blocos do oscilador harmônico controlado por PD discreto no tempo…….....….....280
A.3 – "Root-locus" no plano-s para o sistema totalmente analógico da Figura A.1…….....….....283
A.4 - Resposta ao impulso unitário para o sistema totalmente analógico….....283
A.5 – "Root-locus" no plano-z para T = 0,1 segundos…….....….....…284
A.6 - Resposta ao impulso unitário para T = 0,1 segundos: claramente temos a estabilidade para este caso…….....285
A.7 – "Root-locus" no plano-z para o controle PD discreto por "Tustin", com T = 1,6 segundos: observar o pólo fora do círculo unitário, representando um fator instabilizante…….....….....286
A.8 - Resposta ao pulso unitário para T = 1,6 segundos…….....….....287
A.9 – "Root-locus" do oscilador-harmônico para T = 0,58 segundos….....….....…288
A.10 - Resposta ao pulso unitário para a configuração de pólos do "root-locus" da Figura B.9…….....288
A.11 – "Root-locus" do oscilador-harmônico para T_S = 0,59 segundos….....…289
A.12 - Detalhe instabilizador da Figura A.11…….....290
A.13 - Resposta ao pulso unitário relativa aos dois últimos "root-loci": Figuras A.11 e A.12…….....…290
A.14 – "Root-locus" no plano-z para T = 0,1 segundos…….....….....291
A.15 - Resposta ao impulso unitário para T = 0,1 segundos…….....291
A.16 – "Root-locus" no plano-z para T = 1,6 segundos…….....292
A.17 - Resposta ao pulso unitário para T = 1,6 segundos…….....292
A.18 – "Root-locus" para o caso instável do limiar de estabilidade……...….....293
A.19 - Resposta ao pulso unitário para o "root-locus" da Figura A.18…….....293
A.20 – "Root-locus" do oscilador harmônico amortecido…….....294
A.21 - Resposta ao pulso unitário para o "root-locus" da Figura A.20…….....…294
A.22 – "Root-locus" com o Z.O.H. do oscilador harmônico sobre-amortecido controlado por PD projetado com a regra deTustin…….....296
A.23 – Root-locus para o Z.O.H. do oscilador harmônico sobre-amortecido controlado por PD projetado com a regra deTustin…….....….....297
A.24 – Root-locus para o Z.O.H. do oscilador harmônico sobre-amortecido controlado por PD projetado com a regra deTustin…….....297
A.25 – "Root-locus" do oscilador harmônico amortecido com PD projetado por Schneider…….....298
A.26 – Resposta (instável) ao pulso unitário…….....….....299

A.27 – “Root-locus” do oscilador harmônico amortecido com PD por Schneider…….....299
A.28 – Resposta (instável) ao pulso unitário…….…….....300
A.29 – “Root-locus” do oscilador harmônico não-amortecido com PD por Schneider…….....…..300
A.30 – Resposta (instável) ao pulso unitário…….....301

B.1 – Gráfico de $f_1(T)$…….....…...309
B.2 – Gráfico de $f_2(T)$…….....…...309
B.3 – Gráfico de $f_3(T)$……...…….....…310
B.4 – Gráfico de $f_1(T)$…….…….....311
B.5 – Gráfico de $f_2(T)$…….…….....311
B.6 – Gráfico de $f_3(T)$…….....……..312

C.1 – Eixos de atitude para a simulação com um modelo do satélite CBERS-1….....314
C.2 – Sistema originalmente desenvolvido por Silva (1997) para o caso totalmente analógico (planta e controle), agora com controle discreto no tempo para atender às simulações de atitude…….....318
C.3 – Bloco de controle nos três eixos…….....….318
C.4 – Estrutura interna do bloco de controle da Figura D.3…….....319
C.5 – Detalhe para o controle de “roll”……...….....319
C.6 – Detalhe do controle de “pitch”…….....…..320
C.7 – Detalhe do controle de “yaw”…….....…..320
C.8 – Detalhe de construção de uma “s-function” gráfica…….....….320
C.9 – “S-function” gráfica de teste…….....321
C.10 – “S-funtion” desagrupada…….....…321
C.11 – Ângulos de atitude para o controle PD analógico da planta analógica (CBERS-1)…….....322
C.12 – Sinais de controle de atitude…….....322
C.13 – Ângulos de atitude…….....……323
C.14 – Sinais de controle de atitude……...….....…..323
C.15 – Ângulos de atitude…….....…324
C.16 – Sinais de controle de atitude…….....325
C.17 – Ângulos de atitude…….....……325
C.18 – Sinais de controle de atitude……...….....326
C.19 – Ângulos de atitude…….....…327
C.20 – Sinais de controle de atitude…….....327
C.21 – Ângulos de atitude…….....…328
C.22 – Sinais de controle de atitude……...….....328
C.23 – Ângulos de atitude…….....…329
C.24 – Sinais de controle de atitude…….....329
C.25 – Ângulos de atitude…….....330
C.26 – Sinais de controle da atitude……...….....330
C.27 – Ângulos de atitude…….....……331
C.28 – Sinais de controle de atitude…….....331
C.29 – Ângulos de atitude…….....332

C.30 – Sinais de controle de atitude……..…….....……...332
C.31 – Ângulos de atitude…….....333
C.32 – Sinais de controle de atitude…….....333
C.33 – Ângulos de atitude…….....334
C.34 – Sinais de controle de atitude……..….....334
C.35 – Ângulos de atitude…….....….335
C.36 – Sinais de controle de atitude…….....….335
C.37 – Ângulos de atitude…….....….336
C.38 – Sinais de controle de atitude…….....….336
C.39 – Ângulos de atitude…….....337
C.40 – Sinais de controle de atitude…….….....337
C.41 – Ângulos de atitude……...….....….340
C.42 – Sinais de controle de atitude……..….....….340

D.1 – Controlador PID Aström analógico interferindo no comportamento da planta G(s), também analógica…….....343
D.2 – Implementação em Simulink da estrutura clássica do controlador PID digital interferindo no comportamento da planta analógica…….....344
D.3 – Implementação em Simulink da estrutura do controlador PID Aström digital interferindo no comportamento da planta analógica…….....….344
D.4 – oscilador harmônico excitado por um pulso de duração finita…….....….345
D.5 – diagrama em blocos do controlador PID Aström analógico…….....346
D.6 – Resultados numéricos para a posição x(t) (setor superior da Figura), sinal de controle (setor intermediário da Figura) e pulso de duração 0,01 segundos.....348
D.7 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....348
D.8 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….349
D.9 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….....349
D.10 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....350
D.11 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….....350
D.12 -Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….....351
D.13 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....…….....351
D.14 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....352
D.15 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….....352
D.16 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....353
D.17 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….353

D.18 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….354
D.19 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….354
D.20 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....355
D.21 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....355
D.22 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….356
D.23 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos…….....….....356

E.1 – Resultados numéricos para T = 0,35 segundo…….....….....363
E.2 – Resultados numéricos para T = 1 segundo…….....…..363
E.3 – Resultados numéricos para T = 3 segundos…….....364
E.4 – Resultados numéricos para T = 4,5 segundos…….....364

F.1: aproximações para e^{-t}…….....371
F.2: erro na aproximação da exponencial…….....……371

G.1 - Classificação das funções $f : A \to B$ ……....….....376
G.2 - Pontos interiores de A……...…….....380
G.3 - Ponto de Acumulação x……...….....380
G.4 **– I**nterpretação geométrica de conjunto denso…….....382
G.5 - Raio espectral em função do período de amostragem numa simulação numérica. Para a notação adotada neste trabalho troca-se o ρ da figura por r….....384

LISTA DE TABELAS

B.1 - TABELA DE JURY GENERALIZADA……...…….....….....304

C.1 - MODOS DE VIBRAÇÃO PARA UM MODELO ASSIMÉTRICO DO CBERS-1.....317

D.1 - ANÁLISE DA ESTABILIDADE EM FUNÇÃO DO PERÍODO DE AMOSTRAGEM.....357

E.1 - CONSTANTES USADAS NAS SIMULAÇÕES NUMÉRICAS.....361
E.2 - RESULTADOS NUMÉRICOS OBTIDOS 362

LISTA DE SÍMBOLOS

T - Período de Amostragem

ζ - Razão de Amortecimento (letra grega pronunciada “zéta”)

ω_N - Freqüência Natural Não-Amortecida

k_p - Ganho Proporcional

k_d - Ganho Derivativo

ξ - Coeficiente de Ajuste do denominador da regra ST_1 (letra grega pronunciada “csi”)

ξ_1 - Coeficiente de Ajuste do numerador da regra ST_2

ξ_2 - Coeficiente de Ajuste do denominador da regra ST_2

$\tilde{C}$ - Espaço das funções contínuas por partes

H - Operador "hold" (segurador)

HS - Operador "sampler and hold" (amostrador e segurador)

$x(t)$ - Função contínua em t (tempo)

x_k - Função discreta em k.T

k - $k \in N^*$

λ_∞ - Espaço de Lebesgue das funções discretas

L_∞ - Espaço de Lebesgue das funções contínuas

$\mathcal{L}^{-1}$ - Transformada de Laplace inversa

$\mathcal{Z}$ - Transformada Z

:= - Igual por definição

LISTA DE SIGLAS E ABREVIATURAS

SISO - Modelo com uma Única Entrada e uma Única Saída

A/D - Conversor Analógico-Digital

D/A - Conversor Digital-Analógico

ST - "Sampling Time"

Qubit - "Quantum bit"

CAPITULO 1

1 INTRODUÇÃO, OBJETIVOS, JUSTIFICATIVAS E ORGANIZAÇÃO DA TESE

1.1. Introdução

Neste trabalho é analisada a estabilidade de sistemas de controle analógico-digitais em função do período de amostragem. Uma nova-regra de mapeamento s-z bilinear a dois parâmetros é proposta, testada, analisada e comparada com algumas regras clássicas. Um modelo matemático para o falseamento ("aliasing") também foi proposto. Investigações quanto às fronteiras de estabilidade/instabilidade de sistemas de controle analógicos-digitais é apresentada. Finalmente, por meio da teoria da estabilidade de Liapunov propõem-se teoremas sobre a estabilidade do controle digital de uma planta analógica em função do período de amostragem.

1.2. Objetivos deste trabalho.

Este trabalho tem como objetivo a análise da estabilidade de sistemas de controle analógicos-digitais ("híbridos") em função do período de amostragem. Para atingi-lo, este trabalho propõe-se a gerar como resultados intermediários, no mínimo os seguintes itens:

1. Propor duas novas regras de mapeamento SZ (também chamado de "digital redesign" em **Yamamoto (1994)**, **Hori (1996)**, e em **Kanai (1999)**) que seja de uso geral, que seja melhor que as da literatura (inclusive a citada em **Tredinnick (1999)**). Desenvolver um processo de análise e testes rigorosos com essa eventual nova regra, buscando extrair dela informações importantes a respeito da estabilidade.Mediante um modelo matemático para o problema do mascaramento (falseamento, "aliasing") propor um "majorante" para servir de nível decisório entre eventuais regiões de estabilidade e instabilidade.

2. Apresentar uma análise quanto às fronteiras da estabilidade/instabilidade de sistemas híbridos como uma função do período de amostragem e também da planta usada.

3. Propor, por meio da teoria da estabilidade de Liapunov um meio de analisar a estabilidade de um sistema de controle digital de uma planta analógica sendo desejável o desenvolvimento de teoremas a respeito. Simulações numéricas eventualmente foram executadas e a função exponencial de matriz foi numericamente processada por meio de aproximações via regra de Padé.

1.3. Justificativas deste trabalho.

Torna-se relevante, seja na academia ou na indústria, o estudo de sistemas híbridos dessa natureza, dada a forte tendência de se utilizar controladores digitais (entes físicos cuja dinâmica é discreta no tempo, quantizada na amplitude, e cujas operações de entrada, processamento e saída gastam tempo e produzem atrasos) para guiar/orientar os mais diversos tipos de plantas (entes físicos cuja dinâmica é continuamente definida no tempo, continuamente definida na amplitude e, usualmente, não introduzem atrasos). Isso se dá principalmente devido aos benefícios do controle digital tais como o progressivo aumento do poder de processamento e simultânea diminuição do custo, a capacidade de reprogramação de algoritmos computacionais, do baixo consumo de energia elétrica exigido pelo controlador e da utilização de estruturas leves. Entretanto, existem problemas que surgem com a inserção do controlador digital, como é amplamente citado na literatura (**Frankllin (1981)** e **Tredinnick (1999)**) tais como o mascaramento (falseamento ou “aliasing”), o ruído de quantização, e os atrasos entre a entrada e a saída computacionais).

Entretanto, na literatura atual inexiste um tratamento a respeito da análise da estabilidade de sistemas híbridos de controle limitando-se a estudos a respeito da teoria de controle digital *per si*, onde a dinâmica da planta é discretizada por meio de algum extrapolador discreto, segundo algum critério ou regra de mapeamento SZ (“forward”, “backward”, Tustin, “impulse invariance” (“pole-zero mapping”), “step invariance” (Z.O.H. – “Zero Order Hold”)) desprezando-se assim a informação entre as amostras (provenientes dos Conversores Analógico-Digitais). Essa mutilação da informação da dinâmica da planta, originalmente contínua no tempo, é tecnicamente interpretada como o mascaramento (“aliasing”) e, tal como

mostrado em **Tredinnick (1999)**, pode vir a instabilizar o sistema de controle para valores relativamente altos do período de amostragem quando comparados a valores usuais, para certos tipos de plantas, p.ex.: com modos flexíveis.

Além disso, os atrasos de processamento podem piorar ainda mais o problema do mascaramento e, quanto mais complexo for o algoritmo de controle, maior será o tempo gasto para processá-lo, imaginando um mesmo ambiente computacional em tais eventuais comparações. Naturalmente, o emprego de algoritmos de controle simples torna-se uma clara necessidade industrial. Felizmente tais algoritmos usualmente consomem poucos recursos de processamento computacional e assim não geram grandes atrasos de processamento.

Uma outra necessidade de projeto é a minimização de controle, ou economia de combustível, essencial principalmente para sistemas autônomos, o que força a uma utilização econômica do combustível usado (por ex.: jatos de gás) prolongando a vida útil da máquina (satélite, robô, etc.), o que torna então justificável a utilização de controladores digitais operando com períodos de amostragem elevados.

Assim, avançar nesse obscuro, fascinante e imprevisível novo mundo da teoria de controle, onde se deve utilizar pequenos algoritmos de controle discreto suportando operações com elevados períodos de amostragem e, mesmo assim, garantindo a estabilidade da dinâmica analógica (continuamente definida no tempo, continuamente definida na amplitude) da planta empregada pode vir a proporcionar a criação de uma nova metodologia de projeto de algoritmos de controle para sistemas de controle digitais embarcados.

1.4. ORGANIZAÇÃO DESTE TRABALHO.

Este trabalho está organizado em sete capítulos adicionados de dois apêndices ao final do trabalho. No Capítulo 2 é realizada uma revisão bibliográfica das referências mais relevantes para este trabalho a respeito de Controle Digital de Plantas Analógicas, Regras Clássicas de Mapeamento Analógico-Digital e Teoria

da Estabilidade de Liapunov. No Capítulo 3 são apresentadas as regras de mapeamento analógico-digital com um e com dois parâmetros de sintonia sendo apresentadas as devidas elaborações teóricas. No Capítulo 4 é estudado um modelo matemático para o falseamento (“aliasing”) resultando numa representação em blocos por multicaminhos em paralelo. No Capítulo 5 apresentou-se diagramas de Bode e gráficos fasoriais tendo como base a teoria ensinada em Franklin (1981) a respeito das funções “estreladas” {.}*(s) usando controladores tais como o proporcional, derivativo, integral, proporcional+derivativo (PD), PID, avanço de fase (“Lead”), atraso de fase (“Lag”) e avanço-atraso de fase (“Lead-Lag”) e cujos mapeamentos usados foram: forward, backward, Tustin, ST1, ST2 e Schneider. No Capítulo 6 são estudadas as fronteiras da estabilidade/instabilidade para sistemas analógico/digitais SISO. No Capítulo 7 são apresentados outros Teoremas da estabilidade, Lemas e Corolários para um sistema de controle analógico-digital. Em tal Capítulo 6 utilizou-se aproximações da exponencial de matriz por meio da regra de Padé de terceira ordem até a décima-sexta ordem nas simulações numéricas. Finalmente, no Capitulo 8 são feitas conclusões, sugestões e recomendações. No Apêndice A é apresentada uma revisão baseada na contribuição clássica para o estudo dos projetos de controladores discretos. No Apêndice B é realizada uma aplicação do critério da estabilidade de Jury ao oscilador harmônico com controle PD (Proporcional+Derivativo) mapeado pela regra de Tustin. No Apêndice C foram realizadas simulações numéricas com um modelo do Satélite CBERS-1. Os resultados obtidos no Apêndice C para o controle discreto da atitude do satélite artificial representam a origem experimental na necessidade de se investigar novas regras de mapeamento visto que todas as regras clássicas faharam. No Apêndice D é realizada uma comparação de desempenho entre os controladores PID clássico e PID Aström quando à posição de um oscilador harmônico. O Apêndice E contém diversas simulações comparativas de desempenho do controle PD de posição de um oscilador harmônico amortecido com as regras Tustin, ST1 e ST2. O Apêndice F descreve aproximações da exponencial de matriz por meio de McLaurin-Taylor e Padé mostrando claramente que a regra de

Padé deve ser utilizada. O Apêndice G possui diversas definições elementares de análise funcional úteis especialmente para os Capítulos 5 e 6. O Apêndice H contém discersos "*scripts"* em MatLab utilizados nas simulações desta Tese. Finalmente, nas Referências Bibliográficas constam textos/documentos da literatura úteis no fornecimento das bases teóricas e experimentais relevantes para a tese.

1.5. ORIGINALIDADE, GENERALIDADE E UTILIDADE DESTE TRABALHO.

O presente trabalho é fruto de incessante estudo investigativo que buscou encontrar metodologias de projeto e/ou análise de sistemas de controle satisfatórias para o caso em que a planta a controlar é analógica e o controle é em tempo discreto. A presente literatura do assunto modela uma planta analógica como discreta no tempo, mas é importante lembrar que esta sempre será analógica, na realidade, sendo o modelo matemático empregado apenas um artifício matemático usado na análise de tais sistemas. Dentre toda a literatura técnica da área (Referências Bibliográficas) inexistem colidências nem similaridades com o apresentado nesse presente trabalho. O presente trabalho apresenta no decorrer de seus capítulos análises inspiradas na teoria de controle digital (fundamentalmente na teoria ventilada por Franklin, 1981, Balas, 1982, Naylor, 1982 e Tredinnick, 1999) em alguns casos e em outros casos continuações/avanços na pesquisa de engenharia de controle discreto disposta na literatura, porém até então não realizada / publicada por outros. Todos os Lemas, Teoremas e Corolários desenvolvidos pelo autor desta Tese também inexistem na literatura presente do assunto.

O grau de aplicabilidade da teoria e dos resultados aqui apresentados é de caráter geral visto que fundamentalmente trata de metodologias de projetos de controle discretos no tempo com a estabilidade condicionada ao valor do período de amostragem, ou faixas desse, que mediante a variação de certos parâmetros de controle os tornam aplicáveis a qualquer tipo (ou ao menos uma vasta quantidade) de plantas analógicas. Essa nuance de estudo também é muito mais realista, além de geral, visto que na prática temos o controlador digital interagindo com a planta

analógica, e não “tudo discreto” (controle discreto via regra de mapeamento SZ e planta discretizada por meio de algum tipo de extrapolador) como se encontra na literatura.

A aplicabilidade de grande parte desse trabalho pode se dar por meio do capítulo de análise da estabilidade por meio de diagramas de Bode, o capítulo investigativo sobre o falseamento (“aliasing”) que apresentou resultados quanto a estabilidade de um sistema realimentado como função da planta usada e do período de amostragem, o capítulo em que são propostos avanços sobre o trabalho de Balas (1982) usando as regras ST1 e ST2, bem como fazer uso de resultados numéricos diversos localizados em outros capítulos e nos apêndices. Entretanto, todos os resultados aqui dispostos mostram-se úteis não apenas por serem importantes “benchmarks” para projetos (dado que muitos estudos foram realizados tendo como base osciladores harmônicos de primeira e segunda ordens), apesar de estudos envolvendo um modelo para o satélite artificail CBERS-1 também terem sido realizados com êxito. Finalmente, os resultados deste trabalho são importantes passos rumo ao encontro de uma sólida e futura teoria de controle digital de plantas analógicas.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

2.1. REVISÃO BIBLIOGRÁFICA PRINCIPAL

2.1.1. CONTROLE DIGITAL DE PLANTAS ANALÓGICAS.

Um primeiro problema de um controle digital é a amostragem no tempo, especialmente quando se tratam de estruturas flexíveis. As estruturas oscilam, na teoria, com infinitos *modos de vibração* (máxima componente espectral de freqüência $\omega_{máx} \rightarrow \infty$), mas o conversor A/D amostra o sinal com uma freqüência finita chamada frequência de amostragem ω_S, pequena quando comparada com as maiores freqüências de vibração da estrutura (Ginter, 1978; Tredinnick, 1997). Em casos práticos onde se considera o amortecimento intrínseco dos materiais pode-se considerar que existe um valor finito, porém elevado, para os modos de vibração. Essa interação entre o conversor A/D e a estrutura provoca então uma perda drástica de informação, pelo falseamento (mascaramento, distorção, mutilação ou "aliasing") destas freqüências e pelas oscilações escondidas. O fenômeno do falseamento ocorrerá pois a conhecida freqüência de Nyquist $\omega_N = \omega_S / 2$ será menor que a máxima componente espectral de freqüência $\omega_{máx}$, logo, o sinal amostrado sofrerá distorções por sobreposições de inúmeras freqüências.

Isto pode ser melhor visualizado mediante uma análise espectral: quando ω_N for menor que $\omega_{máx}$ (ver Capítulo 4) parte das bandas de freqüência do sinal amostrado se sobrepõem causando assim o mascaramento. O fenômeno das oscilações escondidas até poderá ocorrer porque ω_N será menor que $\omega_{máx}$, mostrando que toda frequência da planta que for múltipla de ω_N poderá não ser vista pelos sensores, pois possuem banda de passagem limitada, tal como também os conversores A/D.

Estas distorções podem ser praticamente eliminadas impondo-se que a máxima componente espectral $\omega_{máx}$ de um sinal analógico seja bem menor que a freqüência de Nyquist $\omega_N = \pi/T$ (isto é, $\omega_{máx} << \omega_N$) que é a metade da freqüência de amostragem $\omega_S = 2\pi/T$. Isso pode ser conseguido colocando-se filtros passa-baixas nas saídas dos sensores de um sistema de controle em malha-fechada, por exemplo.

Entretanto, quando se tratar de controlar uma estrutura flexível, que tem infinitas freqüências de vibração (modos de vibração), isto é, quando $\omega_{máx} \rightarrow \infty$, teremos que fazer uso de um filtro que vai apenas considerar um certo número de modos de vibração da estrutura flexível, o que pode comprometer o controle. Mas, mesmo assim, considerando-se que as freqüências mais elevadas possuam pequenas amplitudes e razoáveis amortecimentos, elas podem não ser tão críticas ao sistema podendo assim ser negligenciadas. Atualmente formas econômicas de se resolver esse tipo de problema estão em fase de estudo por todo o mundo.

Um segundo problema de um controlador digital se refere aos atrasos no tempo. As conversões A/D e D/A podem demorar, por exemplo até vários microsegundos, o que pode ser desastroso para aplicações que tenham sinais que variem com freqüências superiores a 1 MHz, por exemplo. Somado a estes atrasos característicos dos conversores está o atraso de processamento causado pela execução do programa de computador da simulação. O problema é que todas as operações de entrada, processamento, e saída de dados em um controlador digital têm duração finita e não nula, causando atrasos no tempo e portanto, defasagens crescentes com as freqüências nos vários modos, que pioram o desempenho do controle destes modos até levá-los à instabilidade. Este efeito só é limitado pelo amortecimento passivo da estrutura. Este tipo de problema não será abordado neste trabalho.

Um terceiro problema de um controlador digital é a quantização em amplitude, cujos erros podem ser minimizados se aumentarmos o número de níveis de quantização. Mas isto é um problema sério, pois aumentar o número de níveis de

quantização implica em aumentar o número de bits processados para cada amostra de sinal e portanto, o aumento do tempo de conversão A/D o que resultaria em um tempo maior de processamento, ou a necessidade de microprocessadores mais velozes, encarecendo a missão espacial. Mesmo assim, um controle digital com um pequeno ruído de quantização (uma grande quantidade de níveis de quantização) poderia ser ainda melhor que um controle analógico sujeito a ruídos imprevisíveis que distorceriam muito mais os sinais do sistema, o que poderia provocar ações de controle incorretas. Este tipo de problema também não será estudado neste trabalho.

Muitos estudos abordando tais problemas foram realizados e estão publicados na literatura. Alguns dos mais importantes estudos abordando diretamente o problema do controle digital bem como estudos teóricos periféricos úteis na compreensão de tal problema serão apresentados abaixo.

Von Neumann (1950) estudou a implementação numérica, junto de R.D. Richtmeyer, das equações diferenciais parciais referentes a sistemas hidrodinâmicos na presença de colisões ("shocks"). **Von Neumann (1950)** detectou o problema do falseamento ("aliasing"), sem no entanto tornar claro que se tratava do falseamento, analisando a definição da grade (domínio para elaboração do gráfico) com intervalos de espaço e tempo Δx e Δt, respectivamente, no processo de discretização da equação diferencial parcial do sistema hidrodinâmico utilizado: selecionando Δx e Δt suficientemente pequenos temos resultados muito precisos, próximos da solução real, entretanto, se o comprimento da onda que se simula for da ordem de Δx sempre ocorrem "falsificações" com a resposta numérica (na nomenclatura de Von Neumann). Concluiu inicialmente Von Neumann que para evitar tais "falsificações" (ou falseamentos) é necessário que Δx e Δt sejam pequenos o suficiente. O interessante deste trabalho é o estabelecimento de uma condição de estabilidade numérica como função linear da razão entre Δx e Δt, ou mais precisamente, $L = {s_0 \Delta t}/{\Delta x}$ onde s_0 é a velocidade do som no líquido e a condição de estabilidade

usada foi $L \leq \gamma^{1/2} / 2c$, onde γ é uma constante diretamente relacionadas com as propriedades de compressibilidade do fluido e c é uma constante real representando os choques hidrodinâmicos (onde no caso foi usado c = 1).

Em **Jury (1965)** foi estudada a estabilidade em malha-fechada usando modulação por pulsos na teoria e na prática (implementando inclusive um circuito eletrônico à válvula) tendo claras aplicações em telecomunicações e controle discreto não-linear de servomecanismos. Foram apresentados resultados buscando oferecer a base de um método para encontrar o ponto de equilíbrio do sistema em malha-fechada em questão em relação à função de transferência em malha-fechada.

Dahlin (1968a) estudou o projeto e sintonização (ajuste paramétrico) de controladores digitais utilizando o método "backward" como aproximação da equação diferencial de estados. Foi estudada a simulação computacional de um sistema de controle em malha fechada em função de um parâmetro λ de sintonia, o qual corresponde a uma componente do ganho de controle (PI). Em **Dahlin (1968b)** o problema e o método desenvolvido em **Dahlin (1968a)** foi estendido para o caso multivariável e aplicado ao problema de controle de uma fábrica de produção de papel. **Dahlin (1968a)** apresentou resultados numéricos para a resposta impulsiva para sinal de saída ("basis weight") onde pôde observar a degradação do sinal quando aumentou o período de amostragem de 35 segundos para 100 segundos, sem conseguir explicar o porque dessa degradação, apenas rotulando erroneamente tal degradação de "redução da estabilidade".

Neste trabalho estudou-se a teoria da amostragem segundo **Ayres (1959)** (quanto aos aproximantes numéricos de equações diferenciais); **Ziemer e Tranter (1976)**; **Barradas (1980)**; **Franklin e Powell (1981)**; **Katz (1981)**; **Lathi (1983)**; **Houpis e Lamont (1985)**; **Morari e Zafiriou (1985 e 1986)**; **Ogata (1987)**; **Isermann (1989)**; **Tredinnick (1999),** mostrando matematicamente seus problemas que são verificados na prática.

Da literatura foram elencadas ao menos quatorze referências consideradas importantes a respeito de álgebra linear e análise funcional muito úteis para que seja possível compreender conceitos avançados de estabilidade envolvendo teoria espectral tal como os apresentados em **Filippov (1964)**, **Meirovitch (1980 (b))**, **Barabanov (1989)**, **Maciejowski (1989)**, **Toivonen (1993, 1996 (a,b,c,d), 1998)**, **Akhmetov (1995, 1994, 1992)**; **Francis (1988)**, **Sesekin (2000, 1996, 1994)**, **Oishi (2001, 1997a, 1997b)**, **Yamamoto (2002)**, **Chen (2008)**. Livros que abordam de forma precisa e didática conceitos fundamentais a respeito de álgebra linear são **Fadeeva (1959)**, **Oden (1979)** e **Lax (1997, 2002)**, com destaque para os dois últimos. **Lorch (1962)** é uma referência clássica de análise espectral a qual apresenta de forma muito didática princípios básicos tais como subespaços e ortogonalidade, o teorema de Hahn-Banach (o qual conhecendo uma subestrutura de um determinado operador pode-se compreender a estrutura toda do operador, sob determinadas condições), teoria espectral de transformações lineares e a estrutura espectral de operadores auto-adjuntos. **Sobolev (1963)** é uma referência histórica de análise funcional que apresenta a definição formal do espaço de derivadas que leva o nome do autor, noções elementares a respeito de funcionais lineares, derivadas generalizadas, problema do autovalor, e equações quasilineares. **Zarantonello (1971)** fornece um estudo geral de projeções em conjuntos convexos em espaço de Hilbert, projeção esta sendo um mapeamento de um dado ponto ao mais próximo ponto num conjunto convexo fechado, abordando projeções em cones convexos. Tal teoria pode ser útil para compreender e estender os resultados obtidos em **Thompson (1982)**, onde é ensinado que setores cônicos convexos podem ser usados para representar dispositivos de controle não-lineares e variantes no tempo, onde no caso foi estudado um sistema de controle híbrido, i.e., parte analógico e parte discreto no tempo. **Oden (1979)** buscou abordar quase todo o assunto de análise funcional proporcionando uma visão detalhada e didática tanto de álgebra linear quanto de análise funcional utilizando uma linguagem mais adequada aos engenheiros e físicos e apresentando muitos exemplos práticos. **Lax (1997)** representa um dos melhores e mais didáticos textos sobre álgebra linear abordando brilhantemente

assuntos tais como: dualidade, mapeamentos lineares, teoria espectral, dinâmica e cinemática, convexidade , espaços normados e o teorema dos círculos de Gershgorin. Tal referência ofereceu no capítulo de “Teoria Espectral” a melhor explicação matemática para a amplamente utilizada equação do autovalor (problema do autovalor-autovetor), dentre as demais referências. **Lax (1997)** também apresenta a interessante interpretação de John Von Neumann a respeito do fenômeno conhecido como “avoidance of crossing” em que autovalores de uma matriz quadrada A(t) de dimensão n, em função do tempo, apresentam uma rota de colisão e por um instante de tempo insignificante mudam suas trajetórias automaticamente, evitando a “colisão”. Tal fenômeno foi descoberto pelos físicos no início do desenvolvimento da mecânica quântica e o mesmo ocorre devido à trajetória de um autovalor λ_1 estar num espaço n-dimensional o qual naturalmente evita interseccionar uma superfície dependente de n-2 parâmetros, onde situa-se um outro autovalor λ_2. Peter Lax foi agraciado em 2005 pela Academia Norueguesa de Ciências com a medalha Abel pelas suas contribuições para a teoria da estabilidade de equações diferenciais parciais tendo como principal fruto o seu “princípio da equivalência” **(Lax, 1956)**, o qual estabelece as condições pelas quais a implementação numérica dá uma aproximação válida para a solução de uma equação diferencial parcial. Em **Lax (2002)** temos um texto completo e bastante didático sobre análise funcional abordando assuntos diverrsos tais como: espaços lineares normados, espaços de Hilbert, estudos sobre convergência, exemplos de espectro de operadores, tratamento introdutório completo sobre a teoria espectral de operadores unitários, normais e simétricos, teoria das distribuições. **Balakrishnan (1976)** representa um texto que só deve ser estudado após a correta compreensão e a devida assimilação teórica básica contida, por exemplo, em **Naylor (1982)**, **D’Ambrosio (1977)**, **Lax (2002)** e **Oden (1979)**. **Balakrishnan (1976)** já inicia seu texto com as propriedades dos espaços de Hilbert, passando pelas noções de conjuntos convexos e programação convexa com aplicações à teoria de controle ótimo (LQR e problemas de controle envolvendo otimização do tempo). **D’Ambrosio (1977)** oferece uma excelente introdução às topologias geral e algébrica bem como à análise funcional (em seus

últimos capítulos), servindo como referência básica para aventuras teóricas em textos mais avançados. **Kantorovich (1982)**, prêmio Nobel de economia em 1975 pelas suas contribuições para a teoria da alocação ótima de recursos, escreveu este avançado texto de análise funcional o qual possui um preciso e compreensível tratamento matemático de diversos temas tais como: espaços vetoriais topológicos, espaços normados, funcionais lineares e equações funcionais de segundo tipo com deduções a respeito do espectro. **Naylor (1982)** representa um texto recomendado para o ensino e pesquisa em engenharia do mesmo nível que **Oden (1979)**, porém um pouco melhor organizado, i.e., com as estruturas topológica e algébrica dos operadores claramente identificadas e separadas no início para depois serem combinadas nos estudos de espaços métricos tais como espaços de Banach e de Hilbert. **Naylor (1982)** possui uma parte do livro dedicada à análise espectral com a apresentação de importantes teoremas e exemplos clássicos, com ótimas ilustrações. Talvez uma das partes mais importantes de **Naylor (1982)** seja a "*part B – The Spectrum*", onde o autor define os conjuntos componentes do "spectrum" tais como: espectro pontual (que consiste no conjunto de autovalores), espectro contínuo e o espectro residual, bem como a noção de conjunto resolvente, os quais todos juntos formam o "spectra". **Naylor (1982)** ensina didaticamente a calcular o "spectra" bem como representá-lo geometricamente para diversos operadores tais como o operador a diferenças finitas e o operador diferencial, bem como elaborar o "spectra" de uma equação diferencial ordinária de primeira ordem bem como de um operador de integração. **Hislop (1991)** representa um texto avançado sobre teoria espectral o qual deve ser iniciado somente se a literatura básica sobre o assunto recomendada aqui for muito bem compreendida. Em **Hislop (1991)** métodos geométricos de análise de operadores lineares é apresentada onde o "spectrum" de um operador linear é apresentado através de propriedades tais como o conjunto resolvente, projeções de Riesz e seqüências de Weyl e Zhislin. Em **Hislop (1991)** é apresentada a teoria geral da estabilidade espectral onde é estudado o comportamento semiclássico de autovalores de operadores de Schrödinger com potenciais que crescem até o infinito. A teoria da estabilidade

espectral abordada em **Hislop (1991)** apresenta um importante teorema sobre a estabilidade de um determinado operador A cujo autovalor λ será "estável" caso $\|(A-\lambda I)\underline{v}\| \geq \varepsilon \|\underline{v}\| \geq 0$ for satisfeita, onde $\underline{v}$ é autovetor de A e $\varepsilon > 0$. Apesar de **Arveson (2002)** não fazer referência direta à conceitos importantes como estabilidade e convergência, o conteúdo ensinado nessa referência apresenta-se com uma notação moderna e um tanto avançado perante os demais aqui estudados. **Arveson (2002)** representa uma referência moderna a respeito de teoria espectral fornecendo diversas ferramentas matemáticas para o estudo do espectro de operadores em espaços de dimensão infinita, oferecendo assim as bases matemáticas para a compreensão tanto da mecânica quântica quanto das teorias da estabilidade de sistemas de controle de dimensão infinita. **Moura (2002)** consiste em uma referência com caráter essencialmente prático, de aplicação, dos diversos conceitos de análise funcional com aplicações em modelagem matemática e simulação computacional, com destaque para as noções de convergência de seqüências e para o teorema da projeção. **Moura (2002)** apresenta diversos teoremas e aplicações quanto à convergência e a estabilidade de seqüências apresentando resumidamente o fundamental trabalho de **Lax (1956)** provando que ambos conceitos são equivalentes.

LePage (1961), **Jury (1964)**, **Oppenheim (1975)**, **Muth (1977)**, **Kuhfittig (1978)**, e **Lam (1979)** são excelentes referências básicas a respeito de variáveis complexas, transformadas de Fourier, Laplace e Z bem como de filtros analógicos e digitais que tanto podem servir a estudos básicos quanto avançados possibilitando o aprendizado aprofundado e preciso da teoria de controle analógico e também da teoria de controle digital.

Deets (1972) desenvolveu um estudo a respeito do projeto e ensaios de vôo iniciais do primeiro sistema digital "fly-by-wire" de uma aeronave. O sistema, o qual empregou componentes do sistema de guiagem utilizados no programa APOLLO da NASA, foi instalado num avião F-8. Em **Deets (1972)** o computador de guiagem do módulo lunar é o elemento central de controle digital, multímodo, canal único e em três eixos. Neste trabalho um exemplo interessante de síntese de controle

amostrado é apresentado. Dentre suas principais conlusões podemos destacar: i) é possível e prático implementar leis de controle analógicas na forma digital do controle de vôo; ii) as vantagens de usar funções lógicas na implementação da ação integral de controle foram demosnstradas; iii) a verificação do software de controle digital de vôo exige grande esforço e não pode ser subvalorizada; iv) a habilidade de se realizar alterações paramétricas na lei de controle sem ter que substituir o hardware pode ser descrita com o uma grande vantagem prática.

Ortega (1966) estudou a discretização e diferenciação de operadores com aplicações ao método de Newton. Neste trabalho demostrou-se que na solução numérica de equações da forma $F\underline{x} = \underline{0}$ a discretização da equação e a aplicação do método de Newton resulta no mesmo sistema algébrico linear o que levou ao desenvolvimento de um problema geral para determinar quando duas freqüentemente usadas operações de discretização dadas e a diferenciação de Frechet serão comutativas. **Ortega (1973)** investigou a estabilidade (exponencial) e convergência de equações a diferenças finitas ordinárias em processos iterativos empregando a teoria da estabilidade de Liapunov. Neste trabalho uma equação diferencial ordinária é discretizada pelo método “backward” (método de Euler) e formas equivalentes de garantia da estabilidade usando funções de Liapunov que foram usadas no caso analógico são usadas para o caso discreto. **Ortega (1973)** também investigou o chamado domínio de atração para a evolução iterativa de uma equação a diferenças finitas $x_{k+1} = Gx_k$, sendo G uma matriz, formulando um Teorema a respeito do conjunto correspondente a tal domínio de atração, chamado “D”, e demonstou que se G puder ser aceito como um mapeamento contínuo e existir um ponto fixo x^* para o processo iterativo e se tal x^* for atrativo, então D será obrigatoriamente um conjunto aberto. **Ortega (1973)** também investigou erros de arredondamento e atratores instáveis.

O’Shea (1967) considerou o estudo da estabilidade assintótica de certo tipo de sistema amostrado contendo não-linearidade monotônica simples onde em seu sistema SISO utilizou um equivalente discreto G(z) para a planta G(s) e uma função representando a não-linearidade monotônica do sistema. Os resultados

obtidos se referem ao estudo da estabilidade de sistemas amostrados frente à tal não-linearidade encontrada na malha e não em relação ao período de amostragem.

Boykin (1975), motivado pelas técnicas de guiagem e controle de mísseis e veículos espaciais, desenvolveu uma metodologia de análise a respeito de sistemas amostrados que faziam uso de “multirate-multiloop”. Um método para a determinação da função de resposta de um sistema amostrado “multirate” com taxas de amostragem não-inteiras foi apresentado. Tal método foi baseado na decomposição de uma matriz de operadores utilizada e demonstrou-se que com tal método pode-se realizar cálculos mais velozes, economizando tempo de processamento.

Katz (1975) representa um importante estudo a respeito da seleção da taxa de amostragem para um sistema de controle da vibração das asas de um avião. O propósito deste estudo não foi recomendar qualquer método de projeto de controladores e sim um método para a seleção da taxa de amostragem para sistemas incluindo estruturas flexíveis, como no caso de um avião. A seleção de taxas de amoastragem adequadas para o controlador digital da aeronave é um compromisso que depende de muitos fatores tais como custo e precisão, tal como reconhece **Katz (1975)**. Fatores os quais forçam a baixar a taxa de amostragem são os custos dos conversores A/D e D/A, que seriam também reduzidos. Fatores os quais forçam a elevar a taxa de amostragem são: i) exigências de largura de banda em malha-fechada e resposta no tempo; ii) sensibilidade à variação de parâmetros; e iii) efeitos deletérios de distúrbios de origem aleatória. **Katz (1975)** desenvolveu seu trabalho baseando-se na síntese discreta de um regulador linear ótimo e observador de estados e verificou que se a freqüência de um modo de vibração real do veículo estudado variasse em relação ao pré-estabelecido no modelo, i.e., aumentando o erro em relação ao modo, isto poderia gerar realimentação positiva, compreendida em seu trabalho como a geração de razão de amortecimento negativa do tal modo de vibração, instabilizando o sistema.

Katz (1975) sugeriu um método geométrico para selecionar a taxa de amostragem, visando contornar o problema.

Ziemer (1976) e **Lathi (1983)** fornecem ao leitor um arcabouço teórico aprofundado a respeito de teoria das comunicações; mas, o que é relevante a este trabalho compreende apenas as seguintes seções : a) **Lathi (1983)**: representação de uma função periódica pela série de Fourier em todo o intervalo, existência da transformada de Fourier; o teorema da amostragem, medida de informação, capacidade de canal – lei de Hartley-Shannon (teorema central da teoria da informação); b) **Ziemer (1976)**: séries de Fourier, a transformada de Fourier, teoria da amostragem e o capítulo sobre teoria da informação e codificação.

Ginter (1978) representa uma outra ótima referência para uma introdução ao tema controle digital de estruturas flexíveis. Neste trabalho de mestrado encontra-se uma introdução ao estudo de modelagem newtoniana de estruturas flexíveis e uma introdução ao controle digital. Ginter (1978) apenas constatou o fenômeno da instabilidade em alguns modos e nos sinais de saída devido à inserção de um controlador digital, sem propor nenhuma maneira de solução do problema, usando um modelo de viga de Euler-Bernoulli como estrutura flexível. Em seu trabalho podemos encontrar os efeitos da instabilidade em alguns modos e nos sinais de saída de um sistema controlado digitalmente comparado a um sistema controlado analogicamente.

No trabalho de **Ohkami et alli (1979)** pode-se encontrar resultados de simulações com “hardware” de controle digital de satélites com apêndices flexíveis. Tal trabalho, entretanto, executa as simulações para apenas um modo de vibração além do modo rígido e considerando apenas um eixo de atitude. Como política de controle digital utilizou-se um controlador Proporcional e Derivativo (PD) com um estimador de estados. A comparação de desempenho foi executada em termos da estabilidade e da quantidade de memória exigida e do tempo de processamento. Foram apresentados apenas os resultados estáveis para tais simulações.

Barradas (1980) é uma boa referência para inúmeros elementos teóricos e práticos para a teoria da comunicação e telecomunicações analógicas e digitais. Nesta referência pode-se citar alguns elementos da teoria da informação úteis para o aprofundamento no conhecimento da teoria de controle digital: velocidade de sinal, geração da informação (informação própria e entropia), canais de comunicação com e sem ruído. Também pode-se destacar estudos sobre a teoria das comunicações: teoria da amostragem, amostragem ideal, amostragem real, amostragem de sinais super-rápidos, aproveitamento do tempo de transmissão, modulação em portadora trem-de-pulsos – modulação digital, codificação / decodificação.

Meirovitch (1980a) estudou a digitalização de sistemas de controle analógicos empregando um observador de estados. Neste trabalho de **Meirovitch (1980)** mostrou-se que uma versão discretizada de um sistema de controle analógico, ou simplesmente o controlador, pode ser usada, entretanto podem haver alterações nas características dinâmicas do sistema devido à escolha do período de amostragem.

Franklin (1981) pode ser considerada como uma das melhores introduções ao controle digital que se pode encontrar dentre as demais referências citadas. Esta referência trata desde a introdução teórica até alguns elementos da fronteira atual do conhecimento sobre o tema, podendo-se considerá-la como um texto básico ao mesmo tempo que avançado sobre controle digital. Podemos encontrar nela os elementos teóricos fundamentais da transformada–Z, teoria da amostragem, fenômenos do mascaramento e das oscilações ocultas, projetos de filtros digitais por integradores numéricos, técnicas de projeto de filtros por mapeamento pólo-zero, equivalência “Hold”, filtros de Butterworth e ITAE, “root-locus” discreto, projetos de controladores digitais utilizando métodos tais como transformada-W e de Ragazzini, controlabilidade e observabilidade de sistemas digitais, estimadores digitais, filtro de Kalman, fórmula de Ackermann, quantização, controle digital multivariável, dentre muitos outros itens. Alguns detalhes de pesquisas atuais sobre a teoria de seleção da taxa de amostragem tais como: conceito de

rugosidade e a seleção da taxa de amostragem considerando incerteza nos parâmetros também são mostrados. Tais pesquisas se apresentam em sua fase embrionária podendo-se rotular tal teoria da seleção da taxa de amostragem como “em aberto”, isto é, ainda necessitando de uma enorme quantidade de resultados teóricos e práticos para seu pleno estabelecimento. **Franklin (1981)** desenvolveu um capítulo inteiramente dedicado a sistemas amostrados, i.e., estruturas discretas no tempo que podem ser representadas matematicamente por meio da transformada de Laplace, por exemplo G*(s) é a representação amostrada de G(s), sem ser obrigatório ter que trabalhar com a transformada-Z por onde temos de discretizar todo o sistema. Tal técnica permite trabalhar algebricamente com sistemas amostrados, i.e., que possuem interdependente e simultaneamente subsistemas analógicos e subsistemas em tempo discreto.

Katz (1981) é uma outra excelente referência sobre o tema. Sua introdução teórica não é tão detalhada e profunda quanto a encontrada em **Franklin (1981)** ou em **Ogata (1987)** mas são os resultados de pesquisas experimentais descritos nesta referência que realmente despertam a atenção. Alguns capítulos interessantes são: a) mecanização de algoritmos de controle em microcontroladores, b) análise e implementação de algoritmos numéricos: são apresentados os resultados do trabalho pioneiro de **Shenberg (1977 e 1978)** na implementação de um controlador digital num microprocessador; fala-se sobre o efeito da propagação do ruído de quantização através do sistema; c) seleção da taxa de amostragem: basicamente, corresponde ao mesmo que é exposto em **Franklin (1981)** exceto pela inclusão dos seguintes itens: pré-filtros “anti-aliasing”, seleção prática da taxa de amostragem, e o conceito de fidelidade da resposta proposto por **Peled (1978)**. Entretanto, o conceito de função rugosidade é melhor explorado aqui nesta referência do que em **Franklin (1981)**.

Balas (1982) investigou o controle em tempo discreto de grandes estruturas espaciais empregando realimentação de velocidade, método DVFB (“Direct Velocity Feedback”), desenvolvido em **Balas (1979)**, no qual a informação da velocidade é negativamente realimentada. Entretanto, resultados a respeito da

estabilidade utilizando tal método só eram conhecidas na forma analógica, necessitanto de um estudo na área de controle digital. **Balas (1982)** discretizou sua lei de controle analógica por meio da aproximação “backward” (método de Euler) e assim foi possível obter uma desigualdade, apresentada no Capítulo 6, na qual estabelece uma estrutura condicional entre o período de amostragem e os diversos parâmetros do sistema de forma a garantir a estabilidade, caso a tal desigualdade não seja violada.

Hansen (1983) reconsiderou o problema do falseamento (“aliasing”) quanto à identificação de parâmetros de um processo estocástico analógico a partir de dados amostrados no tempo. Era bem conhecido que processos estocásticos (analógicos) com covariância estacionária sua função densidade espectral não podia, em geral, ser inferida a partir de observações discretas equiespaçadas. **Hansen (1983)** chamou tal problema da identificação de parâmetros de “aliasing phenomenon”. Neste trabalho o desenvolvimento uma matriz racional de densidades espectrais foi utilizada para minimizar o problema sob determinadas circunstâncias.

Glasson (1983) estudou aplicações de controle digital com múltiplos períodos de amostragem, muito conhecido como “multirate digital control”, útil em aplicações complexas onde é necessário utilizar mais de um processador na malha. **Glasson (1983)** apresenta uma retrospectiva histórica dos principais trabalhos a respeito de controle digital focando essencialmente no problema do “multirate”, e em especial apresenta o método de Rattan para o projeto de controle em “multirate-multiloop”. **Glasson (1983)** também apresenta equivalentes em diagramas de blocos da estrutura “multirate”, bem como apresenta como exemplos o piloto automático do ônibus espacial e o controle “fly-by-wire” de um avião F-18.

Em **Morari (1985)** são feitas várias análises de desempenho de algoritmos de controle em tempo discreto para sistemas “SISO” mediante suas características transitórias de resposta tais como: tempo de subida, sobressinal e tempo de acomodação. Mediante tais análises um algoritmo é proposto buscando combinar as vantagens dos demais algoritmos e tornar-se livre de seus inconvenientes. Em

Morari (1986) a estabilidade de um sistema de controle digital e sua performance em termos de saídas analógicas é estudada desconsiderando erros no modelamento, numa primeira etapa, e os considerando, numa segunda etapa. Ainda em **Morari (1986)** os efeitos do período de amostragem sobre o desempenho de controles em tempo discreto e as propriedades de robustez de um sistema apresentado são examinados e os resultados são utilizados como critério de seleção da taxa de amostragem.

Ogata (1987) representa uma boa referência básica para o tema controle discreto, apresentando em riqueza de detalhes todos os princípios básicos a respeito do tema. É nessa referência que se pode encontrar um tratamento detalhado a respeito do critério de estabilidade de Jury.

Diduch (1987) estudou um controle amostrado linear quadrático (LQ) o qual utiliza o valor do período de amostragem como um parâmetro de projeto. Este trabalho baseia-se na minimização de um determinado índice de desempenho *J(T)* como função do período de amostragem T. Em análise das condições de contorno **Diduch (1987)** concluiu que quando $T \to 0$ o controlador ótimo discreto converge para o controlador ótimo analógico linear e invariante no tempo.

Francis (1988) desenvolveu um estudo a respeito da estabilidade de sistemas amostrados expressando matematicamente o problema do controle discreto de uma planta analógica e demonstrando que a margem de ganho pode ser arbitrariamente deslocada para diversos valores de período de amostragem, visando garantir a estabilidade. O ponto de partida deste trabalho baseou-se na exploração na relação entre os zeros de um sistema analógico e sua versão discretizada. Explicando numa forma mais simples, **Francis (1988)** apresentou um método para reposicionar pólos instáveis de certas plantas analógicas em versões discretizadas estáveis evidenciando a dependência do período de amostragem na planta discretizada.

Hagiwara (1988) mostra o que poderia ser considerado elementar na teoria de controle digital: que um sistema amostrado permanece estável se o subsistema

analógico for estável e se os períodos de amostragem utilizados forem pequenos o suficiente. Entretanto, **Hagiwara (1988)** mostrou na conclusão de seu trabalho que caso o subsistema analógico não for estável algumas condições suficientes podem ser aplicadas para a garantia da estabilidade do sistema amostrado, considerando entretanto sempre a condição de que o período de amostragem tem que ser suficientemente pequeno.

Em **Trivelato (1988)** foi realizado um estudo pioneiro no Brasil a respeito de sistemas de controle digitais do torque de rodas de reação, usadas no controle da atitude de satélites artificiais, usando modelos de referência, cujo objetivo foi reduzir os efeitos dos: i) torques perturbadores na roda e ii) ruídos nos seus sensores, sobre o torque produzido pela roda. À época do desenvolvimento de tal trabalho o Brasil seguia progredia com a Missão Espacial Completa Brasileira (MECB), iniciada em 1980, oriundo do Programa Espacial Brasileiro, este, por sua vez, iniciado em 1961. Uma das missões iniciais da MECB foi o desenvolvimento, lançamento e a operação de um satélite artificial de observação da Terra e neste almejava-se um sistema de controle de atitude ativo em três eixos e constituído por sensores, uma unidade eletrônica de estimação e controle e atuadores, incluindo nestes as rodas de reação para controle fino da atitude. Assim, mostrou ser muito importante o estudo do controle de torque de cada roda de reação com precisão tanto maior quanto menores fossem os torques exigidos pelo satélite visando assim o bom funcionamento de tal sistema. Cinco anos após a conclusão o trabalho de **Trivelato (1988)**, em 09 de fevereiro de 1993, o Brasil tornou operacional com sucesso o satélite SCD-1 (Satélite de Coleta de Dados I) em uma órbita circular a 750 km de altitude. Em **Trivelato (1988)** apresentou-se o estudo e o desenvolvimento de sistemas de controle digitiais de torque de rodas de reação usando modelos de referência, com dois controladores: proporcional (P) e proposcional mais integral (PI), ambos com realimentação de velocidade. Em tal trabalho foram realizadas simulações digitais computadorizadas e ensaios de software e hardware em laboratório. Para ambos controladores testados foram obtidos resultados satisfatórios quanto à redução dos efeitos de distúrbios de torque de baixa freqüência na roda, eliminando sua média e reduzindo seu desvio

padrão. Entretando, ambos controladores não reduziram muito intensamente os distúrbios com freqüências superioresa aproximadamente $\frac{1}{10}$ da sua freqüência de amostragem, talvez causados pelo atrito nos mancais. De uma comparação realizada entre a simulação computacional e os resultados obtidos em laboratório com software de controle e roda de reação concluiu-se que: i) o erro médio de velocidade da roda em regime permanente é diferente de zero no primeiro caso (P) e praticamente nulo no segundo (PI); ii) o desvio padrão no caso de PI é praticamente uma ordem de grandeza superior ao caso P, visto que o acréscimo da ação integral provoca um correspondente acréscimo na soma dos resíduos da função de transferência de malha fechada; iii) a diferença entre as correntes para os dois controladores é muito pequena pois o atrito viscoso é pequeno e as velocidades são próximas; iv) o controlador P apresenta resposta mais veloz que o controlador PI, porém apresenta uma ação de controle (corrente) com variações mais bruscas; v) o controlador PI absorve totalmente um distúrbio constante no sistema de controle da roda de reação enquanto o controlador P o faz apenas parcialmente. Assim, a estrutura de controle por modelo de referência mostrou-se eficiente essencialmente pela simplicidade do modelo usado, o que exige pouco tempo de processamento e é baseado num parâmetro praticamente constante: o momento de inércia do volante mais rotor do motor. **Trivelato (1988)** compreendeu que o método de resposta em freqüência mostrou-se adequado na escolha do tipo de controlador, mas que não é suficiente para a escolha dos parâmetros de tais controladores. **Trivelato (1988)** alertou para a importância do problema da amostragem no tempo em sistemas de controle, e de forma complementar chamou a atenção para problemas eventualmente associados com a quantização em amplitude teorizando a respeito da elaboração de novos critérios relacionando estes parâmetros (amostragem no tempo e quantização em amplitude) e os parâmetros do controlador. Para a escolha do período de amostragem as prioridades dos critérios mudam conforme os recursos disponíveis para a sua implementação e para a escolha dos níveis de quantização foram satisfatórios os critérios considerando: i) o modelo de distribuição uniforme para os ruídos de quantização; ii) o teorema de Parseval para a propagação de ruídos em

sistemas discretos e iii) a hipótese de que o ruído de torque provocado na saída devido ás quantizações tem também distribuição uniforme. Para a escolha dos parâmetros do controlador PI foi usado o método do lugar das raízes (“root locus”) e foi necessário também definir o período de amostragem do sistema de controle de atitude do satélite antecipadamente para os dois casos. **Trivelato (1988)** alertou que as maiores desvantagens do sistema de controle da roda simulada e implementada foram: i) aumento da complexidade da eletrônica do sensor; ii) exigência do emprego de palavras (codificadas de comprimento finito) grandes; iii) dificuldades de conciliar o período de amostragem e o nível de quantização (da velocidade); e iv) não possibilita a rejeição de distúrbios e ruídos com freqüências superiores a aproximadamente $1/10$ da freqüência de amostragem do sistema de controle da roda.

Barabanov (1989) estudou a estabilidade das soluções de sistemas de controle contínuos-discretos, no caso se referindo a controles com partes analógica e discreta, por meio do segundo método de Liapunov, propondo uma generalização de tal método por meio de teoremas. Neste trabalho apenas considerações qualitativas profundamente abstratas a respeito da estabilidade foram realizadas.

Isermann (1989) representa outra referência importante a respeito do tema Controle Digital especialmente no que se refere às seções que tratam da análise de estabilidade mediante o posicionamento dos pólos no plano-z e do critério de estabilidade de Schur-Cohn-Jury.

Juang (1990) estudou sistemas contínuos-discretos mostrando dentre vários resultados qualitativos mediante o cálculo do raio espectral de certas matrizes não negativa-definidas. **Juang (1990)** apresentou um teorema de sua autoria no qual afirma que para um sistema dessa natureza ser considerado estável os autovalores da sua matriz de dinâmica devem estar contidos em círculos (ou discos) determinados pelo raio espectral de uma outra matriz. Curiosamente, tal teorema poderia ser interpretado como uma extensão ao teorema de Gershgorin, tal como apresentado em **Lax (1997)**.

Schneider (1991) apresentou um trabalho cuja essência foi a derivação de mapeamentos de ordem elevada que não sofressem problemas de estabilidade. Baseou-se por razões de estabilidade na família de fórmulas de integradores numéricos de Adams-Moulton para desenvolver seus mapeamentos. O procedimento matemático foi relativamente muito simples: apenas aplicar a transformada-Z nas fórmulas de Adams-Moulton. As fórmulas de integração numérica parabólica e cúbica foram as escolhidas para gerar a regra de Schneider-Kaneshige-Groutage (SKG). Nesse trabalho a regra SKG se apresenta como um ótimo integrador quando comparada com regras de grau inferior, trabalho inclusive elogiado por **Bose (1988)**. Entretanto, conforme mostrado em **Tredinnick (1999)** tal regra falha ao ser utilizada não como integrador numérico, mas como uma ação derivativa de controle visto que gera um pólo instabilizante fora do círculo unitário no plano Z.

Hori (1992) propôs um método denominado PIM ("Plant Input Mapping") para discretização de sistemas analógicos busca selecionar a taxa de amostragem dos conversores A/D e D/A de forma que a mesma seja "não-patológica", i.e., que não venha a instabilizar o sistema amostrado, tendo para isso que ser a taxa de amostragem diferente de múltiplos das freqüências naturais da planta. A vantagem deste método foi garantir a estabilidade do sistema de controle amostrado. **Hori (1992)** percebeu que o "uso de valores grandes de período de amostragem leva a degradação no desempenho do sistema de controle amostrado", porém, sem explicar o porquê disso.

Levine (1996) representa outra importante referência básica para o estudo da teoria de controle apresnetando um texto completo a respeito de assuntos tais como a análise e o projeto de sistemas de controle não-lineares e sistemas de controle dedicados a sistemas aeroespaciais, incluindo o controle de atitude de forma bastante diática.

Oishi (1997a, 2001) chamou de "conjectura" a intuição dos engenheiros de que o desempenho de um sistema de controle digital se aproxima até se igualar ao desempenho de um sistema analógico quando o período de amostragem tende à

zero. Em seus estudos, mostrou que existe um caso especial no qual isso não é verdade. **Oishi (1997a, 2001)** demonstrou ser impossível garantir a convergência para todas as plantas, entretanto, concentrou-se em uma classe restrita de plantas de onde desenvolveu condições necessárias e suficientes para assegurar a convergência para certas plantas. **Oishi (1997b)** dedicou-se ao estudo da estabilização robusta de sistemas amostrados e menciona experimentos nos quais um controlador analógico e um discretizado pela regra de Tustin tiveram quase o mesmo desempenho para pequenos períodos de amostragem.

Yamamoto (2002) apresentou um trabalho a respeito da preservação da estabilidade e do espectro para uma dada classe de sistemas de dimensão infinita. Em seu teorema 4.1 Yamamoto estabelece que o sistema analógico-discreto será exponencialmente estável se $\sup\{\mathrm{Re}\,\lambda | \lambda \in \sigma(A)\} < 0$, onde $\sigma(A)$ é o "spectrum" da matriz A. Apesar de Y. Yamamoto ser um estudioso da área de controle robusto e em especial de controle digital, este seu trabalho trata somente de sistemas de controle analógicos e mais especial ainda analisando respostas impulsivas pseudoracionais.

Xiao (2003) desenvolveu um estudo a respeito de estabilidade, controlabilidade e observabilidade de sistemas de controle híbridos contínuos-discretos que possuem o vetor de estados como função do tempo analógico t e também de tempo discreto, chamado de n, onde n = 0, 1, 2, ... Todo seu trabalho foi realizado no domínio da freqüência de Laplace "s", porém **Xiao (2003)** não levou em consideração os efeitos deletérios da amostragem e nem sequer considerou explicitamente o período de amostragem.

Zhai (2004) estudou a estabilidade de um tipo de sistema composto de dois subsistemas: a) sistema analógico linear e invariante no tempo (SALIT) e b) sistema discreto tempo linear e invariante no tempo (SDLIT). **Zhai (2004)** demonstrou que quando ambos subsistemas SALIT e SDLIT são estáveis no sentido de Hurwitz e Schur, respectivamente, e se as matrizes correspondentes a SALIT e SDLIT comutam ou se tais matrizes são simétricas, então existe uma

função de Liapunov comum para os dois subsistemas, sendo que o subsistema SDLIT é exponencialmente estável.

Seyranian (2006) apresentou a teoria de bifurcações em autovalores com interpretações geométricas em duas e três dimensões. Estabilidade e catástrofe de sistemas aeroelásticos e sistemas giroscópicos foi apresentada. Domínios de estabilidade/instabilidade para a equação de Hill submetidas a movimentos periódicos foram apresentados bem como foi apresentado um levantamento de regimes estáveis e instáveis da equação de Duffing harmonicamente excitada. Com resultados similares à análise da posição das raízes ("root locus") da teoria de controle, porém usando autovalores, tal como disposto no Teorema 4.1 de **Yamamoto (2002)**. **Seyranian (2006)** conclui que o problema generalizado do autovalor usando cadeias de Jordan de autovetores gera a estabilidade assintótica se $\mathrm{Re}\{\lambda_i\}<0$, ou "flutter instability" (estabilidade marginal) para $\mathrm{Re}\{\lambda_i\}=0$, com $\mathrm{Im}\{\lambda_i\}\neq 0$, e instabilidade se $\mathrm{Re}\{\lambda_i\}>0$. Para explicar o fenômeno da bifurcação em autovalores **Seyranian (2006)** apresentou os trabalhos originais de dois cientistas russos da década de 60, Visluk e Liusternik, que deram preferência a entender bifurcações de autovalores como variação de parâmetros em matrizes na forma de blocos de Jordan. Foi apresentado também o teorema de Thomson-Tait-Chetaev o qual estabelece que um sistema giroscópico pode ser estabilizado somente por graus pares de instabilidade da matriz A em $M\ddot{q}+G\dot{q}+Aq=0$, i.e., para que os autovalores instáveis possam sair do semiplano direito do plano complexo é preciso que eles coexitam com algum outro autovalor que possa colidir com ele, tal como ocorre no "root locus", por analogia. Foi apresentado o problema da aeroelasticidade de Keldysh o qual relaciona a velocidade transversal de oscilação da asa de um avião com o colapso (quebra) da mesma quando tal velocidade se torna um número complexo. Este trabalho de **Seyranian (2006)** é um grande avanço/extensão em relação ao fenômeno "avoidance of crossing" (**Lax, 1997**), explicado acima, com aplicações à aeroelasticidade, teoria de giroscópios e também construção civil.

Hou (2007) estudou a teoria da estabilidade para sistemas analógicos, sistemas descontínuos, i.e. com descontinuidades no movimento, e sistemas discretos no tempo e demonstou que: i) sistemas discretos no tempo e sistemas descontínuos possuem propriedades de estabilidade idênticas; ii) para determinadas circunstâncias ou hipóteses as condições de estabilidade de sistemas analógicos reduzem-se às condições de estabilidade de sistemas descontínuos; iii) para determinadas circunstâncias ou hipóteses as condições de estabilidade de sistemas discretos no tempo também reduzem-se às condições de estabilidade de sistemas descontínuos. Entretanto, **Hou (2007)** não considerou os efeitos deletérios da amostragem em seu estudo.

Chen (2008) estudou a reconstrução de sistemas analógicos por meio processo de discretização por amostras não-uniformes, i.e., fazendo uso de técnicas de múltiplos períodos de amostragem ("multirate"). Segundo Chen (2008) estudos recentes indicam que dados n+1 períodos de amostragem T_1, T_2, ... , T_{n+1} o modelo analógico G poderá ser reconstruído unicamente a partir dos n+1 modelos discretos. A dificuldade de realizar isso é essencialmente experimental: surge a necessidade de conduzir n+1 configurações amostradas distintas com n+1 diferentes períodos de amostragem. É apresentado um Lema, devido à Bruce Francis, no qual é estabelecido que se um dado sistema analógico G é completamente controlável e completamente observável então o modelo discreto também será completamente controlável e completamente observável se o período de amostragem for "não-patológico", i.e., sem falseamento ("aliasing"). **Chen (2008)** ensina que a taxa de amostragem $\omega_S = \frac{2\pi}{T}$ é patológica, em relação à matriz de dinâmica A, se A tiver dois autovalores com partes reais iguais e partes imaginárias diferindo de um inteiro múltiplo de ω_S. Por outro lado, a taxa de amostragem é dita ser não-patológica caso $\lambda_i - \lambda_j = \pm\frac{2k\pi\sqrt{-1}}{T}$, com k=1, 2, ...

2.1.2. REGRAS CLÁSSICAS DE MAPEAMENTO ANALÓGICO-DIGITAL.

Em Franklin (1981) são ensinadas diversas regras clássicas de mapeamento analógico-digital tal como pode-se ver no Capítulo 3 e no Apêndice A deste trabalho. Todas essas regras de mapeamento são oriundas de equações a diferenças-finitas. Por exemplo, regra de Tustin possui a seguinte equação a diferenças-finitas:

$$e_k = \frac{2}{T}.\nabla u_k - e_{k-1} \tag{2.1}$$

que implica o seguinte mapeamento s-z:

$$s \sim \frac{2}{T}.\frac{z-1}{z+1} \tag{2.2}$$

O seu mapeamento é governado por:

$$z = \exp\left(j.\tan^{-1}\left[\frac{4\omega T}{4-(\omega T)^2} \right] \right) \tag{2.3}$$

$$|z| = 1 \tag{2.4}$$

As equações (2.3) e (2.4) descrevem um círculo unitário no espaço complexo z, i.e., todo semiplano esquerdo do espaço complexo s é mapeado pela regra de Tustin para o interior desse círculo unitário (Figura 2.1).

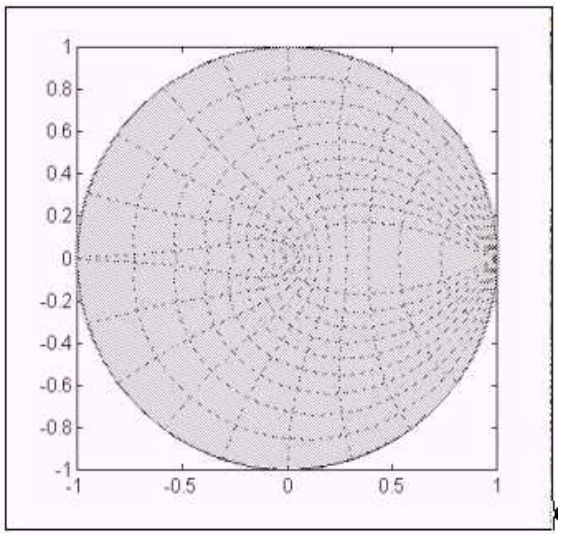


Figura 2.1 – Mapeamento Tustin.

2.1.3. EQUAÇÕES DIFERENCIAIS COM LADO DIREITO DESCONTÍNUO.

Em **Filippov (1964)** é descrito um estudo matemático qualitativo preliminar sobre equações diferenciais com lado direito descontínuo e tal estudo é realizado em duas direções. A primeira delas apresenta equações diferenciais (e sistemas de equações diferenciais) com lado direito descontínuo de forma bastante geral onde teoremas sobre a existência de soluções são provados e as propriedades fundamentais de tais soluções são investigadas. A segunda direção dedica-se a equações diferenciais (e sistemas) com lado direito contínuo por partes. Basicamente a equação estudada por Filippov foi,

$$\ddot{x} + b\ddot{x} + k.\mathrm{sgn}(\dot{x}) + cx = e \qquad (2.5)$$

onde “b”, “k”, “c” são contantes positivas e “e” é uma função periódica. Filippov apresenta um teorema que permite construir geometricamente uma região fora da qual nenhuma solução da equação acima passa e portanto permite estudar a estabilidade da solução, ou sempre que possível, provar a existência de uma solução periódica para um sistema com lado direito descontínuo.

Sesekin (1996) buscou dar continuidade ao trabalho de **Filippov (1964)**. Sesekin considerou um sistema não-linear de equações diferenciais ordinárias com entrada impulsiva limitada e obteve diversos resultados qualitativos a respeito. Seus trabalhos, apesar de ainda estarem em franco desenvolvimento, ainda não apresentam resultados aplicáveis praticamente. Em **Sesekin (1994)** descreve um problema de estudar as propriedades de certas trajetórias descontínuas que segundo o autor seria de muita utilidade em mecânica de vôo, eletromagnetismo, medicina e economia, em especial tratou de estudar sistemas dinâmicos que admitem trajetórias geradas por controles impulsivos. **Sesekin (1999)** investigou um conjunto de trajetórias geradas por controles impulsivos e forneceu noções básicas com equações diferenciais bilineares e modelos de mecânica aeroespacial.

Akhmetov (1992) desenvolveu um estudo matemático qualitativo a respeito de sistemas pulsativos o qual pode ser usado para determinar propriedades de soluções de equações diferenciais com lado direito descontínuo. **Akhmetov (1994)** investigou condições sob as quais soluções de equações diferenciais com lado direito descontínuo dependem de condições iniciais. Neste trabalho Akhmetov apresenta um teorema (sem prova) no qual apresenta a solução $\bar{x}(t)$ de uma equação diferencial com lado direito descontínuo dependendo de uma série cujos termos contém a condição inicial x_0 e funções contínuas por partes. **Akhmetov (1995)** estabelece condições de contorno para que um problema de controle de um sistema pulsativo possua uma solução. Resumidamente, **Akhmetov (1995)** estabeleceu o problema como segue,

$$\begin{aligned} \dot{\underline{x}} &= A(t).\underline{x} + C(t).u(t) + f(t) \quad ; t \neq t_i \\ \Delta \underline{x}\big|_{t=t_i} &= B_i.\underline{x} + D_i.v_i + I_i \end{aligned} \tag{2.6}$$

Com as condições de contorno,

$$\underline{x}(\alpha) = \underline{a}, \qquad \underline{x}(\beta) + \underline{b} \tag{2.7}$$

Aqui $\underline{x} \in R^n$, A e C são matrizes *n x n* e *n x m*, respectivamente, B_i e D_i são matrizes constantes *n x n* e *n x m*, respectivamente, t_i, $i = \overline{1, p}$, $p > 1$. O problema é dito resolvível se para f(t), I_i , a e b existir u(t) e v_i tal que para a condição de contorno estabelecida em (2.7) exista uma solução para (2.6). Conforme disposto num teorema, **Akhmetov (1995)** ensina que, se o problema de controle pulsativo possuir uma solução então o controle poderá ser calculado mediante uma expressão que faz uso do gramiano inverso.

2.1.4. TEORIA DO "LIFTING" CONTÍNUO.

Segundo **Yamamoto (1999)** o preço a se pagar por utilizar transformada-Z em análises de sistemas amostrados de controle é a perda da informação entre as amostras e que existe uma possibilidade de se recuperar essa informação perdida uma vez que conheçamos a lei de controle bem como utilizemos a transformada Z modificada (**Houpis, 1985**). Um problema para a transformada Z modificada é a extrema complexidade de seus cálculos e ainda a limitação da total recuperação da informação entre as amostras (**Houpis, 1985**). Foi a partir daí que Yamamoto procurou investigar uma alternativa. Estudando o controle de sistemas com parâmetros distribuídos, em especial sistemas contendo atrasos de duração L (e^{Ls}) lhe chamou a atenção (**Yamamoto, 1996**) o fato de um sistema de dimensão infinita (o atraso) conseguir controlar um sistema de dimensão finita. No caso o método utilizado foi o "sistema de controle repetitivo". Apesar de não haver uma conexão entre tais mundos, **Yamamoto (1999)** buscou encontrar o caminho durante os anos 80. Um trabalho pioneiro nesse sentido foi **Francis (1991)** onde tentou dar as bases da teoria matemática do controle discreto de plantas analógicas apresentando algumas hipóteses e considerações matemáticas tais como um Teorema o qual estabelece certas condições de convergência de seqüências oriundas do trinômio controle-amostrador-segurador. **Yamamoto (1996)** buscou a partir daí conciliar o conhecimento tradicional de controle com a necessária idéia de obter a informação entre as amostras. Usando o modelo Kalman-Beltram (**Yamamoto, 1996**):

$$x_{k+1} = e^{AT} x_k + \int_0^T e^{A\tau} B u_k \, d\tau \tag{2.8}$$

$$y_k = C x_k \tag{2.9}$$

Naturalmente, com tal modelo introduzimos a amostragem na entrada e portanto ignoramos o comportamento entre as amostras. No verão de 1989, Yamamoto teve a seguinte idéia: “se não queremos perder a informação entre as amostras devemos então tê-la simplesmente como uma função vetorial”. Partindo dessa consideração aparentemente óbvia, Yamamoto chegou a seguinte hipótese: “seja f(t) uma função definida em t em $[0,\infty)$, e T o período de amostragem. Se amostrarmos f em cada período de T unidades de tempo teremos, naturalmente, uma seqüência de funções na forma $\{f(kT)\}_0^\infty$. Isso certamente resulta na perda de informação $f(kT+\theta)$; 0 < θ < T”. É quase trivial então que

$$L : f \;\alpha\; \{f(kT+\theta)\}_{k=0}^{\infty} \quad ;0 < \theta < \mathrm{T} \tag{2.10}$$

Em tais termos, onde há um mapeamento de uma função contínua numa seqüência de funções, podem ser vistos tal como na figura abaixo,

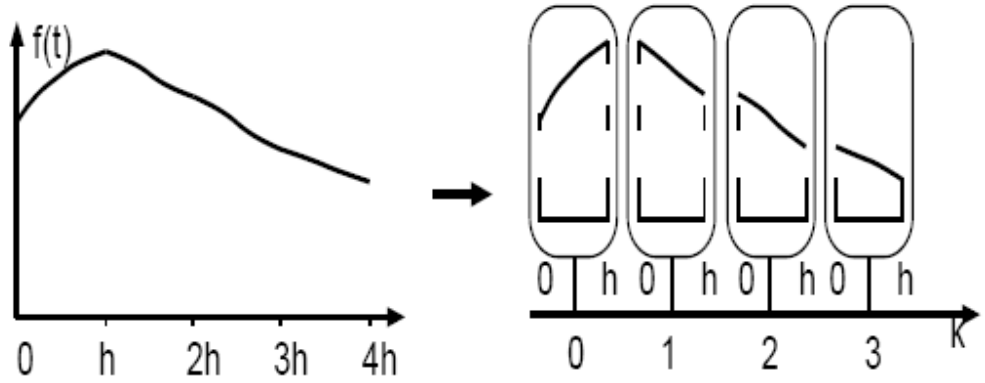


Figura 2.2 – “Lifting” de Yamamoto. Nessa figura original do trabalho de Yamamoto h é o período de amostragem T.

FONTE: Yamamoto (1996).

É a isso que Yamamoto chama de “lifting” contínuo. Uma característica importante é que com tal correspondência plantas analógicas podem ser transformadas em sistemas discretos invariantes no tempo. Entretanto, seus resultados ainda não foram aplicados a casos práticos, principalmente porque ainda não se conseguiu imaginar como realizar na prática tal teoria.

Em **Yamamoto (1990)** existem detalhes dessa teoria que resumidamente será apresentada de forma fiel ao trabalho dele, porém com adaptações à nomenclatura deste trabalho. Ao invés de falar de uma variável de estado num instante da amostragem t = k.T, Yamamoto propôs observar a história passada,

$$x|_{[(k-1)T,kT]} \tag{2.11}$$

(2.11) foi considerada como um estado e a partir daí considerou-se o sistema linear invariante no tempo,

$$\begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t) \\ \underline{y}(t) = C\underline{x}(t) \end{cases} \tag{2.12}$$

(2.12) foi então compreendido como um sistema discreto com espaço-de-estados de dimensão infinita. Isso dá a vantagem de poder controlar uma planta contínua no tempo e o controle digital do mesmo tipo de análise sendo que a maior dificuldade é realmente a obrigação de se trabalhar em dimensão infinita.

Em **Yamamoto (1990)** $\tilde{C}[o,\infty)$ é o espaço das funções contínuas por partes em $[0,\infty)$; $\tilde{C}[o,T)$ é o espaço de todas as funções $\varphi(t)$ em (0,T], denotando-se de U esse espaço. Assim, podemos dizer, de forma correspondente que,

$$\begin{aligned} S:\ & \tilde{C}[0,\infty) \to U^{\infty} \\ & \varphi(t)\ \alpha\ \{\varphi_k\}_{k=1}^{\infty} \qquad ,\varphi_k(\theta) = \varphi((k-1)T+\theta) \end{aligned} \tag{2.13}$$

S é injetora (Apêndice H) onde,

$$\theta \in (0, T] \tag{2.14}$$

O espaço U^{∞} é o espaço formal das séries de potência que **Yamamoto (1990)** define como "$z^{-1}\,\mathrm{Y}\,[[\,z^{-1}]]$" com coeficientes calculados pela transfotmada Z,

$$\begin{aligned} Z: \quad & U^{\infty} \to z^{-1}\,\mathrm{Y}\,[[z^{-1}]] \\ & \{\varphi_k\}_{k=1}^{\infty} \;\alpha\; \sum_{k=1}^{\infty} \varphi_k . z^{-k} \end{aligned} \tag{2.15}$$

Já o operador amostragem *S* (ou operador amostrador),

$$S: \quad U^{\infty} \to R^{\infty} \tag{2.16}$$

é definido como segue,

$$S(\{\varphi_k\}) := \{\varphi_k(T)\}_{k=1}^{\infty} \tag{2.17}$$

O operador "hold" (segurador) *H*, por sua vez,

$$H: \quad R^{\infty} \to U^{\infty} \tag{2.18}$$

é definido como segue,

$$H(\{x_k\}) := \{\varphi_k(\theta)\} \qquad ; \varphi_{k+1}(\theta) \equiv x_k \tag{2.19}$$

por fim, a operação "sampler and hold" (amostrador e segurador),

$$(SH)\;(\{\varphi_k\})_k(\theta) = \begin{cases} 0 & , k = 1 \quad \text{"condição de causalidade"} \\ \varphi_{k-1}(T) & , k \geq 2 \end{cases} \tag{2.20}$$

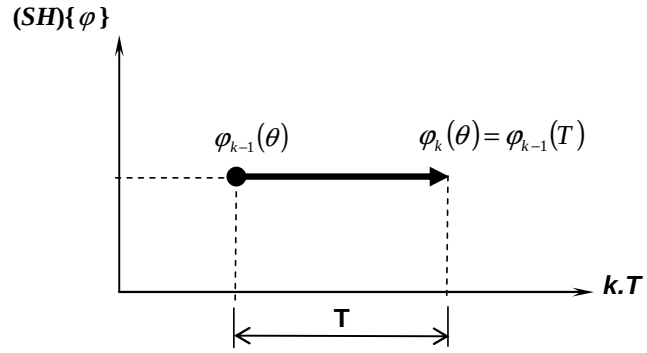


Figura 2.3 – Operador *SH* interpretado geometricamente.

Na Figura 2.3 temos uma representação geométrica do operador *SH*. Até o momento com as equações acima apenas se representou matematicamente o que já se conhece com mais informalidade na teoria de controle digital.

A teoria do "lifting contínuo" (**Yamamoto, 1990**) se construiu partindo do modelo a tempo-discreto com comportamento entre asmostras incorporado associando um "modelo discreto de dimensão infinita" a um sistema contínuo linear. A idéia chave é se trabalhar com a história passada $\underline{x}(t)\big|_{((k-1)T,kT)}$ com oum estado em cada instante amostrado e daí derivar uma transição para dimensão infinita para essa variável.

Consideremos assim o sistema híbrido mostrado na Figura 2.4. Nesse sistema $\sum_d$ e $\sum_C$ são os sistemas discreto e contínuo, respectivamente, (A_d, B_d, C_d) e (A_c, B_c, C_c).

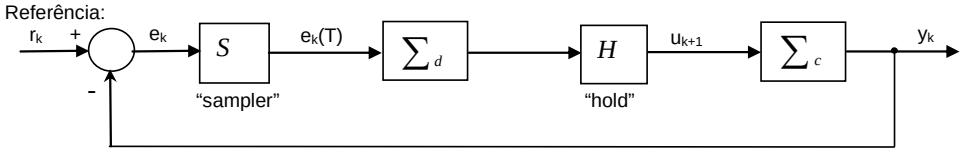


Figura 2.4 – Sistema $\sum_{CL}$ .híbrido em malha-fechada.

Segue que o sistema em malha-fechada $\sum_{CL}$ obedece às equações que seguem (sistema híbrido):

$$\begin{cases} \underline{w}_{k+1} = A_d.\underline{w}_k + B_d\,\underline{e}_k(T) \\ \underline{u}_{k+1}(\theta) = C_d.\underline{w}_k \qquad\qquad ,0 < \theta \le \mathrm{T} \end{cases} \tag{2.21}$$

O vetor $\underline{w}$ é a variável de estado $\underline{x}$ vista com distorção pelo controlador digital, i.e., como se fosse vista por meio de um “espelho fosco” que é o “sampler-hold”.

A equação (2.21) é precisamente o controle amostrado segurado com o “hold” por um período de amostragem de T segundos.

É importante destacar também que o “hold” que trata toda a teoria do Yamamoto é o segurador de ordem zero ou “zero-order hold”.

De volta à equação de estados,

$$\begin{cases} \underline{x}_{k+1}(\theta) = e^{A_C.\theta}\,\underline{x}_k(T) + \int_0^{\theta} e^{A_C.(\theta-\tau)}.B_C.\underline{u}_{k+1}(\tau)d\tau \\ \underline{y}_k(\theta) = C_C.\underline{x}_k(\theta) \qquad\qquad ,0 < \theta \le \mathrm{T} \end{cases} \tag{2.22}$$

Na equação acima há causalidade entre $\underline{x}$ e $\underline{u}$ graças ao “sampler and hold” tal como mostra (2.26). Na Figura 2.5 há uma simples representação envolvendo $\underline{w}$ e $\underline{x}$.

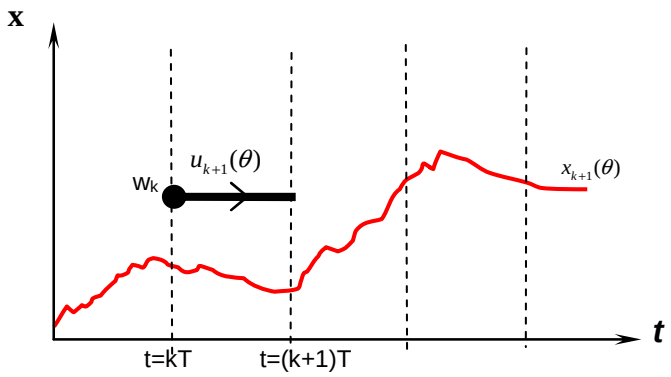


Figura 2.5 – Sinais do sistema híbrido.

A expressão do erro é dada por,

$$e_k(\theta) = r_k(\theta) - y_k(\theta) \tag{2.23}$$

Substituindo (2.23) em (2.21) e (2.21) em (2.22) construiremos o seguinte sistema,

$$\begin{aligned} \underline{w}_{k+1} &= A_d.\underline{w}_k + B_d\left[\underline{r}_k(T) - \underline{y}_k(T)\right] \\ &= A_d.\underline{w}_k + B_d.\underline{r}_k(T) - B_d.\underline{y}_k(T) \\ &= A_d.\underline{w}_k + B_d.\underline{r}_k(T) - B_d.C_C.\underline{x}_k(T) \end{aligned} \tag{2.24}$$

$$\begin{aligned} \underline{x}_{k+1}(\theta) &= e^{A_C.\theta}\underline{x}_k(T) + \int_0^\theta e^{A_C.(\theta-\tau)}.B_C.\underline{u}_{k+1}(\tau)d\tau \quad , \underline{u}_{k+1} \text{ é cte. dentro da integral} \\ &= e^{A_C.\theta}\underline{x}_k(T) + \left(\int_0^\theta e^{A_C.(\theta-\tau)}.B_C.d\tau\right).\underline{u}_{k+1}(\tau) \\ &= e^{A_C.\theta}\underline{x}_k(T) + B(\theta).\underline{u}_{k+1}(\tau) \\ &= e^{A_C.\theta}\underline{x}_k(T) + B(\theta).C_d.\underline{w}_k \end{aligned} \tag{2.25}$$

onde,

$$B(\theta) = \int_0^\theta e^{A_C.(\theta-\tau)}.B_C.d\tau \tag{2.26}$$

Juntando os resultados das equações (2.24) e (2.25),

$$\begin{cases} \underline{w}_{k+1} = A_d.\underline{w}_k - B_d.C_C.\underline{x}_k(T) + B_d.r_k(T) \\ \underline{x}_{k+1}(\theta) = B(\theta)C_d.\underline{w}_k + \mathrm{e}^{A_C\theta}.\underline{x}_k(T) \end{cases} \tag{2.27}$$

Como $x_k(T)$ e $r_k(T)$ são constantes dentro do intervalo $0 < \theta \leq \mathrm{T}$ podemos rescrevê-los com o uso do operador trem de impulsos unitário δ_T com período T:

$$x_k(T) = \delta_T.x_k(\theta) \tag{2.28}$$

$$r_k(T) = \delta_T.r_k(\theta) \tag{2.29}$$

Porquê foi feito isso? Porque o operador δ_T "pinça" (amostra, colhe) o valor de $x_k(T)$ e $r_k(T)$ nos instantes da amostragem.

Daí,

$$\begin{cases} \underline{w}_{k+1} = A_d.\underline{w}_k - B_d.C_C.\delta_T\,\underline{x}_k(\theta) + B_d.\delta_T.r_k(\theta) \\ \underline{x}_{k+1}(\theta) = B(\theta)C_d.\underline{w}_k + \mathrm{e}^{\mathrm{A_C}\theta}.\delta_T\,\underline{x}_k(\theta) \end{cases} \tag{2.30}$$

Finalmente, a representação híbrida do espaço dee estados será,

$$\begin{pmatrix} \underline{w}_{k+1} \\ \underline{x}_{k+1}(\theta) \end{pmatrix} = \begin{pmatrix} A_d & -B_d.C_C.\delta_T \\ B(\theta)C_d & \mathrm{e}^{\mathrm{A_C}\theta}.\delta_T \end{pmatrix}.\begin{pmatrix} \underline{w}_k \\ \underline{x}_k(\theta) \end{pmatrix} + \begin{pmatrix} B_d.\delta_T \\ 0 \end{pmatrix}.r_k(\theta) \tag{2.31}$$

De onde operador A acima, de dimensão infinita, é definido como segue,

$$A := \begin{pmatrix} A_d & -B_d.C_C.\delta_T \\ B(\theta)C_d & \mathrm{e}^{\mathrm{A_C}\theta}.\delta_T \end{pmatrix} \tag{2.32}$$

E o sinal de erro,

$$\underline{e}_k = \begin{pmatrix} 0 & -C_C \end{pmatrix}.\begin{pmatrix} \underline{w}_k \\ \underline{x}_k(\theta) \end{pmatrix} + r_k(\theta) \tag{2.33}$$

onde,

$$\underline{y}_k = \begin{pmatrix} 0 & -C_C \end{pmatrix}.\begin{pmatrix} \underline{w}_k \\ \underline{x}_k(\theta) \end{pmatrix} \tag{2.34}$$

Yamamoto (1990) também investigou algumas propriedades envolvendo estabilidade em sua teoria. O seguinte teorema sugere uma caracterização espectral do sistema híbrido em malha-fechada,

Teorema 2.1 (**Yamamoto, 1990**): o espectro do sistema $\sum_{CL}$ descrito matematicamente pelo conjunto de equações (2.31) a (2.34) é a união de {0} e o espectro da matriz,

$$A_0 := \begin{pmatrix} A_d & -B_d.C_C \\ B(T)C_d & \mathrm{e}^{\mathrm{A_C}T} \end{pmatrix} \tag{2.35}$$

[devendo os seus autovalores estarem dentro do círculo unitário]

Entretanto, exceto o 0, todos são autovalores, tal como ensinado em **Yamamoto (1990)**. Em outras palavras, o espectro do sistema híbrido pode ser inteiramente determinado fazendo $\theta = T$.

□

O espectro do operador A de dimensão infinita é um espectro finito formado apenas por autovalores de $A_0 \, \mathrm{Y} \{0\}$.

Tal primeiro teorema do Yamamoto tem importantes conseqüências para a estabilidade de sistemas desse tipo, entretanto, fica claro que Yamamoto apenas tateou a importância do período de amostragem para controladores digitais em sistemas híbridos sem entender explicitamente nem sequer citar o problema do falseamento (mascaramento ou "aliasing"). Ele tornou explícita a influência do período de amostragem na estabilidade de um sistema híbrido usando autovalores que dependerão do período de amostragem T.

Teorema 2.2 (Yamamoto, 1990): o sistema $\sum_{CL}$ é internamente estável se e somente se A_0 (equação 2.35) é uma matriz estável.

Observação: para A_0 ser internamente estável é necessário que $\lim_{n \to \infty} \left\| A_0^n \right\| = 0$.

O teorema 2.2 mostra que se o sistema $\left(e^{A_c . T}, B(T), C_C \right)$ é estabilizado por um controlador digital $\left(A_d, B_d, C_d \right)$ de dimensão finita então o sistema em malha-fechada também será internamente estável.

2.2. REVISÃO BIBLIOGRÁFICA SECUNDÁRIA

2.2.1. RASBAND (1990).

Rasband (1990) descreve a complexidade de sistemas amostrados por meio de um exemplo de um oscilador harmônico não-amortecido e forçado (controlado) por impactos periódicos ("kicked harmonic oscillator") onde x(t) é a posição, f(x) é uma

função arbitrária de x, T é o período em que surgem os impactos com intensidade (amplitude) α e ω_o é o modo de vibração do oscilador (freqüência característica). Esse tipo de sistema que, dependendo de certas condições a respeito dos impactos pode gerar caos, possui a seguinte expressão,

$$\frac{d^2x(t)}{dt^2} + \omega_o^2 x(t) = \alpha . f(x) . \sum_{n=1}^{\infty} \delta(t - nT) \qquad (2.36)$$

Como uma alternativa para o gráfico de trajetórias do espaço de fase, o qual se torna impraticável após um considerável número de períodos, é conveniente estudar a evolução desse sistema gerando um ponto no plano de fase $(x, \dot{x})$ para valores de t = T, 2T, 3T, ..., isto é, nos valores de t correspondendo a múltiplos inteiros do período do termo forçante (de controle). Um gráfico desse tipo é denominado de “seção de Poincaré”.

Num intervalo entre os impactos o termo do lado direito de (2.16) é zero e temos a solução muito familiar, para intervalos (k-1)T < t < kT,

$$x(t) = A_k \cos(\omega_o t) + B_k \sin(\omega_o t) \qquad (2.37)$$

e

$$\dot{x}(t) = -\omega_o A_k \sin(\omega_o t) + \omega_o B_k \cos(\omega_o t) \qquad (2.38)$$

onde k = 1, 2, ... Pode-se calcular uma forma iterativa para os coeficientes nos intervalos em k e (k+1),

$$A_{k+1} = A_k - \frac{\alpha}{\omega_o} f(x_k) \sin(\omega_o kT) \qquad (2.39)$$

$$B_{k+1} = B_k + \frac{\alpha}{\omega_o} f(x_k) \cos(\omega_o kT) \qquad (2.40)$$

onde,

$$x_k = A_k \cos(\omega_o kT) + B_k \sin(\omega_o kT) \tag{2.41}$$

$$\dot{x}_k = -\omega_o A_k \sin(\omega_o kT) + \omega_o B_k \cos(\omega_o kT) \tag{2.42}$$

onde o subscrito k se refere ao instante kT. Partindo daí podemos escrever a função iterativa que segue,

$$\begin{pmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{pmatrix} = \begin{pmatrix} \cos(\omega_o kT) & \omega_o^{-1} \sin(\omega_o kT) \\ -\omega_o \sin(\omega_o kT) & \cos(\omega_o kT) \end{pmatrix} \begin{pmatrix} x_k \\ \dot{x}_k + \alpha.f(x_k) \end{pmatrix} \tag{2.43}$$

Com α não nulo e f(x) = 1 ou x o mapeamento (2.43) é linear e inversível e nenhuma dinâmica caótica foi observada (Rasband, 1990), porém foi detectada a ressonância (instabilidade) do sistema quando a freqüência dos impactos torna-se igual à freqüência do modo de vibração da estrutura, i.e., $\frac{2\pi}{T} = \omega_o$. Para esse caso onde f(x) = 1 temos a seção de Poincaré mostrada na Figura 2.6.

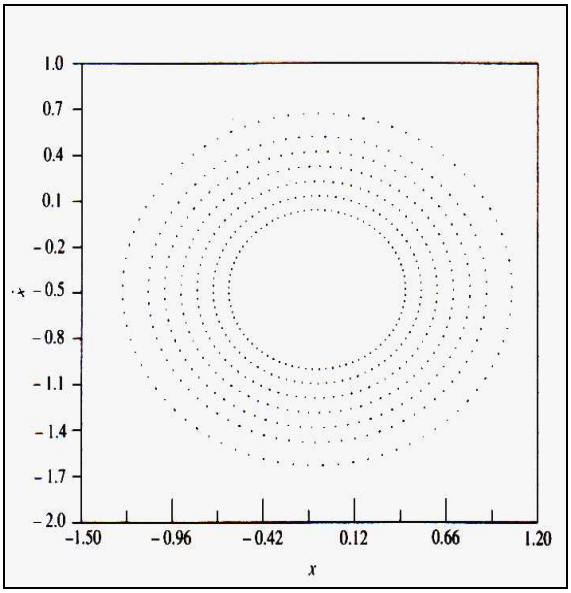


Figura 2.6 – Seção de Poincaré $(x, \dot{x})$ para o oscilador harmônico excitado por impactos periódicos independentes de x.

FONTE: Rasband (1990, p. 7).

Quando alteramos f(x) =1 para f(x) = x^4, em (2.36), preservando as mesmas condições iniciais, podemos observar na Figura 2.7 o caos que é produzido onde o mesmo pode ser observado em pequenas camadas de órbitas caóticas (onde uma delas é mostrada ampliada no destaque).

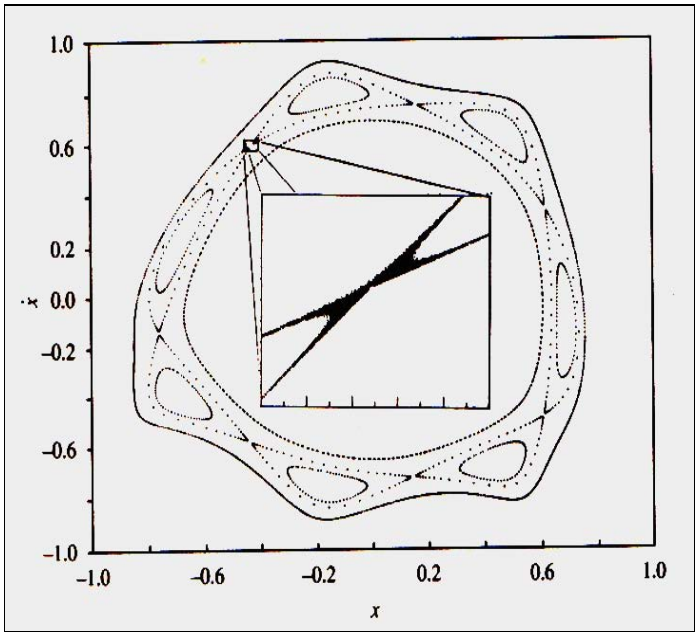


Figura 2.7 – Caos: seção de Poincaré $(x, \dot{x})$ para o oscilador harmônico excitado por impactos periódicos onde f(x) = x^4.

FONTE: Rasband (1990, p. 8).

CAPÍTULO 3

MAPEAMENTOS ANALÓGICO-DIGITAIS CLÁSSICOS E PROPOSTOS

3.1 INTRODUÇÃO

Neste Capítulo são fornecidos conceitos fundamentais da teoria de controle digital bem como técnicas clássicas de mapeamento analógico-digital usuais em integração numérica e muito utilizadas em projetos de controladores discretos e, finalmente, dois Teoremas descrevendo novas regras de mapeamento analógico-digital que foram construídos tendo como base diversas experiências numéricas realizadas e relatadas nos Apêndices desta Tese.

Um sistema com controle digital de uma planta qualquer pode ser apresentado como mostra a Figura 3.1.

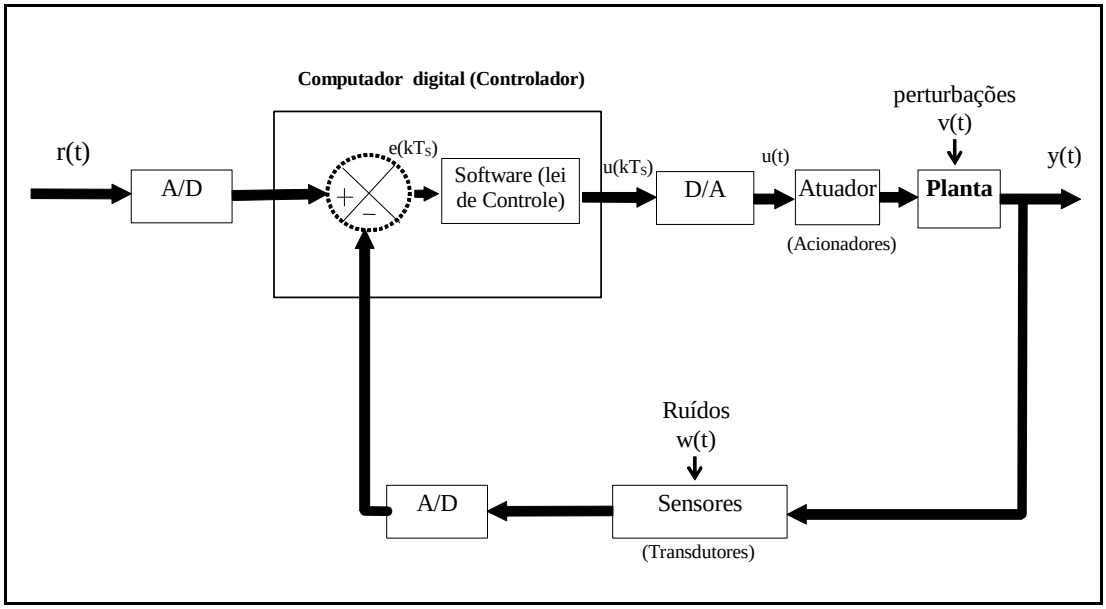


Figura 3.1 - Diagrama de blocos de um sistema com controle digital.

Até antes de 1947 somente existiam computadores analógicos que faziam uso de algoritmos lógicos iterativos que necessitavam de grandes e pesados equipamentos dotados de grandes circuitos elétricos com válvulas para executá-los, o que implicava um enorme consumo de energia elétrica. Isso era definitivamente inviável para a aplicação da computação em sistemas

aeroespaciais, onde qualquer aumento de massa ou de consumo de energia era extremamente significativo. Por esse motivo foi só após a invenção do transistor em 17 de novembro de 1947 por J. Bardeen, W. H. Brattain e W. B. Shokley (laureados por isso com o prêmio Nobel de 1956), do circuito integrado em em julho de 1958 por J. S. Kilby (laureado por isso com o prêmio Nobel de 2000) e, principalmente, dos microprocessadores em 15 de novembro de 1971, com a invenção do microprocessador 4004 (Intel, 1971), contendo uma CPU de apenas 4 bits, 16 pinos e “clock” de 740 kHz por Federico Faggin (patente norte-americana nº 3821715 concedida em 28 de junho de 1974), ver Figuras 3.2 e 3.3, e também dos microcontroladores oficialmente inventados em 17 de junho de 1988 por G. P. Hyatt cuja patente norte-americana nº 4942516 foi concedida em 17 de julho de 1990. Convém observar que a primeira tentativa de depósito dessa patente ocorreu em 24 de novembro de 1969 (USPTO, 1990), Figura 3.4. Em cerca de quarenta anos tais invenções promoveram uma incontestável revolução tecnológica que mudou para sempre o nosso mundo principalmente com a rápida difusão da computação digital. Isso, entretanto, não descartou de imediato a computação analógica que continuou sendo empregada por várias décadas e em diversos países. Em especial, o emprego da computação digital na tecnologia aeroespacial tornou-se viável, permitindo avanços tecnológicos antes apenas imaginados na ficção científica. Exemplos de tais aplicações aeroespaciais são: sondas espaciais autônomas, robôs (robô “soujourney” para exploração do solo marciano e manipulador robótico da estação espacial internacional), controle automático de aeronaves, sistemas microprocessados de controle de balões atmosféricos, etc.

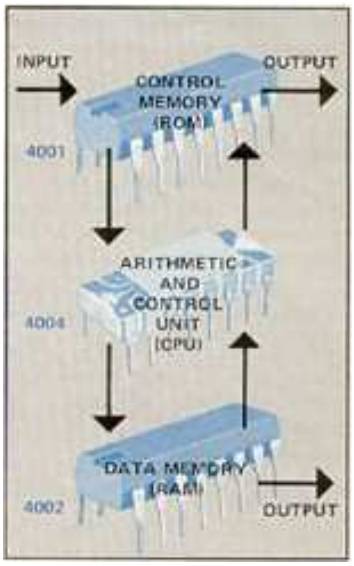


Figura 3.2 - Microprocessador Intel 4004.

FONTE: INTEL (1971).

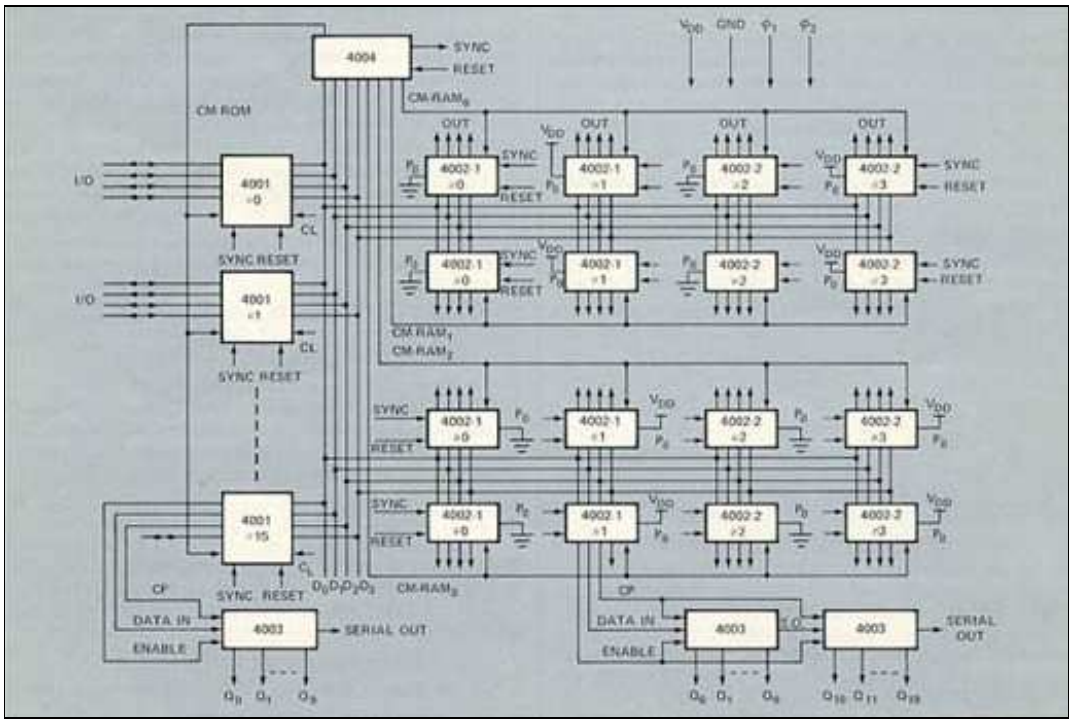


Figura 3.3 - Interconexão do sistema MCS-4.

FONTE: INTEL (1971).

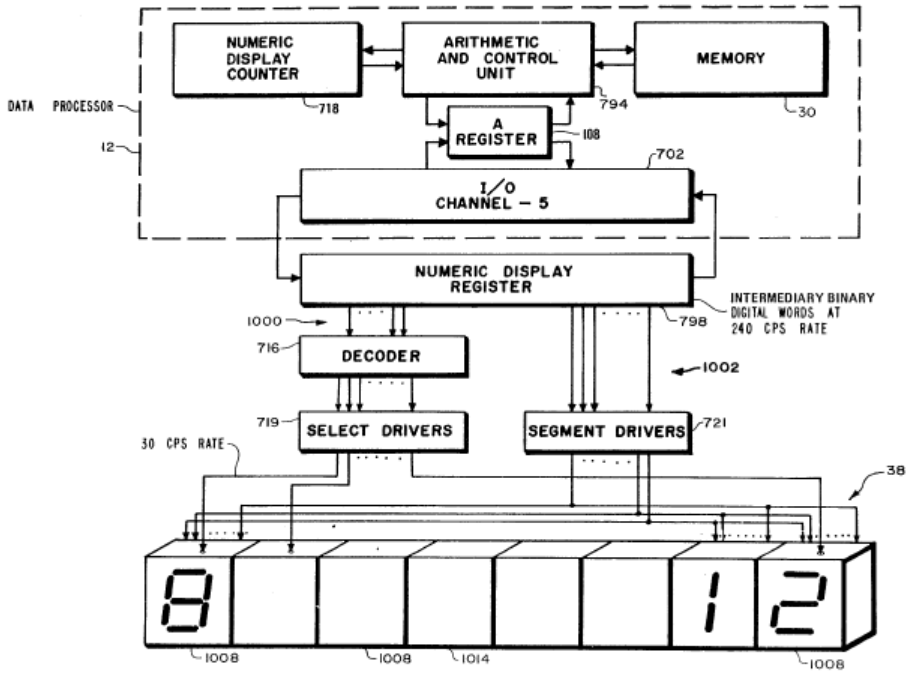


Figura 3.4 - Microcontrolador de Hyatt.
FONTE: USPTO (1990).

Mesmo assim, resolvidos os problemas de minimizar o custo nos lançamentos de satélites artificiais, ou ônibus espaciais, etc., reduzindo a massa desses objetos com a utilização da computação digital, nos deparamos com outra série de problemas técnicos a resolver. Tais problemas são características imanentes à tecnologia digital, conforme veremos mais adiante neste trabalho.

Um sistema digital é caracterizado por: amostragem no tempo, atrasos em entradas/saídas e processamento, e a quantização em amplitude. Da amostragem no tempo surgem os seguintes problemas: mascaramento (“aliasing”) e oscilações escondidas (“hidden oscillations”). Entendamos o problema da seguinte maneira: dizer que um dado sistema é discreto no tempo significa dizer que todas as variáveis utilizadas nesse sistema são atualizadas em instantes periódicos de tempo bem definidos cujo intervalo de tempo é denominado “período de amostragem” T. Analogamente, a freqüência ou taxa de amostragem ω_S é dada

por $2\pi/T_S$. Uma entidade fictícia porém extremamente importante no mundo digital é a freqüência de Nyquist $\omega_N = \omega_S / 2$.

Seja um sistema com uma largura de faixa finita $\omega_{máx}$. Haverá o mascaramento (mutilação ou distorção) na reconstrução (ou recuperação) do sinal a partir de suas amostras se $\omega_N < \omega_{máx}$ (ou $\omega_S < 2.\omega_{máx}$). Isso ocorrerá pois no domínio da freqüência não haverá espaço suficiente para dispor comodamente as repetições da largura de faixa finita provocadas pela amostragem. Para que isso não ocorra é necessário que $\omega_N > \omega_{máx}$ (ou $\omega_S > 2.\omega_{máx}$) que é o conhecido teorema de Nyquist (1928) como podemos ver na Figura 4.2. As oscilações escondidas também ocorrem se $\omega_N < \omega_{máx}$ (ou $\omega_S < 2.\omega_{máx}$) pois haverá freqüências do espectro do sinal de entrada que não serão vistas pelo amostrador.

A Figura 3.5 mostra um sinal senoidal submetido a um processo de amostragem.

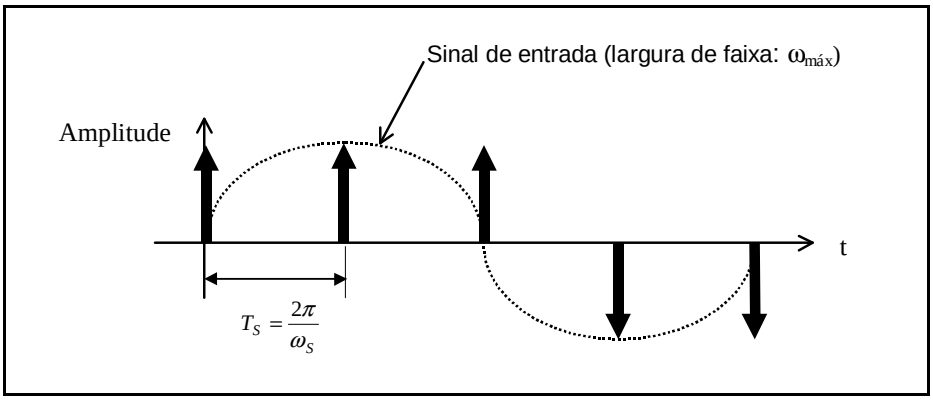


Figura 3.5 - Amostragem de um sinal.

Conforme pode-se notar na Figura 3.5, nos nodos tem-se valor nulo de amostragem, nos picos tem-se valor máximo e nos vales, valor mínimo, mostrando os principais elementos relativos à informação de entrada (picos, vales e nodos), por meio dos quais é possível reconstruir o sinal tendo apenas suas amostras. Vejamos mais adiante situações em que nem sempre podemos realizar tal esperada reconstrução do sinal a partir de suas amostras.

O mais interessante desses fenômenos do mascaramento e das oscilações ocultas é que eles podem ser uma das causas de instabilidade em sistemas digitais, como mostra este trabalho. Neste trabalho também é sugerido um método de análise que tem apresentado bons resultados para a compreensão do problema e de sua solução.

No domínio da freqüência teríamos, pela transformada de Fourier F(ω) do sinal de entrada f(t) seu espectro de freqüência tal como podemos conferir na Figura 3.6.

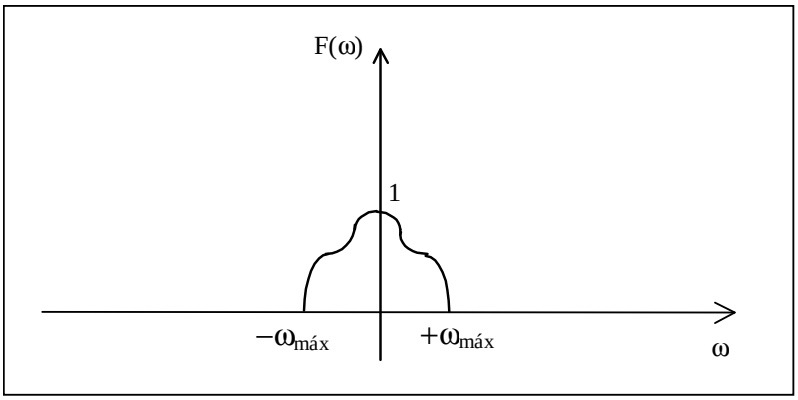


Figura 3.6 - Espectro normalizado de Fourier de um sinal de entrada com largura de banda finita.

Quando temos uma amostragem do sinal, sua transformada de Fourier tomará a forma da Figura 3.7 (se o sinal de amostragem puder ser modelado num trem de pulsos). Como se vê na Figura 3.7, ω_S deve ser no mínimo igual a 2. $\omega_{máx}$ para não haver mutilação (distorção) do sinal (ou $\omega_S \geq 2\,\omega_{máx}$).

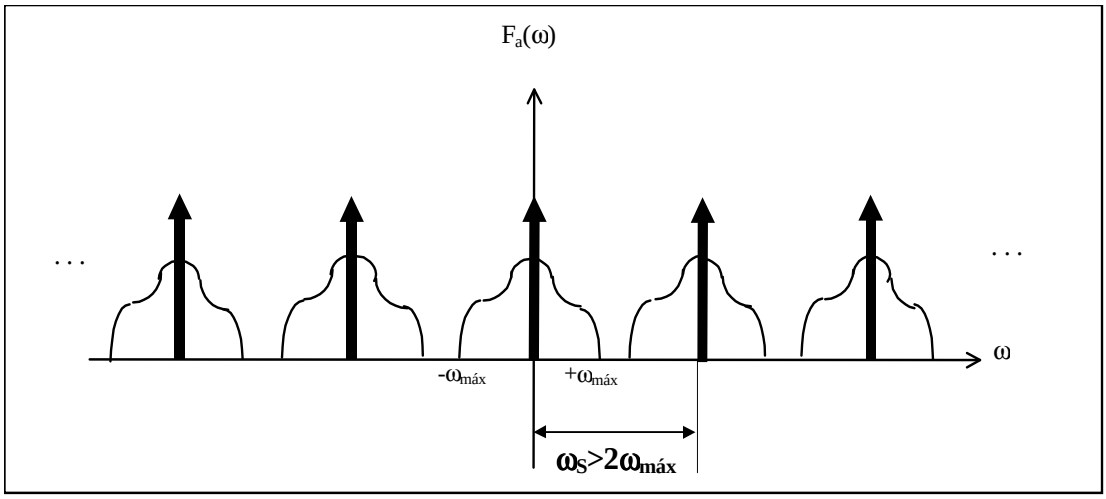


Figura 3.7 - Sinal amostrado no domínio da frequência.

Os atrasos em entradas/saídas são gerados nos conversores A/D e D/A, respectivamente, e os atrasos de processamento são causados pelo tempo de processamento que são pertinentes a programas de computador responsáveis pela política de controle. Em suma, pode-se dizer que os atrasos em entradas/saídas e processamento geram defasagens negativas e a instabilidade pode ocorrer para modos de freqüências elevadas. Conforme pode-se verificar no Apêndice A deste trabalho, um filtro Butterworth usado como filtro “anti-aliasing” gera atrasos de resposta que devem se somar aos demais atrasos mencionados. Também há o problema do controle digital com incerteza nos parâmetros da planta como elemento instabilizador, o qual está documentado em Katz (1975). Tais nuances da teoria de controle digital não serão abordadas neste trabalho o qual tem por objetivo investigar problemas inerentes à amostragem no tempo.

A quantização em amplitude de constantes e variáveis em 2^n níveis de quantização de palavras digitais (“digital words”) com n bits gera o chamado erro de quantização que se interpreta como a mutilação das amplitudes das amostras devido aos ajustes das amplitudes das amostras aos níveis de quantização.

Isto é reduzido pelo crescimento do n, mas isto eleva o tempo de processamento, demandando microprocessadores, conversores A/D e D/A mais rápidos e caros, etc.

O processo de discretização pode ser muito facilmente analisado por diagrama de blocos como podemos ver a seguir (**Franklin, 1981**):

a) Seja o sistema analógico da Figura 3.8:

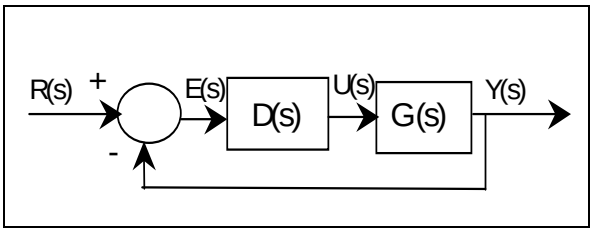


Figura 3.8 - Representação analógica em malha-fechada.

b) Tendo por base o sistema da Figura 3.8, no domínio digital teremos a Figura 3.9:

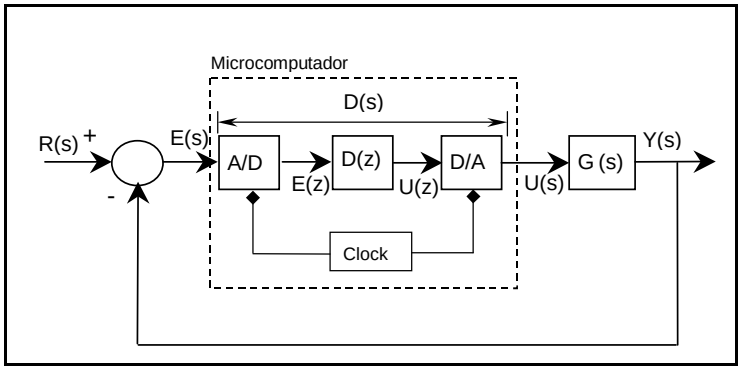


Figura 3.9 - Equivalência digital da Figura 3.8.

As próximas Figuras serão simplificadas, para facilitar.

c) Uma simples manipulação dos blocos é necessária para se obter o equivalente discreto $G_{Hn}(z)$ da planta analógica G(s) mostrado na Figura 3.10:

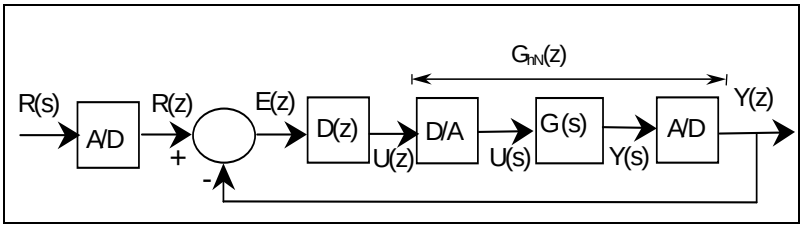


Figura 3.10 - Sistema digital manipulado.

Notemos que $G_{hN}(z)$ é o equivalente discreto por segurador (“Hold”) de ordem enésima da planta analógica.

O conversor A/D pode ser representado pela seguinte configuração generalizada da Figura 3.11:

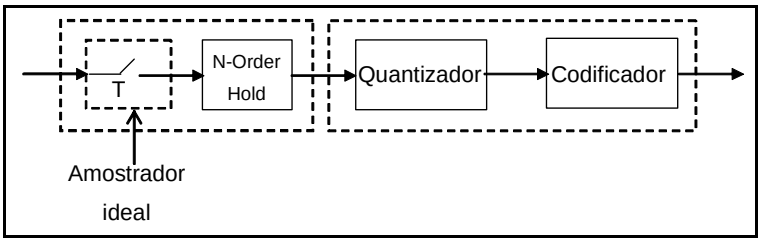


Figura 3.11 - A/D generalizado.

O princípio de funcionamento é deveras simples: um sinal analógico (contínuo no tempo e na amplitude) é amostrado de T em T segundos (periodicamente). Podemos dizer que a partir daí o sinal é de tempo discreto. Os valores destas amostras são “segurados” por um segurador (“Hold”); que teoricamente pode ter qualquer ordem durante T segundos. Após isso, o sinal passa por um quantizador que se responsabiliza por ajustar tais sinais de amplitude analógica a certos níveis de amplitude muito bem definidas. Estes sinais ajustados em amplitude ou quantizados serão finalmente convertidos em códigos binários pelo codificador que serão interpretados por um algoritmo digital ou programa de computador o qual gerará saídas digitais.

O conversor D/A possui um funcionamento complementar ao do A/D como podemos verificar na Figura 3.12.

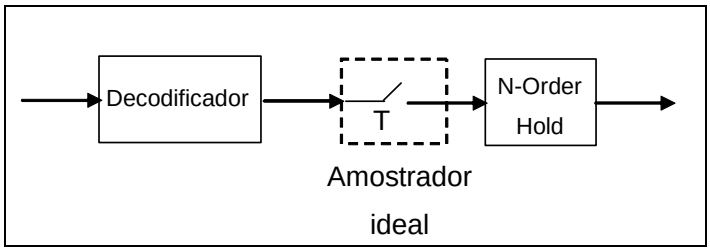


Figura 3.12 - D/A generalizado.

O decodificador interpreta códigos binários em amplitudes quantizadas. Tais sinais da saída do D/A serão os sinais de controle.

d) Uma simplificação com segurador de ordem zero (“Zero-Order Hold”) e sistema discreto está na Figura 3.13.

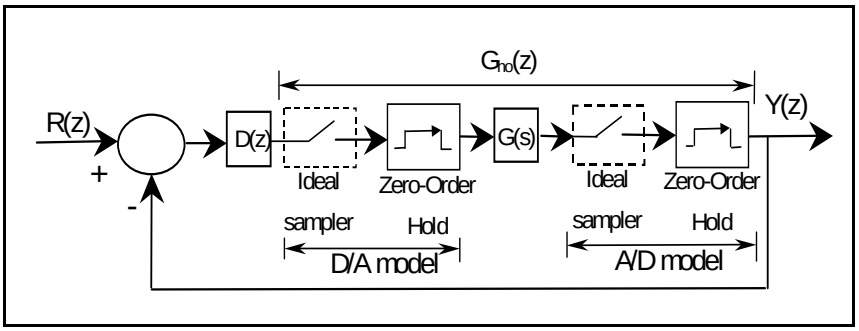


Figura 3.13 - Simplificação do modelo digital generalizado.

Aqui são feitas duas simplificações. Numa delas retira-se a quantização e a codificação/decodificação do sistema. Isso é feito com o objetivo de analisar unicamente os efeitos dos sinais discretos no tempo. Na outra simplificação o segurador de ordem-N (“N-Order Hold”) é transformado num Zero-Order Hold, isto é, um “segurador” de ordem zero. Esta é a equivalência “Hold” de forma mais simples e foi escolhida para descrever o modelo do segurador neste trabalho.

Da Figura 3.13 construímos finalmente a Figura 3.14.

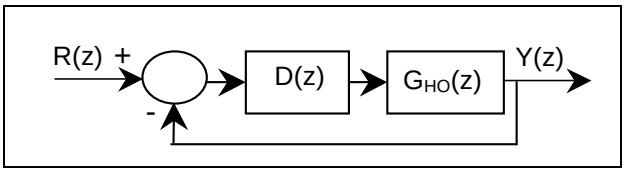


Figura 3.14 - Equivalente discreto do sistema analógico.

Na Figura 3.14 temos a política de controle D(z) que corresponde ao algoritmo de controle do sistema e o equivalente discreto “Zero-Order Hold” da planta $G_{HO}(z)$.

3.2. A TRANSFORMADA-Z E EQUIVALÊNCIAS DISCRETAS: FILTROS DISCRETOS

Sejam $\{...,-x_k,..., x_0, x_1, .., x_k, ...\}$ uma sequência, eventualmente obtida por amostragem periódica $x_k = x(k.T_S)$ de um dado sinal $x(t)$ nos instantes $t_k = k.T_S \quad , \forall k \in \boldsymbol{Z}$, onde $\boldsymbol{Z}$ é o conjunto dos números inteiros. Define-se a transformada-Z de dois lados deste sinal por,

$$X(z) \stackrel{\Delta}{=} Z\{x_k\} \stackrel{\Delta}{=} \sum_{k=-\infty}^{+\infty} x_k.z^{-k} \qquad , r_0 \le |z| \le R_0 \tag{3.1}$$

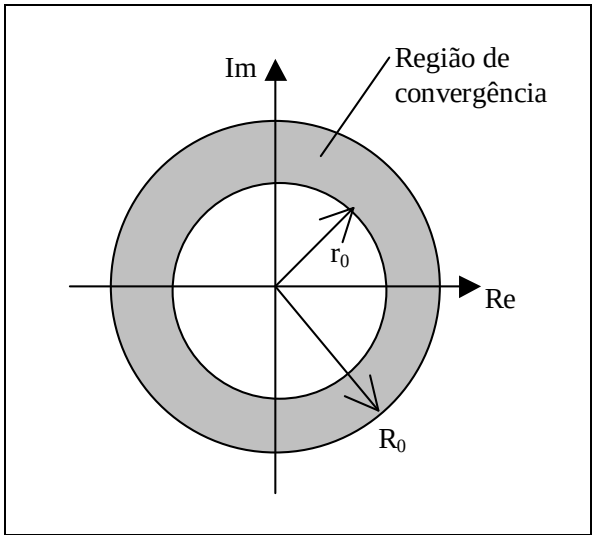


Figura 3.15 - Região de convergência para a equação 3.1.

que é conhecida como a transformada de dois lados, e a desigualdade mostrada descreve a região de convergência da transformação mostrada na Figura 3.15. Podemos definir também a transformada de um lado só,

$$X(z) \stackrel{\Delta}{=} Z\{x_k\} \stackrel{\Delta}{=} \sum_{k=0}^{+\infty} x_k.z^{-k} \qquad , r_0 \le |z| \le \infty \tag{3.2}$$

mudando apenas a região de convergência conforme mostrado na Figura 3.16.

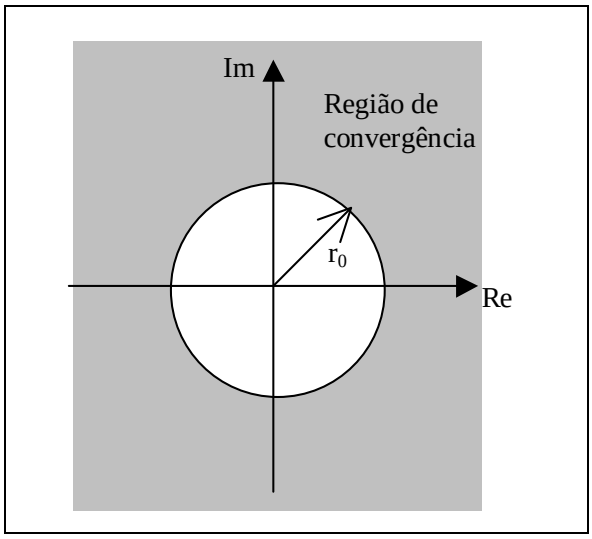


Figura 3.16 - Região de convergência da equação 3.2.

Uma operação muito usual em cálculos da transformada Z é a do “delay” de um período de amostragem,

$$Z\{x_{k-1}\} = \sum_{k=-\infty}^{\infty} x_{k-1}.z^{-k} = z^{-1}.X(z) \tag{3.3}$$

o que nos será útil para entender futuros desenvolvimentos. A obtenção de (3.3) é matematicamente muito simples tal como segue: fazendo j=k-1 temos,

$$\sum_{k=-\infty}^{\infty} x_{k-1}.z^{-(k-1+1)} = \sum_{k=-\infty}^{\infty} x_{k-1}.z^{-(k-1)}.z^{-1} = z^{-1}.\sum_{j=-\infty}^{\infty} x_j.z^{-j} = z^{-1}.X(z) \tag{3.4}$$

3.2.1. Integração Numérica: Mapeamento de Equações a Diferenças Finitas

Analisemos a seguir, detalhadamente, os seguintes métodos de integração numérica: “backward”, “forward”, Tustin e Schneider:

a) “Backward”: dado um sinal de entrada e_k (que no consecutivo instante passado, foi e_{k-1}) determine a saída (sua integral) u_k. Este método é também conhecido como método de Euler. Matematicamente isso é expresso pela seguinte equação a diferenças finitas (equivalente discreto de uma equação diferencial) conforme mostra a Figura 3.17:

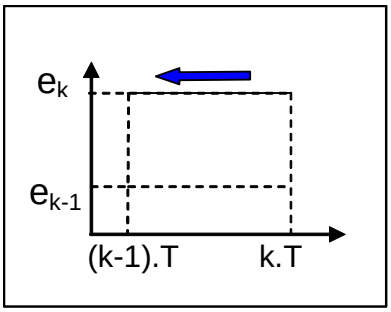


Figura 3.17 - Integrador “backward” no domínio do tempo. A seta da direita para a esquerda simboliza a diferença “backward” (**Franklin, 1981**).

$$e_k = \frac{\nabla u_k}{T} \tag{3.5}$$

$$\nabla u_k = u_k - u_{k-1} \tag{3.6}$$

Agora faremos uso do que aprendemos a respeito da transformada-Z de x_{k-1} por (3.3) calculando a transformada-Z da equação acima:

$$T.E(z) = U(z) - z^{-1}.U(z) \tag{3.7}$$

$$\frac{E(z)}{U(z)} = \frac{1 - z^{-1}}{T} = \frac{z-1}{T.z} \tag{3.8}$$

Portanto, a equivalência entre o plano-s e o plano-z da Figura 3.18 é dada por:

$$\frac{E(s)}{U(s)} = s \sim \frac{E(z)}{U(z)} = \frac{z-1}{T.z} \tag{3.9}$$

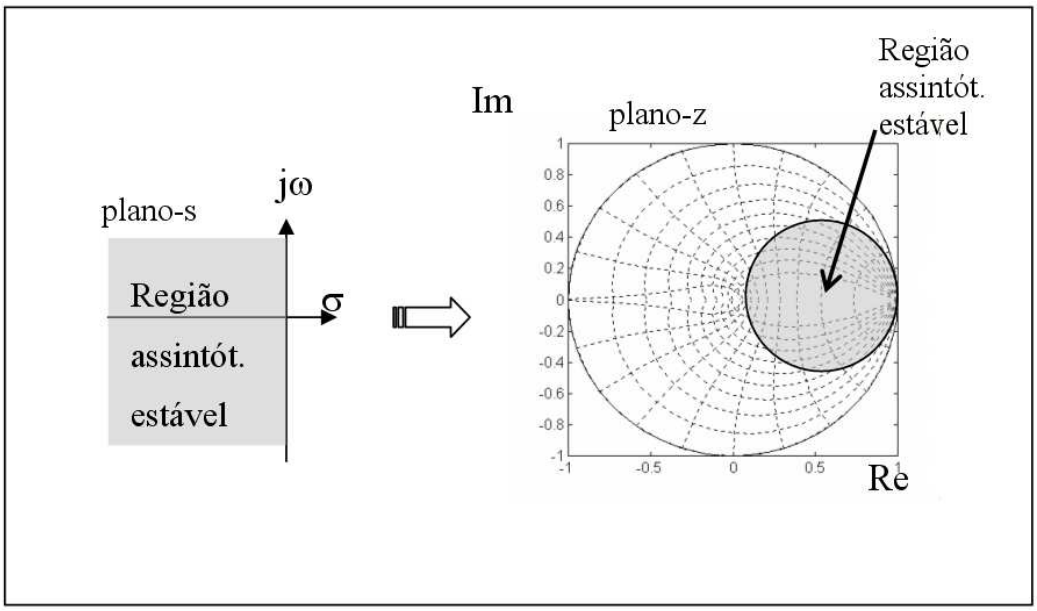


Figura 3.18 - Mapeamento do plano s para o plano z pela a equivalência "backward".

Exemplo 3.1: mapeamento "backward" atuando como integrador, conforme mostrado na Figura 3.19.

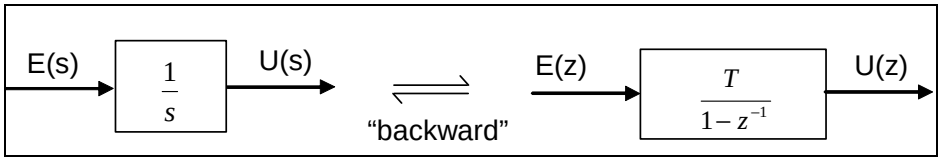


Figura 3.19 - Mapeamento "backward" atuando como integrador em z.

$$\frac{U(z)}{E(z)} = \frac{T}{1-z^{-1}} \tag{3.10}$$

$$U(z) - z^{-1}.U(z) = T.E(z) \tag{3.11}$$

Aplicando a transformação inversa teremos:

$$u_k - u_{k-1} = T.e_k \tag{3.12}$$

$$u_k = u_{k-1} + T.e_k \tag{3.13}$$

A representação gráfica da equação (3.13) está na Figura 3.20.

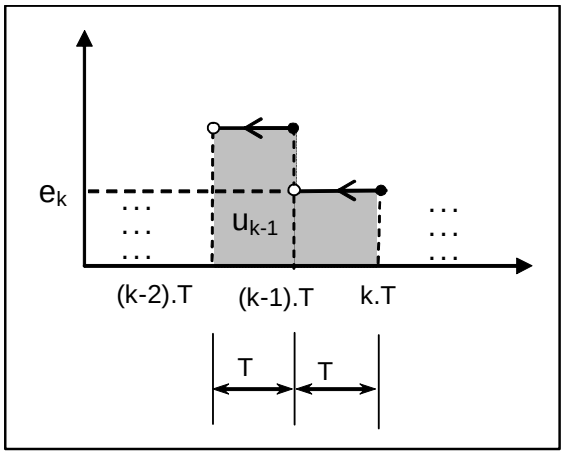


Figura 3.20 - Equivalência “Backward” atuando como integrador em k.T.

b) “Forward”: é uma representação não-causal ao construir uma saída (derivada) presente com uma informação de entrada (integral) futura. É importante salientar que filtros projetados com este mapeamento geram instabilidades.

$$e_{k-1} = \frac{1}{T}.\nabla u_k = \frac{u_k - u_{k-1}}{T} \tag{3.14}$$

Após a aplicação da transformada – Z na equação (3.14) teremos (Figuras 3.21 e 3.22):

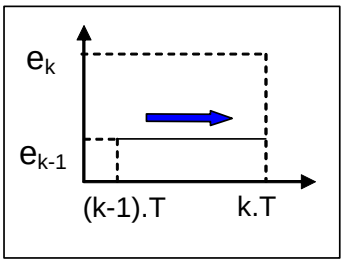


Figura 3.21 - Integrador “forward” no domínio do tempo.

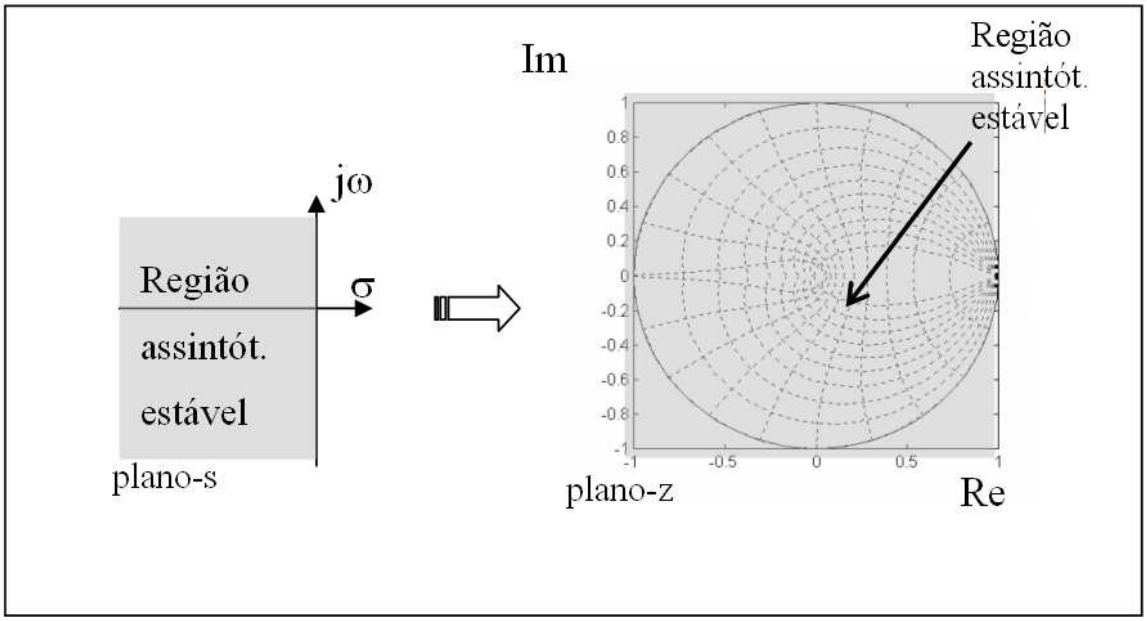


Figura 3.22 - Mapeamento do plano s para o plano z pela a aproximação "forward".

$$\frac{E(s)}{U(s)} = s \sim \frac{E(z)}{U(z)} = \frac{1 - z^{-1}}{T.z^{-1}} = \frac{z-1}{T} \qquad (3.15)$$

Pela equação (3.15) pode-se perceber a não-causalidade da aproximação "forward" notando-se que o número de zeros é maior que o número de pólos (a rigor um zero e nenhum pólo) e isso não pode ser real, destacando assim a não-causalidade da expressão. Entretanto, essa não causalidade do filtro "forward" pode enganar o projetista que fizer uso desta aproximação numa simulação em tempo virtual (tempo não real ou tempo mais que real) produzindo até alguns resultados satisfatórios. Entretanto, se este filtro for utilizado como derivador numa simulação em tempo real, com sensores na malha, por exemplo, ele provavelmente não funcionará porque necessita de uma entrada futura para produzir uma saída presente, ferindo assim o Princípio da Causalidade: para que um efeito presente seja gerado por um sistema qualquer é necessário que tenha havido uma causa passada que obrigou o sistema a produzir tal efeito.

Exemplo 3.2: aproximação "forward" atuando como integrador conforme mostra a Figura 3.23.

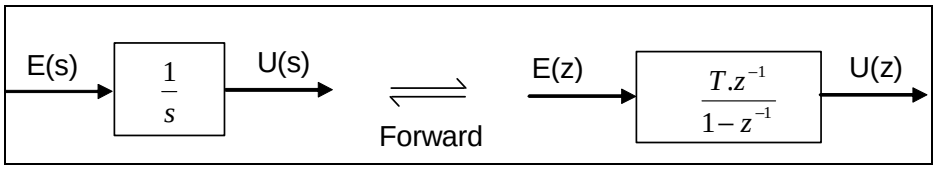


Figura 3.23 - Aproximação “forward” atuando como integrador.

$$\frac{U(z)}{E(z)} = \frac{T.z^{-1}}{1-z^{-1}} \tag{3.16}$$

$$U(z) - z^{-1}.U(z) = T.z^{-1}.E(z) \tag{3.17}$$

Aplicando a transformação inversa vem (Figura 3.23):

$$u_k - u_{k-1} = T.e_{k-1} \tag{3.18}$$

$$u_k = u_{k-1} + T.e_{k-1} \tag{3.19}$$

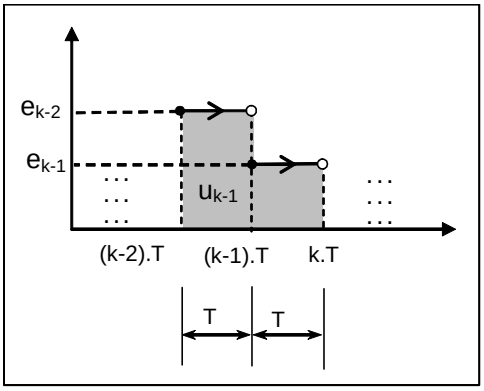


Figura 3.24 - Aproximação “forward” atuando como integrador.

Provavelmente esta perda de informação da entrada presente seja responsável pela fragilidade de segurança nos projetos com esta aproximação discreta.

c) Aproximação Tustin, bilinear ou trapezoidal: de todas as técnicas de integração numérica apresentadas até aqui, esta é a melhor por descrever um mapeamento fiel, isto é, mapear toda a região assintoticamente estável do plano s (semiplano esquerdo) para o interior do círculo unitário do plano z. Este método de integração utiliza a entrada e a saída passadas e a entrada presente para construir a saída presente, tal como podemos verificar na equação (3.21).

A regra de Tustin pode ser retirada de uma equação a diferenças finitas de Adams-Moulton de segunda ordem (**Schneider, 1991**) e pode ser escrita como segue:

$$e_k = \frac{2}{T}.\nabla u_k - e_{k-1} = \frac{2}{T}.(u_k - u_{k-1}) - e_{k-1} \tag{3.20}$$

$$e_k = \frac{2}{T}.u_k - \frac{2}{T}.u_{k-1} - e_{k-1} \tag{3.21}$$

Aplicando a transformada-Z vem (Figuras 3.25 e 3.26):

$$E(z) + z^{-1}.E(z) = \frac{2}{T}.\left(U(z) - z^{-1}.U(z)\right) \tag{3.22}$$

$$\frac{E(s)}{U(s)} = s \sim \frac{E(z)}{U(z)} = \frac{2}{T}.\frac{z-1}{z+1} \tag{3.23}$$

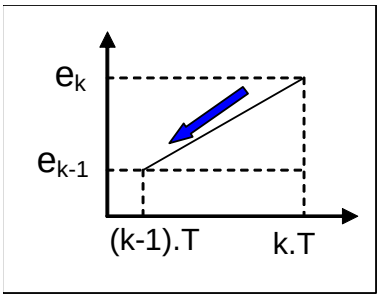


Figura 3.25 - Integrador “Tustin” no domínio do tempo.

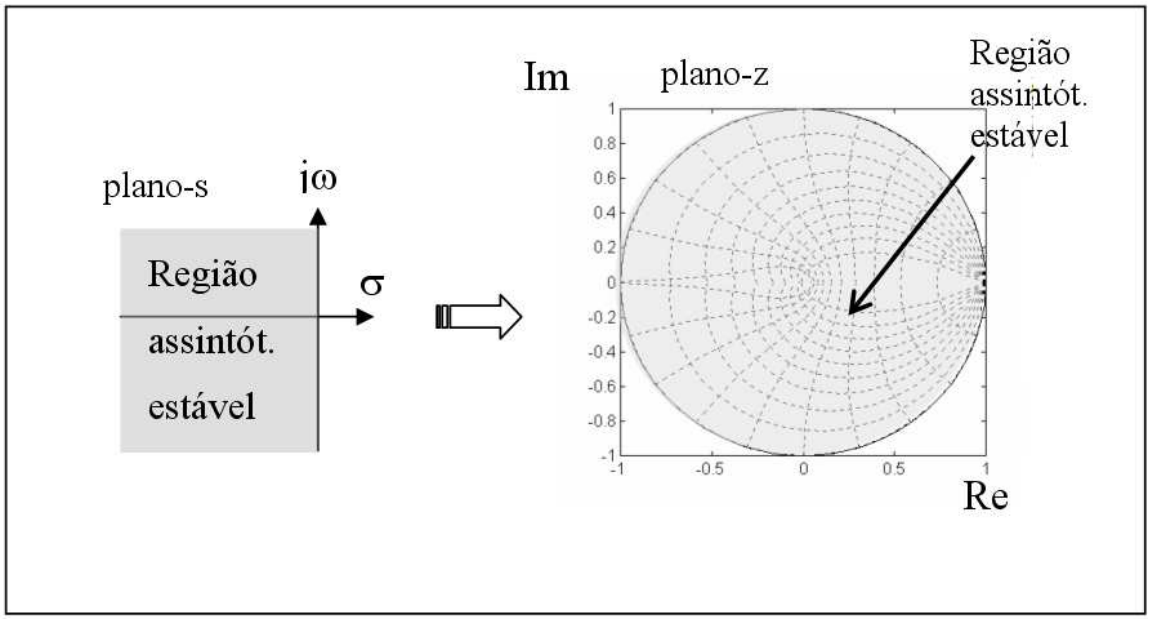


Figura 3.26 - Mapeamento entre os planos s e z pelo integrador “Tustin”.

Exemplo 3.3: equivalência “Tustin” atuando como integrador (Figura 3.27).

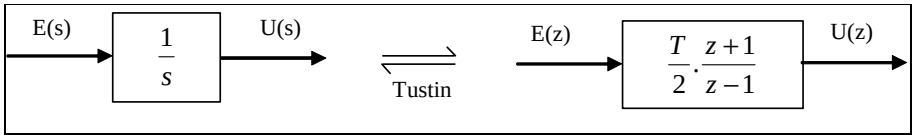


Figura 3.27 - Diagrama em blocos básico do integrador por “Tustin”.

$$\frac{U(z)}{E(z)} = \frac{T}{2}.\frac{z+1}{z-1} = \frac{T}{2}.\frac{1+z^{-1}}{1-z^{-1}} \tag{3.24}$$

$$U(z) - z^{-1}.U(z) = \frac{T}{2}.\left(E(z) + z^{-1}.E(z)\right) \tag{3.25}$$

Aplicando uma transformação inversa vem:

$$u_k - u_{k-1} = \frac{T}{2}.\left(e_k + e_{k-1}\right) \tag{3.26}$$

$$u_k = u_{k-1} + \frac{T}{2}.\left(e_k + e_{k-1}\right) \tag{3.27}$$

Notemos que $T.\frac{(e_k + e_{k-1})}{2}$ é a área de um trapézio de altura T e bases e_k e e_{k-1} (Figura 3.28).

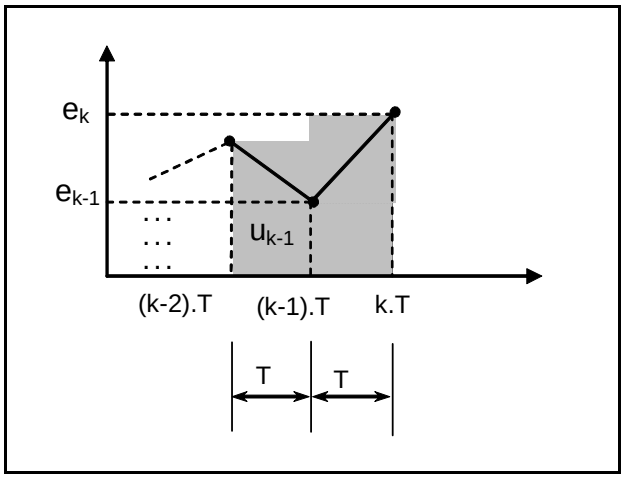


Figura 3.28 - “Tustin” como integrador.

d) A equivalência de Schneider:

A regra de Schneider pode ser obtida a partir de uma equação a diferenças finitas de Adams-Moulton de terceira ordem (**Schneider, 1991**):

$$u_k = u_{k-1} + \frac{T}{12}.(5.e_k + 8.e_{k-1} - e_{k-2}) \tag{3.28}$$

a qual pode ser escrita na forma:

$$e_k = \frac{1}{5}.\left(\frac{12}{T}.\nabla u_k - 8.e_{k-1} + e_{k-2}\right) \tag{3.29}$$

que, após a aplicação da transformada-Z vem:

$$E(z) = \frac{1}{5}.\left(\frac{12}{T}.(1 - z^{-1})U(z) - 8.z^{-1}.E(z) + z^{-2}.E(z)\right) \tag{3.30}$$

$$\frac{E(s)}{U(s)} = s \sim \frac{E(z)}{U(z)} = \frac{12}{T}.\frac{z^2 - z}{5z^2 + 8z - 1} \tag{3.31}$$

suas propriedades relevantes e limitações para este trabalho serão discutidas no Apêndice A (seção A.5.3.e).

3.3. Regras de mapeamentos analógico-digitais com parâmetros de sintonia

3.3.1. Regra “ST1” a um parâmetro de sintonia:

As limitações dos métodos clássicos apresentados na literatura visando preservar a estabilidade na presença de elevados ganhos de controle e elevados valores de período de amostragem levou à proposta de um novo método de mapeamento analógico-digital (“sz”) em **Tredinnick (1999)**. Essa regra de mapeamento possui a seguinte equação a diferenças-finitas:

$$e_k = \frac{2}{T}.\nabla u_k - \xi.e_{k-1} \tag{3.32}$$

Aplicando a transformada-Z sobre (3.32) obtemos:

$$s \sim \frac{2}{T}.\frac{z-1}{z+\xi} \qquad ;0<\xi<1 \tag{3.33}$$

A equação (3.33) mostra claramente um deslocamento do pólo z = -1 da regra de Tustin para z’ = -ξ, 0 ≤ ξ ≤ 1. Isto evita ou retarda a instabilização em sistemas de malha-fechada usando ξ como um novo parâmetro de projeto (além dos ganhos de controle e do período de amostragem. A regra ST a um parâmetro se reduz à regra de Tustin se $\xi = 1$ ou à regra “backward” se $\xi = 0$,

se $\xi = 1 \rightarrow$ centro = 0 ; raio = 1; (Tustin); (3.34)

se $\xi = 0 \rightarrow$ centro = ½ ; raio = ½; (backward); (3.35)

Sua inversa é dada por:

$$s = \frac{2}{T}.\frac{z-1}{z+\xi} \quad \Rightarrow \quad z = \frac{2+s.T.\xi}{2-s.T} \qquad (0 \le \xi \le 1) \tag{3.36}$$

A observação de tais características levou a elaboração do seguinte Teorema devido ao autor desta Tese,

Teorema 3.1: a regra ST1 mapeia a o semiplano esquerdo do plano s para um círculo com centro dado pela equação:

$$\text{centro} = \frac{1}{2}.(1-\xi) \tag{3.37}$$

e raio dado por,

$$\text{raio} = \frac{1}{2}.(1+\xi) \tag{3.38}$$

□

Prova: para provar o Teorema 3.1 reescrevamos (3.36) para s = j.ω tendo o cuidado de inserir no numerador de (3.36) a expressão "auto-cancelável" $\frac{1}{2}.(1-\xi).[2-\text{j}.\omega.\text{T}]-\frac{1}{2}.(1-\xi).[2-\text{j}.\omega.\text{T}]$,

$$z = \frac{\frac{1}{2}.(1-\xi).[2-j.\omega.T]-\frac{1}{2}.(1-\xi).[2-j.\omega.T]+2+j.\omega.T.\xi}{2-j.\omega.T} \tag{3.39}$$

$$z = \frac{1}{2}.(1-\xi).\frac{[2-j.\omega.T]}{[2-j.\omega.T]}+\frac{-\frac{1}{2}.(1-\xi).[2-j.\omega.T]+2+j.\omega.T.\xi}{2-j.\omega.T} \tag{3.40}$$

$$z-\frac{1}{2}.(1-\xi)=\frac{1+\xi+j.\omega.T.\frac{1}{2}.(1+\xi)}{2-j.\omega.T} \tag{3.41}$$

$$z-\frac{1}{2}.(1-\xi)=\frac{1}{2}.(1+\xi).\exp\left(j\tan^{-1}\left(\frac{4.\omega.T}{4-(\omega.T)^2}\right)\right) \tag{3.42}$$

A equação (3.42) já prova o Teorema 1 visto que inclui (3.37) e (3.38). Para melhorar a visualização podemos reescrever (3.42) apenas em módulo:

$$\left| z - \frac{1}{2}.(1-\xi) \right| = \frac{1}{2}.(1+\xi) \qquad (3.43)$$

□

Graficamente, a Figura 3.29 apresenta o mapeamento da regra ST1 em comparação com as regras de Tustin e “backward”. Assim, conclui-se que a regra ST1 é uma generalização que inclui as regras de Tustin e “backward”.

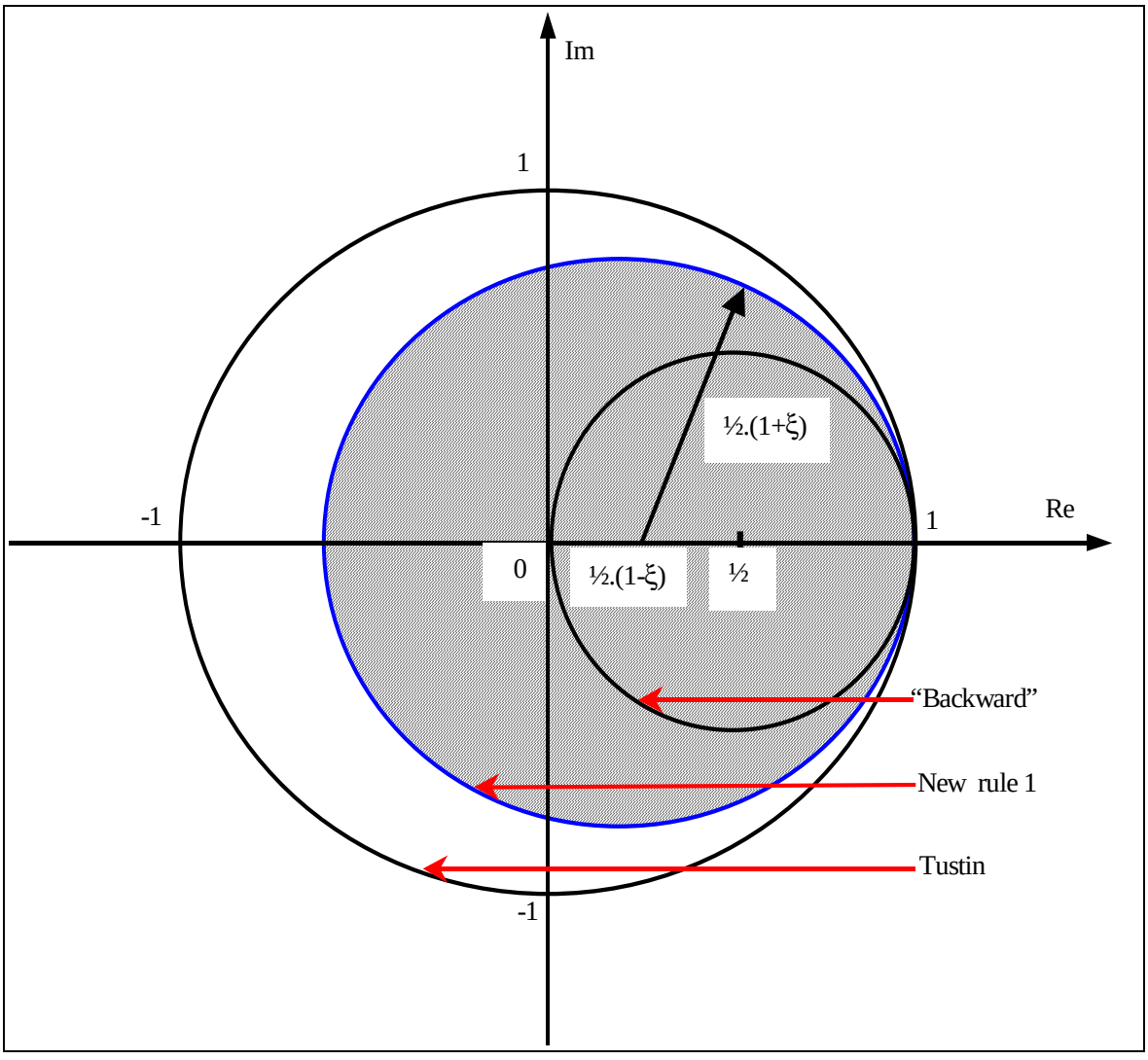


Figura 3.29 - Mapeamentos das regras ST1, Tustin e “Backward”.

3.3.2. Regra de mapeamento analógico-digital com dois parâmetros de sintonia

Uma generalização da equação a diferenças finitas (3.32) pode ser dada por,

$$e_k = \frac{2}{T}.(u_k - \xi_1 u_{k-1}) - \xi_2.e_{k-1} \tag{3.44}$$

Esta é a regra de mapeamento ST2 onde, diferentemente da ST1, temos dois parâmetros de sintonia. Aplicando a transformada Z em (3.44) teremos o seguinte resultado,

$$s \sim \frac{2}{T}.\frac{z-\xi_1}{z+\xi_2} \qquad ;\begin{cases} 0 < \xi_1 < 1 \\ 0 < \xi_2 < 1 \end{cases} \tag{3.45}$$

Um estudo similar ao realizado no Teorema 3.1 tendo como base (3.45) leva ao Teorema 3.2, também devido ao autor desta Tese.

Teorema 3.2: a regra ST2 mapeia o semiplano esquerdo do plano s para um círculo com centro dado pela equação:

$$\frac{1}{2}.(\xi_1 - \xi_2) \tag{3.46}$$

e raio dado por,

$$\frac{1}{2}.(\xi_1 + \xi_2) \tag{3.47}$$

□

Prova:

$$z = \frac{\frac{1}{2}.(\xi_1 - \xi_2).[2 - j.\omega.T] - \frac{1}{2}.(\xi_1 - \xi_2).[2 - j.\omega.T] + 2\xi_1 + j.\omega.T.\xi_2}{2 - j.\omega.T} \tag{3.48}$$

$$z = \frac{1}{2}.(\xi_1 - \xi_2).\frac{[2 - j.\omega.T]}{[2 - j.\omega.T]} + \frac{-\frac{1}{2}.(\xi_1 - \xi_2).[2 - j.\omega.T] + 2\xi_1 + j.\omega.T.\xi_2}{2 - j.\omega.T} \tag{3.49}$$

$$z - \frac{1}{2}.(\xi_1 - \xi_2) = \frac{\xi_2 - \xi_1 - \frac{\xi_1}{2}.j.\omega.T + \frac{\xi_2}{2}.j.\omega.T + 2\xi_1 + \xi_2\, j.\omega.T.\xi}{2 - j.\omega.T} \tag{3.50}$$

$$z - \frac{1}{2}.(\xi_1 - \xi_2) = \frac{\xi_1 + \xi_2 + j.\omega.T.\frac{1}{2}.(\xi_1 + \xi_2)}{2 - j.\omega.T} \tag{5.1}$$

$$z - \frac{1}{2}.(\xi_1 - \xi_2) = \frac{1}{2}.(\xi_1 + \xi_2).\frac{2 + j.\omega.T}{2 - j.\omega.T}.\frac{2 + j.\omega.T}{2 + j.\omega.T} \tag{3.51}$$

$$z - \frac{1}{2}.(\xi_1 - \xi_2) = \frac{1}{2}.(\xi_1 + \xi_2).\frac{(2 + j.\omega.T)^2}{4 + (\omega.T)^2} \tag{3.52}$$

$$z - \frac{1}{2}.(\xi_1 - \xi_2) = \frac{1}{2}.(\xi_1 + \xi_2).\exp\left(j\tan^{-1}\left(\frac{4.\omega.T}{4 - (\omega.T)^2}\right)\right) \tag{3.53}$$

$$\left|z - \frac{1}{2}.(\xi_1 - \xi_2)\right| = \frac{1}{2}.(\xi_1 + \xi_2) \tag{3.54}$$

□

A equação (3.54) descreve uma família de círculos de raio $\frac{1}{2}.(\xi_1 + \xi_2)$ dentro do círculo unitário com centro $\frac{1}{2}.(\xi_1 - \xi_2)$ posicionado sobre o eixo real, tal como mostrado na Figura 3.30.

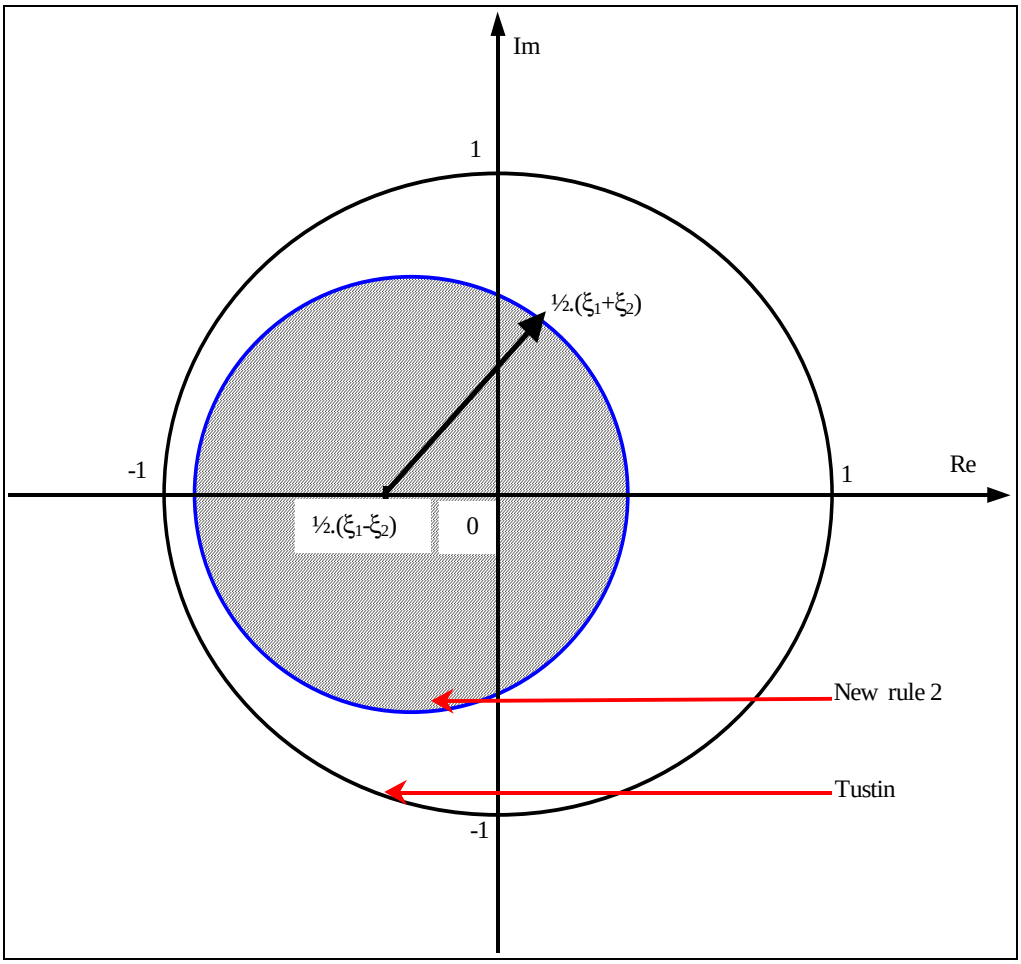


Figura 3.30 - Mapeamentos das regras ST2 (região hachurada) e Tustin.

Uma comparação entre as Figuras 3.29 e 3.30 nos levará a conclusão que as regras ST1, Tustin e “backward” são casos particulares da nova regra ST2 e fornecem mais robustez ao sistema de controle que as regras clássicas.

Com os dois parâmetros de sintonia temos então dois graus de liberdade adicionais permitindo o posicionamento do mapeamento concentricamente ao círculo unitário e com círculo menor que 1, garantindo um enorme amortecimento. Tal situação é realizada na prática por diversos engenheiros, porém sem um tratamento teórico adequado. O superamortecimento introduzido é realizado sobre os pólos e não sobre os zeros.

CAPÍTULO 4

PROPOSTA DE MODELO MATEMÁTICO PARA O FALSEAMENTO

Neste Capítulo apresentar-se-á uma representação em blocos para o problema do falseamento (“aliasing”) que foi denominada de representação por multicaminhos em paralelo, tendo como base o modelo matemático para o falseamento.

4.1. UM MODELO MATEMÁTICO PARA O FALSEAMENTO

Damos continuidade às investigações a respeito da influência do período de amostragem no desempenho e na estabilidade de sistemas amostrados tais como o apresentado na Figura 4.1, propondo um modelo matemático para o falseamento (“aliasing” ou mascaramento).

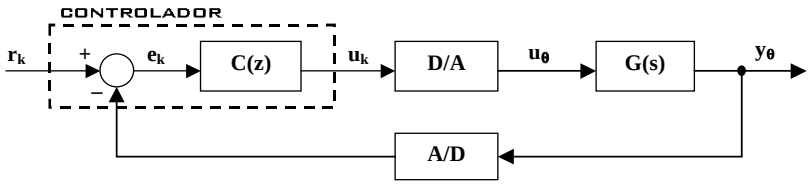


Figura 4.1 - Sistema de controle digital de uma planta analógica.

Usando reprojeto digital (“digital redesign”) um controle analógico C(s) pode ser discretizado por algum método de mapeamento s-z para gerar C(z):

$$\left[G(s),C(s)\right]_T \xrightarrow{\substack{método\\ de\ discretização}} C(z) \tag{4.1}$$

É utilizada usualmente em controle digital a digitalização da planta G(s) discretizando-a por “step-invariance” (“ZOH equivalence”), preferencialmente, e a partir daí calcular C(z):

$$G(s)\big|_T \xrightarrow{\substack{método\\ de\ discretização}} G_*(z) \xrightarrow{projeto\ do\ controlador} C(z) \tag{4.2}$$

Tal como na Figura 3.5, é mostrado na Figura 4.2 no domínio da freqüência um sinal analógico G(s) e seu equivalente discretizado Gd(s) onde apresentamos

neste a freqüência de amostragem ω_S ($\omega_S = \frac{2\pi}{T}$), a freqüência de Nyquist ω_N ($\omega_N = \frac{\omega_S}{2} = \frac{\pi}{T}$) e o período de amostragem T. Notamos que no caso mostrado não ocorreu o falseamento dado que os lóbulos repetidos pela amostragem não se sobrescreveram. Segundo **Franklin (1981)**:

$$G^*(s) = \frac{1}{T}\sum_{k=-\infty}^{\infty} G(s - jk\omega_s) = \frac{1}{T}[G(s) + \Delta G^*(s)] \tag{4.3}$$

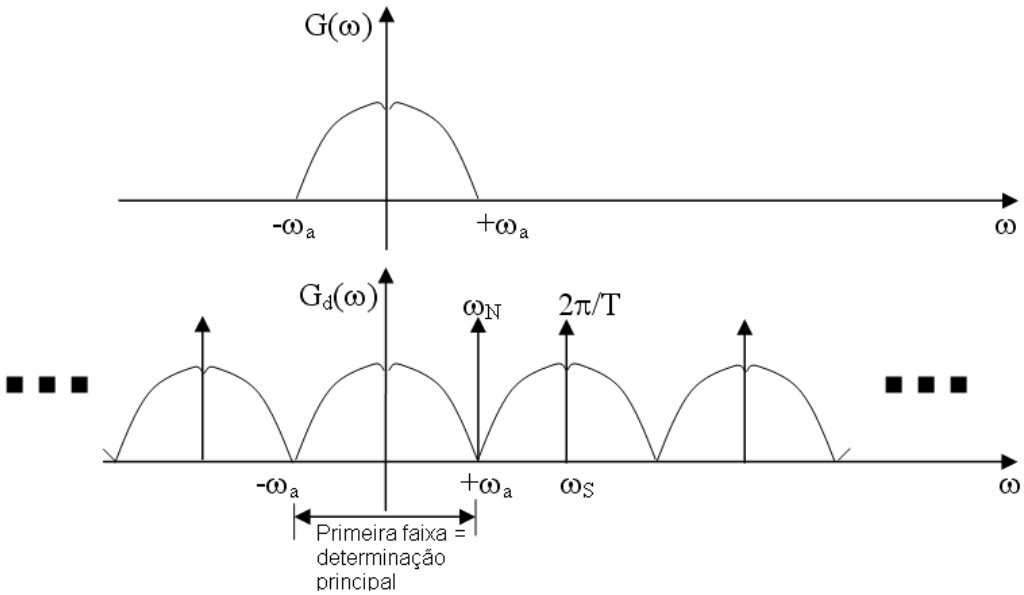


Figura 4.2 - Processo de amostragem sem falseamento.

Na equação (4.3), o $\frac{1}{T}$ vem do amostrador, um modelo do amostrador ideal, e operador “zero-order hold” (segurador de ordem-zero) é representado por $1 - e^{-sT}$ (**Franklin, 1981**, Capítulo 4).

$$\Delta G^*(s) = \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} G(s - jk\omega_s) \tag{4.4}$$

$$G_d(s) = \frac{1-e^{-sT}}{s} G^*(s) = \frac{1-e^{-sT}}{s}\frac{1}{T}\sum_{k=-\infty}^{\infty} G(s - jk\omega_s) = \frac{1-e^{-sT}}{s}.\frac{1}{T}[G(s) + \Delta G^*(s)] \tag{4.5}$$

$$G_T(z) = G_d(s)\big|_{z=e^{sT}} = \frac{1-e^{-sT}}{s} G^*(s)\big|_{z=e^{sT}} = \frac{1-e^{-sT}}{s} . \frac{1}{T} [G(s) + \Delta G^*(s)]\big|_{z=e^{sT}} \tag{4.6}$$

$$G_T(z) = G_{*,T}(z) + \Delta G_{*,T}(z) \tag{4.7}$$

$$G_{*,T}(z) = \frac{1-e^{-sT}}{s} . \frac{1}{T} G(s)\big|_{z=e^{sT}} \tag{4.8}$$

$$\Delta G_{*,T}(z) = \frac{1-e^{-sT}}{s} . \frac{1}{T} \sum_{\substack{k\neq 0 \\ k=-\infty}}^{\infty} G(s - jk\omega_s)\big|_{z=e^{sT}} \tag{4.9}$$

A transformada-Z do sinal analógico $G_d(s)$ tem sua determinação principal (ou domínio principal) entre $-\pi/T$ e $+\pi/T$ para a chamada "freqüência discreta" ω_d ao passo que a "freqüência analógica" ω_a é contínua e assim vai de $-\infty$ a $+\infty$. Na Figura 4.3 apresenta-se um gráfico buscando exemplificar a relação existente entre a base de freqüência do caso analógico que chamaremos de ω_a com a do caso discreto ω_d.

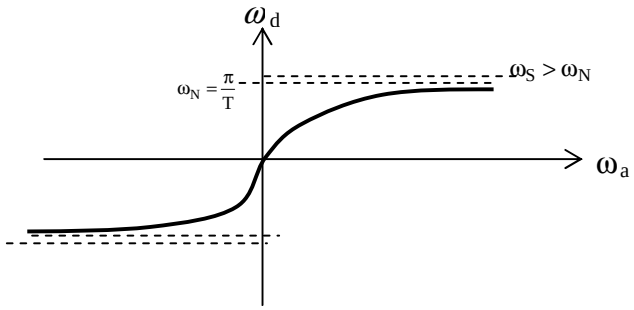


Figura 4.3 - Gráfico ω_d *x* ω_a.

Assim conclui-se que ao fazermos projetos usando a transformada-Z estamos nessa faixa $\omega_d \in \left(\frac{-\pi}{T}, \frac{\pi}{T}\right)$. O mapeamento Bilinear é biunívoco do plano complexo no plano complexo, mas não é um isomorfismo nesse plano visto que ele não conserva somas e produtos; assim, não é um isomorfismo da álgebra dos complexos. Ele conserva limites de seqüências, então é um isomorfismo topológico e, nesse caso, dizemos que ele é um homomorfismo. Já o mapeamento

exponencial é um mapeamento multívoco, pois a exponencial complexa é periódica de período 2πj levando a cada uma das infinitas faixas (definidas entre assíntotas horizontais dadas por $\frac{\pi}{T}$(2k+1), sendo k=0, 1, 2, ...). O falseamento ("aliasing") surge dessa incapacidade matemática dos projetos de o considerarem sendo assim inerente aos mesmos, porém havendo a possibilidade de projetar compensadores que diminuam seus efeitos deletérios à estabilidade.

Partindo de tal raciocínio temos a seguinte equação representando o modelo da discretização completa da planta G, i.e., considerando todas as faixas inclusive a determinação principal:

$$G_T(z) = G_{*,T}(z) + \Delta G_{*,T}(z) \qquad (4.10)$$

Onde, na equação (4.10):

a) $G_T(z)$: "discreto verdadeiro", a "representação" discreta completa da planta.

b) $G_{*,T}(z)$: mapeamento feito na faixa de $-\pi/T$ a $+\pi/T$, sendo o valor π/T igual à freqüência de Nyquist, $\omega_N = \omega_S/2 = \frac{2\pi}{T}.\frac{1}{2} = \pi/T$, sendo ω_S é a freqüência de amostragem. O que é o resultado obtido do ZOH ("step invariance") de G, o mapeamento feito na determinação principal. Esse termo é uma "representação" extremamente limitada que o controlador discreto tem da planta.

c) $\Delta G_{*,T}(z)$: é o termo que gera o falseamento ("aliasing"). Aqui ocorre o enrolamento de todo o eixo imaginário jω na circunferência unitária, exceto para a faixa de $-\pi/T$ a $+\pi/T$:

$$\Delta G_{*,T}(z) = \frac{1}{T}\sum_{\substack{k\neq 0 \\ k=-\infty}}^{\infty} \frac{1-e^{-sT}}{s}.\frac{G(s)}{1}\Bigg|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \qquad (4.11)$$

A equação (4.11) é então o modelo matemático para o fenômeno do falseamento usando o ZOH ("step invariance").

4.2. REPRESENTAÇÃO DO MODELO DISCRETO DA PLANTA POR MULTICAMINHOS EM PARALELO

As equações (4.10) e (4.11) sugerem uma representação do modelo discreto de uma planta tal como mostrado na Figura 4.4.

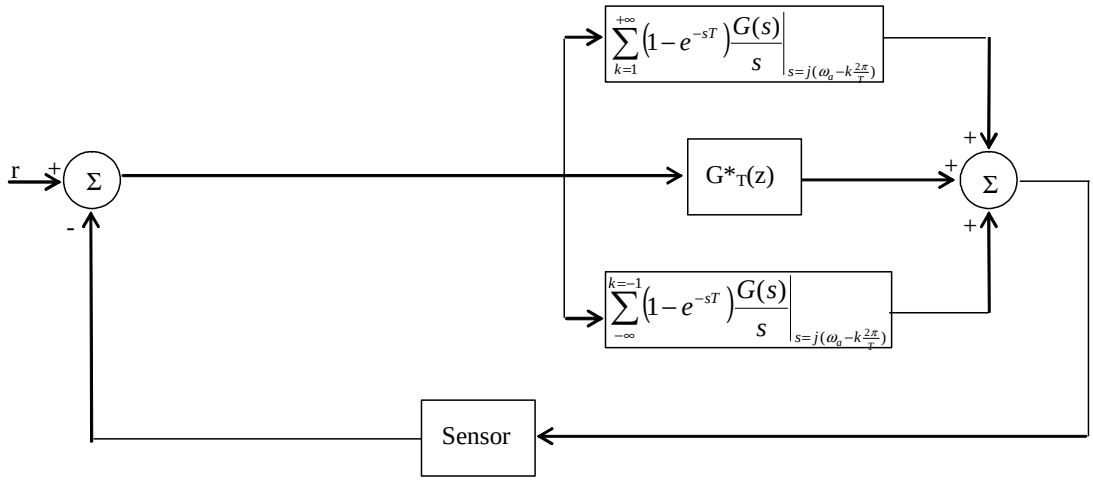


Figura 4.4 - Modelo de multicaminhos em forma fechada.

Na Figura 4.5 podemos ver a identificação do falseamento no modelo de multicaminhos apresentado na figura anterior.

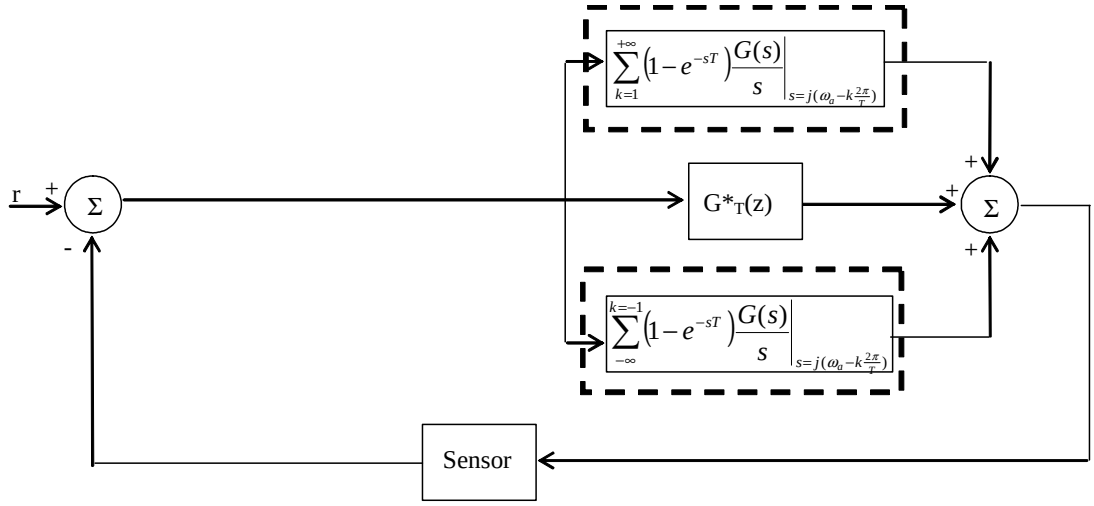


Figura 4.5 - Identificação do falseamento no modelo de multicaminhos (blocos contidos dentro da linha tracejada).

Num tratamento do modelo de multicaminhos por forma aberta podemos representar o disposto nas Figuras (4.10) e (4.11) tal como apresentado na Figura 4.6.

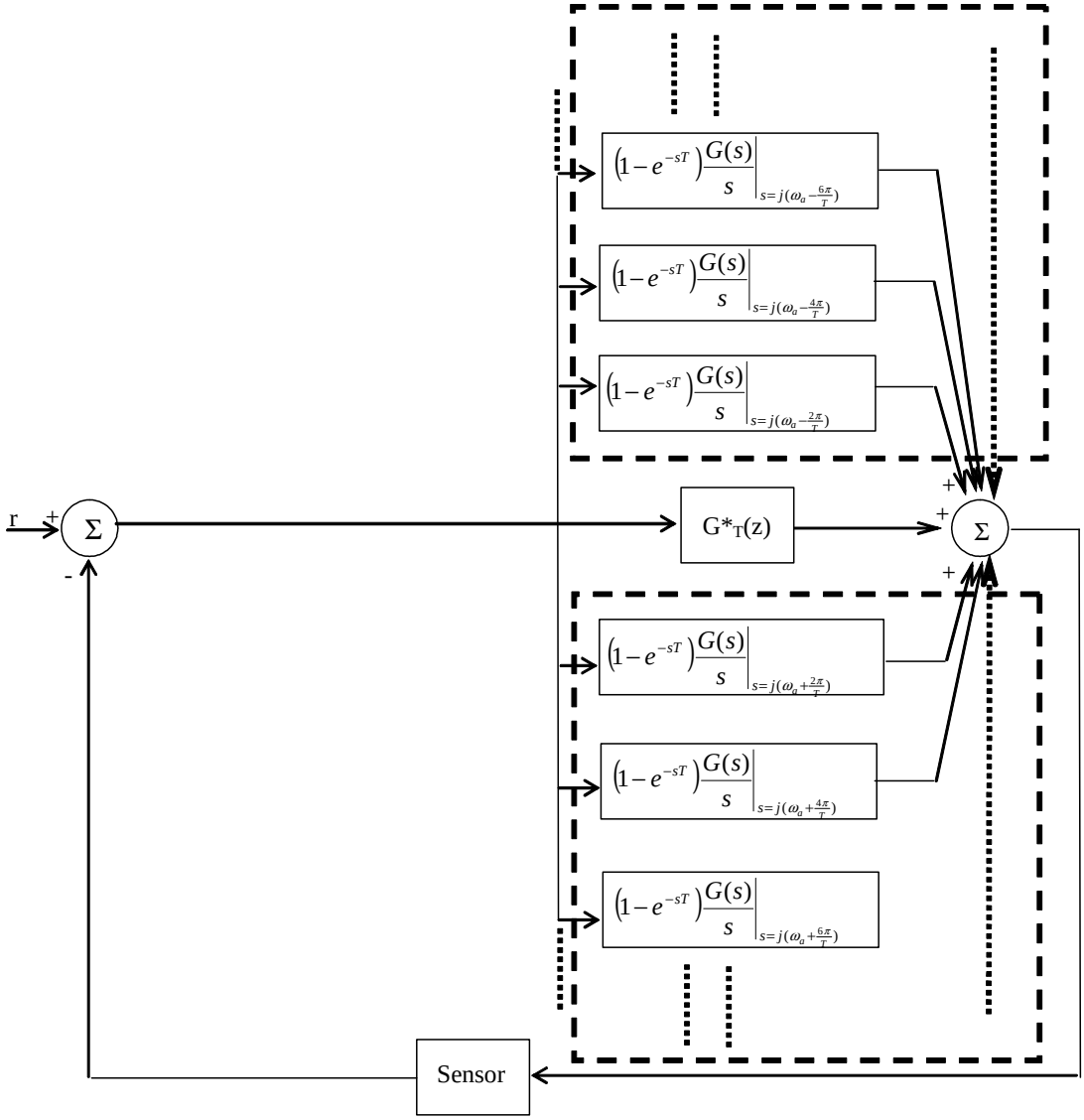


Figura 4.6 - Identificação do falseamento no modelo de multicaminhos de forma aberta (blocos contidos dentro da linha tracejada).

O falseamento representado na Figura 4.6, algo invisível ao olho humano, porém presente no mundo real, pode ser interpretado como fatores de ganho adicionais que, mediante o período de amostragem T e a planta G, podem alterar a posição dos pólos do "root-locus" de $G^*_T(s)$ de tal sorte que podem levar tais pólos à região de instabilidade. Tal modelo pode ser útil em simulações numéricas para prever o comportamento de determinados sistemas na presença do falseamento. Nas simulações do Capítulo 6 o sensor é feito unitário por simplicidade.

CAPÍTULO 5

DIAGRAMAS DE BODE E GRÁFICOS FASORIAIS

Neste Capítulo revela como o período de amostragem altera as curvas representativas para diversos sistemas de controle em diagramas de Bode, mostrando que para determinados valores de período de amostragem o sistema em malha-fechada pode ser assintoticamente estável e para outros valores pode ser instável. Além da análise por diagramas de Bode, gráficos fasoriais também são apresentados.

5.1. INTRODUÇÃO À ANÁLISE DE SISTEMAS DE CONTROLE DISCRETO DE PLANTAS ANALÓGICAS

Tal como ensinado em Franklin (1981), se temos que {.}*(s) é o sinal amostrado de {.} então,

$$Y^*(s) = \frac{H^*(s)}{1+H^*(s)} R^*(s) \tag{5.1}$$

Onde Y* é o sinal amostrado de saída e R* é o sinal amostrado de referência e

$$H^*(s) = \left(1-e^{-sT}\right).D^*(s).\left(\frac{G(s)}{s}\right)^* \tag{5.2}$$

sendo D* a função de transferência do controlador discreto, G(s) a função de transferência da planta considerada e T é o período de amostragem.

Tal como ensinado em Franklin (1981) podemos verificar que,

$$Y(s) = R^*(s).\frac{D^*(s)}{1+H^*(s)}\frac{\left(1-e^{-sT}\right)}{s}.G(s) \tag{5.3}$$

a qual permite calcular a resposta do sistema entre os instantes amostrados, i.e., ao inserirmos a função exponencial estamos incluindo não apenas o domínio

principal coberto pela transformada Z mas também os domínios auxiliares, ou seja, o mascaramento (ver figura abaixo).

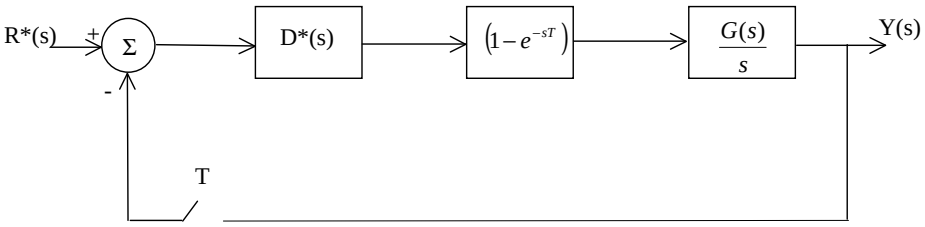


Figura 5.1 - Modelo do sistema de controle amostrado.

Finalmente, se em (5.3) H*(s) tiver módulo igual a 1 e fase igual a -180 graus será uma indicação de que o sistema se encontra “condicionalmente estável”, podendo instabilizar-se ou estabilizar-se dependendo de uma variação no período de amostragem. Assim, seguem diversos diagramas de Bode para diversos casos usando plantas de primeira e segunda ordens e oito controladores distintos sendo mapeados por seis tipos de regras distintas. Os controladores usados foram: proporcional, derivativo, integral, proporcional+derivativo (PD), PID, avanço de fase (“Lead”), atraso de fase (“Lag”) e avanço-atraso de fase (“Lead-Lag”). Os mapeamentos usados foram: forward, backward, Tustin, ST1, ST2 e Schneider.

A planta de primeira ordem já com o segurador de primeira ordem é a seguinte (obtida de **Franklin, 1981**, página 89, onde foi atribuído valor numérico para a constante a),

$$G(s) = \frac{a}{s+a} \tag{5.4}$$

$$\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*} = \frac{\frac{1}{2}}{e^{sT}-\frac{1}{2}} \tag{5.5}$$

A planta de segunda ordem usada foi,

$$G(s) = \frac{a}{s.(s+a)} \tag{5.6}$$

$$\left(\frac{G(s)}{s}\right)^{*} = \left(\frac{a}{s^2(s+a)}\right)^{*} \tag{5.7}$$

Fazendo uso da Tabela B2 (equação 13) existente em Franklin (1981),

$$\left(\frac{G(s)}{s}\right)^{*} = \frac{e^{sT}\left[\left(a.T-1+e^{-aT}\right)e^{sT} + \left(1-e^{-aT}-a.T.e^{-aT}\right)\right]}{a.\left(e^{sT}-1\right)^2.\left(e^{sT}-e^{-aT}\right)} \tag{5.8}$$

Inserindo o segurador de primeira ordem temos,

$$\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*} = \left(1-e^{-sT}\right)\frac{e^{sT}\left[\left(a.T-1+e^{-aT}\right)e^{sT} + \left(1-e^{-aT}-a.T.e^{-aT}\right)\right]}{a.\left(e^{sT}-1\right)^2.\left(e^{sT}-e^{-aT}\right)} \tag{5.9}$$

$$\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*} = \frac{\left(e^{sT}-1\right)}{e^{sT}}\frac{e^{sT}\left[\left(a.T-1+e^{-aT}\right)e^{sT} + \left(1-e^{-aT}-a.T.e^{-aT}\right)\right]}{a.\left(e^{sT}-1\right)^2.\left(e^{sT}-e^{-aT}\right)} \tag{5.10}$$

$$\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*} = \frac{\left(a.T-1+e^{-aT}\right)e^{sT} + \left(1-e^{-aT}-a.T.e^{-aT}\right)}{a.\left(e^{sT}-1\right).\left(e^{sT}-e^{-aT}\right)} \tag{5.11}$$

Para as diversas simulações realizadas na próxima seção foram utilizadas as seguintes equações adicionais para os controles:

a) Controle proporcional:

$$D^{*}(s) = k_p \tag{5.12}$$

b) Controle derivativo: (ganho k_d)

i. Mapeamento forward:

$$D^{*}(s) = \frac{1}{T}\left(e^{sT}-1\right) \tag{5.13}$$

ii. Mapeamento backward:

$$D^*(s) = \frac{\left(e^{sT} - 1\right)}{Te^{sT}} \tag{5.14}$$

iii. Mapeamento Tustin:

$$D^*(s) = \frac{2}{T}\frac{\left(e^{sT} - 1\right)}{\left(e^{sT} + 1\right)} \tag{5.15}$$

iv. Mapeamento ST1:

$$D^*(s) = \frac{2}{T}\frac{\left(e^{sT} - 1\right)}{\left(e^{sT} + \xi\right)} \quad ;0 < \xi < 1 \tag{5.16}$$

v. Mapeamento ST2:

$$D^*(s) = \frac{2}{T}\frac{\left(e^{sT} - \xi_1\right)}{\left(e^{sT} + \xi_2\right)} \quad ;0 < \xi_1 < 1;\ 0 < \xi_2 < 1 \tag{5.17}$$

vi. Mapeamento Schneider:

$$D^*(s) = \frac{12}{T}\frac{\left(e^{2sT} - e^{sT}\right)}{\left(5e^{2sT} + 8e^{sT} - 1\right)} \tag{5.18}$$

c) Controle integral: (ganho k_i)

i. Mapeamento forward:

$$D^*(s) = \frac{T}{e^{sT} - 1} \tag{5.19}$$

ii. Mapeamento backward:

$$D^*(s) = \frac{Te^{sT}}{e^{sT} - 1} \tag{5.20}$$

iii. Mapeamento Tustin:

$$D^*(s) = \frac{T}{2}\frac{\left(e^{sT}+1\right)}{\left(e^{sT}-1\right)} \tag{5.21}$$

iv. Mapeamento ST1:

$$D^*(s) = \frac{T}{2}\frac{\left(e^{sT}+\xi\right)}{\left(e^{sT}-1\right)} \quad ;0 < \xi < 1 \tag{5.22}$$

v. Mapeamento ST2:

$$D^*(s) = \frac{T}{2}\frac{\left(e^{sT}+\xi_2\right)}{\left(e^{sT}-\xi_1\right)} \quad ;0 < \xi_1 < 1;\ 0 < \xi_2 < 1 \tag{5.23}$$

vi. Mapeamento Schneider:

$$D^*(s) = \frac{T}{12}\frac{\left(5e^{2sT}+8e^{sT}-1\right)}{\left(e^{2sT}-e^{sT}\right)} \tag{5.24}$$

d) Controle Proporcional+Derivativo (PD):

i. Mapeamento forward:

$$D^*(s) = k_p + k_d\frac{1}{T}\left(e^{sT}-1\right) \tag{5.25}$$

ii. Mapeamento backward:

$$D^*(s) = k_p + k_d\frac{\left(e^{sT}-1\right)}{Te^{sT}} \tag{5.26}$$

iii. Mapeamento Tustin:

$$D^*(s) = k_p + k_d\frac{2}{T}\frac{\left(e^{sT}-1\right)}{\left(e^{sT}+1\right)} \tag{5.27}$$

iv. Mapeamento ST1:

$$D^*(s) = k_p + k_d \frac{2}{T} \frac{\left(e^{sT} - 1\right)}{\left(e^{sT} + \xi\right)} \quad ;0 < \xi < 1 \tag{5.28}$$

v. Mapeamento ST2:

$$D^*(s) = k_p + k_d \frac{2}{T} \frac{\left(e^{sT} - \xi_1\right)}{\left(e^{sT} + \xi_2\right)} \quad ;0 < \xi_1 < 1;\ 0 < \xi_2 < 1 \tag{5.29}$$

vi. Mapeamento Schneider:

$$D^*(s) = k_p + k_d \frac{12}{T} \frac{\left(e^{2sT} - e^{sT}\right)}{\left(5e^{2sT} + 8e^{sT} - 1\right)} \tag{5.30}$$

e) Controle Proporcional+Integral+Derivativo (PID):

i. Mapeamento forward:

$$D^*(s) = k_p + k_i \frac{T}{e^{sT} - 1} + k_d \frac{1}{T}\left(e^{sT} - 1\right) \tag{5.31}$$

ii. Mapeamento backward:

$$D^*(s) = k_p + k_i \frac{Te^{sT}}{e^{sT} - 1} + k_d \frac{\left(e^{sT} - 1\right)}{Te^{sT}} \tag{5.32}$$

iii. Mapeamento Tustin:

$$D^*(s) = k_p + k_i \frac{T}{2} \frac{\left(e^{sT} + 1\right)}{\left(e^{sT} - 1\right)} + k_d \frac{2}{T} \frac{\left(e^{sT} - 1\right)}{\left(e^{sT} + 1\right)} \tag{5.33}$$

iv. Mapeamento ST1:

$$D^*(s) = k_p + k_i \frac{T}{2} \frac{\left(e^{sT} + \xi\right)}{\left(e^{sT} - 1\right)} + k_d \frac{2}{T} \frac{\left(e^{sT} - 1\right)}{\left(e^{sT} + \xi\right)} \quad ;0 < \xi < 1 \tag{5.34}$$

v. Mapeamento ST2:

$$D^*(s) = k_p + k_i \frac{T}{2}\frac{\left(e^{sT}+\xi_2\right)}{\left(e^{sT}-\xi_1\right)} + k_d \frac{2}{T}\frac{\left(e^{sT}-\xi_1\right)}{\left(e^{sT}+\xi_2\right)} \quad ;0 < \xi_1 < 1;\ 0 < \xi_2 < 1 \qquad (5.35)$$

vi. Mapeamento Schneider:

$$D^*(s) = k_p + k_i \frac{T}{12}\frac{\left(5e^{2sT}+8e^{sT}-1\right)}{\left(e^{2sT}-e^{sT}\right)} + k_d \frac{12}{T}\frac{\left(e^{2sT}-e^{sT}\right)}{\left(5e^{2sT}+8e^{sT}-1\right)} \qquad (5.36)$$

f) Controle usando rede de avanço de fase (rede “lead”):

A idéia básica de se usar uma rede de avanço de fase é fazer ascender o diagrama de fase aumentando assim a margem de fase. A expressão analógica para uma rede de avanço de fase tipo derivador é, segundo **Dorf (1990)**,

$$D(s) = k\frac{s+z}{s+p} \qquad (5.37)$$

onde $|p| > |z|$.

i. Mapeamento forward:

$$s \approx \frac{1}{T}\left(e^{sT}-1\right) \qquad (5.38)$$

$$D^*(s) = k.\frac{\frac{1}{T}\left(e^{sT}-1\right)+z}{\frac{1}{T}\left(e^{sT}-1\right)+p} \qquad (5.39)$$

$$D^*(s) = k.\frac{e^{sT}+Tz-1}{e^{sT}+Tp-1} \qquad ,|p| > |z| \qquad (5.40)$$

ii. Mapeamento backward:

$$D^*(s) = k.\frac{(1+Tz)e^{sT} - 1}{(1+Tp)e^{sT} - 1} \qquad ,|p| > |z| \qquad \text{(A.41)}$$

iii. Mapeamento Tustin:

$$D^*(s) = k.\frac{(2+Tz)e^{sT} + Tz - 2}{(2+Tp)e^{sT} + Tp - 2} \qquad ,|p| > |z| \qquad \text{(5.42)}$$

iv. Mapeamento ST1:

$$D^*(s) = k.\frac{(2+Tz)e^{sT} + Tz\xi - 2}{(2+Tp)e^{sT} + Tp\xi - 2} \qquad ,|p| > |z|,\ 0 < \xi < 1 \qquad \text{(5.43)}$$

v. Mapeamento ST2:

$$D^*(s) = k.\frac{(2+Tz)e^{sT} + Tz\xi_2 - 2\xi_1}{(2+Tp)e^{sT} + Tp\xi\xi_2 - 2\xi_1} \qquad ,|p| > |z|,\ 0<\xi_1<1,\ |,\ 0<\xi_2<1 \qquad \text{(5.44)}$$

vi. Mapeamento Schneider:

$$D^*(s) = k.\frac{\left(1+\frac{5zT}{12}\right)e^{2sT} + \left(\frac{2zT}{3}\right)e^{sT} - \frac{zT}{12}}{\left(1+\frac{5pT}{12}\right)e^{2sT} + \left(\frac{2pT}{3}\right)e^{sT} - \frac{pT}{12}} \qquad ,|p| > |z| \qquad \text{(5.45)}$$

g) Controle usando rede de atraso de fase (rede “lag”):

Corresponde a um conjunto de equações estruturalmente iguais ao caso anterior só que para |z| > |p|.

h) Controle usando rede de avanço-atraso (rede “lead-lag”):

Corresponde a associação dos casos f e g acima.

5.2. RESULTADOS NUMÉRICOS OBTIDOS COM PLANTAS DE PRIMEIRA E SEGUNDA ORDENS UTILIZANDO CONTROLADORES E MAPEAMENTOS DIVERSOS

As simulações realizadas nessa seção mostram casos de plantas de primeira e segunda ordem controladas por diversos tipos de controle discreto onde são plotados os diagramas de Bode das respectivas funções de transferência em malha aberta H*(s). A partir desses diagramas pode-se verificar que a dependência do período de amostragem modifica os diagramas de tal sorte que, dependendo do período de amostragem, o sistema pode vir de uma situação de estabilidade assintótica para estabilidade marginal ou então tornar-se instável e vice-versa. Ainda por cima tal situação pode ser verificada no diagrama de Bode de forma periódica.

Abaixo está uma figura representativa do diagrama de Bode de um sistema hipotético onde, mediante margens de ganho e fase, podemos lembrar que quando a tais margens forem negativas ($\mu_g = \frac{1}{|H^*|}$ ou em dB $\mu_g = -20\log|H^*|$ e μ_f = 180^o - |fase(H*)|) o sistema será instável e, caso contrário, será estável.

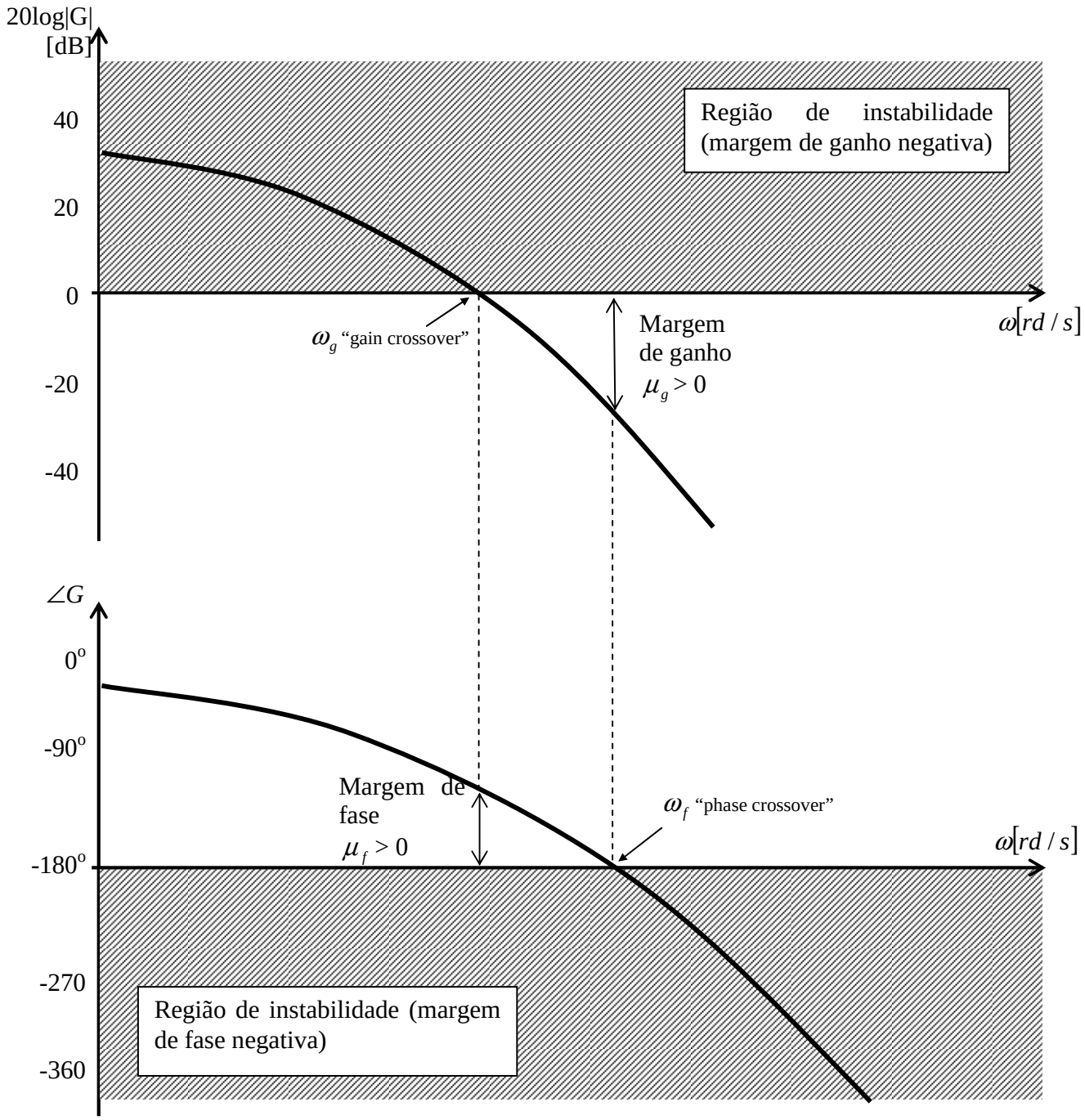


Figura 5.2 - margens de ganho e fase de um diagrama de Bode.

O que ocorre quando o período de amostragem T varia são situações similares ao mostrado na próxima figura em que podemos ter um sistema originalmente estável para um determinado T que perde a sua estabilidade, tornando-se instável, para outro valor de T, ou vice-versa.

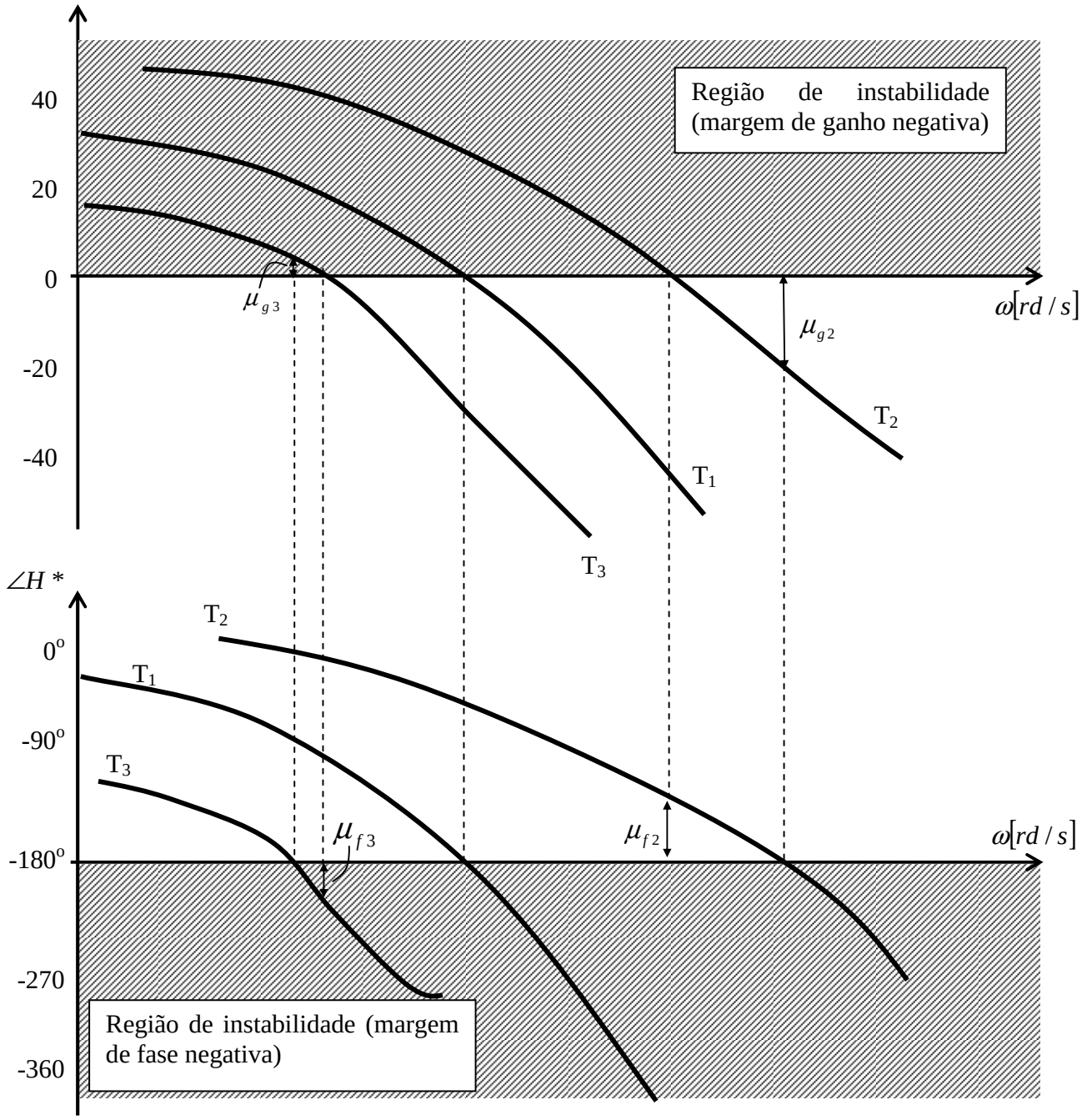


Figura 5.3 - Margens de ganho e fase de um diagrama de Bode em função do período de amostragem.

Na figura T3 > T1 >T2 e as curvas correspondentes a T1 se referem a um sistema amostrado marginalmente estável, as curvas correspondentes a T2 correspondem

a um sistema assintoticamente estável e as curvas correspondentes a T3 correspondem a um sistema instável.

5.2.1. Diagrama de Bode e gráfico fasorial para o caso da planta de primeira ordem e controle integral mapeado pelo método forward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1 segundo e ω= 0,723 rad/seg, ou seja ωT= 0,723 radianos ou 41,42^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

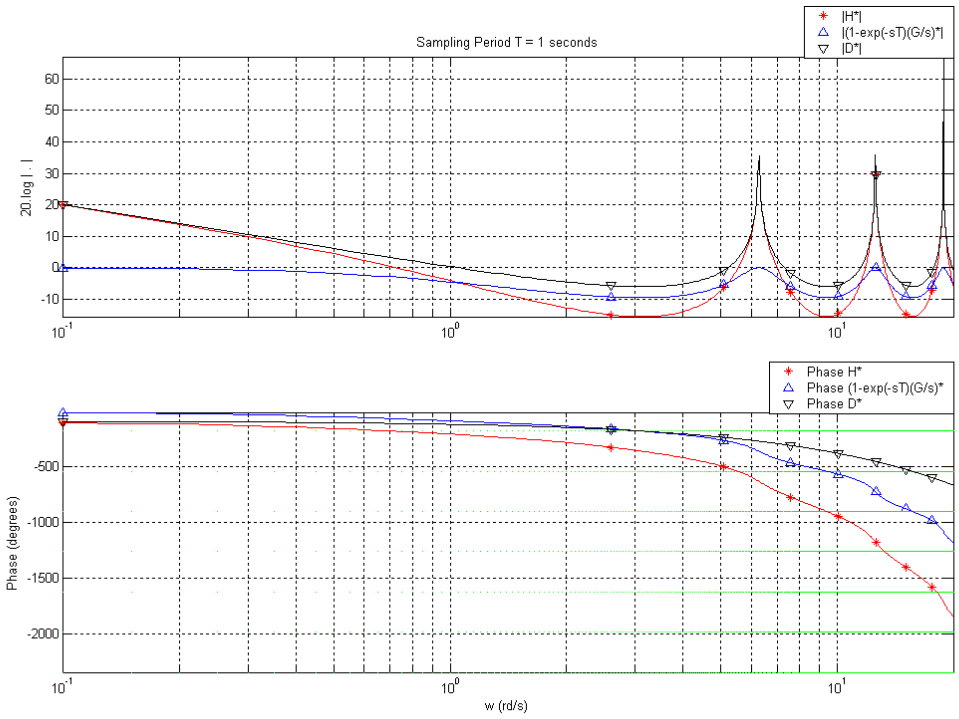


Figura 5.4 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$. As linhas tracejadas horizontais (cor verde) indicam múltiplos ímpares de -180^o.

Na figura acima pode-se perceber nitidamente a periodicidade existente no gráfico dos módulos provocado pela periodicidade da função exponencial e^{sT}.

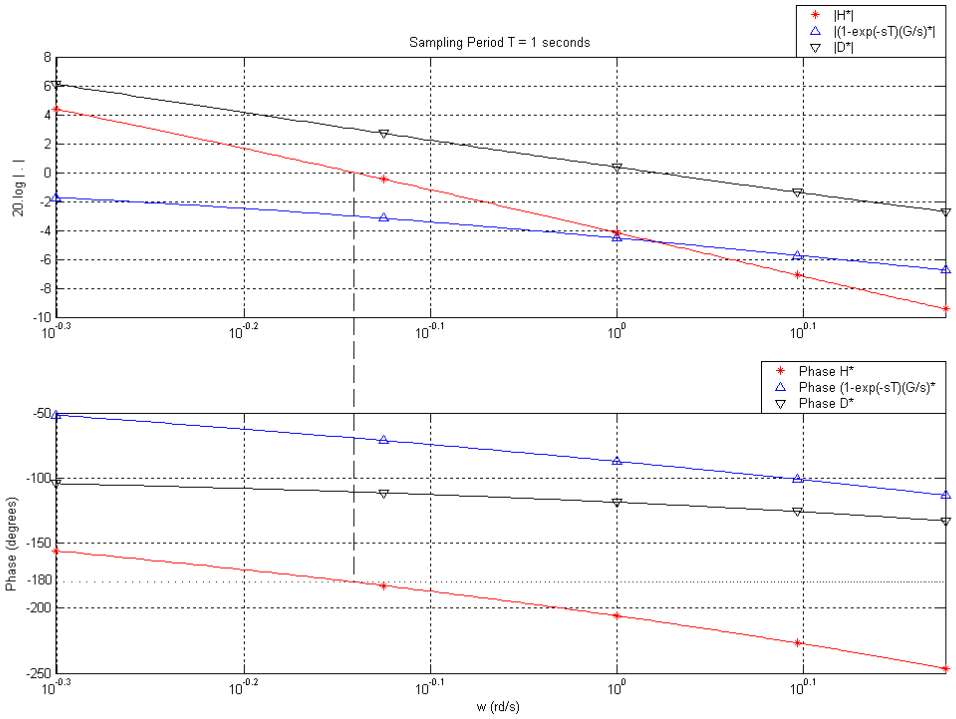


Figura 5.5 - Detalhe do diagrama de Bode para |H*(s)|=1 e fase = -180^{o}, T=1 segundo.

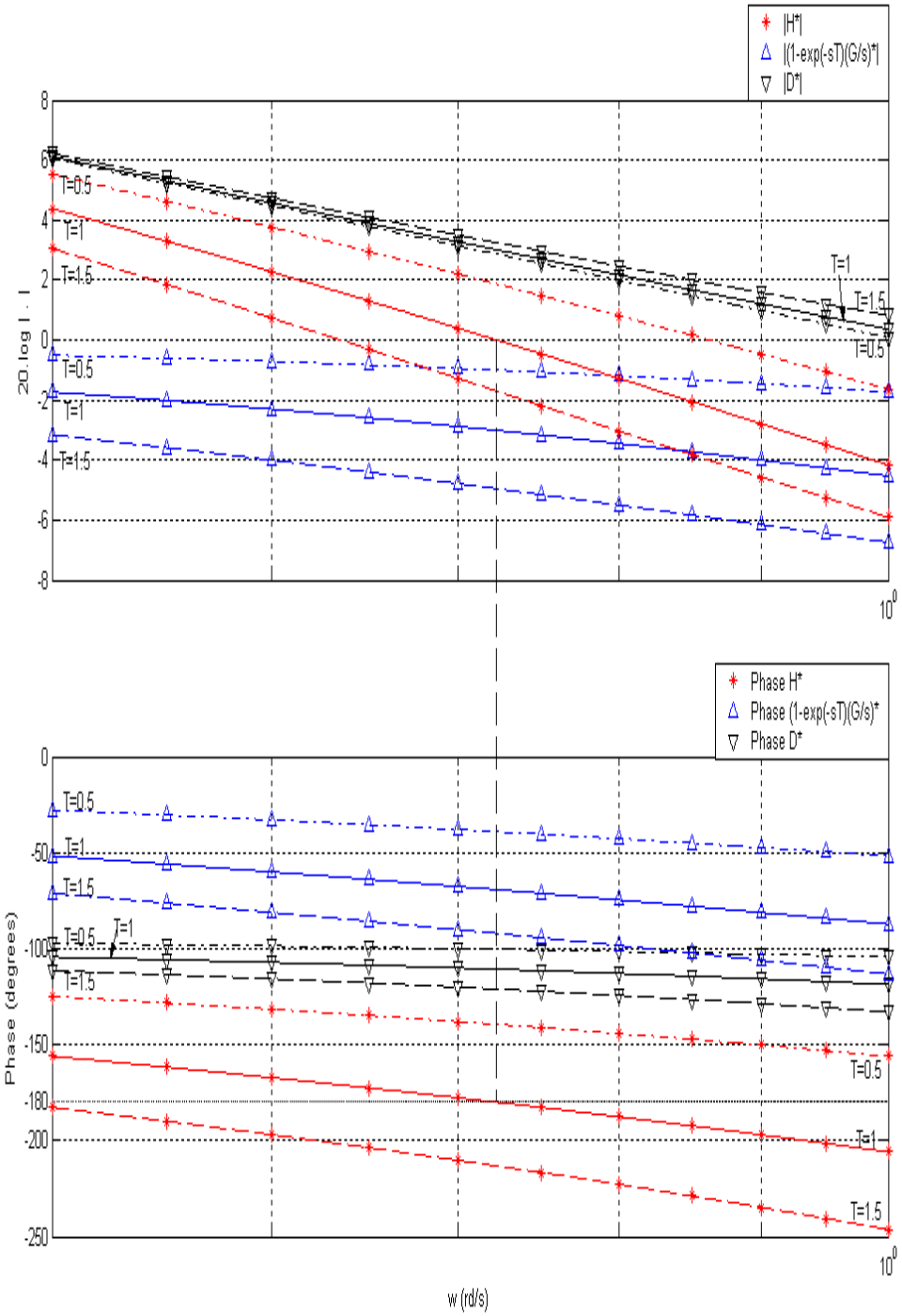


Figura 5.6 - Detalhe dos diagramas de Bode para T=0,5 segundo (assintoticamente estável); 1 segundo (marginalmente estável) e 1,5 segundo (instável); faixa $0.5 \leq \omega \leq 1$.

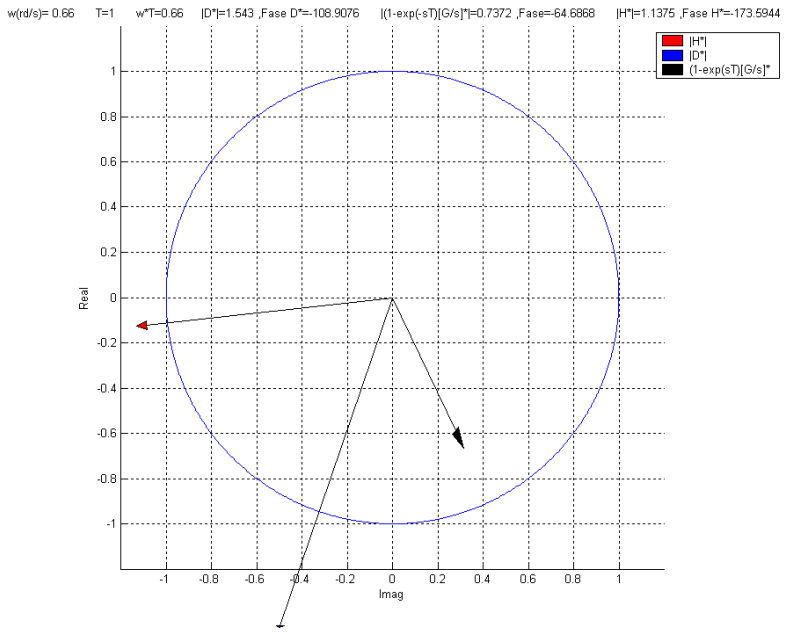


Figura 5.7 - **D**iagrama fasorial para T=1 segundo.

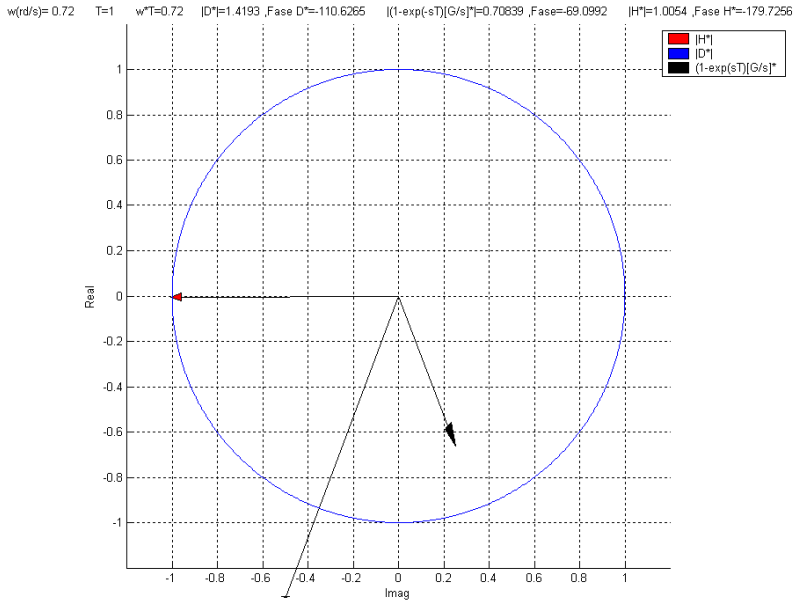


Figura 5.8 - Diagrama fasorial para T=1 segundo apresentando uma situação próxima da instabilidade onde $|H*|$ = 1 e a fase de H* é -180^{o}.

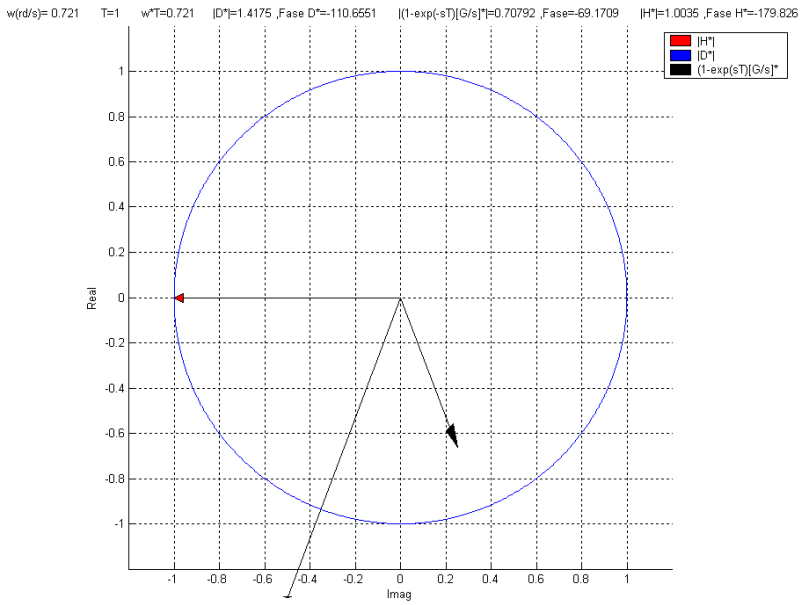


Figura 5.9 - Diagrama fasorial para T=1 segundo.

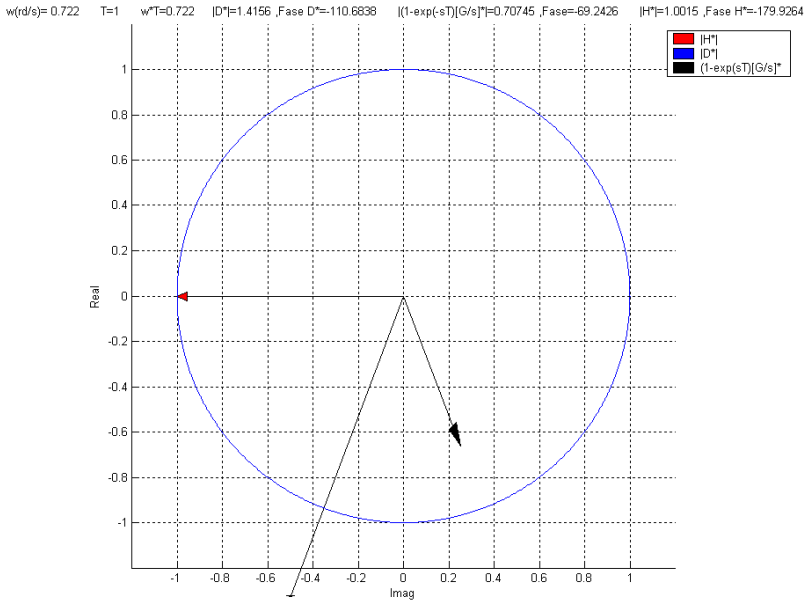


Figura 5.10 - Diagrama fasorial para T=1 segundo.

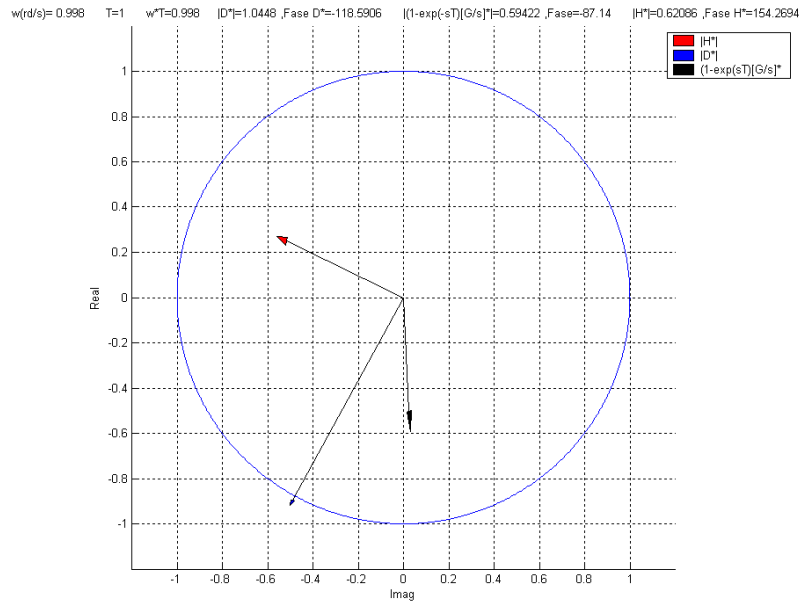


Figura 5.11 - Diagrama fasorial para T=1 segundo.

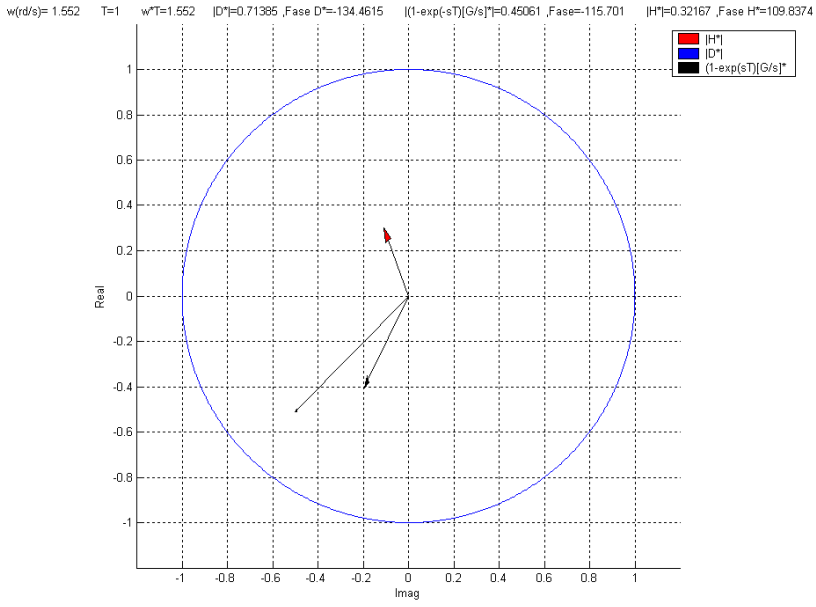


Figura 5.12 - Diagrama fasorial para T=1 segundo.

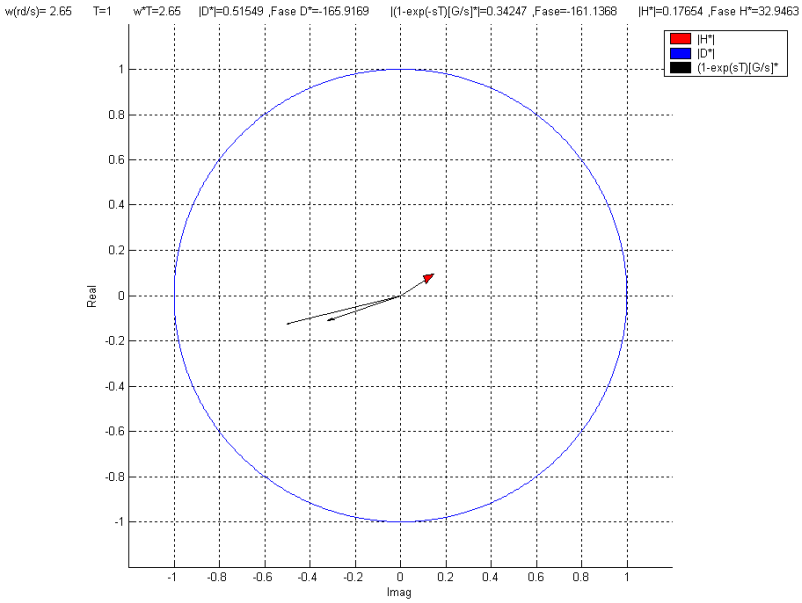


Figura 5.13 - Diagrama fasorial para T=1 segundo.

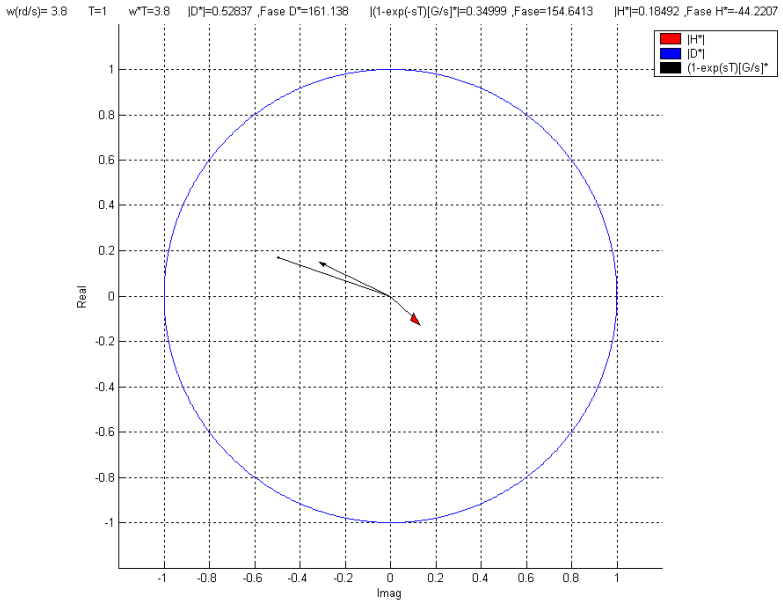


Figura 5.14 - Diagrama fasorial para T=1 segundo.

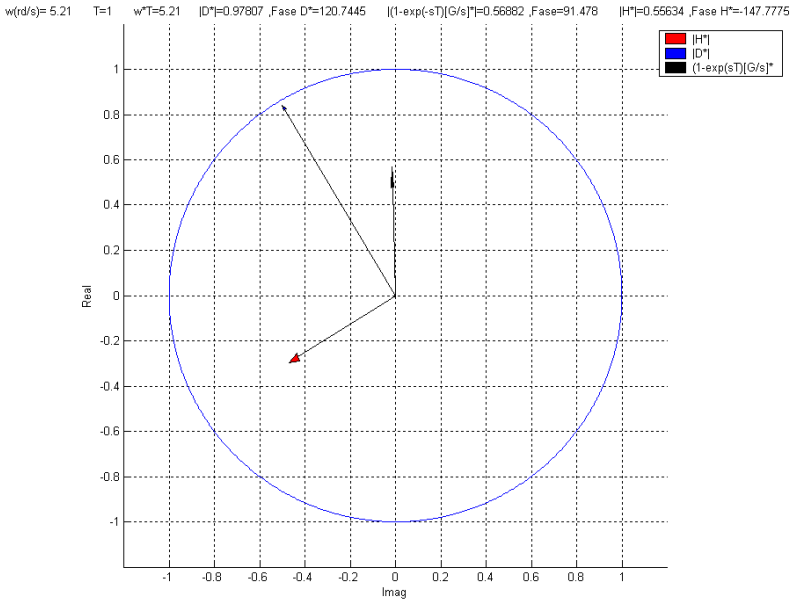


Figura 5.15 - Diagrama fasorial para T=1 segundo.

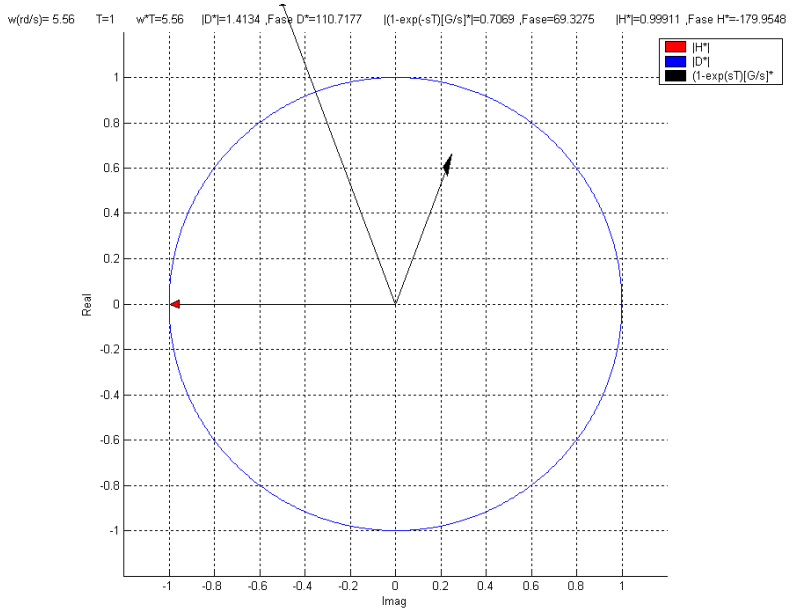


Figura 5.16 - Diagrama fasorial para T=1 segundo.

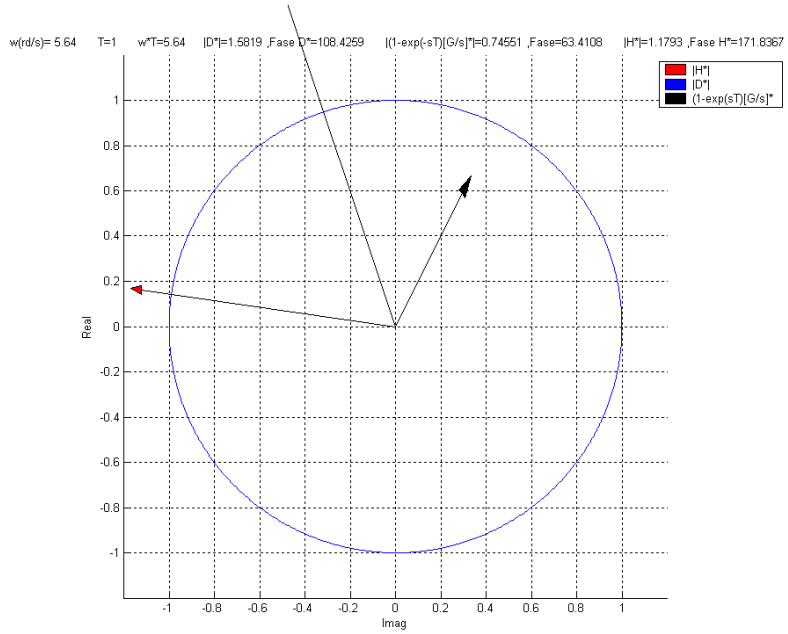


Figura 5.17 - Diagrama fasorial para T=1 segundo.

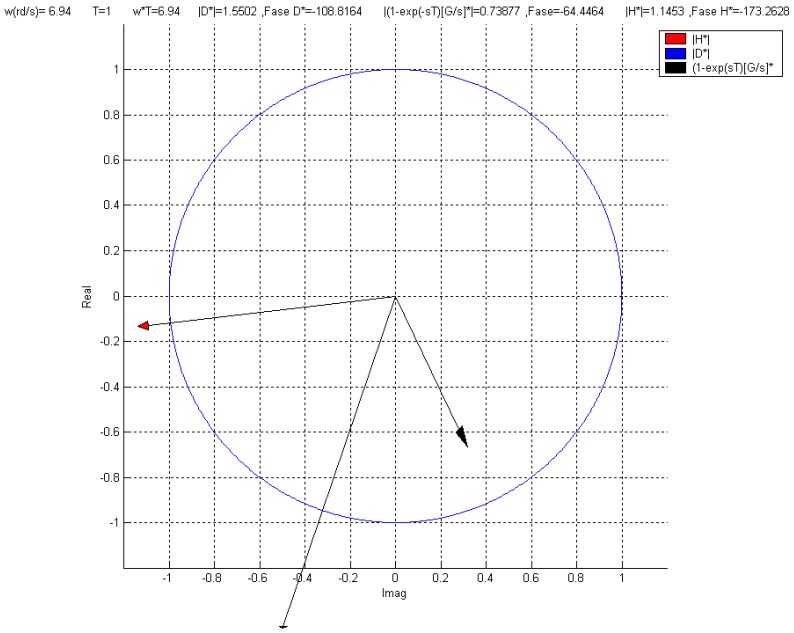


Figura 5.18 - Diagrama fasorial para T=1 segundo.

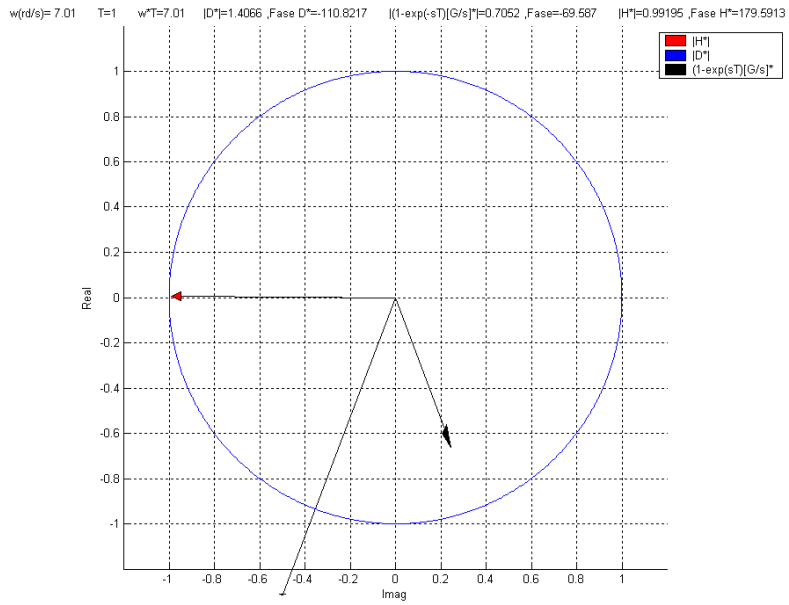


Figura 5.19 - Diagrama fasorial para T=1 segundo.

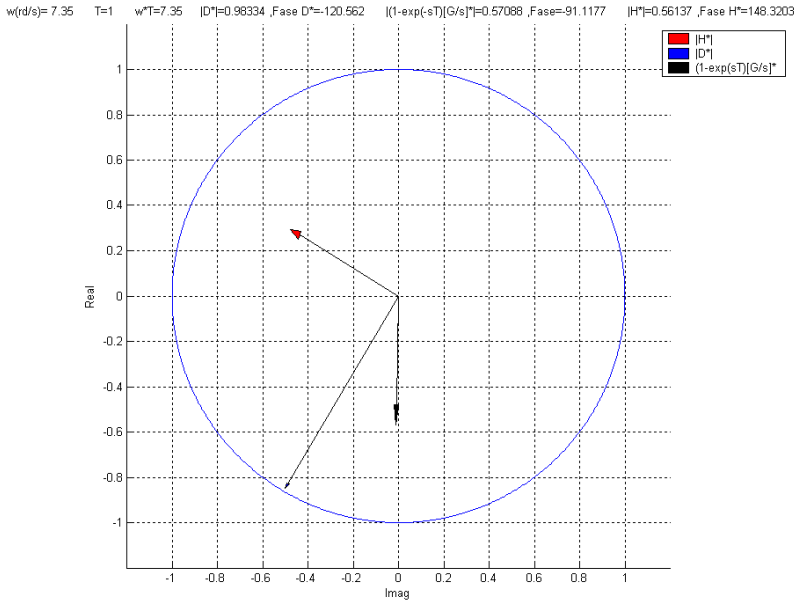


Figura 5.20 - Diagrama fasorial para T=1 segundo.

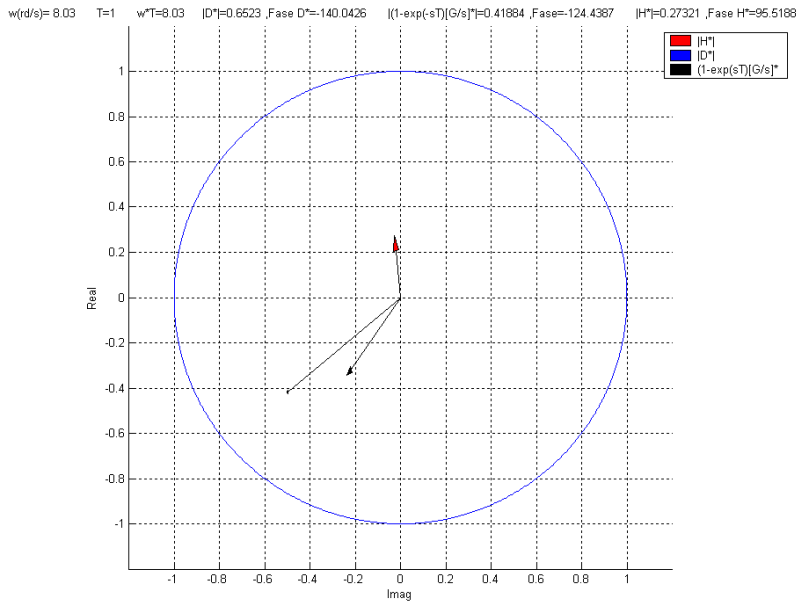


Figura 5.21 - Diagrama fasorial para T=1 segundo.

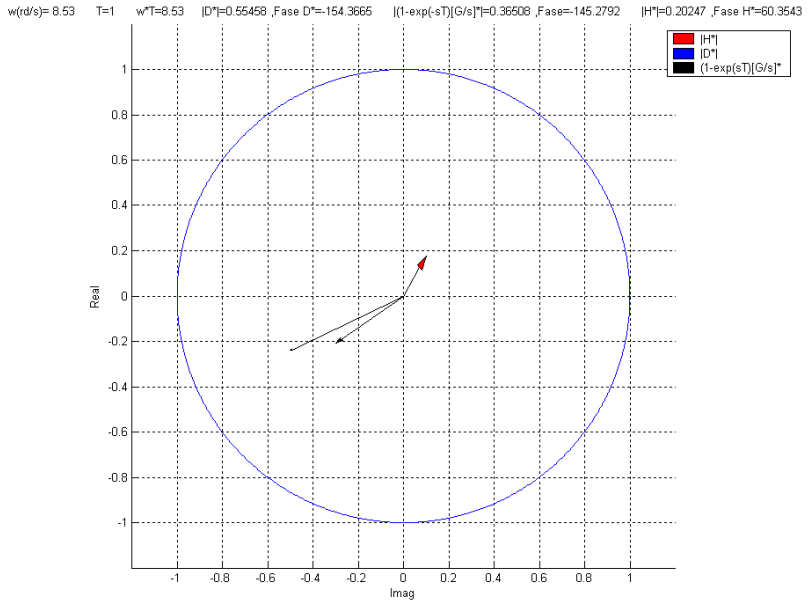


Figura 5.22 - Diagrama fasorial para T=1 segundo.

5.2.2. Diagrama de Bode e gráfico fasorial para o caso da planta de primeira ordem e controle integral mapeado pelo método Tustin

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2 segundos e ω= 2,618 rad/seg, ou seja ωT= 5,236 radianos ou 300^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

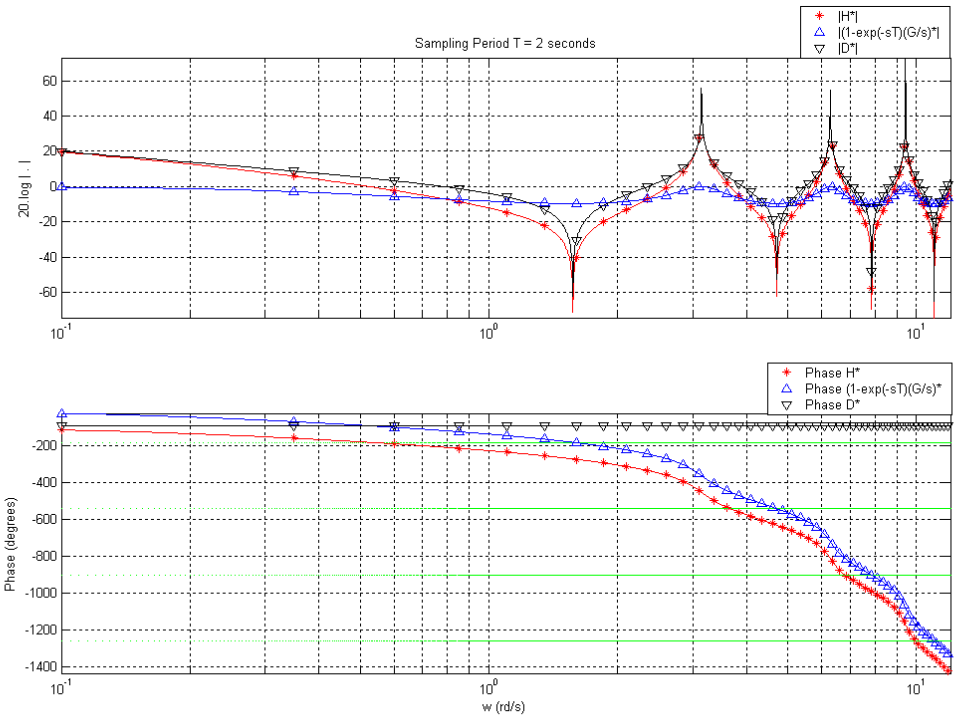


Figura 5.23 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$; a fase de D* fixou-se em 90^o.

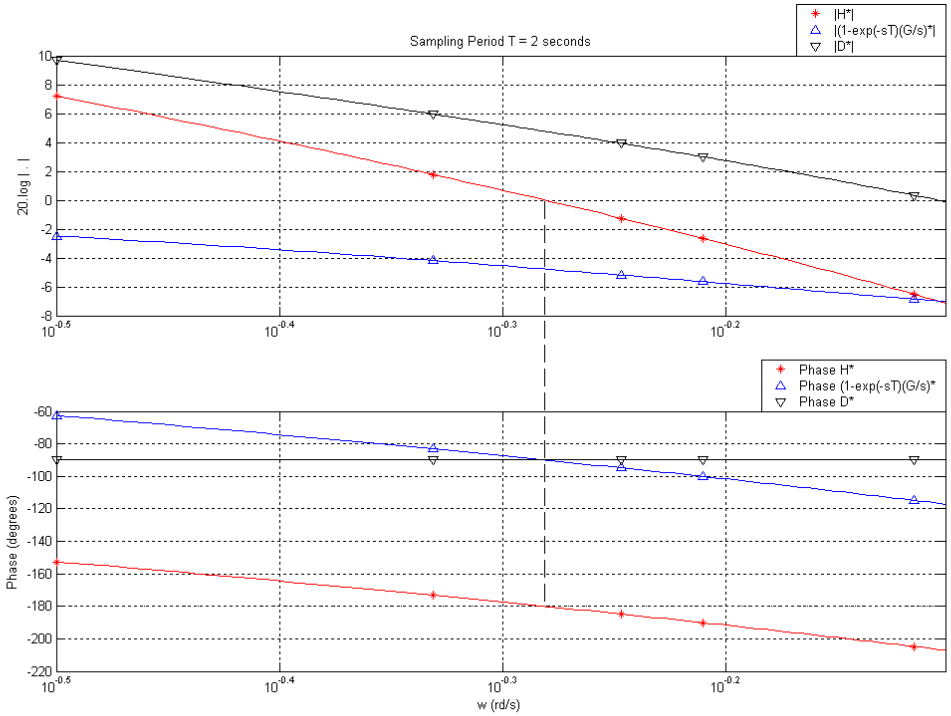


Figura 5.24 - Detalhe do diagrama de Bode para |H*(s)|=1 e fase = -180^{o}, T=2 segundos.

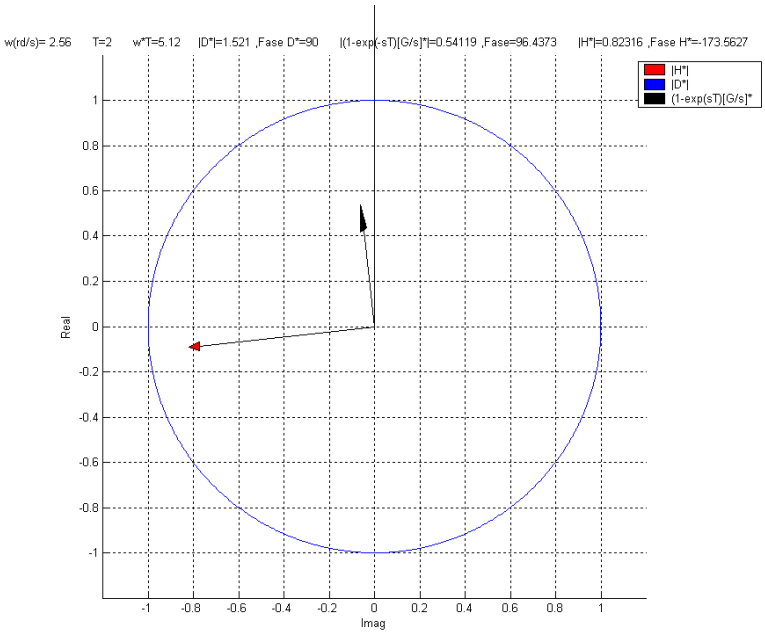


Figura 5.25 - Diagrama fasorial para T=2 segundos.

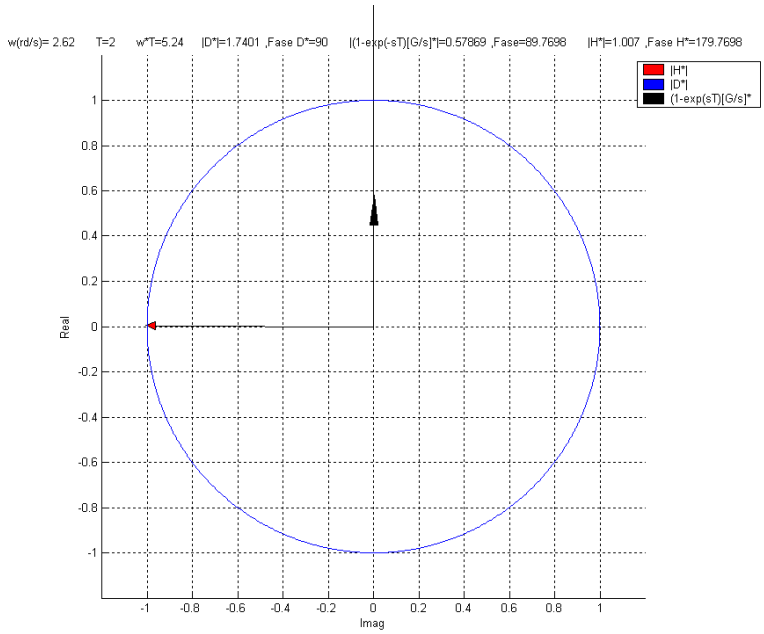


Figura 5.26 - Diagrama fasorial para T=2 segundos.

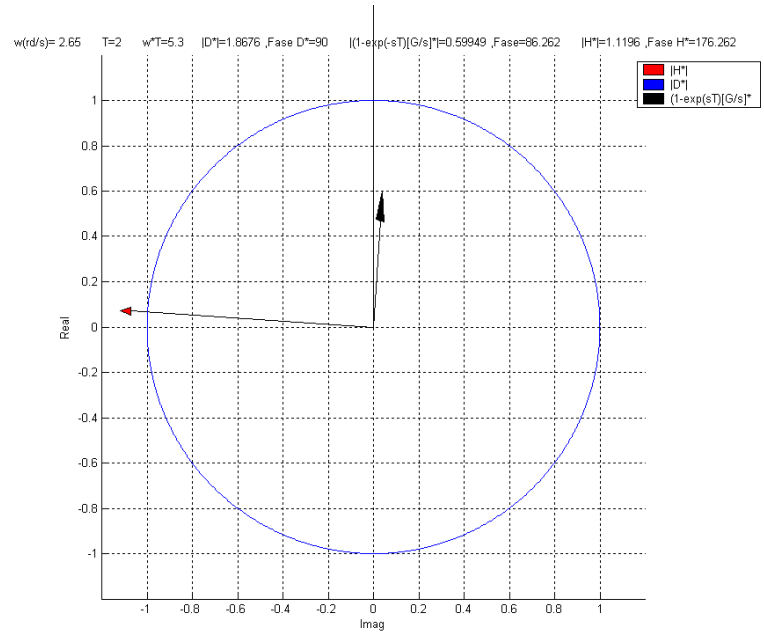


Figura 5.27 - Diagrama fasorial para T=2 segundos.

5.2.3. Diagrama de Bode e gráfico fasorial para o caso da planta de primeira ordem e controle usando rede de atraso de fase (rede "lag") mapeado pelo método forward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,22 segundos e ω= 2,303 rad/seg, ou seja ωT= 5,1126 radianos ou 292,9^{o}. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. O pólo da rede situa-se em -0,1 e o zero em -1 (caso analógico).

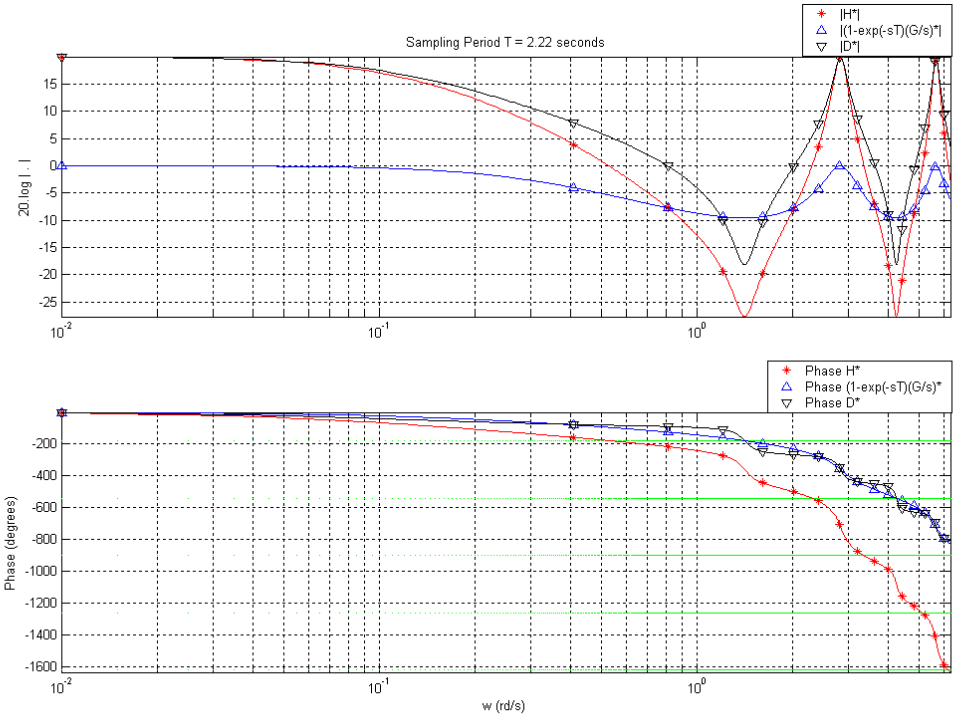


Figura 5.28 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$.

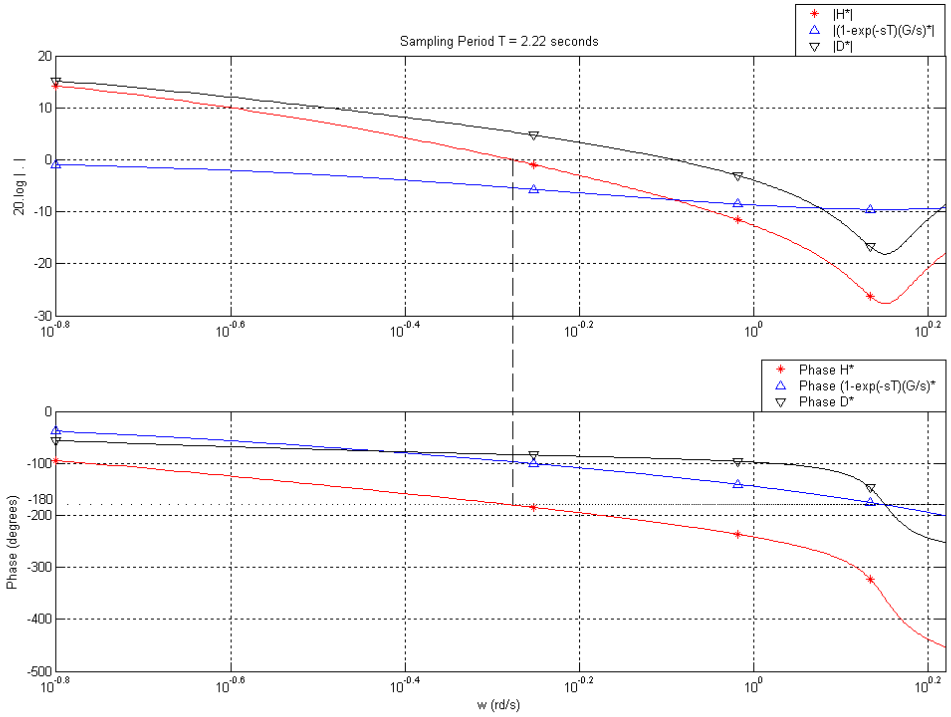


Figura 5.29 - Detalhe do diagrama de Bode para |H*(s)|=1 e fase = -180^{o}, T=2,22 segundos.

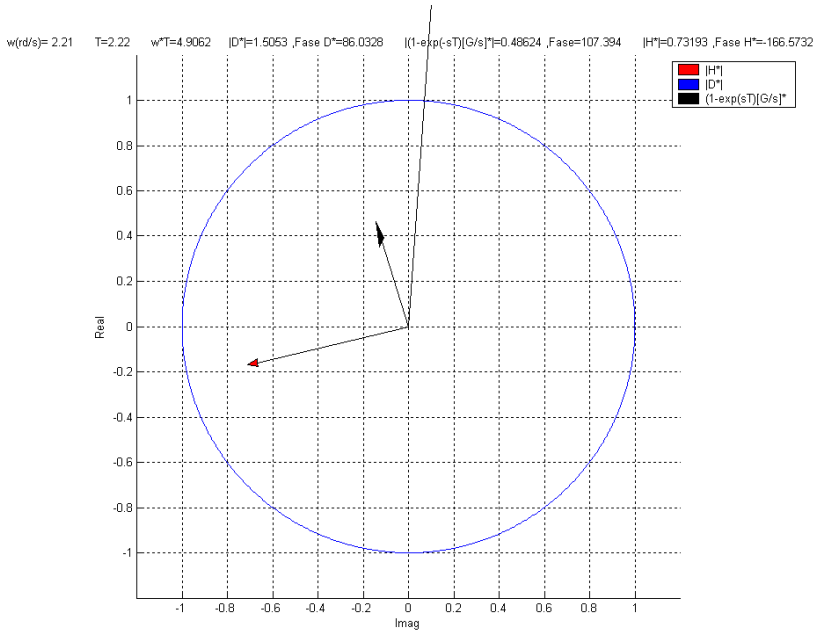


Figura 5.30 - Diagrama fasorial para T=2,22 segundos.

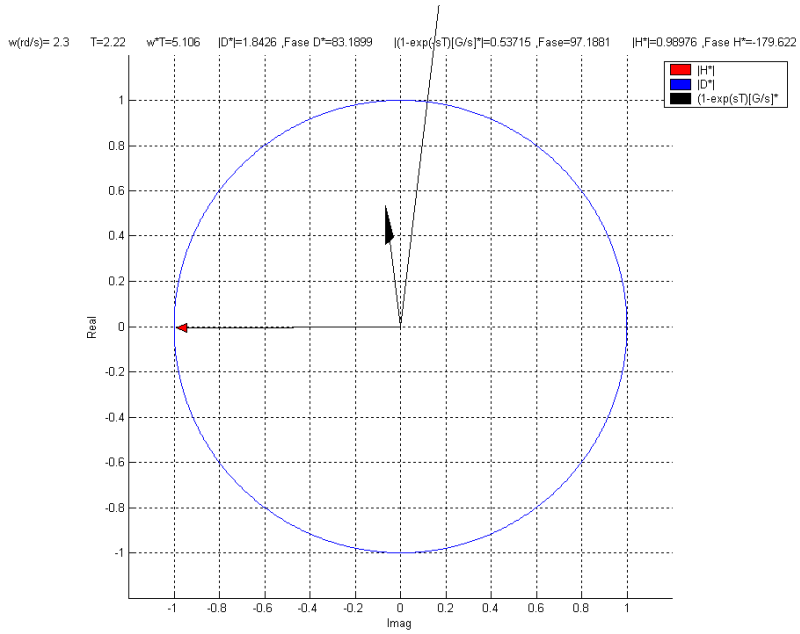


Figura 5.31 - Diagrama fasorial para T=2,22 segundos.

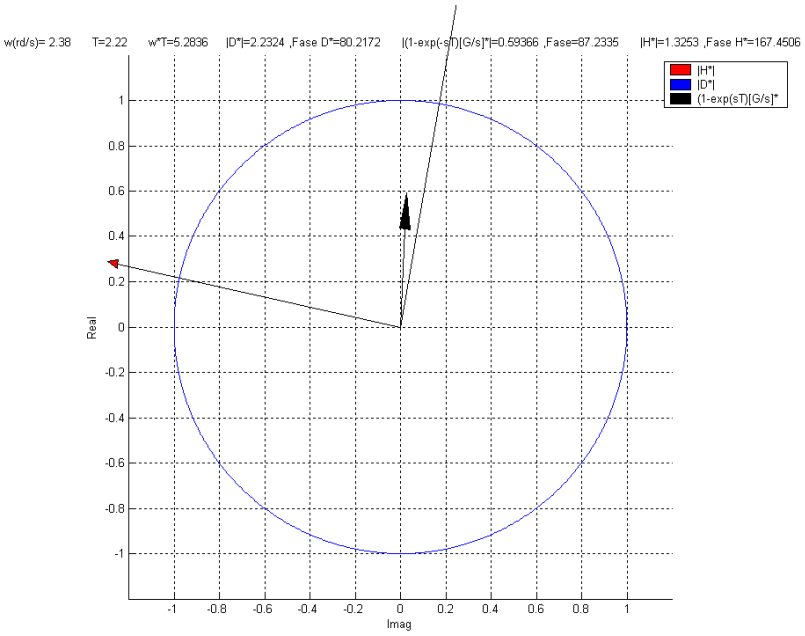


Figura 5.32 - Diagrama fasorial para T=2,22 segundos.

5.2.4. Diagrama de Bode para o caso da planta de primeira ordem e controle usando controle PD mapeado pelo método backward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1 segundo e $\omega = \pi$ rad/seg, ou seja $\omega T = \pi$ radianos ou 180°. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

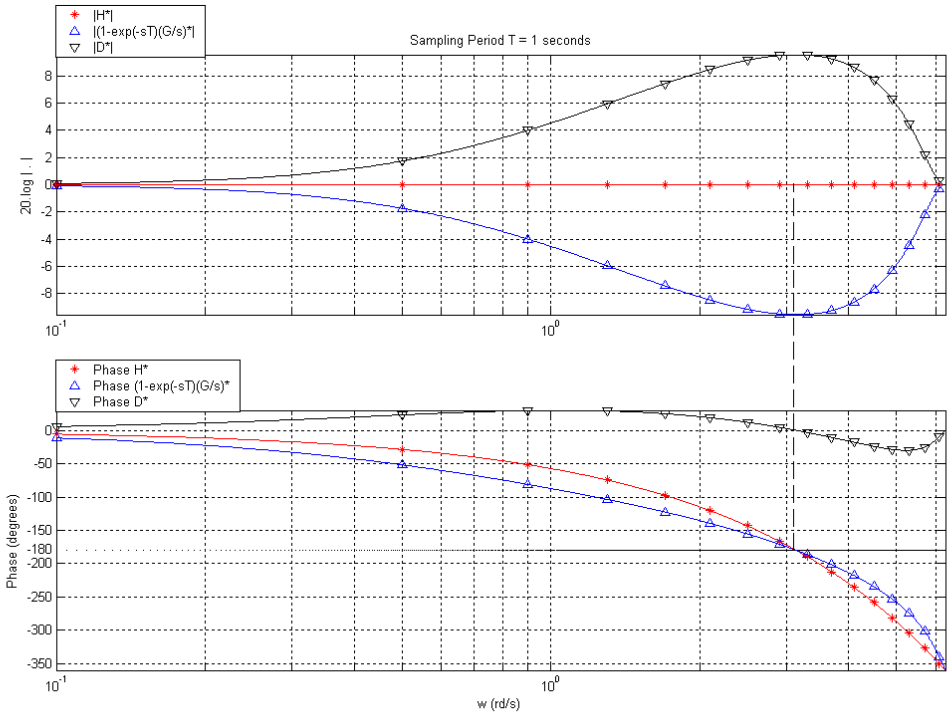


Figura 5.33 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180°, T=1 segundo.

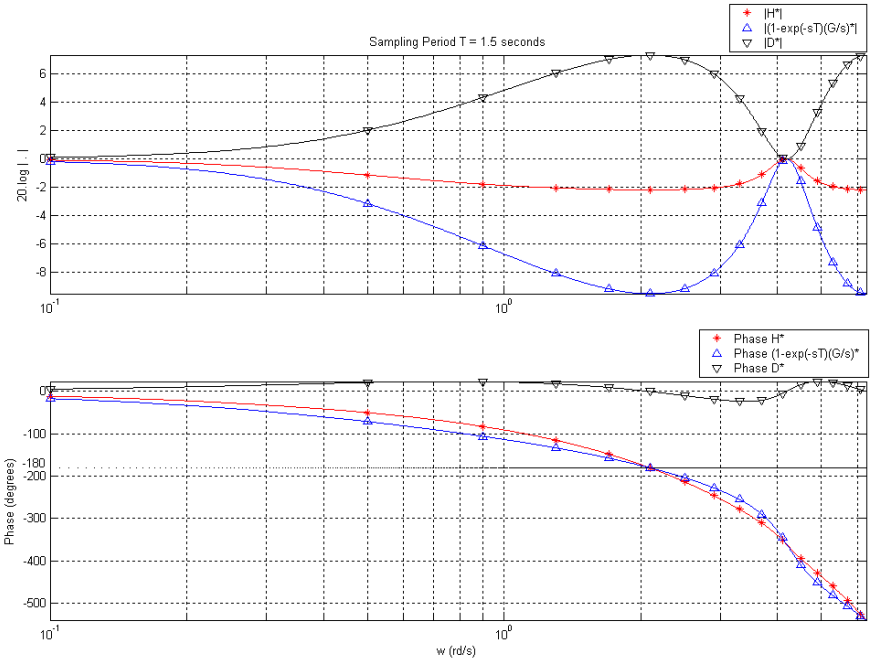


Figura 5.34 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$,com T=1,5 segundos.

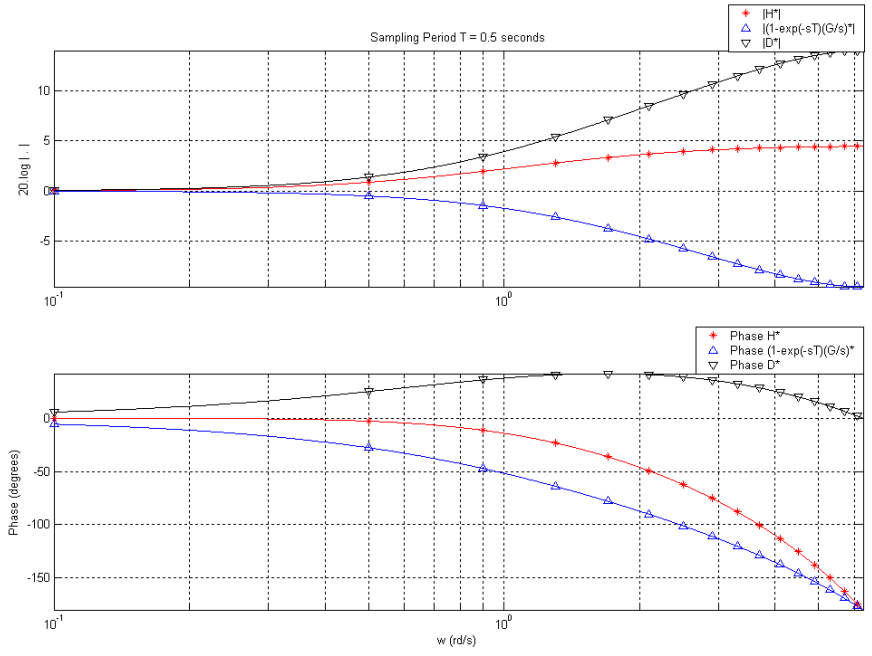


Figura 5.35 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$,com T=0,5 segundo.

5.2.5. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método backward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1,89 segundos e ω= 1,661 rad/seg, ou seja ωT= 3,08 radianos ou 177,02^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

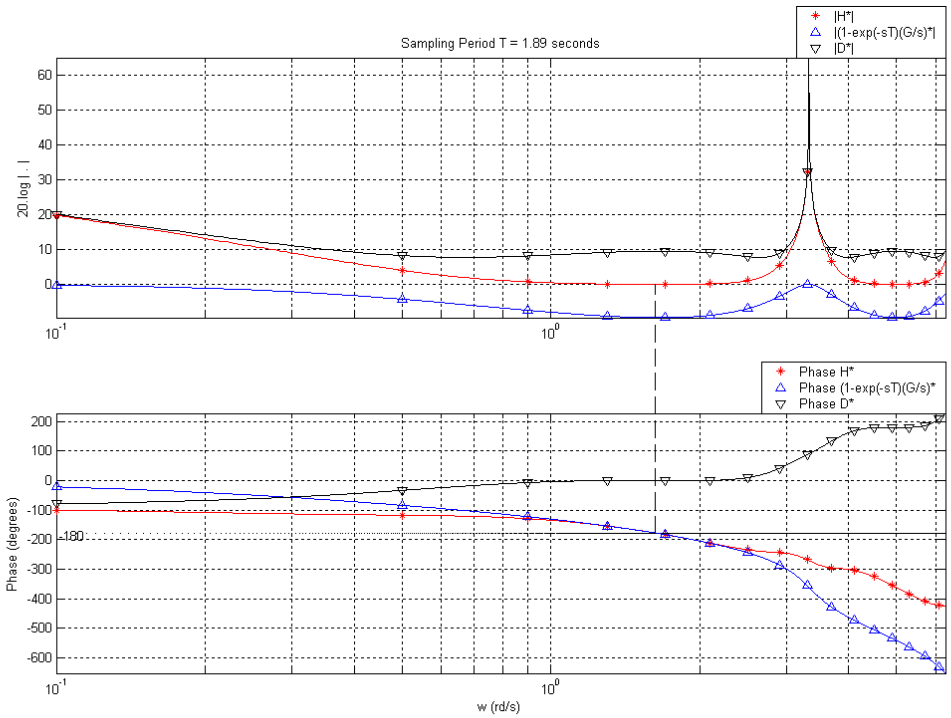


Figura 5.36 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=1,89 segundos.

5.2.6. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método forward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2 segundos e ω= 0,4624 rad/seg, ou seja ωT= 0.9248 radianos ou 52,98^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

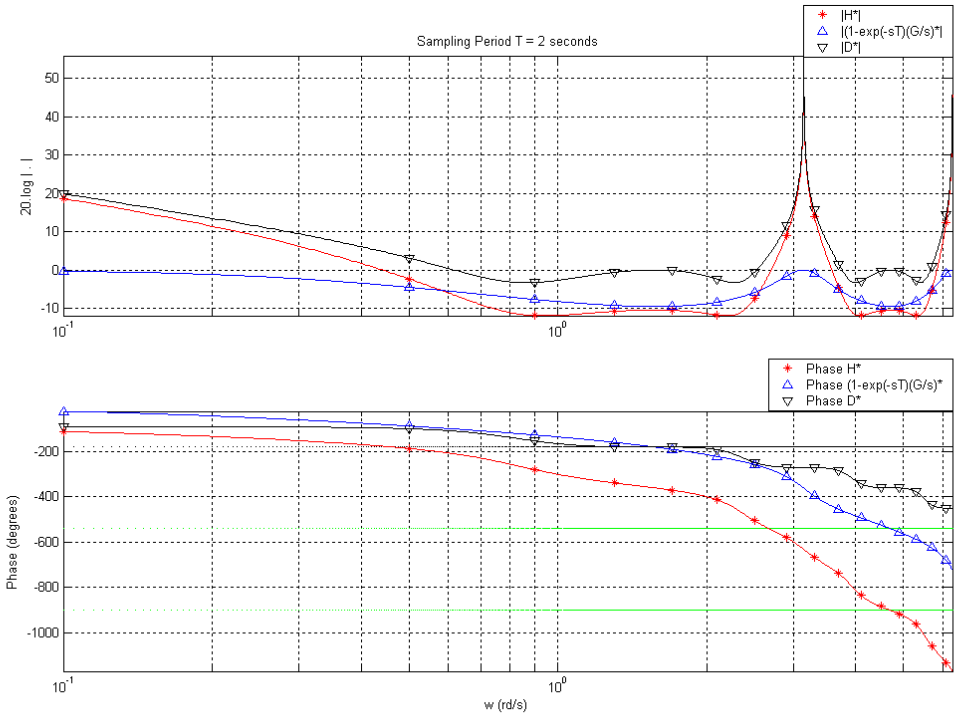


Figura 5.37 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2 segundos.

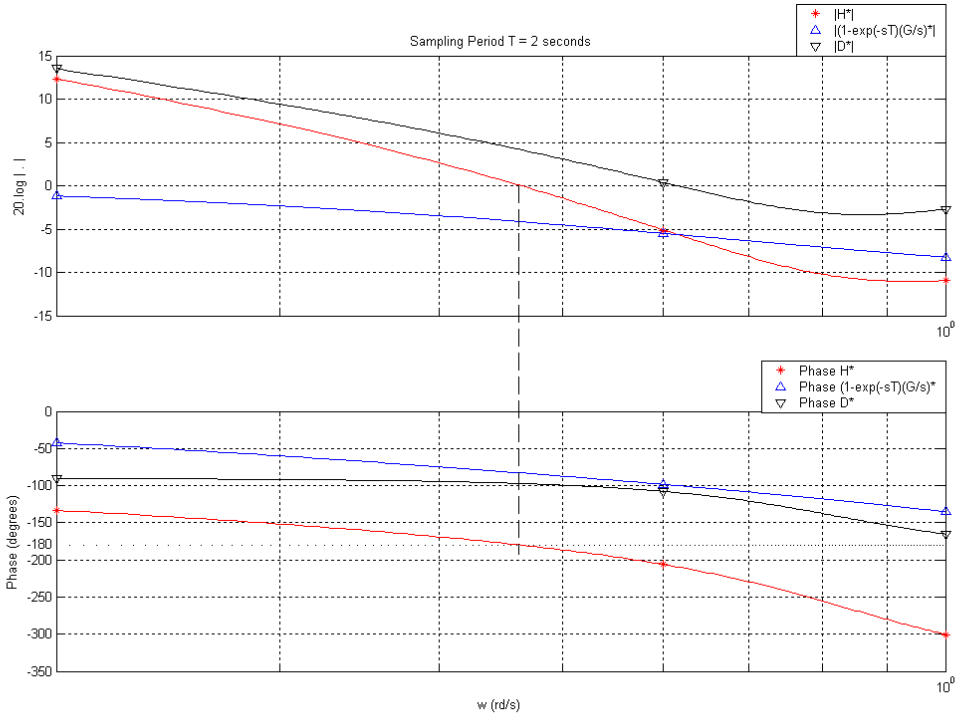


Figura 5.38 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2 segundos.

5.2.7. Diagrama de Bode para o caso da planta de segunda ordem e controle integral mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1,48 segundos e ω= 3,52 rad/seg, ou seja ωT= 5,2 radianos ou 298,48^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. Os parâmetros escolhidos foram ξ_1 = 0,2 e ξ_2 = 0,3.

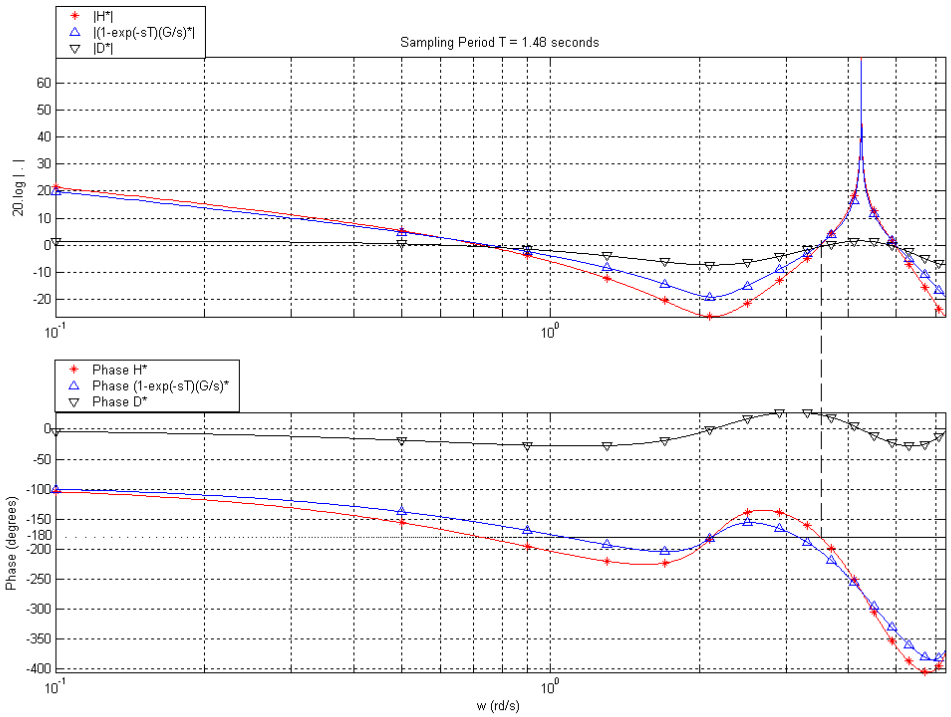


Figura 5.39 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=1,48 segundos.

5.2.8. Diagrama de Bode para o caso da planta de segunda ordem e controle usando rede de atraso de fase ("lag") mapeado pelo método backward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 0,35 segundos e ω = 1,029 rad/seg, ou seja ωT= 0,36 radianos ou 20,64^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. Pólo da rede (caso analógico) em -0,1 e zero em -1.

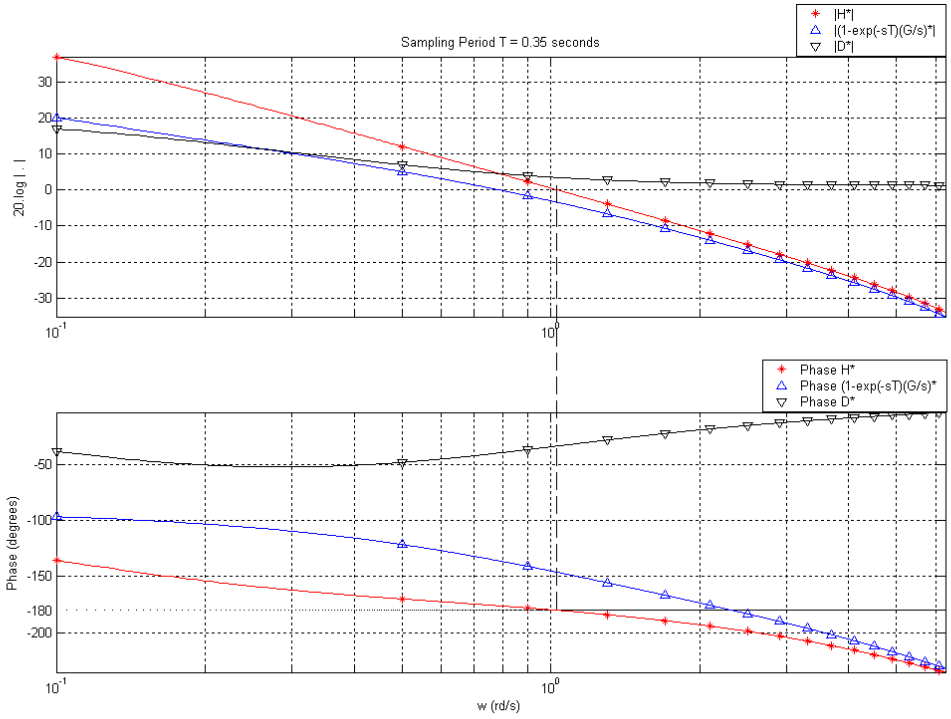


Figura 5.40 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=0,35 segundos.

5.2.9. Diagrama de Bode para o caso da planta de segunda ordem e controle usando rede de atraso de fase ("lag") mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 0,77 segundos e ω = 1,029 rad/seg, ou seja ωT= 0,36 radianos ou 20,64^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. Pólo da rede (caso analógico) em -0,1 e zero em -1. O parâmetro ξ = 0,3.

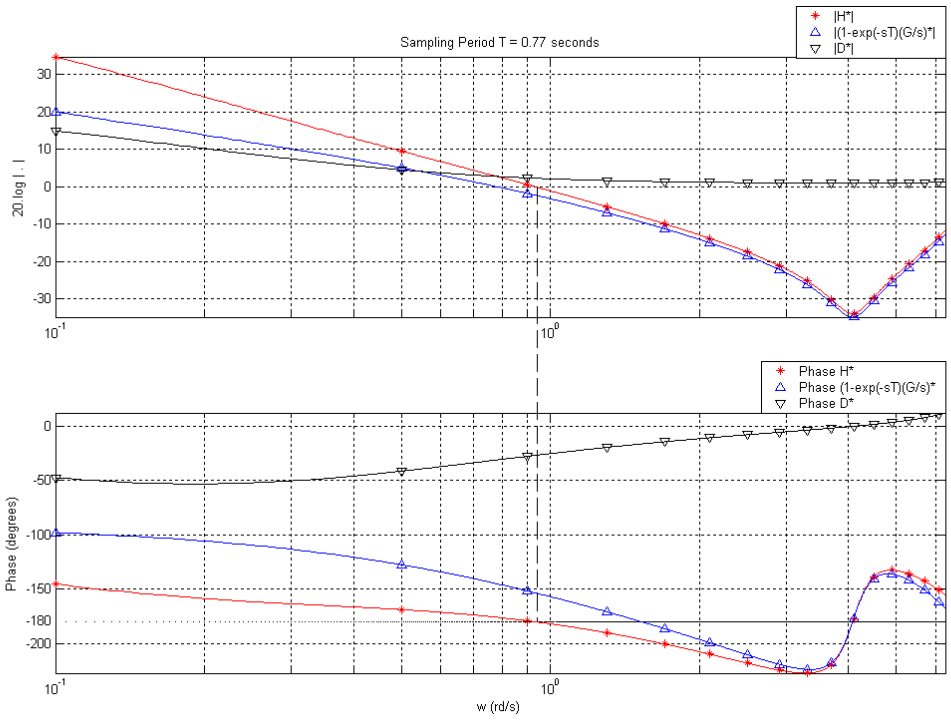


Figura 5.41 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=0,77 segundos.

5.2.10. Diagrama de Bode para o caso da planta de segunda ordem e controle usando rede de atraso de fase ("lag") mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1,23 segundos e ω= 1,024 rad/seg, ou seja ωT= 1,259 radianos ou 72,17^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. Pólo da rede (caso analógico) em -0,1 e zero em -1. Os parâmetros escolhidos foram ξ_1 = 0,2 e ξ_2 = 0,3.

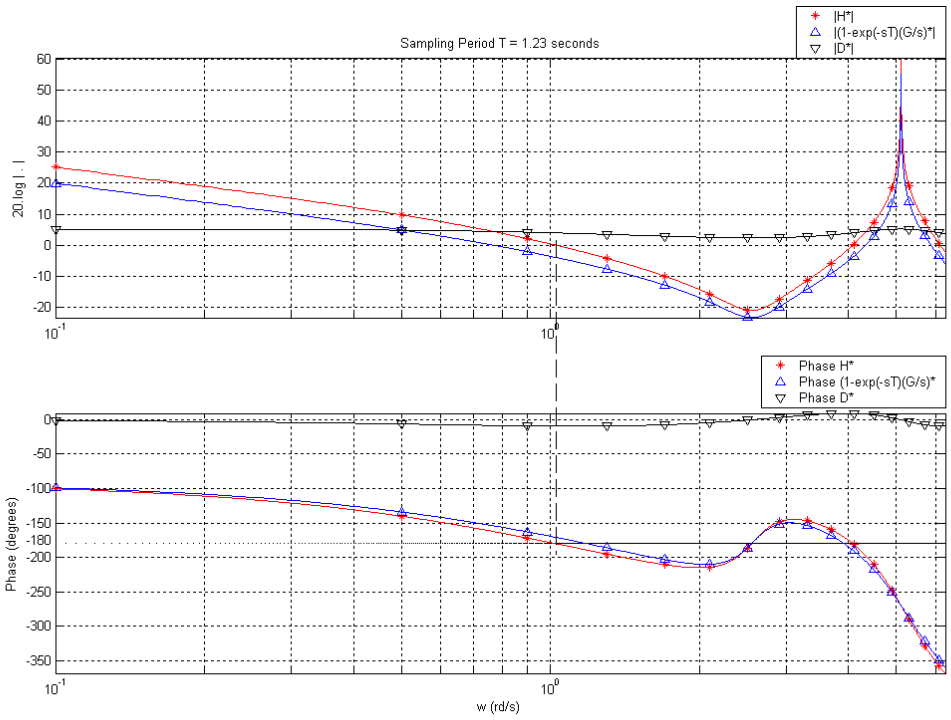


Figura 5.42 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=1,23 segundos.

5.2.11. Diagrama de Bode para o caso da planta de segunda ordem e controle PID mapeado pelo método de Schneider

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1,15 segundos e ω= 0,761 rad/seg, ou seja ωT= 0,875 radianos ou 50,14^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

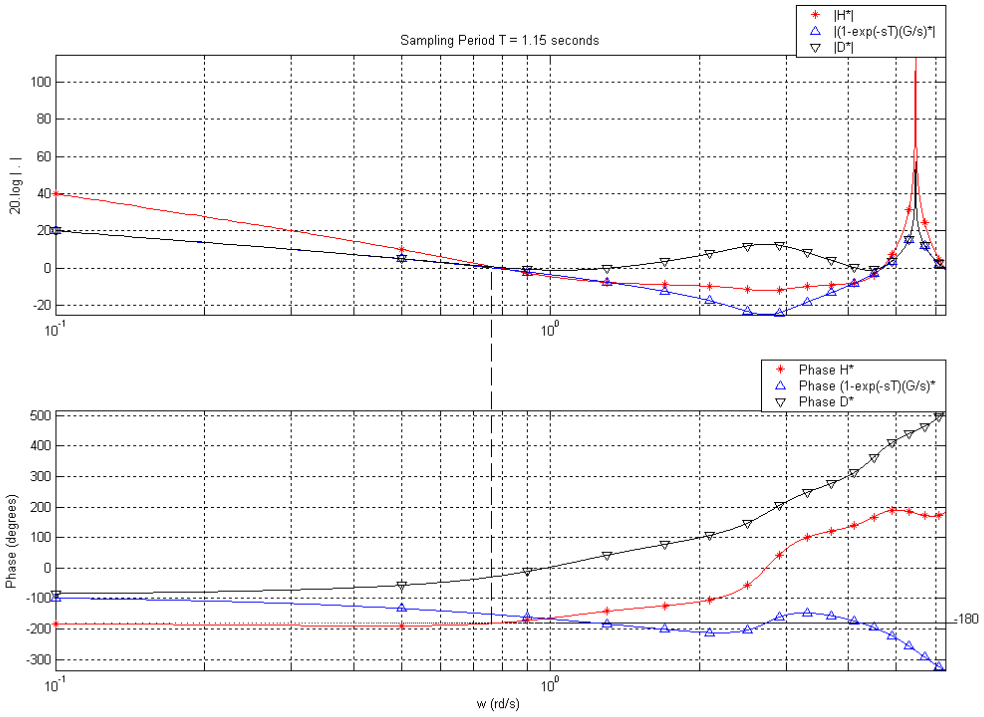


Figura 5.43 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=1,15 segundos.

5.2.12. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PID mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1,65 segundos e ω= 1,499 rad/seg, ou seja ωT= 2,47 radianos ou 141,7^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. O parâmetro ξ = 0,3.

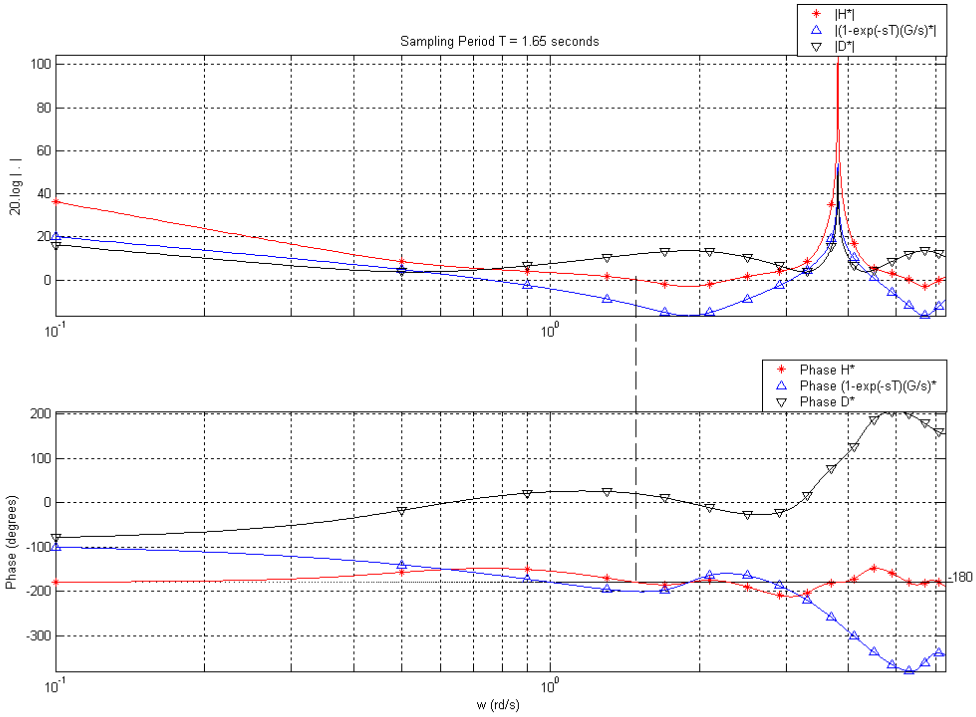


Figura 5.44 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=1,65 segundos.

5.2.13. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PID mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 1,07 segundos e ω= 1,522 rad/seg, ou seja ωT= 1,628 radianos ou 93,31^{o}. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. Os parâmetros escolhidos foram ξ_1 = 0,2 e ξ_2 = 0,3.

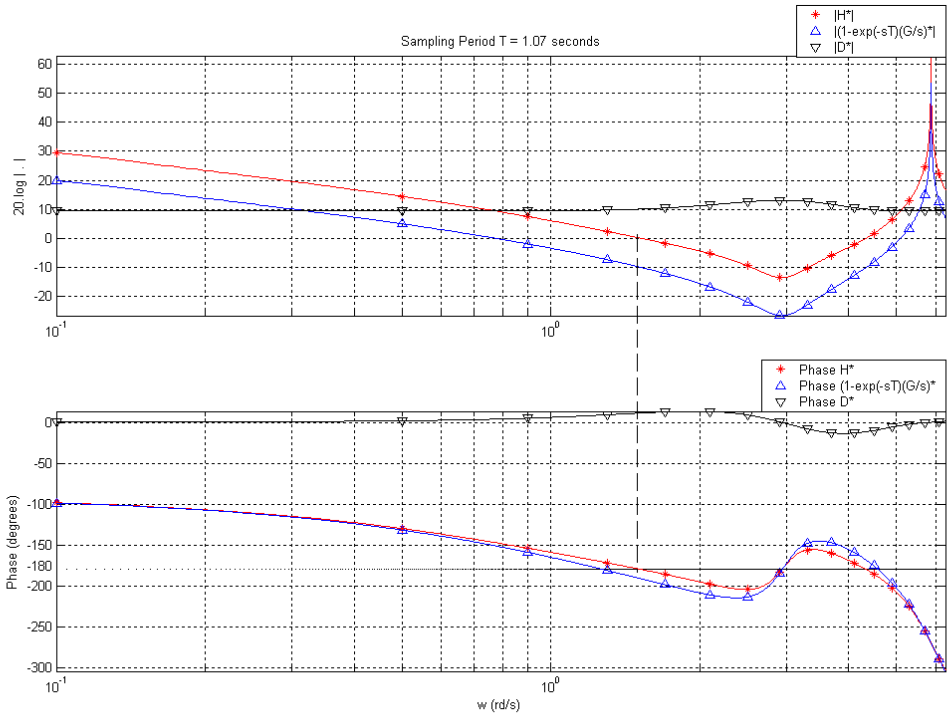


Figura 5.45 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=1,07 segundos.

5.2.14. Diagrama de Bode para o caso da planta de primeira ordem e controle usando controle PD mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,86 segundos e ω= 1,098 rad/seg, ou seja ωT= π radianos ou 180^{o}. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo. O parâmetro ξ = 0,3.

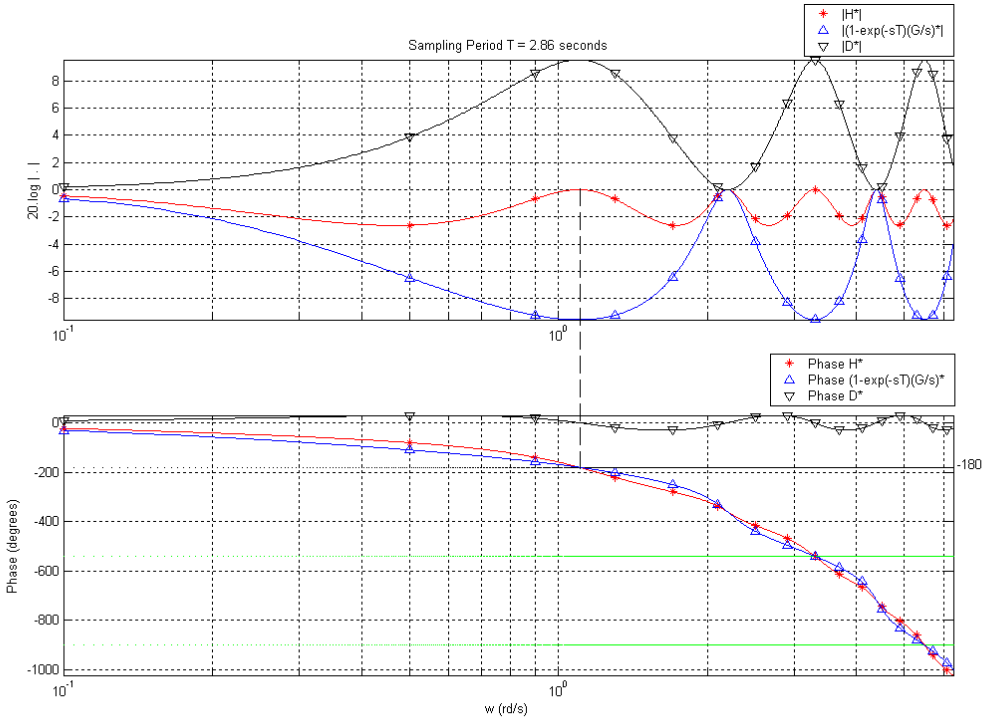


Figura 5.46 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2,86 segundos.

5.2.15. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método schneider

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,6 segundos e ω= 0.3945 rad/seg, ou seja ωT= 1,42 radianos ou 81,37^{o}. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

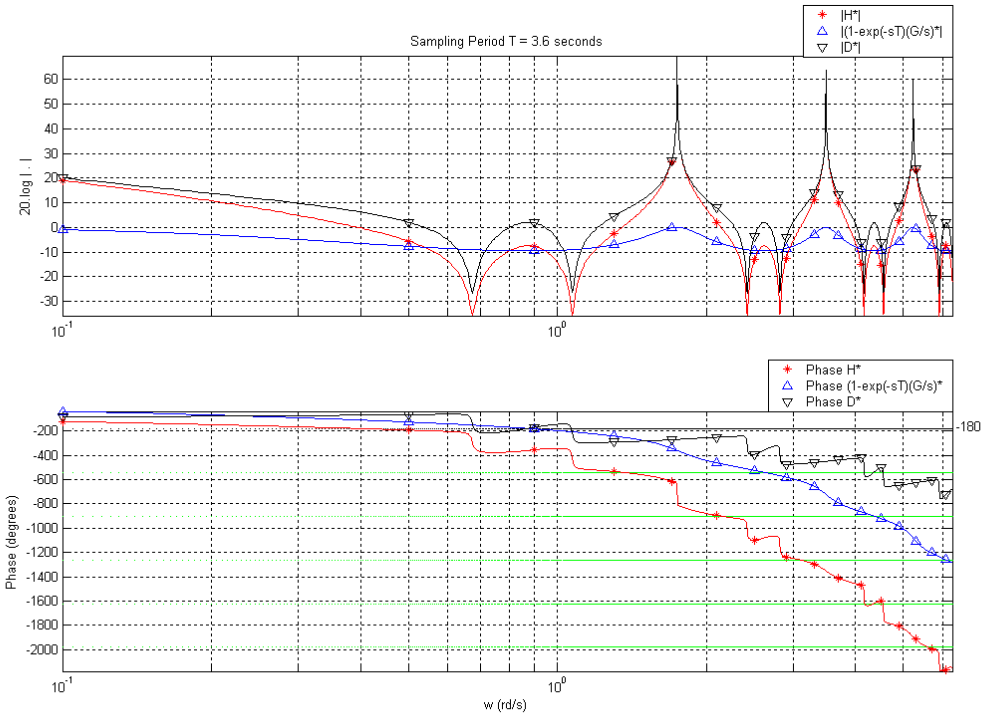


Figura 5.47 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3,6 segundos.

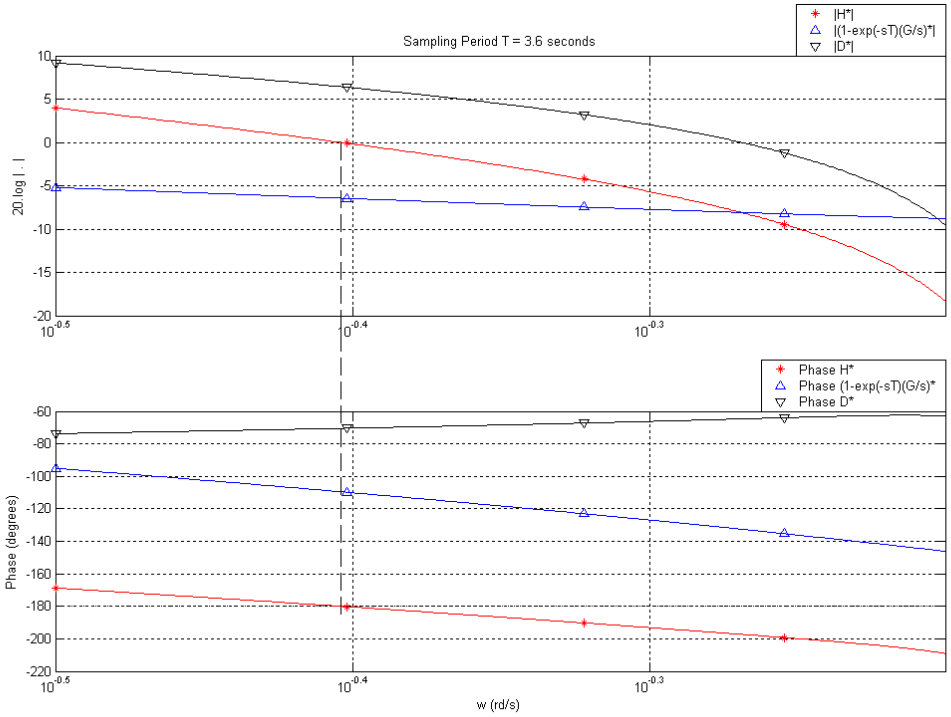


Figura 5.48 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3,6 segundos.

5.2.16. Diagrama de bode para o caso da planta de segunda ordem e controle “lead-lag” mapeado pelo método backward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,63 segundos e ω= 0.7608 rad/seg, ou seja ωT= 2 radianos ou 114,6^{o}. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

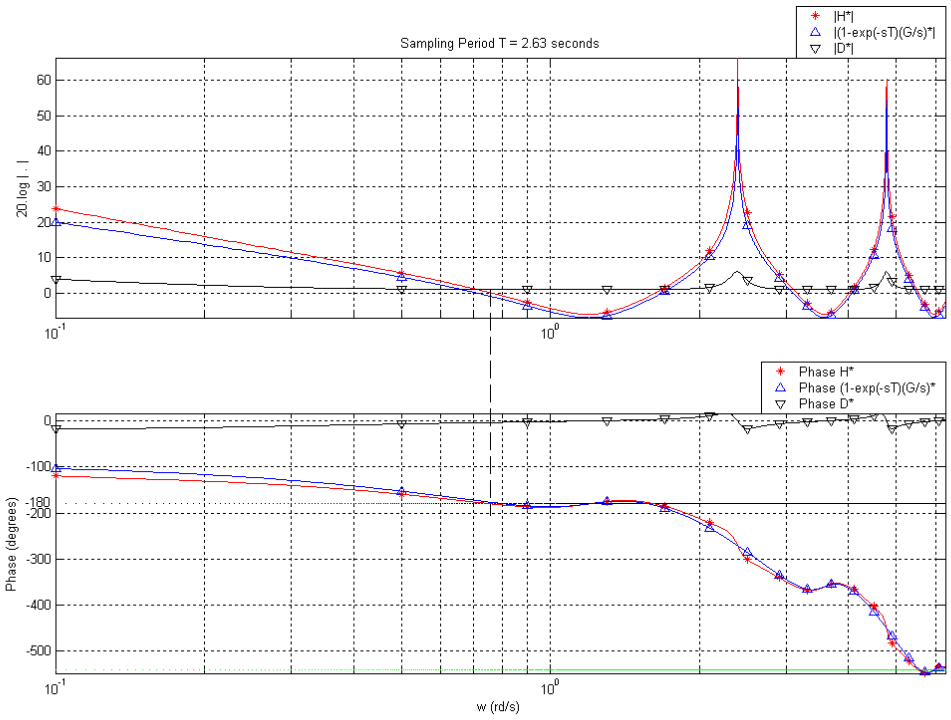


Figura 5.49 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2,63 segundos.

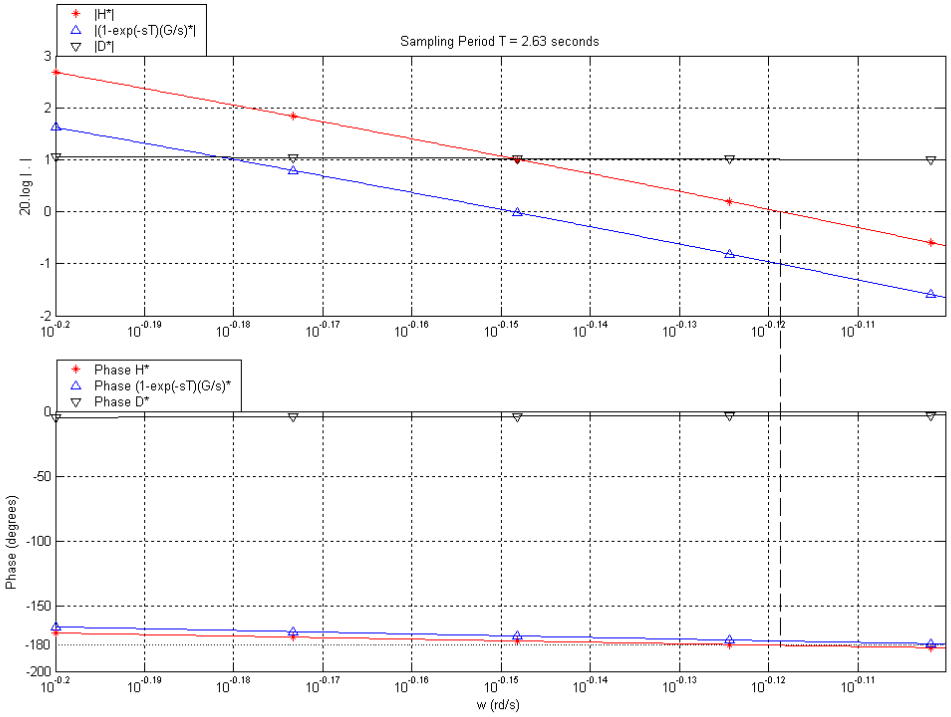


Figura 5.50 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2,63 segundos.

5.2.17. Diagrama de Bode para o caso da planta de segunda ordem e controle “lead-lag” mapeado pelo método forward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,01 segundos e ω= 1,4531 rad/seg, ou seja ωT= 4,3739 radianos ou 250,61^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

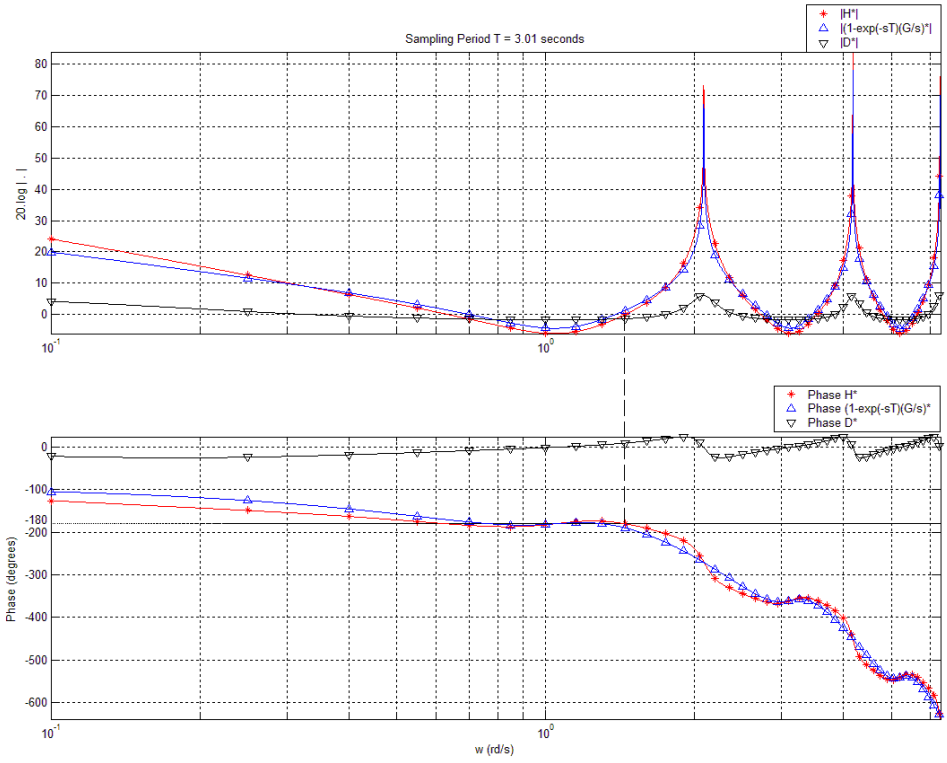


Figura 5.51 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,01 segundos.

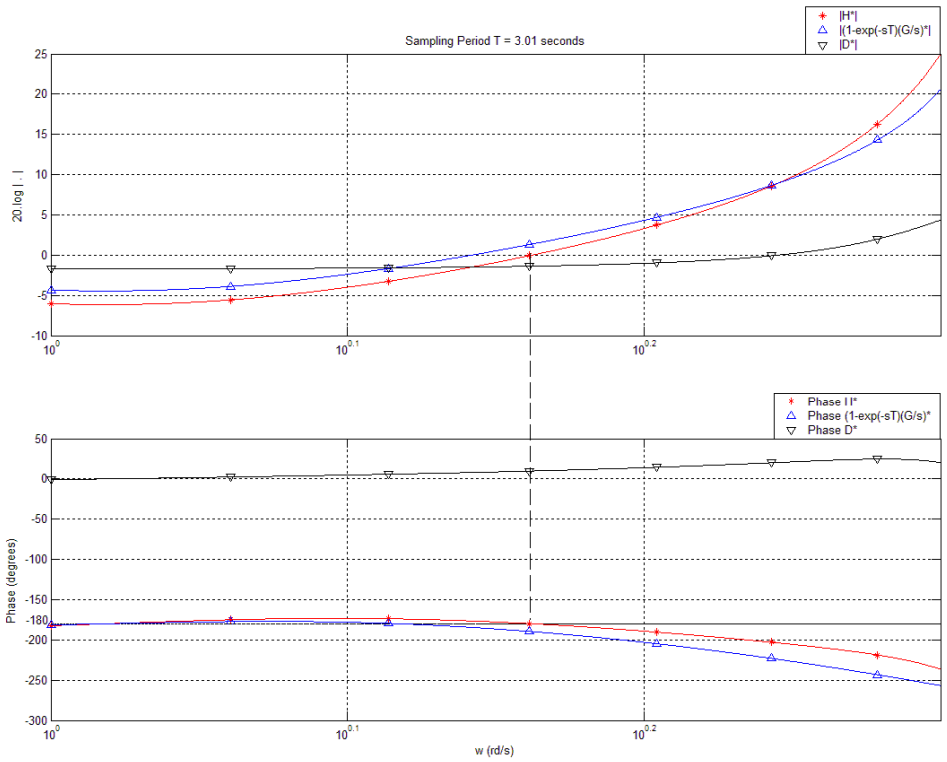


Figura 5.52 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3,01 segundos.

5.2.18. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle "lead-lag" mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,11 segundos e ω= 0,7235 rad/seg, ou seja ωT= 2,25 radianos ou 128,92^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

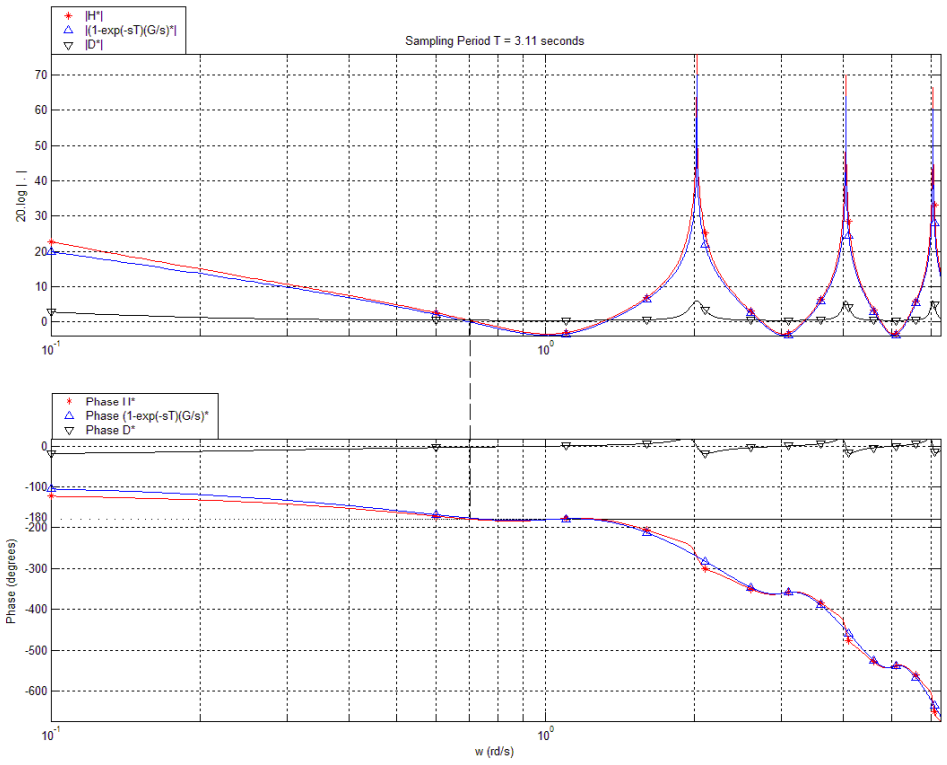


Figura 5.53 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,11 segundos.

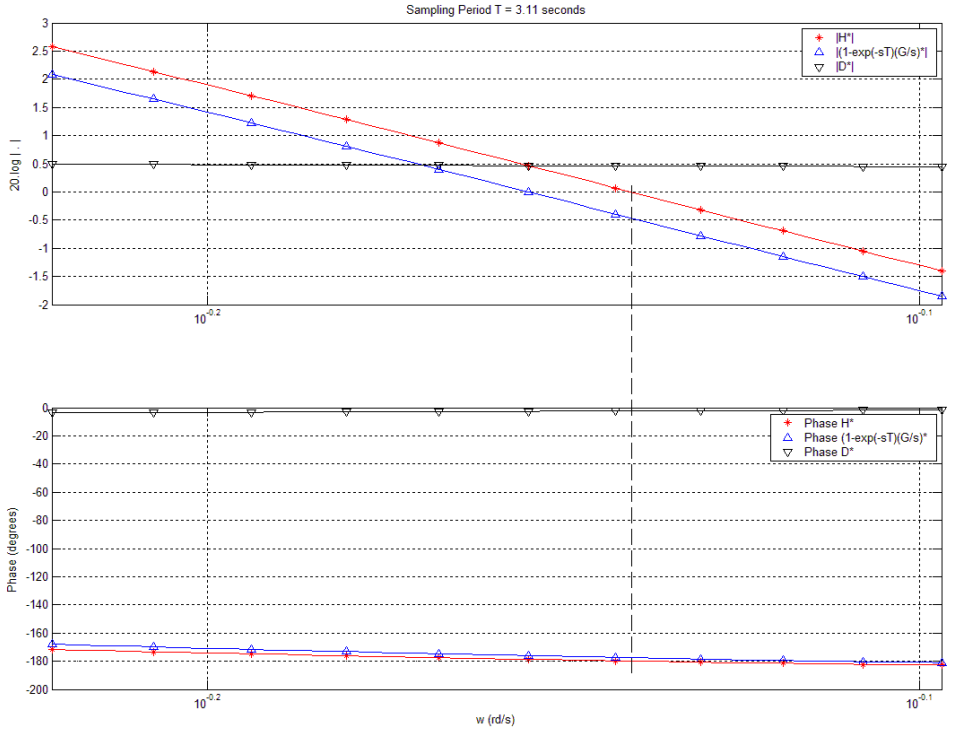


Figura 5.54 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3,11 segundos.

5.2.19. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle "lead-lag" mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,97 segundos e ω= 0,742 rad/seg, ou seja ωT= 2,21 radianos ou 126,265^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

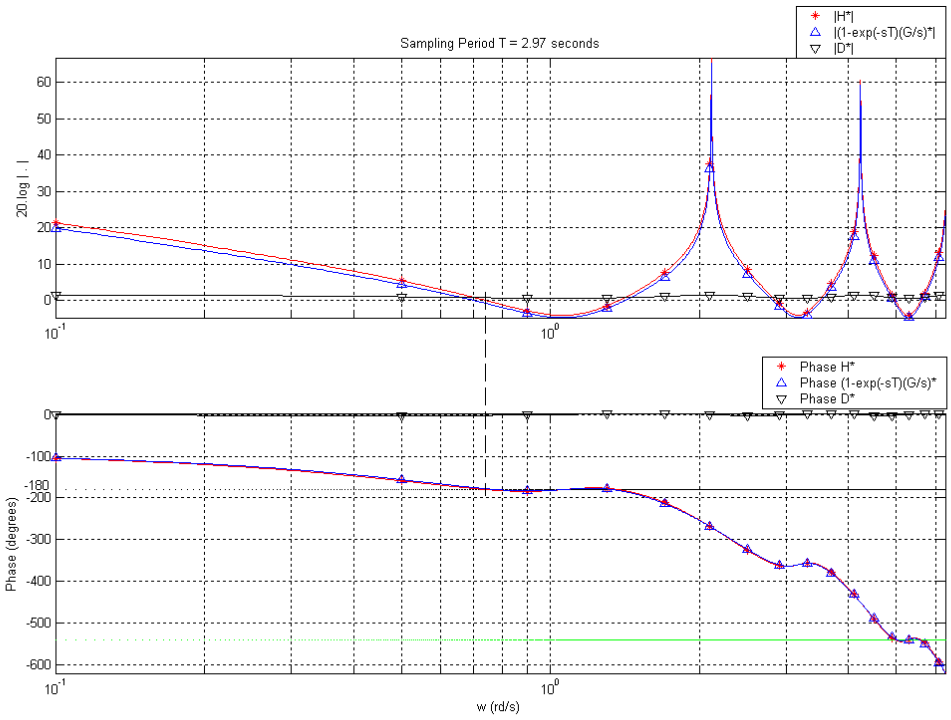


Figura 5.55 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,97 segundos.

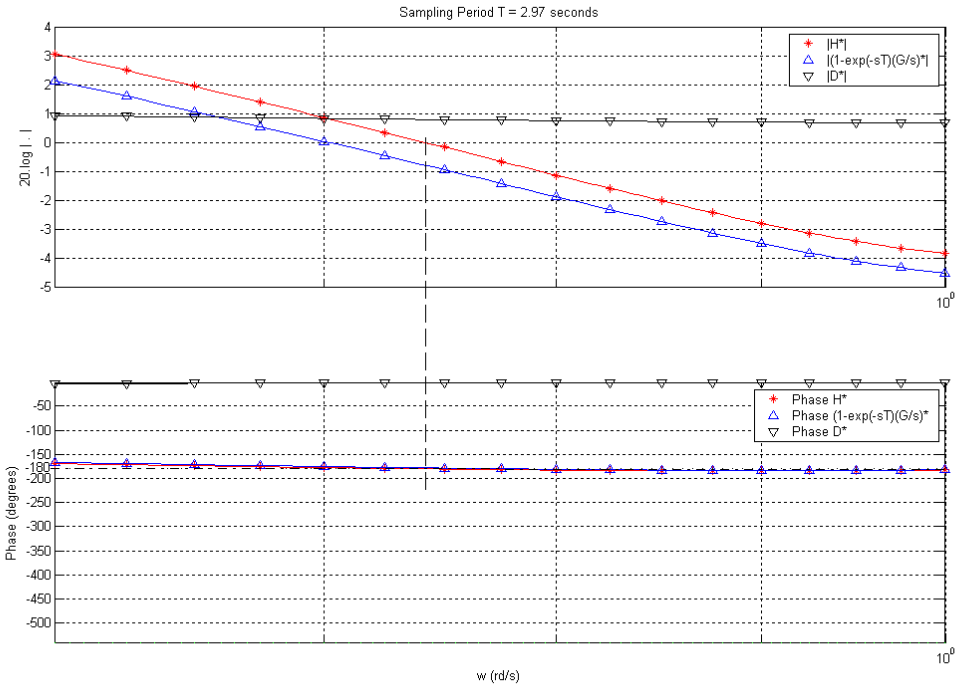


Figura 5.56 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2,97 segundos.

5.2.20. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle "lead-lag" mapeado pelo método Tustin

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,86 segundos e ω= 0,709 rad/seg, ou seja ωT= 2,03 radianos ou 116,18^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

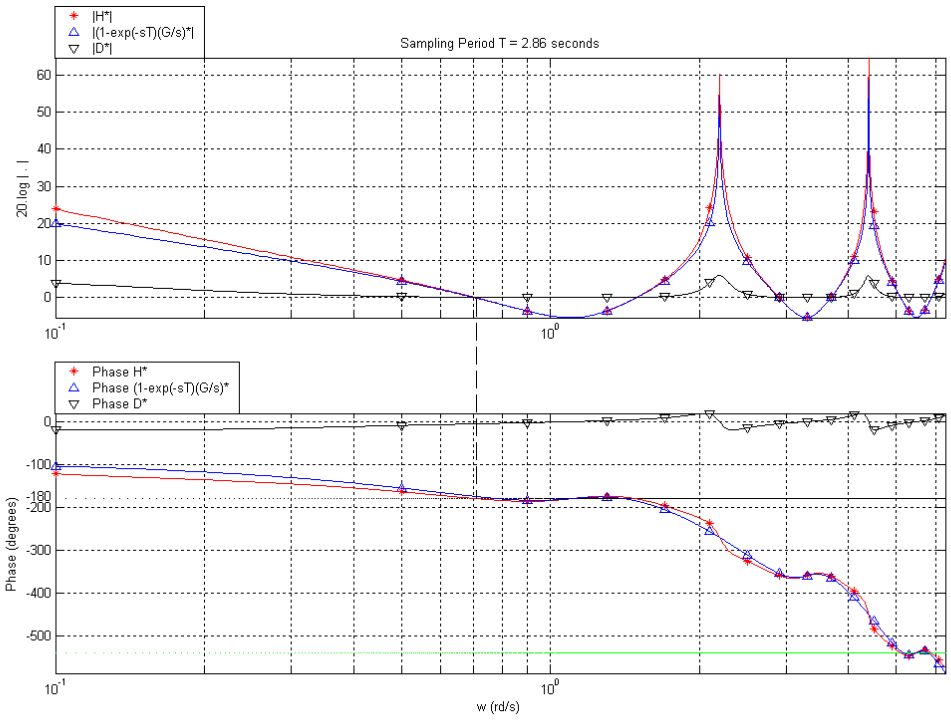


Figura 5.57 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,97 segundos.

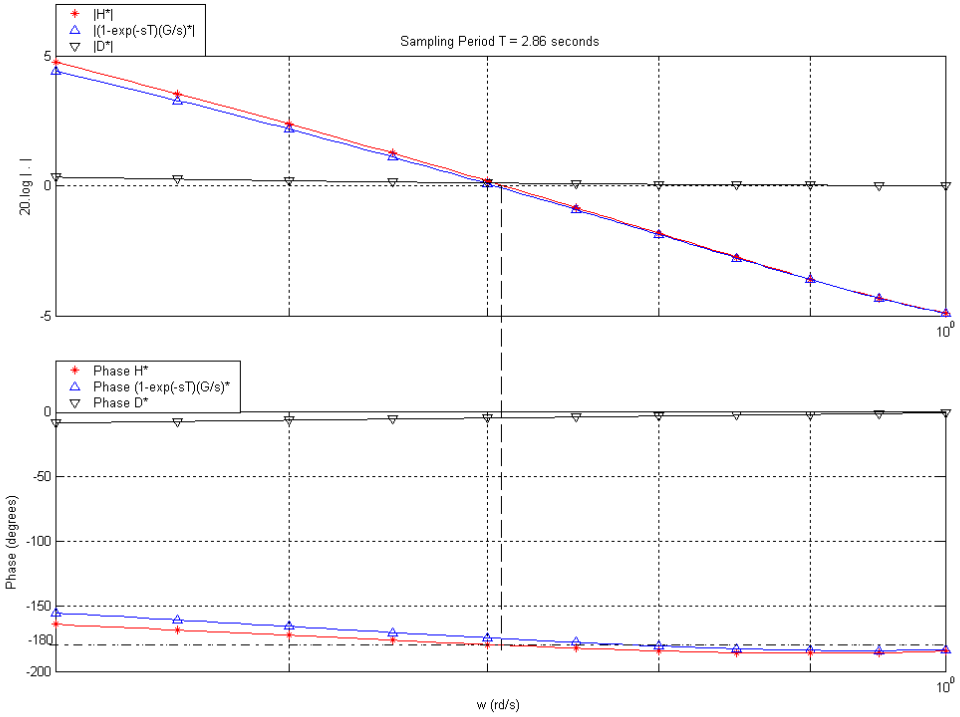


Figura 5.58 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2,86 segundos.

5.2.21. Diagrama de Bode e gráfico fasorial para o caso da planta de segunda ordem e controle usando controle PD mapeado pelo método backward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,01 segundos e ω= 1,016 rad/seg, ou seja ωT= 3,058 radianos ou 175,22^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

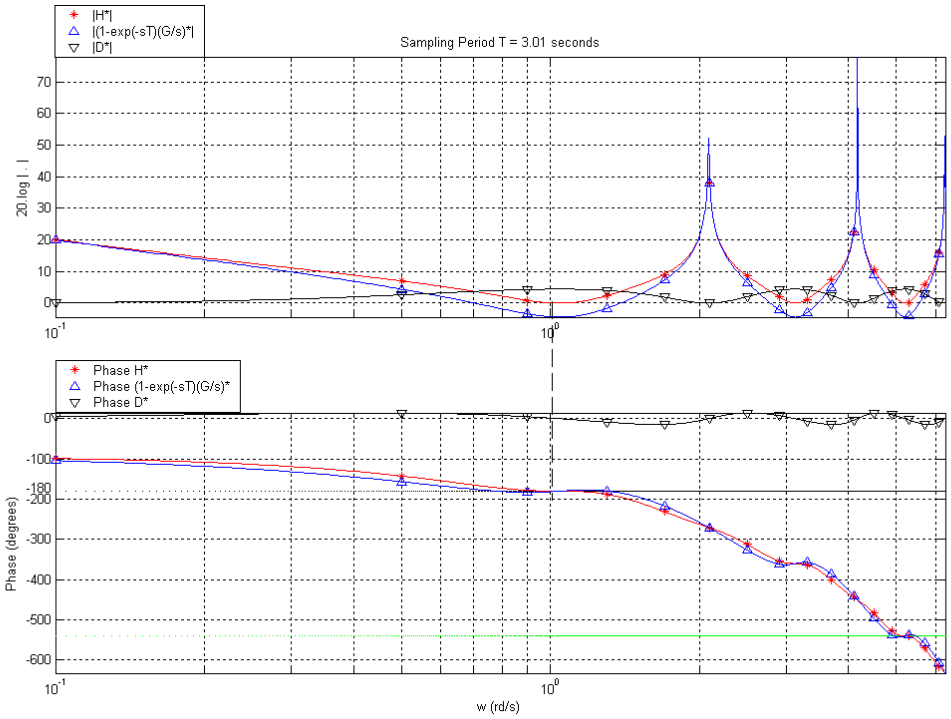


Figura 5.59 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,01 segundos.

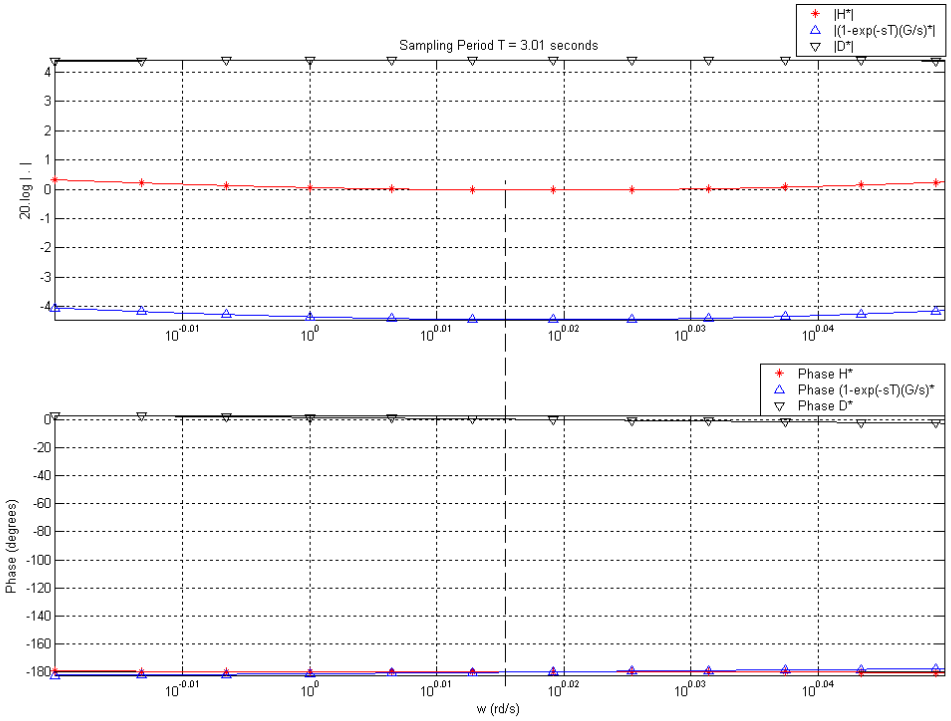


Figura 5.60 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=3,01 segundos.

5.2.22. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PD mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,23 segundos e ω= 1,129 rad/seg, ou seja ωT= 2,52 radianos ou 144,25^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

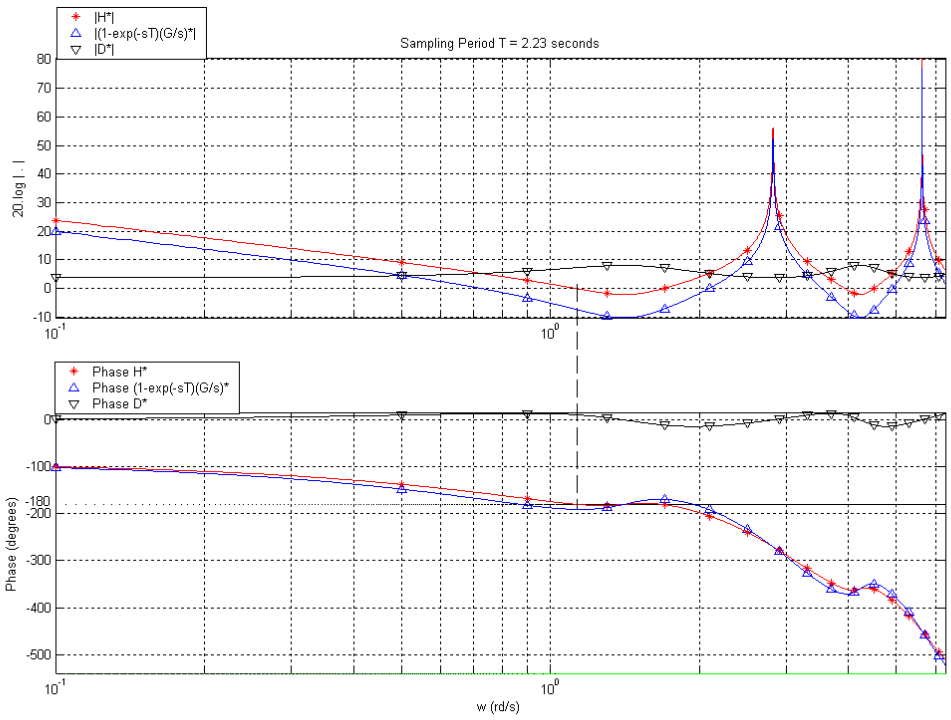


Figura 5.61 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,23 segundos.

5.2.23. Diagrama de Bode para o caso da planta de primeira ordem e controle usando rede “lag” mapeado pelo método forward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,22 segundos e ω= 0,5274 rad/seg, ou seja ωT= 1,17 radianos ou 67,09^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

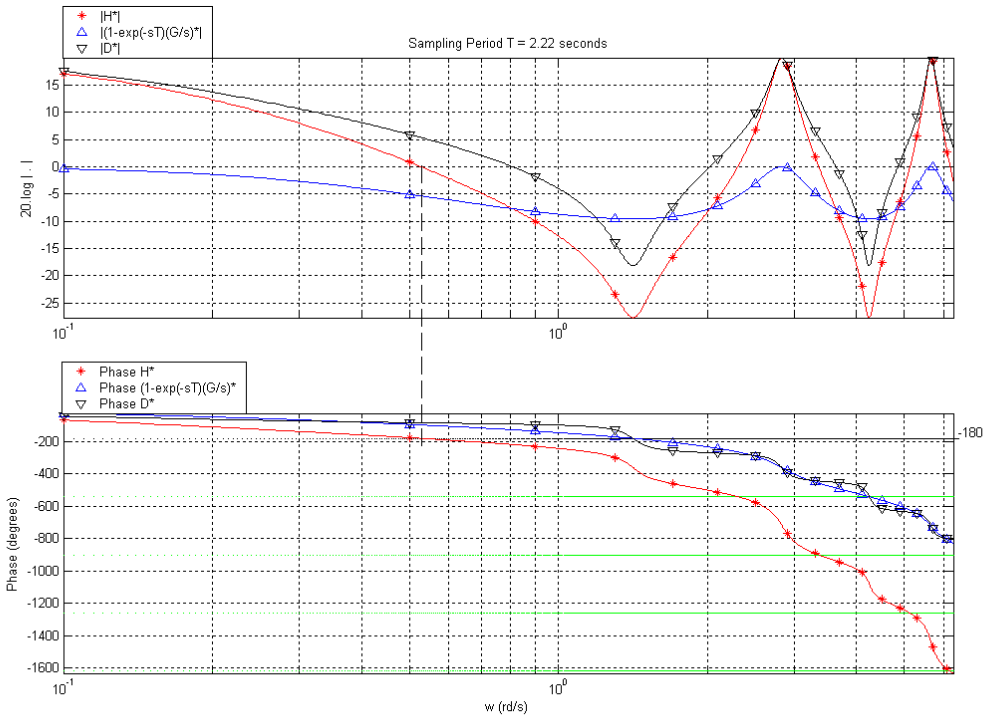


Figura 5.62 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,22 segundos.

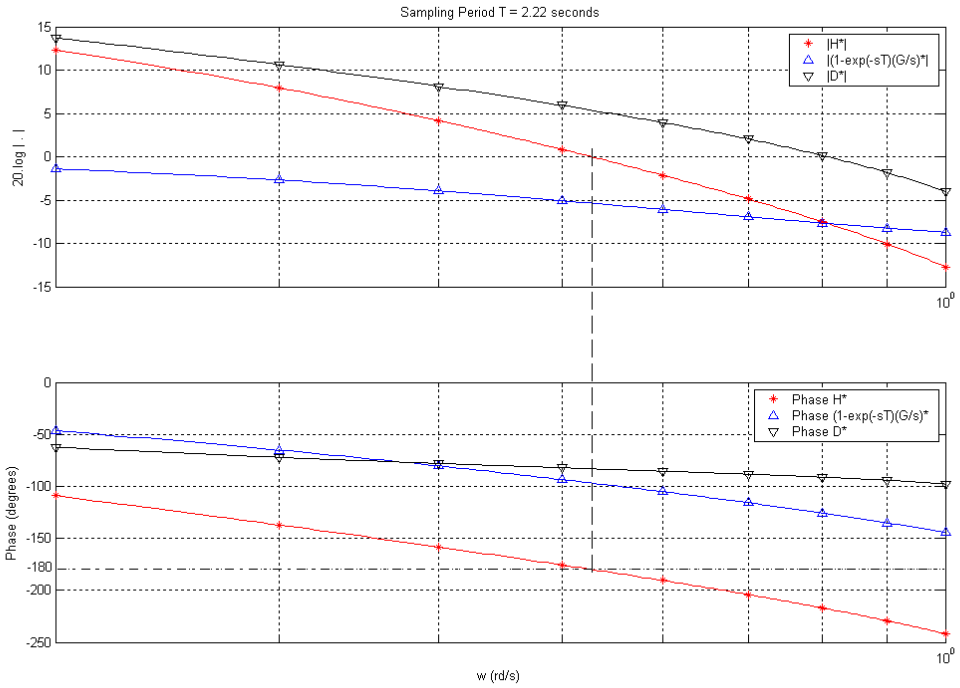


Figura 5.63 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=2,22 segundos.

5.2.24. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,25 segundos e $\omega \approx$ 0,9666 rad/seg, ou seja $\omega T \approx \pi$ radianos ou 180^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

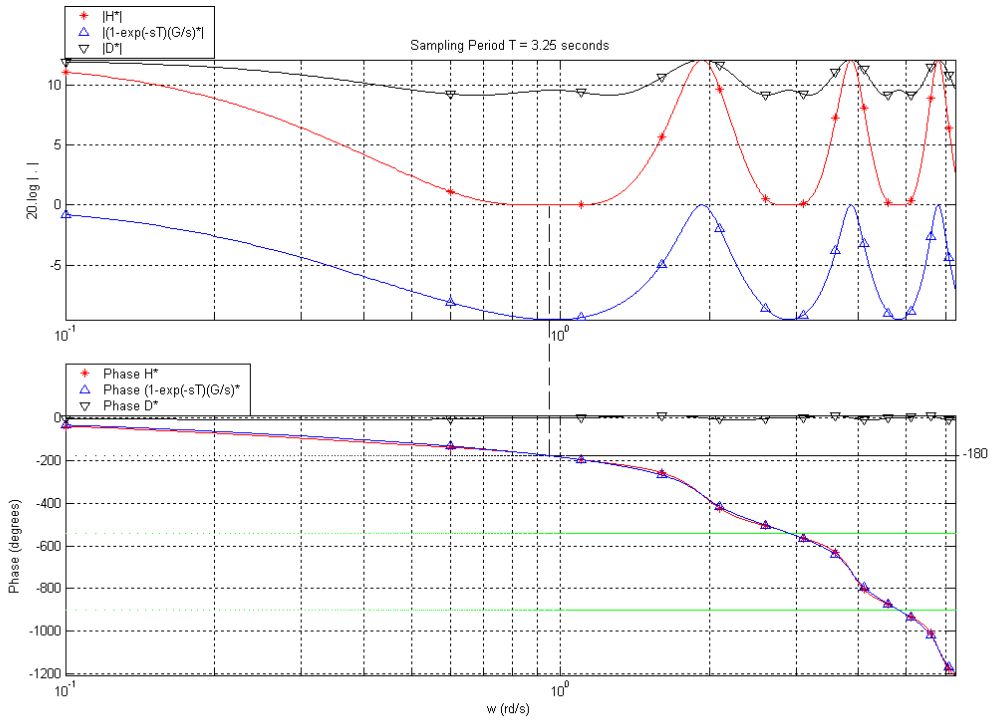


Figura 5.64 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,25 segundos.

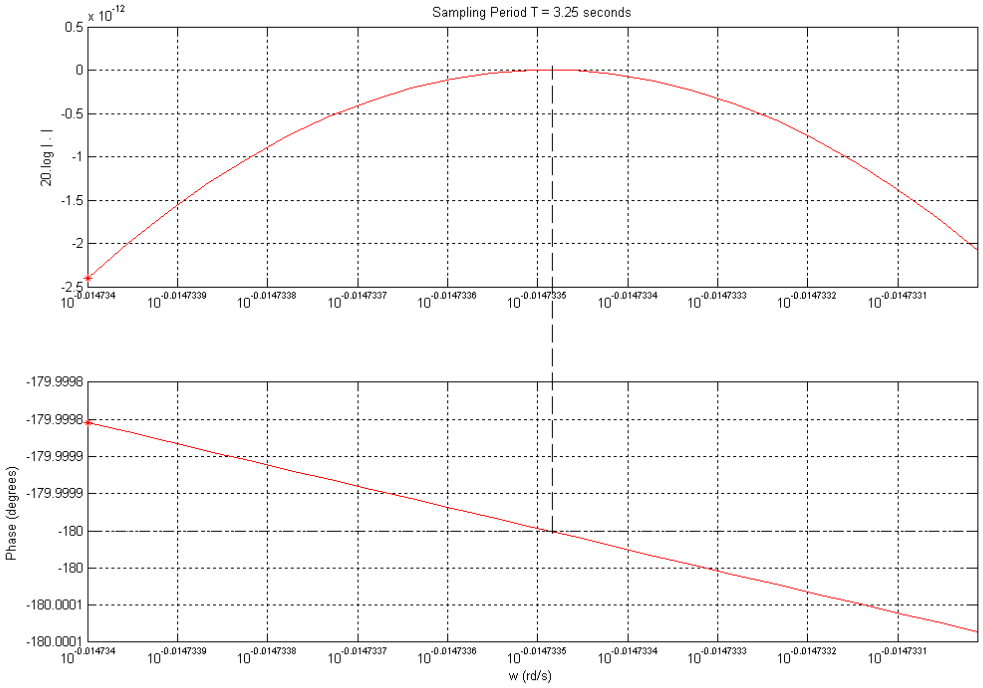


Figura 5.65 - Detalhe do diagrama de Bode para H*(s) para |H*(s)|=1 e fase = -180^{o}, em T=3,25 segundos.

5.2.25. Diagrama de Bode para o caso da planta de primeira ordem e controle usando controle PID mapeado pelo método Tustin

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 4,82 segundos e $\omega \approx$ 0,3256 rad/seg, ou seja ωT $\approx$ 1,569 radianos ou 89,92^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

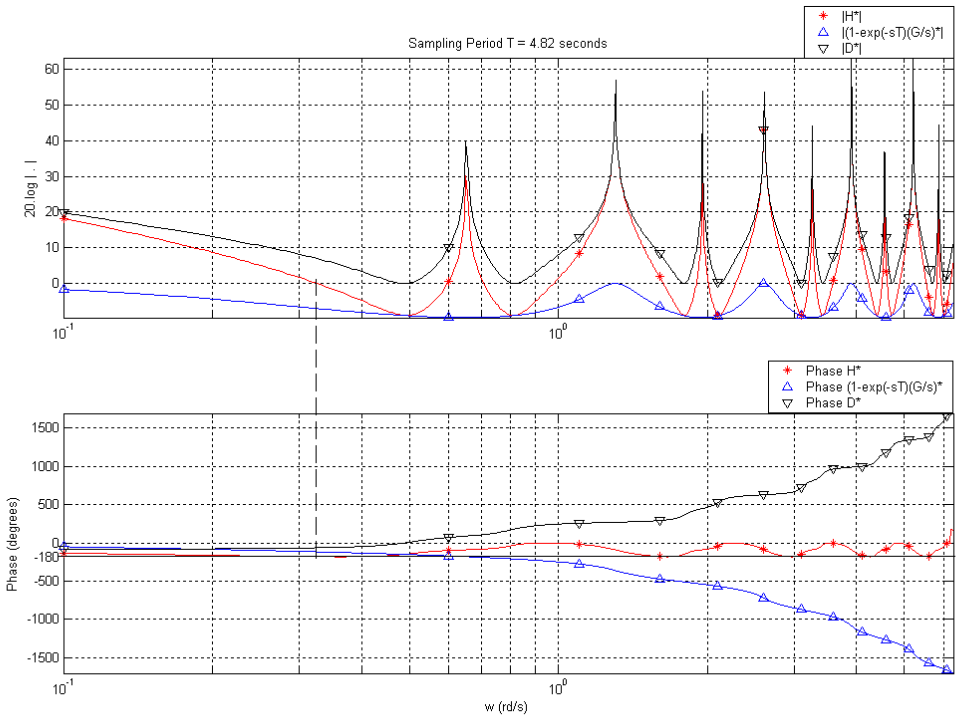


Figura 5.66 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=4,82 segundos.

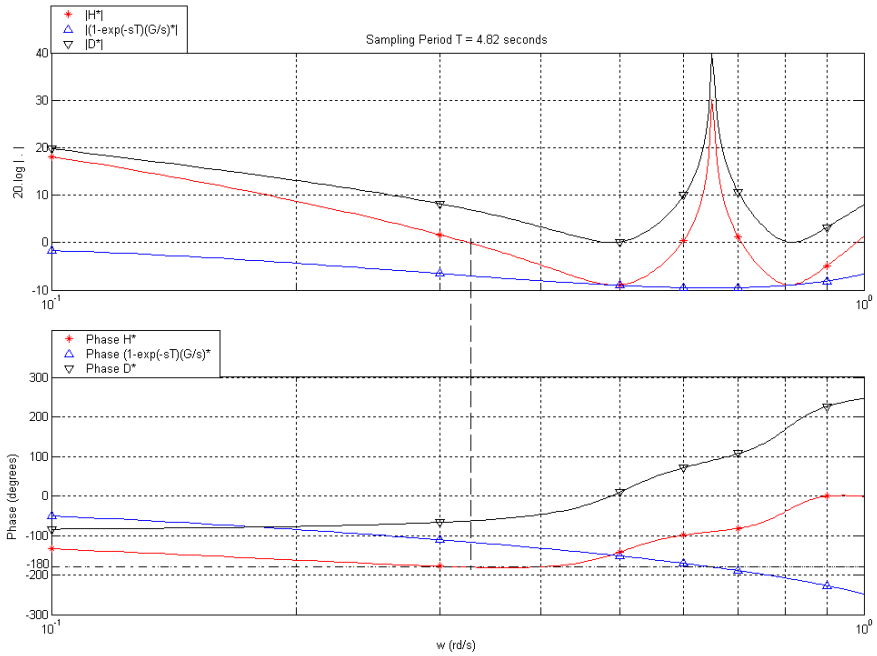


Figura 5.67 - Detalhe dos diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T=4,82 segundos.

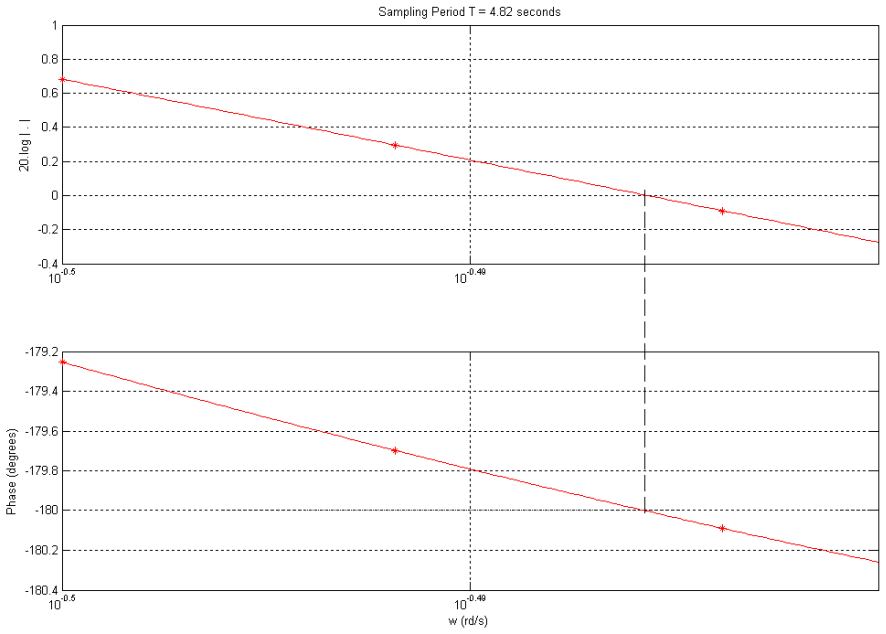


Figura 5.68 - Detalhe do diagrama de Bode para H*(s) para |H*(s)|=1 e fase = -180^{o}, em T=4,82 segundos.

5.2.26. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle derivativo mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 4,71 segundos e $\omega \approx$ 0,667 rad/seg, ou seja $\omega T \approx \pi$ radianos ou 180^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

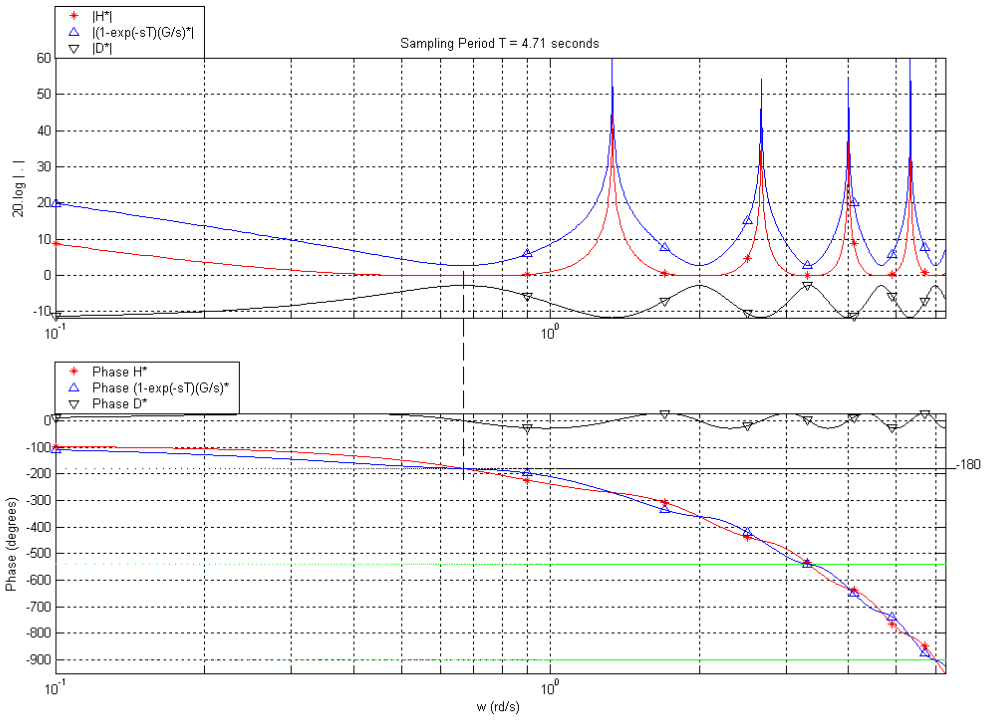


Figura 5.69 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=4,71 segundos.

5.2.27. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle "lag" mapeado pelo método Tustin

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,92 segundos e $\omega \approx$ 0,801 rad/seg, ou seja ωT $\approx$ 3,141 radianos ou 179,99^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

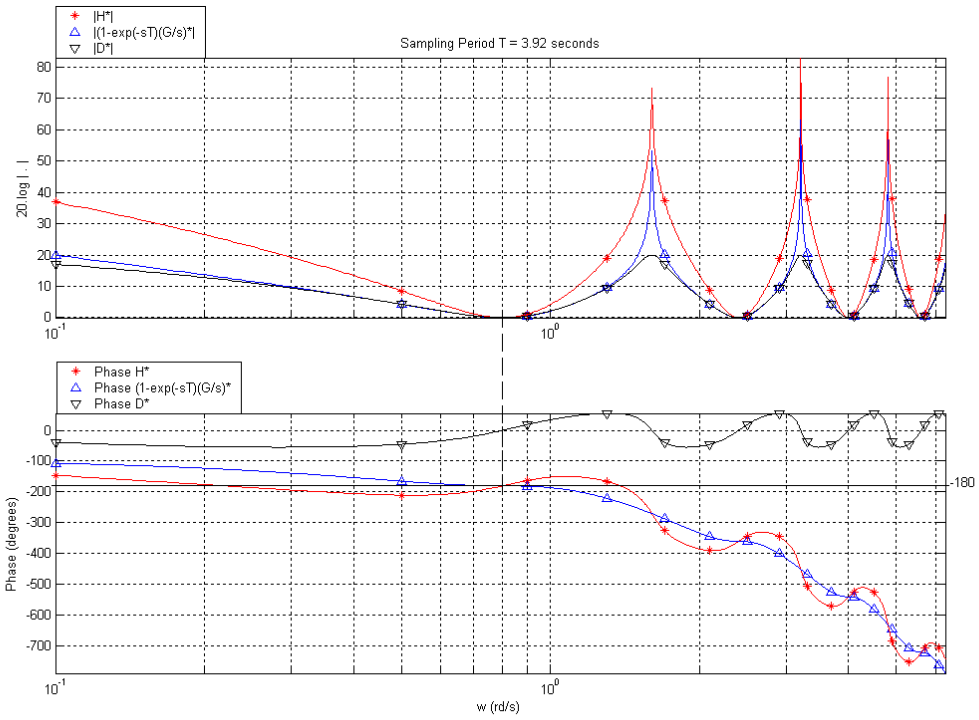


Figura 5.70 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,92 segundos.

5.2.28. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle "lead" mapeado pelo método Tustin

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,92 segundos e $\omega \approx$ 0,801 rad/seg, ou seja ωT $\approx$ 3,141 radianos ou 179,99^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

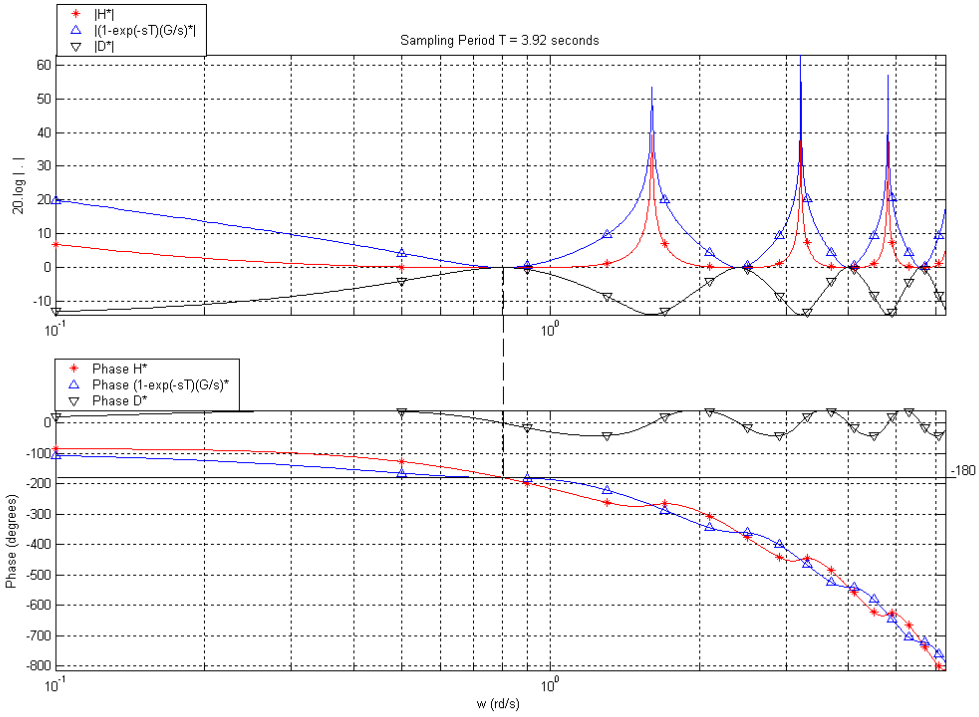


Figura 5.71 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3,92 segundos.

5.2.29. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle PD mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 2,1 segundos e $\omega \approx$ 1,476 rad/seg, ou seja ωT $\approx$ 3,1 radianos ou 177,6^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

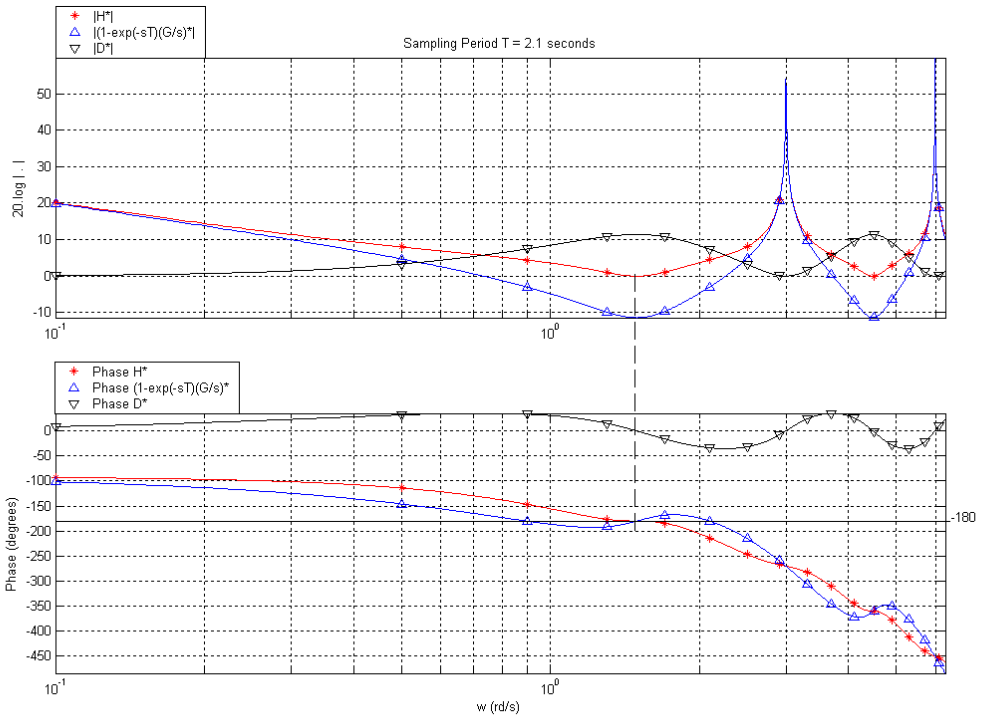


Figura 5.72 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=2,1 segundos.

5.2.30. Diagrama de Bode para o caso da planta de segunda ordem e controle usando controle proporcional

Naturalmente, como o controle é o proporcional não será necessário o uso de mapeamentos. Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 3,92 segundos e $\omega \approx$ 0,79 rad/seg, ou seja ωT $\approx$ 3,09 radianos ou 117,41^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

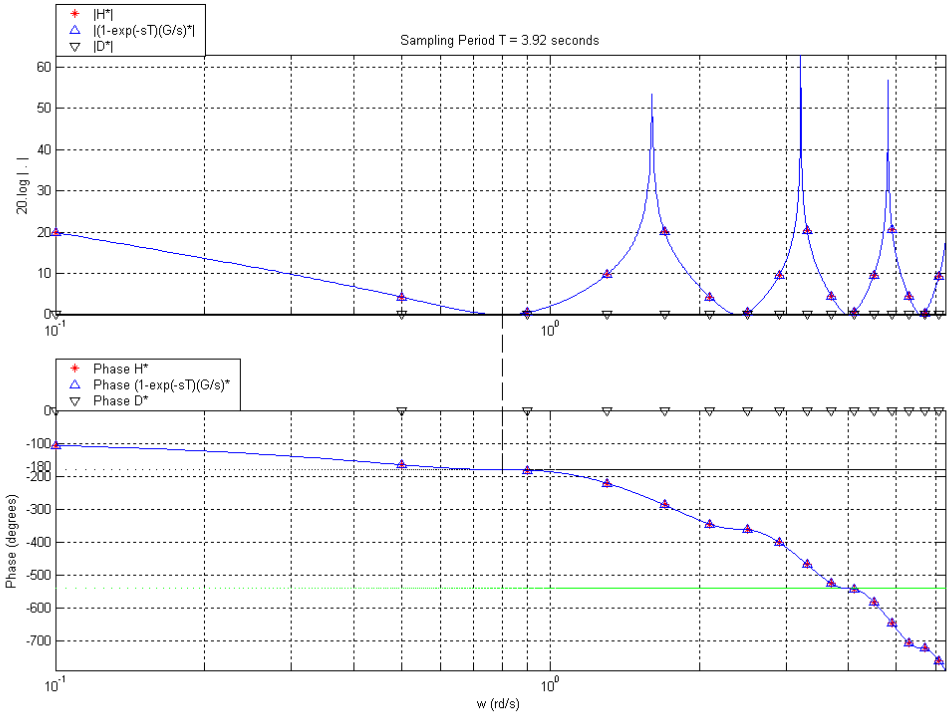


Figura 5.73 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=3.92 segundos.

5.2.31. Diagrama de Bode para o caso da planta de primeira ordem e controle integral mapeado pelo método backward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 6 segundos e $\omega \approx$ 0,522 rad/seg, ou seja ωT $\approx$ 3,132 radianos ou 179,4^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

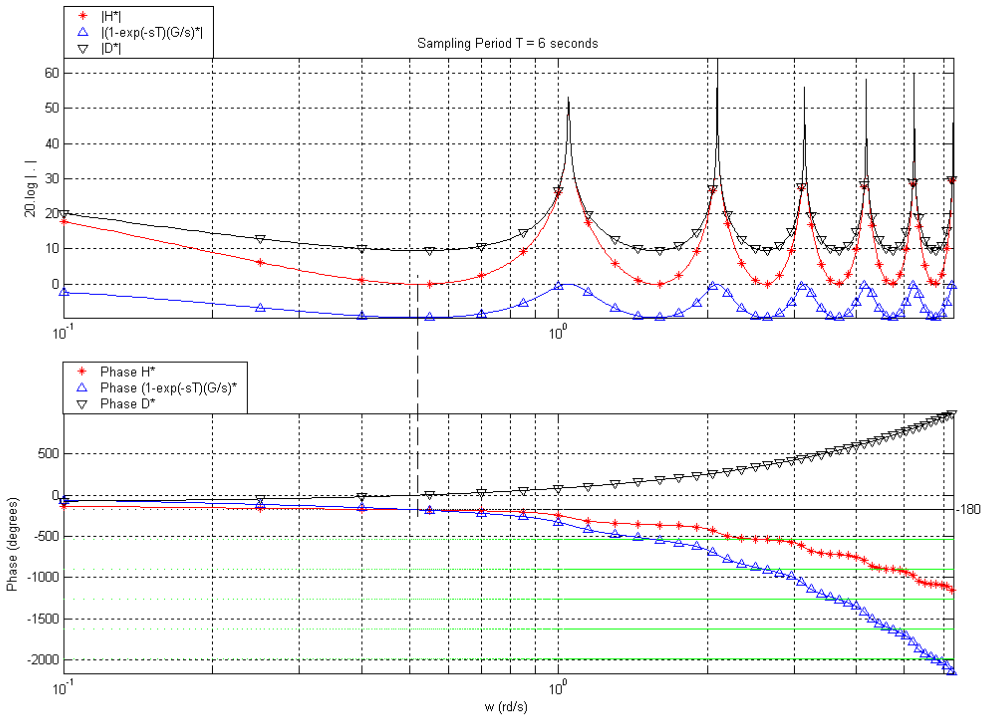


Figura 5.74 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=6 segundos.

5.2.32. Diagrama de Bode para o caso da planta de primeira ordem e controle usando controle integral mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 6,67 segundos e $\omega \approx$ 0,248 rad/seg, ou seja ωT $\approx$ 1,654 radianos ou 94,76^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

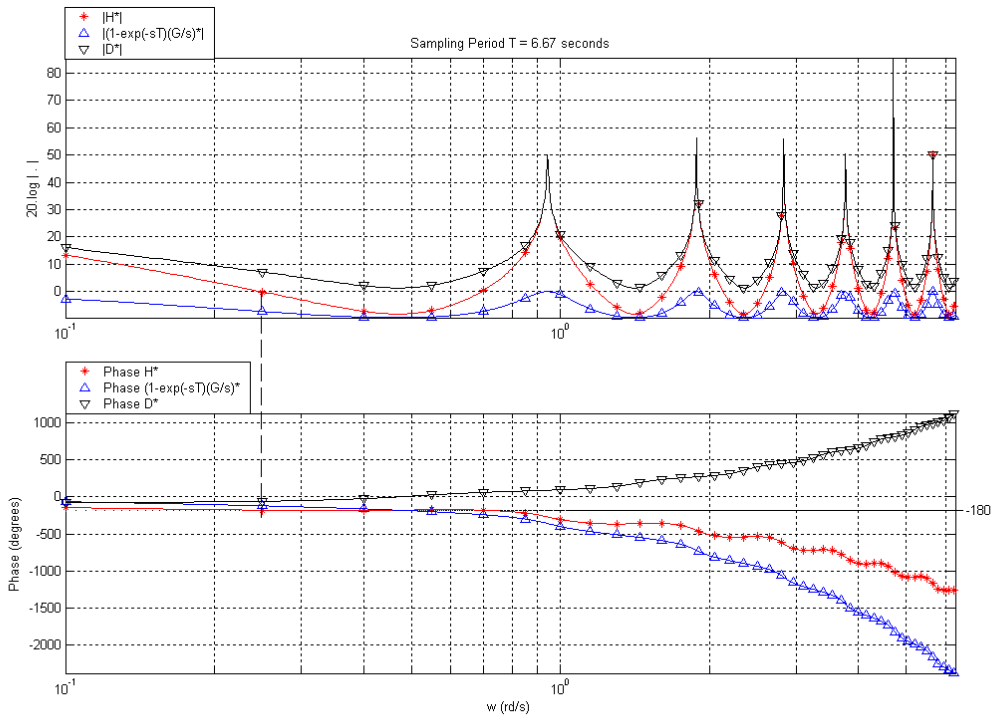


Figura 5.75 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T=6,67 segundos.

5.2.33. Diagrama de Bode para o caso da planta de primeira ordem e controle usando rede lag mapeado pelo método backward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 5,71 segundos e $\omega \approx$ 0,55 rad/seg, ou seja ωT $\approx$ 3.1405 radianos ou 179,94^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

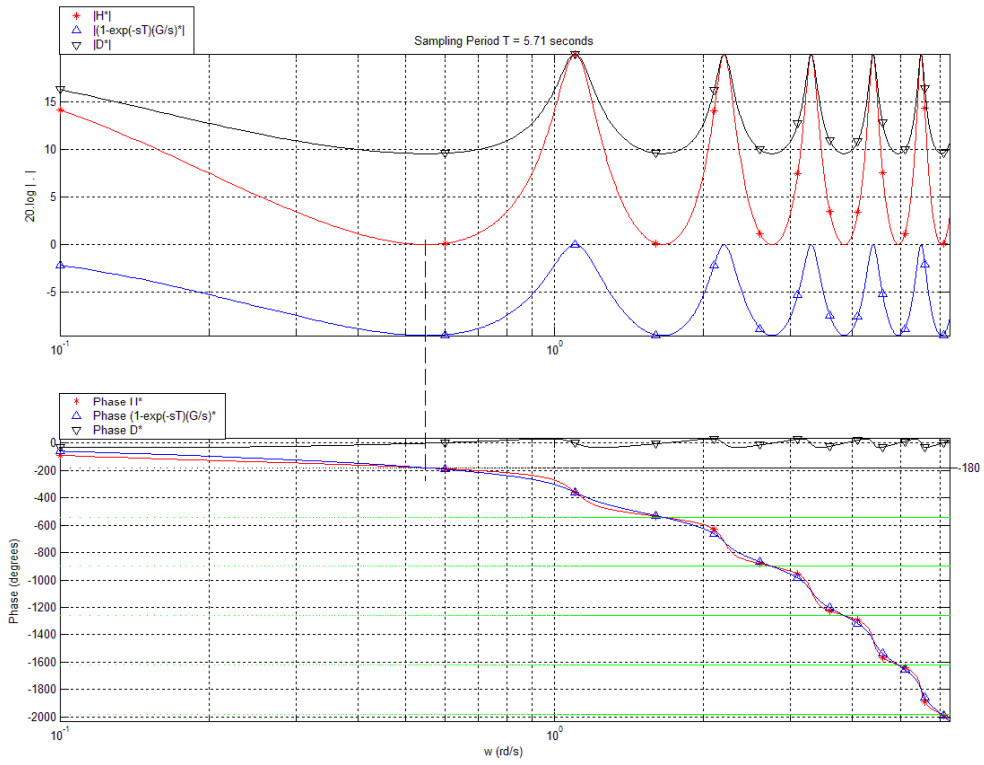


Figura 5.76- Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T= 5,71 segundos.

5.2.34. Diagrama de Bode para o caso da planta de primeira ordem e controle usando rede lag mapeado pelo método Tustin

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 5,72 segundos e $\omega \approx$ 0,314 rad/seg, ou seja ωT $\approx$ 1,796 radianos ou 102,9^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

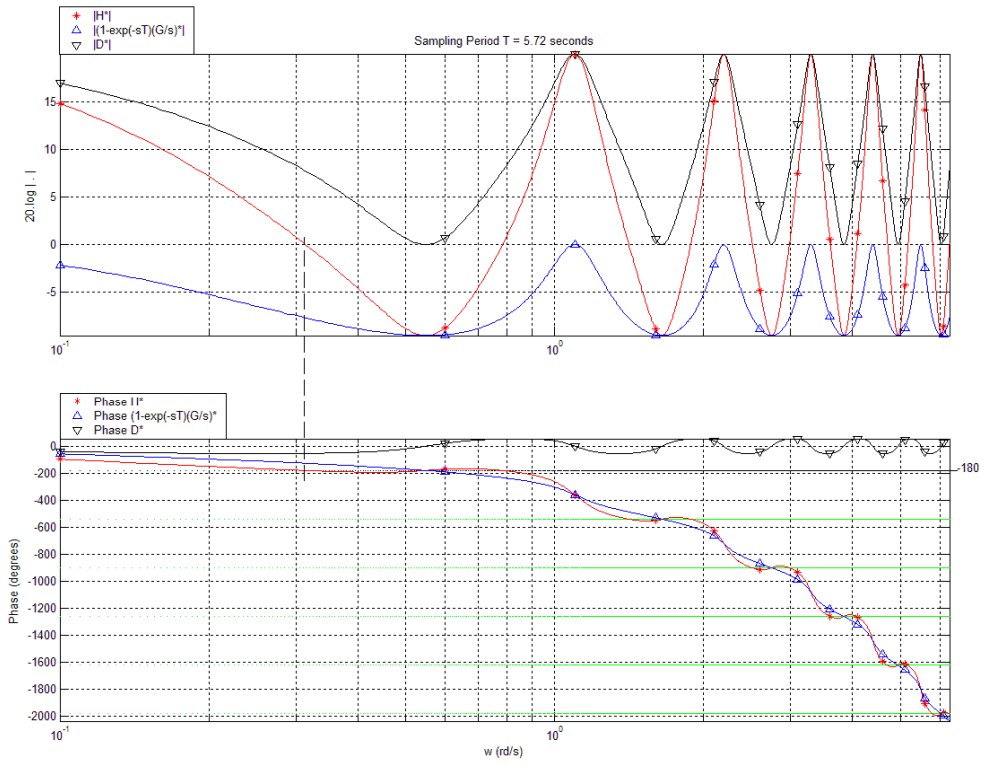


Figura 5.77 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T= 5,72 segundos.

5.2.35. Diagrama de Bode para o caso da planta de primeira ordem e controle PID mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 5,39 segundos e $\omega \approx$ 0,583 rad/seg, ou seja ωT $\approx$ 3,142 radianos ou 180,04^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

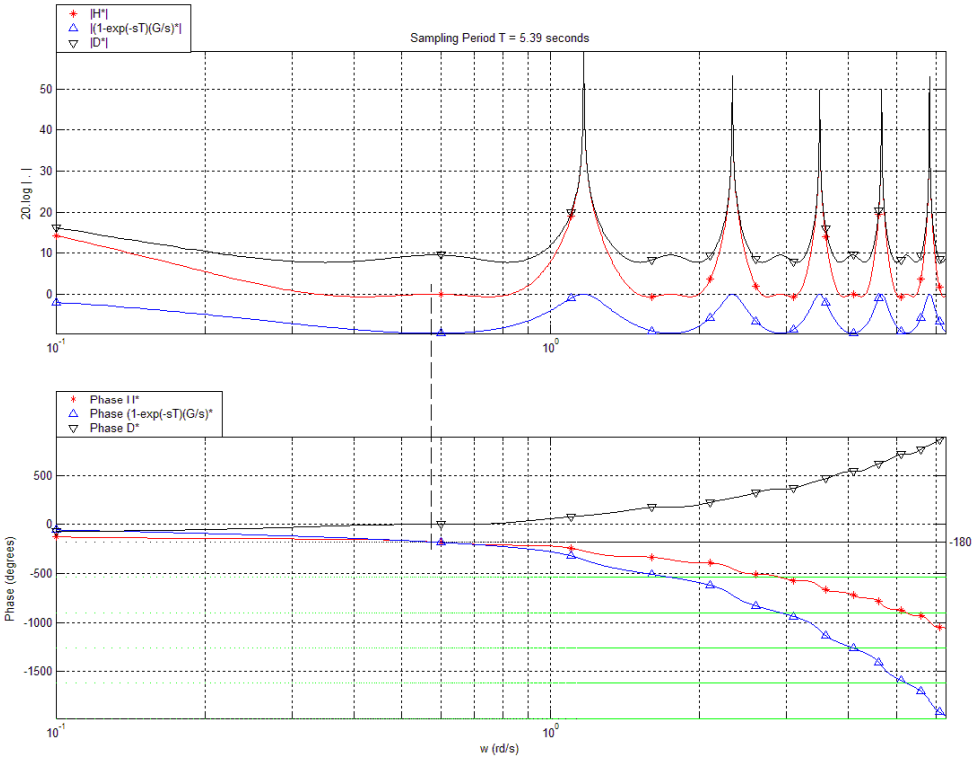


Figura 5.78 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T= 5,39 segundos.

5.2.36. Diagrama de Bode para o caso da planta de segunda ordem e controle com rede de avanço de fase (“lead”) mapeado pelo método ST1

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 5,21 segundos e $\omega \approx$ 0,603 rad/seg, ou seja ωT $\approx$ 3,141 radianos ou 180,0023^{o}. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

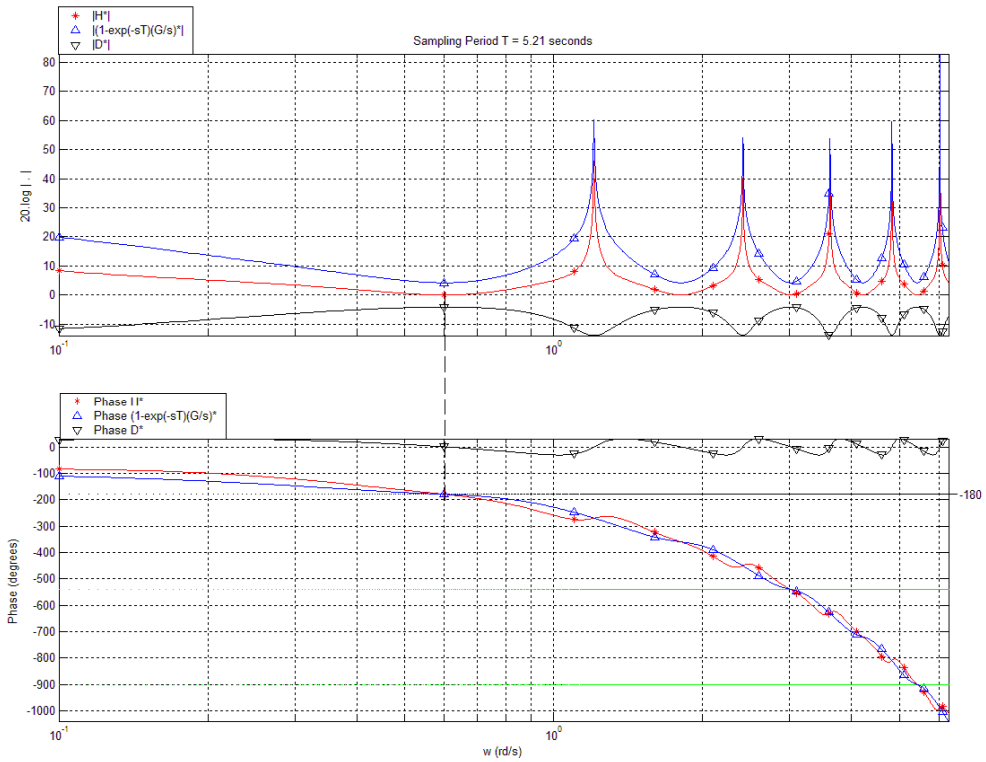


Figura 5.79 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^{*}$ para |H*(s)|=1 e fase = -180^{o}, T= 5,21 segundos.

5.2.37. Diagrama de Bode para o caso da planta de segunda ordem e controle com rede de avanço de fase ("lead") mapeado pelo método ST2

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 6,09 segundos e $\omega \approx$ 0,516 rad/seg, ou seja ωT $\approx$ 3,142 radianos ou 180,05^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

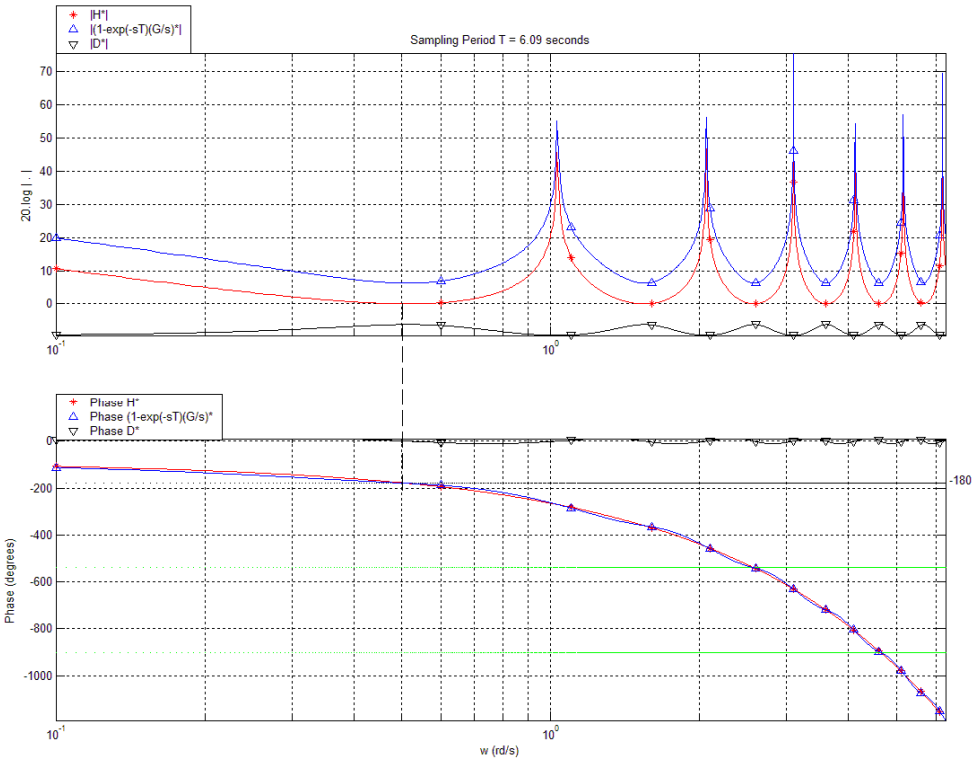


Figura 5.80 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T= 6,09 segundos.

5.2.38. Diagrama de Bode para o caso da planta de segunda ordem e controle PD mapeado pelo método forward

Para tal caso |H*| = 1 e a fase de H* é -180 graus ocorreu para um T = 5,22 segundos e $\omega \approx$ 0,602 rad/seg, ou seja ωT $\approx$ 3,142 radianos ou 180,05^o. Tal primeiro caso ocorre na determinação principal, i.e., no primeiro ciclo.

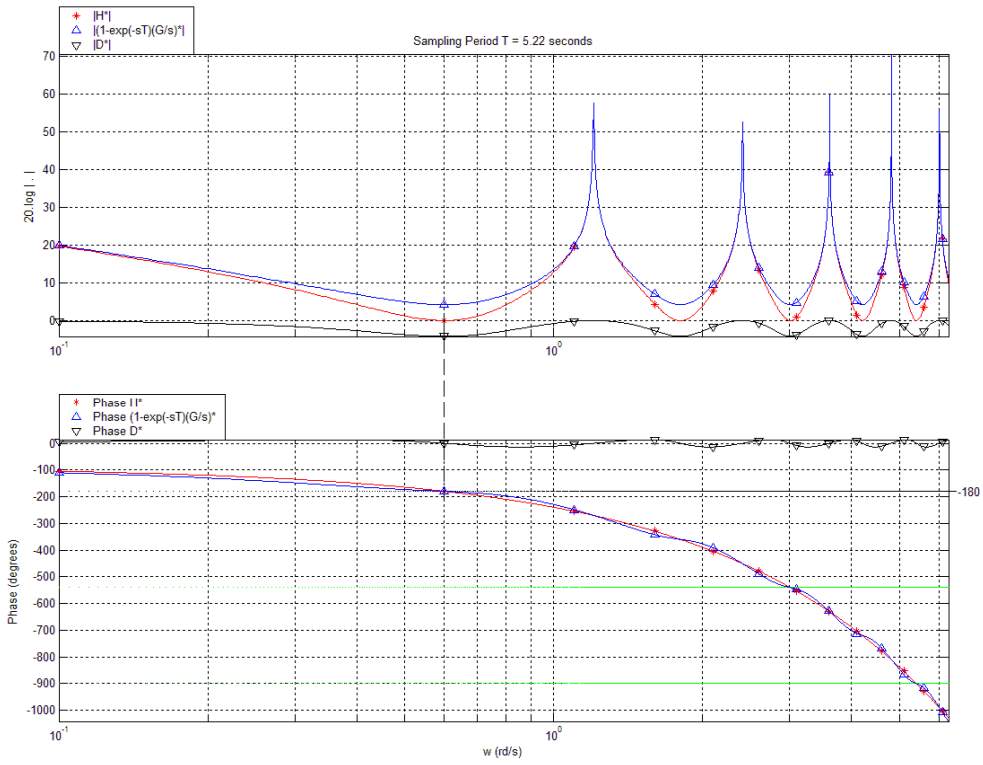


Figura 5.81 - Diagramas de Bode para H*(s), D*(s) e $\left(1-e^{-sT}\right)\left(\frac{G(s)}{s}\right)^*$ para |H*(s)|=1 e fase = -180^o, T= 5,22 segundos.

CAPÍTULO 6

FRONTEIRAS DA ESTABILIDADE/INSTABILIDADE PARA SISTEMAS DE CONTROLE ANALÓGICOS-DIGITAIS SISO

Em **Tredinnick (1999),** e em especial nos resultados apresentados no Apêndice C desta Tese, foi mostrado que o falseamento ("aliasing") e o controlador digital podem instabilizar o sistema de controle analógico/digital em malha-fechada, para uma dada planta usada. Levantou-se no entanto a suspeita de que o falseamento ("aliasing") *per si* poderia instabilizar um sistema em malha-fechada, mesmo sem a presença do controlador, isto é, existindo apenas na malha a planta e o amostrador. A inserção do controlador digital seria um problema adicional à estabilidade sistema o qual, se mal dimensionado, poderia gerar a instabilidade numérica provocando a instabilidade do sistema em malha fechada. A questão até então não respondida era como dimensionar este fenômeno (falseamento) de forma a fazê-lo ser função do período de amostragem, dos conversores A/D e D/A, bem como da planta usada e a partir dessa técnica poder prever até qual valor de período de amostragem o sistema em malha-fechada se manteria assintoticamente estável, ficando instável a partir desse determinado valor de período de amostragem. Considerando um sistema linear invariante no tempo e SISO este Capítulo apresenta resultados que permitiram tal análise.

6.1. INTRODUÇÃO.

Assim, tal como explicado acima, projetar um controlador discreto que suporte as perturbações do falseamento sem instabilizar-se é uma necessidade de projeto.

Se o somatório de $\Delta G_{*,T}(z)$ em (4.11) vier a divergir ou mesmo manter um valor finito indesejável então o controle numérico poderá se instabilizar, pois o somatório de (4.11) não convergirá. Para esse caso (4.10) se tornaria muito distinta da representação discreta da planta como percebida pelo controlador discreto, gerando problemas dado que o controle estaria tentando controlar uma planta que não mais corresponde à realidade. Porém, caso o controle numérico se

mantenha assintoticamente estável será sinal de que o somatório de $\Delta G_{*,T}(z)$ em (4.11) convergiu.

Assim, torna-se relevante estudar a convergência da série $\Delta G_{*,T}(z)$ em (4.11) para diversas plantas distintas e como função do período de amostragem visando dimensionar, medir o valor de tal somatório. Essa medida é um primeiro passo para conectar o falseamento ao problema da instabilidade/estabilidade em sistemas de controle analógicos/digitais.

6.2. ABORDAGEM ANALÍTICA DE MAJORANTES REAIS.

Podemos investigar matematicamente se existe um número real positivo M_0 chamado majorante, que seja maior que o valor obtido quando $\Delta G_{*,T}(z)$ converge para uma dada planta G e para um determinado período de amostragem T. Assim, podemos escrever, por exemplo:

$$\left\|\Delta G_{*,T}(z)\right\|_{\infty} < M_0 \tag{6.1}$$

ou

$$\left\| \frac{1}{T} \sum_{\substack{k\neq 0 \\ k=-\infty}}^{\infty} \frac{1-e^{-sT}}{s}.G(s)\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_{\infty} < M_0 \tag{6.2}$$

Supondo a planta G(s) representada por uma função racional própria estável podemos supor que haverá uma convergência da série. Então M_0 é finito, i.e., $M_0<\infty$. Dessa premissa do majorante M_0 chegamos no seguinte Teorema devido ao autor desta Tese,

<u>Teorema 6.1</u>: o falseamento é limitado superiormente pelo somatório dos valores supremos da planta G(s) nos instantes de amostragem, ou seja:

$$\left\| \frac{1}{T} \sum_{\substack{k\neq 0 \\ k=-\infty}}^{\infty} \frac{1-e^{-sT}}{s}.G(s)\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_{\infty} < \sum_{\substack{k\neq 0 \\ k=-\infty}}^{\infty} \left\| G(s)\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_{\infty} \tag{6.3}$$

□

Prova:

Recordemos que $\|\cdot\|_\infty$ representa a norma H_∞ que é definida para um vetor $\underline{x} = (x_1, x_2, ..., x_n) \in R^n$ como,

$$\|\underline{x}\|_\infty = \max_{0 \le j \le n} |x_j| \tag{5.4}$$

ou, para vetores de dimensão infinita $\underline{x} = (x_1, x_2, ...) \in \lambda_\infty$:

$$\|\underline{x}\|_\infty = \sup_{0 \le j \le \infty} |x_j| \tag{6.5}$$

ou, para funções contínuas $f(\omega) \in L_\infty$:

$$\|f(\omega)\|_\infty = \sup_{\omega} |f(\omega)| \tag{6.6}$$

Onde a norma “sup” fornece o supremo de f(ω) em (5.4), isto é, o “menor valor superior” (“least upper bound”) do valor absoluto de f(ω).

Voltando aos cálculos a partir de (6.2) temos,

$$\frac{1}{T} \left\| \sum_{\substack{k \neq 0 \\ k = -\infty}}^{\infty} \Delta G_k \right\|_\infty < M_0 \tag{6.7}$$

onde denominou-se,

$$\Delta G_k := \frac{1 - e^{-sT}}{s} . \frac{G(s)}{1} \Bigg|_{s = j\left(\omega_a - k\frac{2\pi}{T}\right)} \tag{6.8}$$

Da desigualdade triangular $\|f + g\| \le \|f\| + \|g\|$, onde f e g são funções contínuas hipotéticas, teremos para (6.7):

$$\frac{1}{T}\left\|\sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty}\Delta G_k\right\|_{\infty} \leq \frac{1}{T}\sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty}\left\|\Delta G_k\right\|_{\infty} \tag{6.9}$$

A desigualdade acima permite investigar a convergência da série do falseamento por meio da série disposta do lado direito, i.e., caso a série do lado direito da desigualdade convergir então a série do lado esquerdo convergirá.

Estabeleçamos agora um segundo majorante real M_1:

$$\frac{1}{T}\sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty}\left\|\Delta G_k\right\|_{\infty} < M_1 \tag{6.10}$$

onde $M_0 < M_1$. Daí,

$$\frac{1}{T}\sum_{\substack{k\neq 0\\ k=-\infty}}^{+\infty}\left\|\frac{1-e^{-sT}}{s}.G(s)\Big|_{s=j\left(\omega_a-k\frac{2\pi}{T}\right)}\right\|_{\infty} < M_1 \tag{6.11}$$

pela desigualdade de Cauchy-Schwartz (ou de Kantorovich, ou de Buniakovsky), $\|f.g\| \leq \|f\|.\|g\|$, temos:

$$\frac{1}{T}\sum_{\substack{k\neq 0\\ k=-\infty}}^{+\infty}\left\|\frac{1-e^{-sT}}{s}.G(s)\Big|_{s=j\left(\omega_a-k\frac{2\pi}{T}\right)}\right\|_{\infty} \leq \frac{1}{T}\sum_{\substack{k\neq 0\\ k=-\infty}}^{+\infty}\left\|\frac{1-e^{-sT}}{s}\Big|_{s=j\left(\omega_a-k\frac{2\pi}{T}\right)}\right\|_{\infty}.\left\|G(s)\Big|_{s=j\left(\omega_a-k\frac{2\pi}{T}\right)}\right\|_{\infty} \tag{6.12}$$

Daí temos,

$$\frac{1}{T}\sum_{\substack{k\neq 0\\ k=-\infty}}^{+\infty}\left\|\frac{1-e^{-sT}}{s}\Big|_{s=j\left(\omega_a-k\frac{2\pi}{T}\right)}\right\|_{\infty}.\left\|G(s)\Big|_{s=j\left(\omega_a-k\frac{2\pi}{T}\right)}\right\|_{\infty} < M_2 \tag{6.13}$$

onde M_2 é outro majorante que satisfaz $M_1 < M_2$.

Numericamente pode-se mostrar que $\left\| \frac{1-e^{-sT}}{s}\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty$ equivale ao valor do período de amostragem, como ilustrado na Figura abaixo para um hipotético valor de T = 2 segundos, tomado como exemplo. Exaustivas simulações foram efetuadas buscando confirmar tal equivalência.

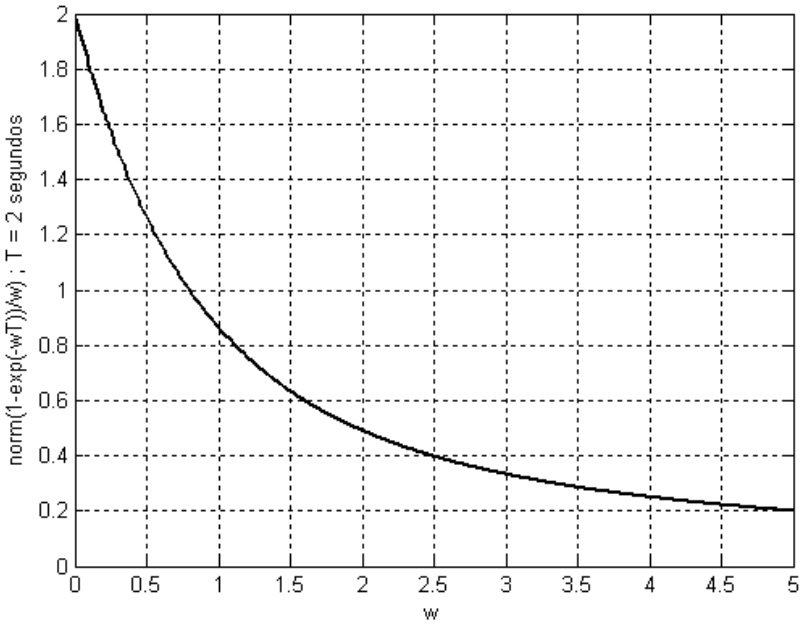


Figura 6.1 – Prova da equivalência da norma "sup" do segurador-amostrador ("*sampler-hold*") ao período de amostragem T, no caso igual a 2 segundos.

Assim,

$$\left\| \frac{1-e^{-sT}}{s}\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty := T \qquad ,\forall \omega \in [0,\infty) \tag{6.14}$$

Daí,

$$\frac{1}{T}.T\sum_{\substack{k\neq 0\\ k=-\infty}}^{+\infty}.\left\| G(s)\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty < M_2 \tag{6.15}$$

$$\sum_{\substack{k\neq 0\\ k=-\infty}}^{+\infty} . \left\| G(s)\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty < M_2 \tag{6.16}$$

Como $M_0 < M_1 < M_2$ pode-se concluir que,

$$\left\| \frac{1}{T} \sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty} \frac{1-e^{-sT}}{s} . G(s)\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty < \sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty} \left\| G(s)\Big|_{s=j\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty \tag{6.17}$$

□

O seguinte Corolário é devido ao autor desta Tese.

<u>Corolário 6.1</u>: Caso possamos rescrever G, uma função racional, própria e estável como $G(s) = \frac{N(s)}{D(s)}$ teremos,

$$\left\| \frac{1}{T} \sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty} \frac{1-e^{-sT}}{s} . \frac{N\left(\omega_a - k\frac{2\pi}{T}\right)}{D\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty < \sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty} \left\| \frac{N\left(\omega_a - k\frac{2\pi}{T}\right)}{D\left(\omega_a - k\frac{2\pi}{T}\right)} \right\|_\infty \tag{6.18}$$

□

<u>Prova</u>:

A prova se dá por trivial substituição de $G(s)$ por $\frac{N(s)}{D(s)}$ no Teorema 6.1.

□

<u>Exemplo 1</u>: se $G(s) = \frac{1}{Js}$, onde *J* é o momento de inércia polar (feito unitário por simplicidade), analisemos a convergência do falseamento gerado por um processo de amostragem cujo período de amostragem é T,

$$\left\| \frac{1}{T} \sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty} \Delta G_{*_T} \right\|_\infty < \sum_{\substack{k\neq 0\\ k=-\infty}}^{\infty} \left\| \frac{1}{\omega - k\frac{2\pi}{T}} \right\|_\infty \tag{6.19}$$

converge absolutamente. Lembrando do teste da razão dos termos da série temos que para uma dada série $\sum f_k$ com $\lim_{k\to\infty}\frac{|f_{k+1}|}{|f_k|}=L$ converge absolutamente se L<1 e diverge se L>1 ou se L tender ao infinito.

Assim,

$$L=\lim_{k\to\infty}\frac{\left|\frac{1}{\omega-(k+1)\frac{2\pi}{T}}\right|}{\left|\frac{1}{\omega-k\frac{2\pi}{T}}\right|}<1 \qquad (6.20)$$

o que foi mostrado numericamente para qualquer T > 0.

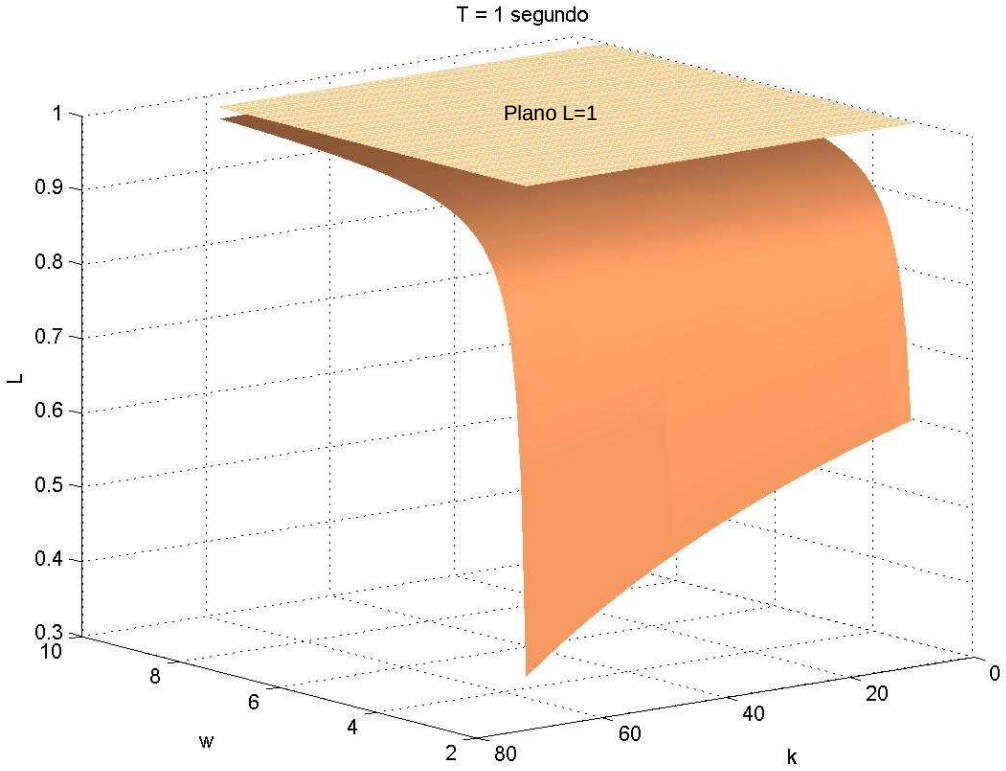


Figura 6.2 – Teste de convergência pelo método da razão, em função de k e ω, para T = 1 segundo, como exemplo, onde L < 1.

Exemplo 2: se $G(s) = \frac{1}{s+a}$, onde *a* é uma constante real não nula (no caso a = 2), analisemos a convergência do falseamento gerado por um processo de amostragem cujo período de amostragem é T,

$$\left\| \frac{1}{T} \sum_{\substack{k \neq 0 \\ k=-\infty}}^{\infty} \Delta G_{*T} \right\|_{\infty} < \sum_{\substack{k \neq 0 \\ k=-\infty}}^{\infty} \left\| \frac{1}{\omega - k\frac{2\pi}{T} + a} \right\|_{\infty} \tag{6.21}$$

$$L = \lim_{k \to \infty} \frac{\left| \dfrac{1}{\omega - (k+1)\dfrac{2\pi}{T} + a} \right|}{\left| \dfrac{1}{\omega - k\dfrac{2\pi}{T} + a} \right|} < 1 \qquad (6.22)$$

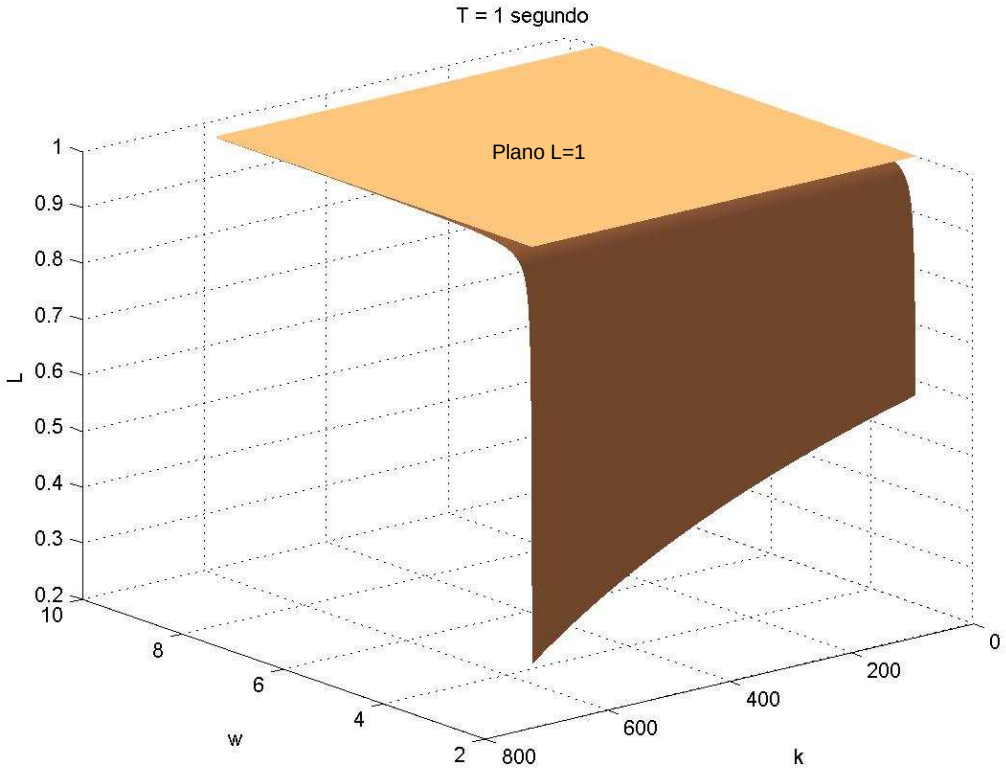


Figura 6.3 – Teste de convergência pelo método da razão para T = 1 segundo obtivemos L < 1.

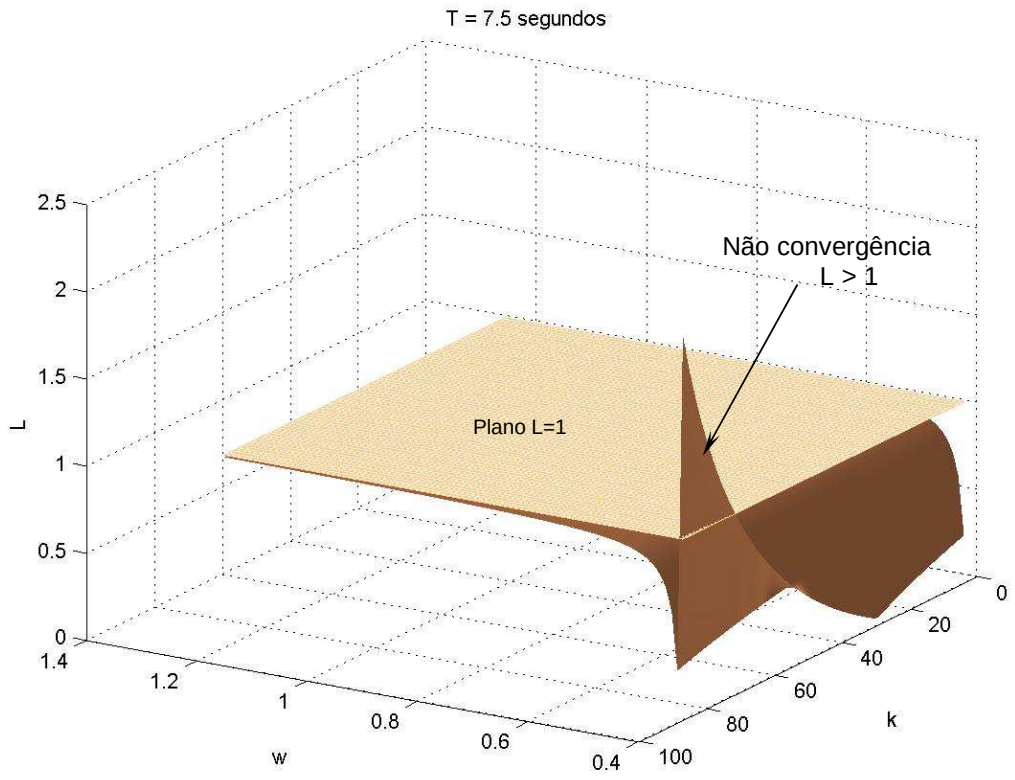


Figura 6.4 – Teste de convergência pelo método da razão para T = 7.5 segundos obtivemos L > 1, isto é, a série do falseamento diverge.

Verificou-se que para valores de T superiores a 7,5 segundos L > 1, i.e., a série do falseamento diverge, o que pode-se dizer que o sistema de controle em tempo discreto da planta analógica tornou-se instável devido à perturbação divergente do falseamento.

Simulações considerando grandes varreduras de ω e k para o caso instável geraram oscilações periódicas com grandes valores de pico acima do plano unitário (L = 1).

Exemplo 3: se $G(s)=\frac{s+a}{s+b}$, onde *a* e *b* são constantes reais não nulas e a $\neq$ b, analisemos a convergência do falseamento gerado por um processo de amostragem cujo período de amostragem é T,

$$L = \lim_{k\to\infty} \frac{\left|\dfrac{\omega-(k+1)\dfrac{2\pi}{T}+a}{\omega-(k+1)\dfrac{2\pi}{T}+b}\right|}{\left|\dfrac{\omega-k\dfrac{2\pi}{T}+a}{\omega-k\dfrac{2\pi}{T}+b}\right|} < 1 \qquad (6.23)$$

Neste caso, para certos a e b testados numericamente, o falseamento mostrou-se ser absolutamente convergente se e somente se a < b e divergente, em caso contrário, i.e., a > b. Utilizando a condição estruturalmente estável a < b a instabilidade ocorreu para T ≥ 5 segundos. As simulações mostradas nas Figuras (5.5) e (5.6) ilustram os dois casos.

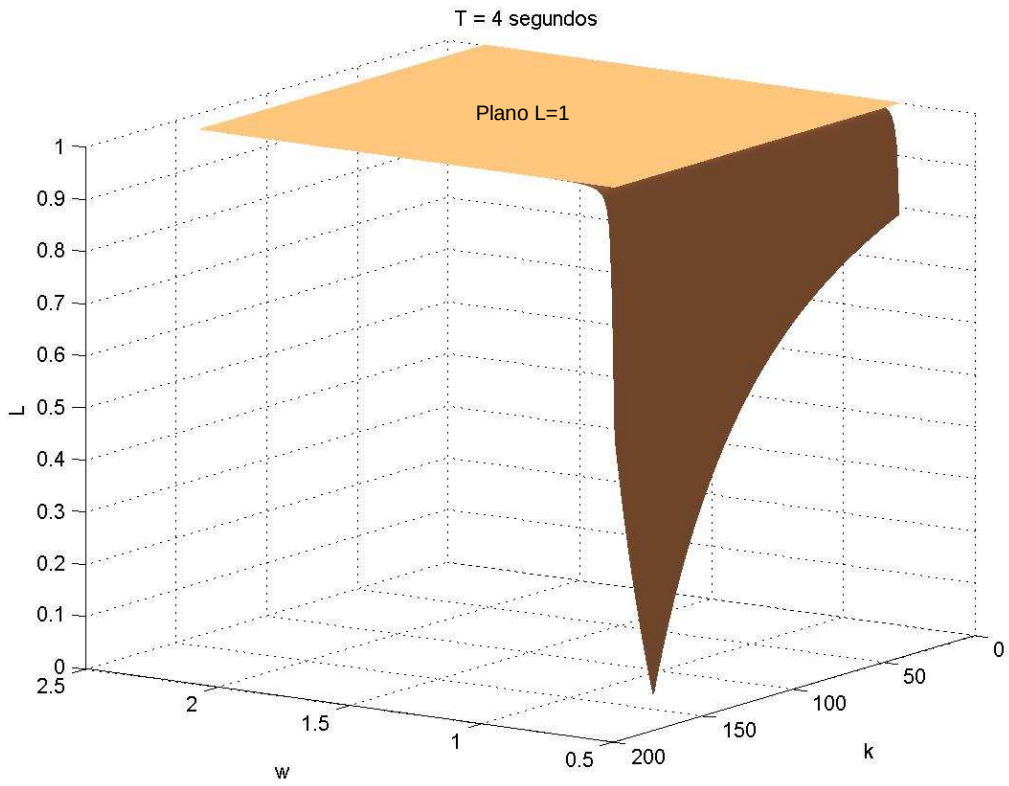


Figura 6.5 – Teste de convergência com T = 4 segundos para a < b.

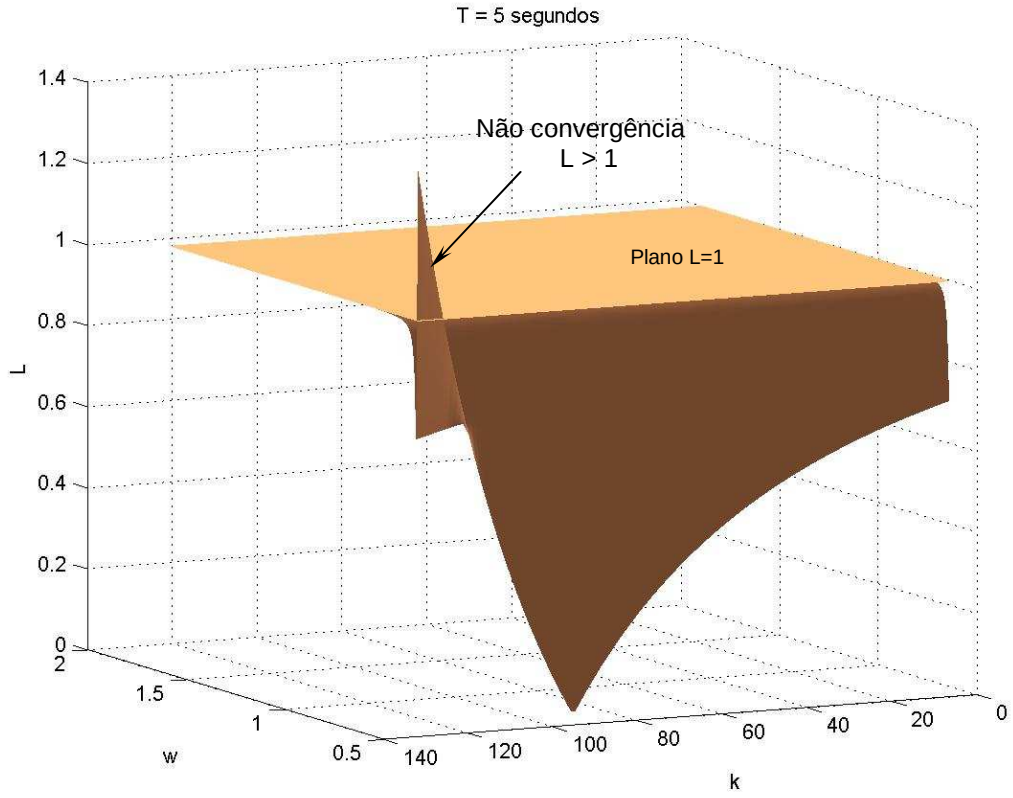


Figura 6.6 – Teste de convergência com T = 5 segundos, para a < b.

Exemplo 4: se $G(s)=\frac{1}{s^2+as+b}$, onde *a* e *b* são constantes reais não nulas, analisemos a convergência do falseamento gerado por um processo de amostragem cujo período de amostragem é T,

$$L=\lim_{k\to\infty}\frac{\left|\frac{1}{\left(\omega-(k+1)\frac{2\pi}{T}\right)^2+a.\left(\omega-(k+1)\frac{2\pi}{T}\right)+b}\right|}{\left|\frac{1}{\left(\omega-k\frac{2\pi}{T}\right)^2+a.\left(\omega-k\frac{2\pi}{T}\right)+b}\right|}<1 \qquad (6.24)$$

é absolutamente convergente, independendo do período de amostragem T e dos parâmetros a e b.

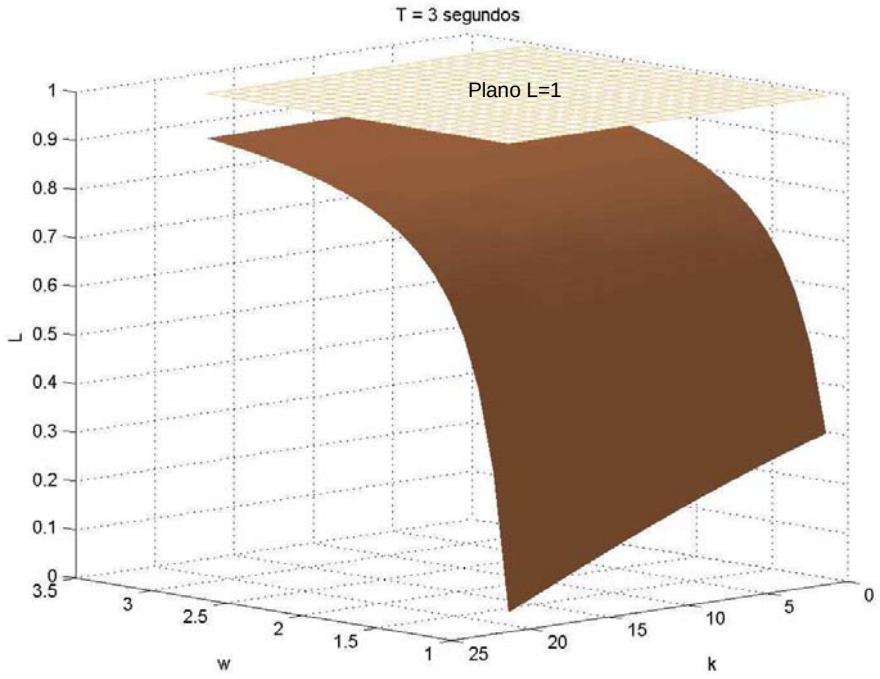


Figura 6.7 – Teste de convergência do falseamento função de do período de amostragem T e da planta $G(s) = \frac{1}{s^2 + as + b}$.

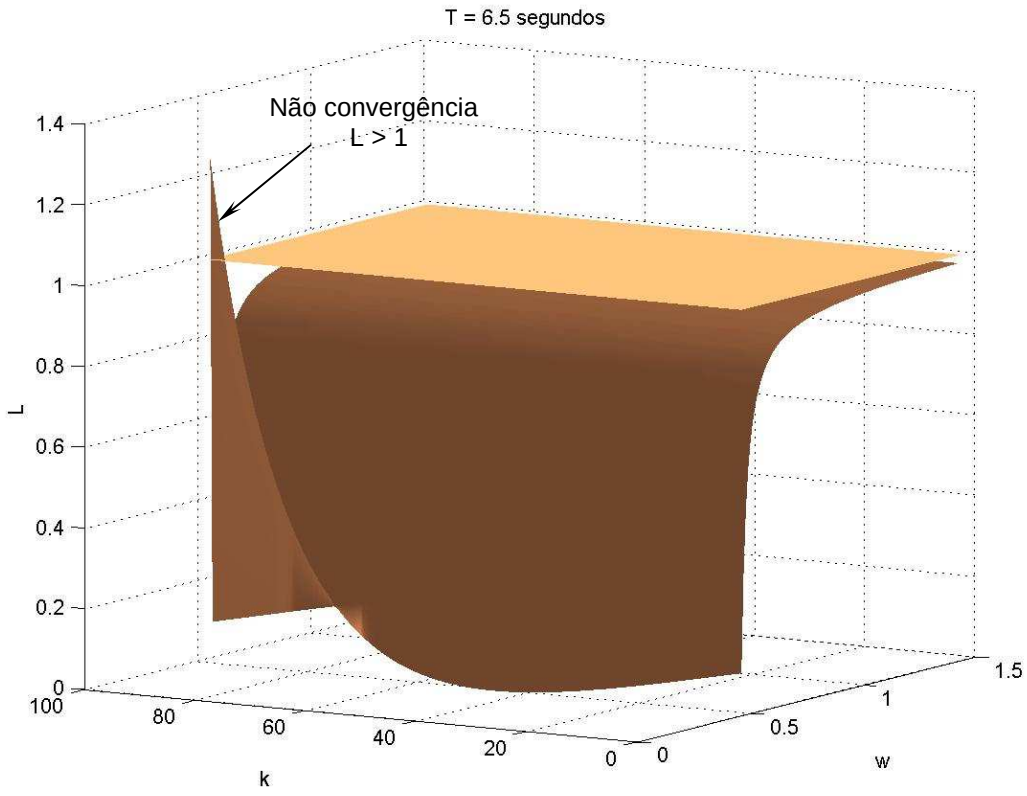


Figura 6.8 – Teste de convergência com $G(s)=\frac{1}{s^2+as+b}$ mostrando que para T=6,5 segundos L > 1.

Portanto, o falseamento (“aliasing”), o qual depende do período de amostragem T e da planta analógica G usada, pode divergir, instabilizando o sistema de controle discreto de tal planta analógica. Uma alternativa ao método da razão aqui utilizado seria utilizar o teste de Cauchy, enunciado abaixo.

Lema 6.1 (**Naylor, 1982**)**:** (*Teste de Cauchy*) seja X um espaço de Banach. Uma série infinita $\sum_{k=1}^{+\infty} x_k$ converge em X se e somente se para cada $\varepsilon > 0$ existe um inteiro N tal que,

$$\left\| \sum_{j=m}^{n} x_j \right\| \leq \varepsilon \qquad (6.25)$$

sempre que, $n \geq m > N$.

Prova: a prova deste Lema é trivial segundo **Naylor (1982)**, visto que ele meramente estabelece que a série converge se e somente se a seqüência de somas parciais é uma seqüência de Cauchy. Uma seqüência de Cauchy $\{x_m\}$ num espaço métrico (X,d) é dita ser uma seqüência de Cauchy se para cada $\varepsilon > 0$ existe um N tal que $d(x_n,x_m) \leq \varepsilon$ para quaisquer escolha de $n,m \geq N$, sendo que N depende de ε. Outra forma de explicar o que é uma seqüência de Cauchy é por meio do limite:

$$\lim_{n,m\to\infty} d(x_n, x_m) = 0 \qquad (6.26)$$

CAPÍTULO 7

TEOREMAS DA ESTABILIDADE DO SISTEMA DE CONTROLE ANALÓGICO-DIGITAL

Neste capítulo será usada a teoria da estabilidade de Liapunov para ajudar a calcular as regiões de instabilidade e estabilidade que aparecem num sistema de controle discreto de uma planta analógica.

7.1 – Introdução.

Como mostrado em **Balas (1982)** uma classe de sistemas com parâmetros distribuídos, os quais podem incluir estruturas flexíveis, podem ser descritos por

$$\dot{v} = Av + Bu \tag{7.1}$$

com v(0) = v_o ; e u=-QB'v sendo o "*Direct Velocity Feedback Control*" (DVFB) o que dá,

$$\dot{v} = A_c v \tag{7.2}$$

onde, $A_c = A - BQB'$ sendo Q positiva-definida.

Usando mapeamento "backward" obtemos a expressão equivalente discreta para tal sistema de controle, conforme veremos abaixo. Como ensina **Franklin (1981)**, a equivalência "backward" é dada por,

$$z = \frac{1}{1 - T.s} \tag{7.3}$$

ou

$$s = \frac{1 - z}{T.z} \tag{7.4}$$

$$s = \frac{z^{-1} - 1}{T} \tag{7.5}$$

Partindo da equação acima como aproximação discreta do operador derivada podemos reescrever,

$$\left(\frac{z^{-1} - 1}{T}\right).V(z) = A_c V(z) \tag{7.6}$$

$$\left(\frac{1 - z^{-1}}{T.z^{-1}}\right).V(z) = A_c V(z) \tag{7.7}$$

$$\left(1 - z^{-1}\right).V(z) = T.A_c.z^{-1}.V(z) \tag{7.8}$$

$$V(z) - z^{-1}V(z) = T.A_c.z^{-1}V(z) \tag{7.9}$$

Aplicando a Transformada-Z inversa acima teremos,

$$v_k - v_{k-1} = T.A_c.v_{k-1} \tag{7.10}$$

$$v_k = v_{k-1} + T.A_c.v_{k-1} \tag{7.11}$$

Apenas fazendo um truque matemático simples onde substituimos k por k+1 obtemos,

$$v_{k+1} = v_k + T.A_c.v_k \tag{7.12}$$

E usando uma função de Liapunov que tem aspecto de “energia do sistema” dada por,

$$V(v_k) = \tfrac{1}{2} v_k^{'} P v_k \tag{7.13}$$

onde P é uma matriz positiva-definida, temos,

$$V(v_k) = \tfrac{1}{2} v_k^{'} P v_k \tag{7.14}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_{k+1}^{'} P v_{k+1} \tag{7.15}$$

Utilizando a expressão obtida para v_{k+1} temos o seguinte desenvolvimento algébrico,

$$V(v_{k+1}) = \tfrac{1}{2}\left[v_k + TA_c v_k\right]^{'} P\left[v_k + TA_c v_k\right] \tag{7.16}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_k^{'} \left(I + TA_c\right)^{'} P\left(I + TA_c\right) v_k \tag{7.17}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_k^{'} \left(I + TA_c^{'}\right) P\left(I + TA_c\right) v_k \tag{7.18}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_k^{'} \left(P + TA_c^{'} P\right)\left(I + TA_c\right) v_k \tag{7.19}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_k^{'} \left(P + TA_c^{'} P + TPA_c + T^2 A_c^{'} PA_c\right) v_k \tag{7.20}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_k^{'} \left(P + T\left[A_c^{'} P + PA_c\right] + T^2 A_c^{'} PA_c\right) v_k \tag{7.21}$$

Observando o termo entre colchetes acima percebemos que podemos fazer uso da equação de Liapunov apresentada em **Balas (1982)**,

$$A_c^{'} P + PA_c = -Q \tag{7.22}$$

daí,

$$V(v_{k+1}) = \tfrac{1}{2} v_k^{'} \left(P - T.Q + T^2 A_c^{'} PA_c\right) v_k \tag{7.23}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_k^{'} P v_k - \frac{T}{2} v_k^{'} \left(Q - TA_c^{'} PA_c\right) v_k \tag{7.24}$$

A taxa de variação dessa função de Liapunov é dada por,

$$\Delta V_k = V\left(v_{k+1}\right) - V\left(v_k\right) \tag{7.25}$$

o que dá,

$$\Delta V_k = -\tfrac{T}{2} v_k' \left[Q - TA_c' P A_c \right] v_k \tag{7.26}$$

Onde temos que ter $\Delta V_k < 0$ para a estabilidade assintótica, i.e., para que a "energia" do sistema esteja se reduzindo, tal como pode ser resumido no seguinte Lema:

Lema 7.1 (Balas, 1982): o sistema em tempo-discreto v_{k+1} permanece estável enquanto for satisfeita a condição

$$T < \frac{\lambda_{\min}(Q)}{\lambda_{\max}(A_c' P A_c)} \tag{7.27}$$

□

Onde λ são os autovalores.

Prova: esta prova foi elaborada pelo autor desta Tese visto que a mesma não foi apresentada em **Balas (1982)**,

De (6.26) temos,

$$-v_k' Q v_k + v_k' \left(TA_c' P A_c \right) v_k < 0 \tag{7.28}$$

Tal como ensinado em **Mostofi (2008)**, o primeiro termo pode ser reescrito por meio da desigualdade de Rayleigh-Ritz e o segundo termo reescrito por meio da norma induzida e aplicando-se também a desigualdade de Cauchy-Schwartz,

$$-\left| v_k' Q v_k \right| + \left| v_k' \left(TA_c' P A_c \right) v_k \right| \le -\lambda_{\min}(Q) \left\| v_k \right\|_\infty^2 + \left\| v_k' \right\|_\infty \left\| TA_c' P A_c \right\|_\infty \left\| v_k \right\|_\infty \tag{7.29}$$

Daí,

$$-\left| v_k' Q v_k \right| + \left| v_k' \left(TA_c' P A_c \right) v_k \right| \le -\lambda_{\min}(Q) \left\| v_k \right\|_\infty^2 + \left\| v_k \right\|_\infty^2 \left\| TA_c' P A_c \right\|_\infty \tag{7.30}$$

Voltando à desigualdade (7.28),

$$-\lambda_{\min}(Q)\|v_k\|_\infty^2 + \|v_k\|_\infty^2 \left\|TA_c^{'}PA_c\right\|_\infty < 0 \tag{7.31}$$

$$-\lambda_{\min}(Q) + \left\|TA_c^{'}PA_c\right\|_\infty < 0 \tag{7.32}$$

Como $\lambda_{\max}(\cdot) \le \|\cdot\|_\infty$ (Oden, 1979) podemos escrever,

$$-\lambda_{\min}(Q) + \lambda_{\max}\left(TA_c^{'}PA_c\right) < 0 \tag{7.33}$$

$$-\lambda_{\min}(Q) + T.\lambda_{\max}\left(A_c^{'}PA_c\right) < 0 \tag{7.34}$$

O que resulta em,

$$\mathrm{T} < \frac{\lambda_{\min}(\mathrm{Q})}{\lambda_{\max}(\mathrm{A_c^{'}PA_c})} \tag{7.35}$$

□

Caso usemos a regra de Tustin como mapeamento s-z podemos obter uma forma mais precisa e geral para estudar a estabilidade de sistemas de controle discretos de plantas analógicas e isso é descrito por meio do seguinte teorema. Curiosamente, neste Teorema o cálculo inicialmente realizado com a regra de Tustin levou a uma generalização permitindo o emprego da exponencial de matriz e^{AcT} como podemos ver no desenvolvimento da prova do mesmo.

Cálculos visando utilizar uma regra mais geral que a de “backward”, usada por **Balas (1982)**, tal como foi o caso da regra de Tustin, levaram acidentalmente à matriz de transição e^{AcT}. Entretanto, tais cálculos foram apenas um exercício algébrico visto que bastaria aplicar a matriz de transição para elaborar o Teorema 7.1.

□

Tal como ensinado por **Franklin (1981)** a equivalência analógico-digital usando a regra de Tustin é dada por,

$$z = \frac{1 + \frac{T}{2}.s}{1 - \frac{T}{2}.s} \tag{7.36}$$

ou

$$s = \frac{2}{T}.\frac{z-1}{z+1} \tag{7.37}$$

A equação $\dot{v} = A_c v$ mapeada pela regra de Tustin é então processada da seguinte forma,

$$\frac{2}{T}\left(\frac{z-1}{z+1}\right).V(z) = A_c V(z) \tag{7.38}$$

$$\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right).V(z) = A_c V(z) \tag{7.39}$$

$$\frac{2}{T}\left(1-z^{-1}\right).V(z) = \left(1+z^{-1}\right)A_c V(z) \tag{7.40}$$

aplicando a Transformada-Z inversa teremos, após substituir k por k+1,

$$\tfrac{2}{T}\left(v_{k+1} - v_k\right) = A_c\left(v_{k+1} + v_k\right) \tag{7.41}$$

$$v_{k+1} = \left[\frac{2I - TA_c}{T}\right]^{-1}\left[\frac{2I + TA_c}{T}\right]v_k \tag{7.42}$$

$$v_{k+1} = \left[2I - TA_c\right]^{-1}\left[2I + TA_c\right]v_k \tag{7.43}$$

Fazendo agora uso de uma aproximação por Padé (ver Apêndice G) para a função exponencial teremos o que segue,

$$e^{A_c T} = \left[e^{-\frac{T}{2}A_c}\right]^{-1}\left[e^{\frac{T}{2}A_c}\right] = \left[I - A_c\frac{T}{2} + \ldots\right]^{-1}\left[I + A_c\frac{T}{2} + \ldots\right] \tag{7.44}$$

$$e^{A_c T} \approx \left[I - A_c \frac{T}{2} \right]^{-1} \left[I + A_c \frac{T}{2} \right] \tag{7.45}$$

Assim, verifica-se ser possível reconstruir a forma racional apresentada pela expressão exponencial exata,

$$v_{k+1} = e^{A_c T} v_k \tag{7.46}$$

O seguinte teorema devido ao autor desta Tese.

Teorema 7.1: o sistema discreto $v_{k+1} = e^{A_c T} v_k$ permanecerá estável enquanto a seguinte condição for satisfeita

$$\lambda_{\min}(P) > \lambda_{\max}((e^{A_c T})' P e^{A_c T}) \tag{7.47}$$

□

Prova:

Assuma P > 0 na função de Liapunov abaixo,

$$V(v_k) = \tfrac{1}{2} v_k' P v_k \tag{7.48}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_{k+1}' P v_{k+1} \tag{7.49}$$

$$V(v_{k+1}) = \tfrac{1}{2} v_k' \left(e^{A_c T}\right)' P e^{A_c T} v_k \tag{7.50}$$

Daí, podemos escrever a expressão da diferença,

$$\Delta V_k = \tfrac{1}{2} v_k' \left(e^{A_c T}\right)' P e^{A_c T} v_k - \tfrac{1}{2} v_k' P v_k \tag{7.51}$$

$$\Delta V_k = -\tfrac{1}{2} v_k' \left[P - \left(e^{A_c T}\right)' P e^{A_c T} \right] v_k < 0 \tag{7.52}$$

Similarmente ao desenvolvido na prova do Lema 7.1, para satisfazer a desigualdade $\Delta V_k < 0$ é necessário que o termo entre colchetes em (7.52) seja

positivo definido, logo para tal consideração é necessário que $\lambda_{\min}(P) > \lambda_{\max}((e^{A_c T})' P e^{A_c T})$.

□

Pode ser mostrado que se expandirmos a função exponencial matricial em seus termos lineares no Teorema 7.1 poderemos obter o resultado do Lema 7.1, mostrando assim que o Lema 7.1 (**Balas, 1982**) é um caso particular do Teorema 7.1.

A desigualdade (7.47) pode também ser expressa por meio do corolário 7.1, o que permite uma curiosa interpretação geométrica.

O seguinte Corolário é devido ao autor desta Tese.

Corolário 7.1: o sistema discreto $v_{k+1} = e^{A_c T} v_k$ permanecerá estável enquanto a seguinte condição for satisfeita,

$$\rho\left[(e^{A_c . T})' P e^{A_c . T}\right] < \min_{\lambda} P \qquad (7.53)$$

Onde ρ é o raio espectral da matriz entre colchetes.

□

Prova:

Do Teorema 7.1, reescrevemos,

$$\min_{\lambda} P > \max_{\lambda} (e^{A_c . T})' P e^{A_c . T} \qquad (7.54)$$

Tal como consta no Apêndice H e foi ensinado em **Oden (1979)** e também em **Sobolev (1963)**, **Lorch (1962)**, **Hislop (1991)**, **D'Ambrosio (1977)**, **Balakrishnan (1976)**, **Arveson (2002)**, **Naylor (1982)** e **Feintuch (1982)** o raio espectral é definido como,

$$\rho(\Xi) = \max_{\lambda} \Xi \qquad (7.55)$$

onde Ξ é uma matriz nxn e a operação $\max_{\lambda} \Xi$ se refere ao máximo módulo de autovalor de Ξ. Assim sendo,

$$\min_{\lambda} P > \rho\left[(e^{A_c.T})' P e^{A_c.T}\right] \tag{7.56}$$

ou

$$\rho\left[(e^{A_c.T})' P e^{A_c.T}\right] < \min_{\lambda} P \tag{7.57}$$

□

Por (7.54) temos uma interpretação geométrica desta teoria considerando o raio espectral tal como mostra a figura abaixo..

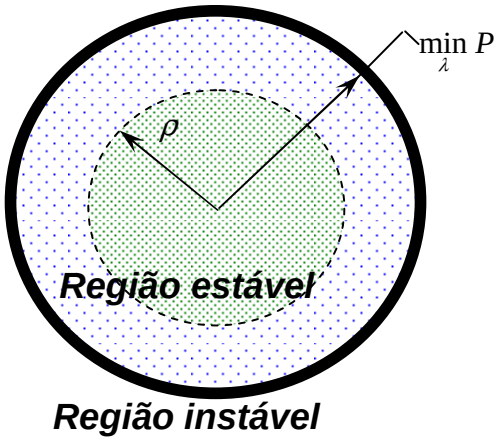


Figura 7.1: representação geométrica da estabilidade usando o conceito de raio espectral de $(e^{A_c.T})' P e^{A_c.T}$ onde T é o período de amostragem do sistema de controle digital, A_c é a matriz de dinâmica do sistema e P é uma matriz positiva-definida.

Se fizermos uso da regra ST1 obteremos o seguinte Teorema, devido também ao autor desta Tese:

Teorema 7.2: o sistema $\dot{v} = A_c v$, mapeamento pela regra ST1, permanecerá estável enquanto a seguinte condição for satisfeita,

$$\lambda_{\min}(P) > \lambda_{\max}\left\{\left[\frac{2}{T}I + \xi.A_c\right]'\left(\left[\frac{2}{T}I - A_c\right]^{-1}\right)'.P.\left[\frac{2}{T}I - A_c\right]^{-1}\left[\frac{2}{T}I + \xi.A_c\right]\right\} \quad (7.58)$$

sendo T o período de amostragem e $0 \le \xi \le 1$,.

□

Prova:

$$\frac{2}{T}\left(\frac{z-1}{z+\xi}\right).V(z) = A_c V(z) \quad (7.59)$$

$$\frac{2}{T}\left(\frac{1-z^{-1}}{1+\xi.z^{-1}}\right).V(z) = A_c V(z) \quad (7.60)$$

Para $0 \le \xi \le 1$. Similarmente, obteremos,

$$\tfrac{2}{T}\left(v_{k+1} - v_k\right) = A_c\left(v_{k+1} + \xi.v_k\right) \quad (7.61)$$

$$v_{k+1} = \left[\frac{2}{T}I - A_c\right]^{-1}\left[\frac{2}{T}I + \xi.A_c\right]v_k \quad (7.62)$$

$$V(v_k) = \tfrac{1}{2}v_k' P v_k \quad (7.63)$$

$$V(v_{k+1}) = \tfrac{1}{2}v_{k+1}' P v_{k+1} \quad (7.64)$$

$$V(v_{k+1}) = \frac{1}{2}\left(\left[\frac{2}{T}I - A_c\right]^{-1}\left[\frac{2}{T}I + \xi.A_c\right]v_k\right)'.P.\left[\frac{2}{T}I - A_c\right]^{-1}\left[\frac{2}{T}I + \xi.A_c\right]v_k \quad (7.65)$$

$$V(v_{k+1}) = \frac{1}{2}v_k'\left[\frac{2}{T}I + \xi.A_c\right]'\left(\left[\frac{2}{T}I - A_c\right]^{-1}\right)'.P.\left[\frac{2}{T}I - A_c\right]^{-1}\left[\frac{2}{T}I + \xi.A_c\right]v_k \quad (7.66)$$

$$\Delta V_k = V(v_{k+1}) - V(v_k) \tag{7.67}$$

$$\Delta V_k = \frac{1}{2} v_k^{'} \left[\frac{2}{T} I + \xi . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2}{T} I + \xi . A_c \right] v_k - \frac{1}{2} v_k^{'} P v_k \tag{7.68}$$

$$\Delta V_k = \frac{-1}{2} v_k^{'} \left\{ P - \left[\frac{2}{T} I + \xi . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2}{T} I + \xi . A_c \right] \right\} v_k \tag{7.69}$$

Para satisfazer a desigualdade $\Delta V_k < 0$,

$$\lambda_{\min}(P) > \lambda_{\max} \left\{ \left[\frac{2}{T} I + \xi . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2}{T} I + \xi . A_c \right] \right\} \tag{7.70}$$

□

O seguinte Corolário é devido ao autor desta Tese.

Corolário 7.2: o sistema $\dot{v} = A_c v$, mapeado pela regra ST1, permanecerá estável enquanto a seguinte condição for satisfeita,

$$\rho \left[\left[\frac{2}{T} I + \xi . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2}{T} I + \xi . A_c \right] \right] < \min_{\lambda} P \tag{7.71}$$

Onde ρ é o raio espectral da matriz entre colchetes.

□

Prova:

Por similaridade à prova do Corolário 7.1 tendo como base o Teorema 7.2, prova-se o Teorema 7.2.

□

Se fizermos uso da regra ST2 obteremos o seguinte Teorema da estabilidade, também devido ao autor desta Tese:

Teorema 7.3: o sistema $\dot{v} = A_c v$, mapeado pela regra ST2, permanecerá estável enquanto a seguinte condição for satisfeita, para $0 \le \xi_1 \le 1$ e $0 \le \xi_2 \le 1$,

$$\lambda_{\min}(P) > \lambda_{\max}\left\{\left[\frac{2\xi_1}{T}I + \xi_2.A_c\right]'\left(\left[\frac{2}{T}I - A_c\right]^{-1}\right)'.P.\left[\frac{2}{T}I - A_c\right]^{-1}\left[\frac{2\xi_1}{T}I + \xi_2.A_c\right]\right\} \tag{7.72}$$

sendo T o período de amostragem.

□

Prova: do equivalente discreto da derivada usando a regra ST2, já aplicando na equação diferencial de estado discreta no domínio da freqüência Z, temos:

$$\frac{2}{T}\left(\frac{1-\xi_1.z^{-1}}{1+\xi_2.z^{-1}}\right).V(z) = A_c V(z) \tag{7.73}$$

Para $0 \le \xi_1 \le 1$ e $0 \le \xi_2 \le 1$. Similarmente aos teoremas anteriores, obteremos,

$$\tfrac{2}{T}\left(v_{k+1} - \xi_1.v_k\right) = A_c\left(v_{k+1} + \xi_2.v_k\right) \tag{7.74}$$

$$v_{k+1} = \left[\frac{2}{T}I - A_c\right]^{-1}\left[\frac{2}{T}\xi_1 I + \xi_2.A_c\right]v_k \tag{7.75}$$

$$V(v_k) = \tfrac{1}{2}v_k' P v_k \tag{7.76}$$

$$V(v_{k+1}) = \tfrac{1}{2}v_{k+1}' P v_{k+1} \tag{7.77}$$

$$V(v_{k+1}) = \frac{1}{2} v_k^{'} \left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right] v_k \tag{7.78}$$

$$\Delta V_k = V(v_{k+1}) - V(v_k) \tag{7.79}$$

$$\Delta V_k = \frac{-1}{2} v_k^{'} \left\{ P - \left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right] \right\} v_k \tag{7.80}$$

Para satisfazer a desigualdade $\Delta V_k < 0$,

$$\lambda_{\min}(P) > \lambda_{\max} \left\{ \left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right] \right\} \tag{7.81}$$

□

Tal seguinte Corolário é devido ao autor desta Tese.

Corolário 7.3: o sistema $\dot{v} = A_c v$, mapeado pela regra ST2, permanecerá estável enquanto a seguinte condição for satisfeita,

$$\rho \left[\left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right]^{'} \left(\left[\frac{2}{T} I - A_c \right]^{-1} \right)^{'} . P . \left[\frac{2}{T} I - A_c \right]^{-1} \left[\frac{2\xi_1}{T} I + \xi_2 . A_c \right] \right] < \min_{\lambda} P \tag{7.82}$$

Onde ρ é o raio espectral da matriz entre colchetes.

□

Prova:

Por similaridade à prova do Corolário 7.1 tendo como base o Teorema 7.3.

□

7.2 – Resultados numéricos.

A seguir serão apresentados resultados numéricos baseados nos Teoremas da seção anterior onde as aproximações numéricas fizeram uso de aproximações por Padé de ordem elevada para garantir grande precisão no caso do Teorema 7.1.

Neste trabalho usamos Padé [3/3] (terceira ordem), Padé [6/6] (sexta ordem), Padé [8/8] (oitava ordem), Padé [10/10] (décima ordem), Padé [12/12] (décima segunda ordem), Padé [14/14] (décima quarta ordem) e Padé [16/16] (décima sexta ordem) para aproximar a exponencial de matriz $e^{A_c T}$ e foi suficiente para verificar a presença de regiões de instabilidade e estabilidade do sistema de controle discreto de planta analógica como função do período de amostragem T.

Exemplo 1: Como exemplo, usou-se uma matriz A_c na forma companheira tal como segue,

$$A_c = \begin{pmatrix} 0 & 1 \\ -3 & -5 \end{pmatrix} \tag{7.83}$$

Para calcular a matriz P positiva-definida usou-se a equação de Liapunov $A_cP+PA_c'=Q$; onde Q é outra matriz positiva-definida que no caso é dada por,

$$Q = \begin{pmatrix} 50 & 100 \\ 40 & 100 \end{pmatrix} \tag{7.84}$$

Partindo da equação de Liapunov usando (6.19) obtemos,

$$P = \begin{pmatrix} 1{,}4667 & -0{,}38 \\ -0{,}62 & 0{,}5 \end{pmatrix} \tag{7.85}$$

Os resultados numéricos para o Teorema 7.1 estão apresentados na Figura 7.2. Em tal figura podemos notar que à medida que aumentamos a precisão da aproximação por séries de Padé a curva do autovalor permanece por um tempo maior na faixa estável o que nos permite salientar que para a exponencial exata teríamos uma faixa instabilizante logo no começo e após isso teríamos a faixa

estabilizante. Entretanto, convém lembrar que não estamos aplicando expoenciais exatas em nossos microcontroladores e sim aproximações da exponencial, tal como é o caso das regras de Tustin, Schneider, “backward”, ST1 e ST2. Assim, a exemplo do que vemos na Figura 7.2 nota-se facilmente que para tais aproximações haverá uma região de instabilidade seguida de uma região de estabilidade e terminando numa região de instabilidade, em função do período de amostragem. O método é útil então para identificar estas regiões de estabilidade.

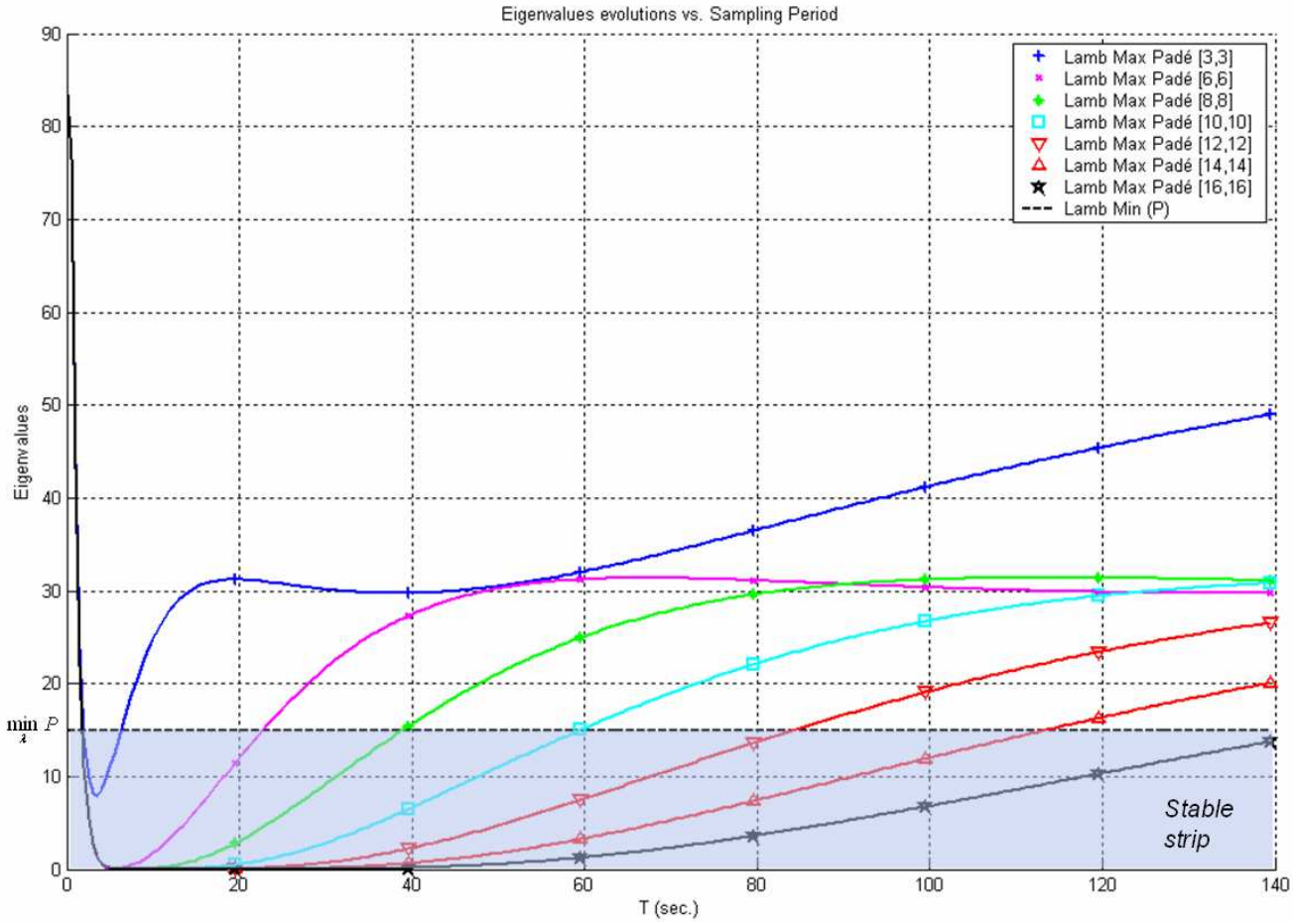


Figura 7.2: resultados numéricos do Teorema 7.1 apresentando regiões aproximadas de estabilidade/instabilidade como função do período de amostragem.

Quanto ao resultado numérico para o Teorema 7.2 podemos verificar na figura abaixo o raio espectral ρ (“Rho”) (ver Corolário 7.2) como função do período de

amostragem T e do parâmetro ξ que existe uma faixa de valores para ξ que garante a estabilidade do sistema de controle digital para valores de T superiores a 140 segundos, confirmando tal regra ST1 melhor do que a regra de Tustin.

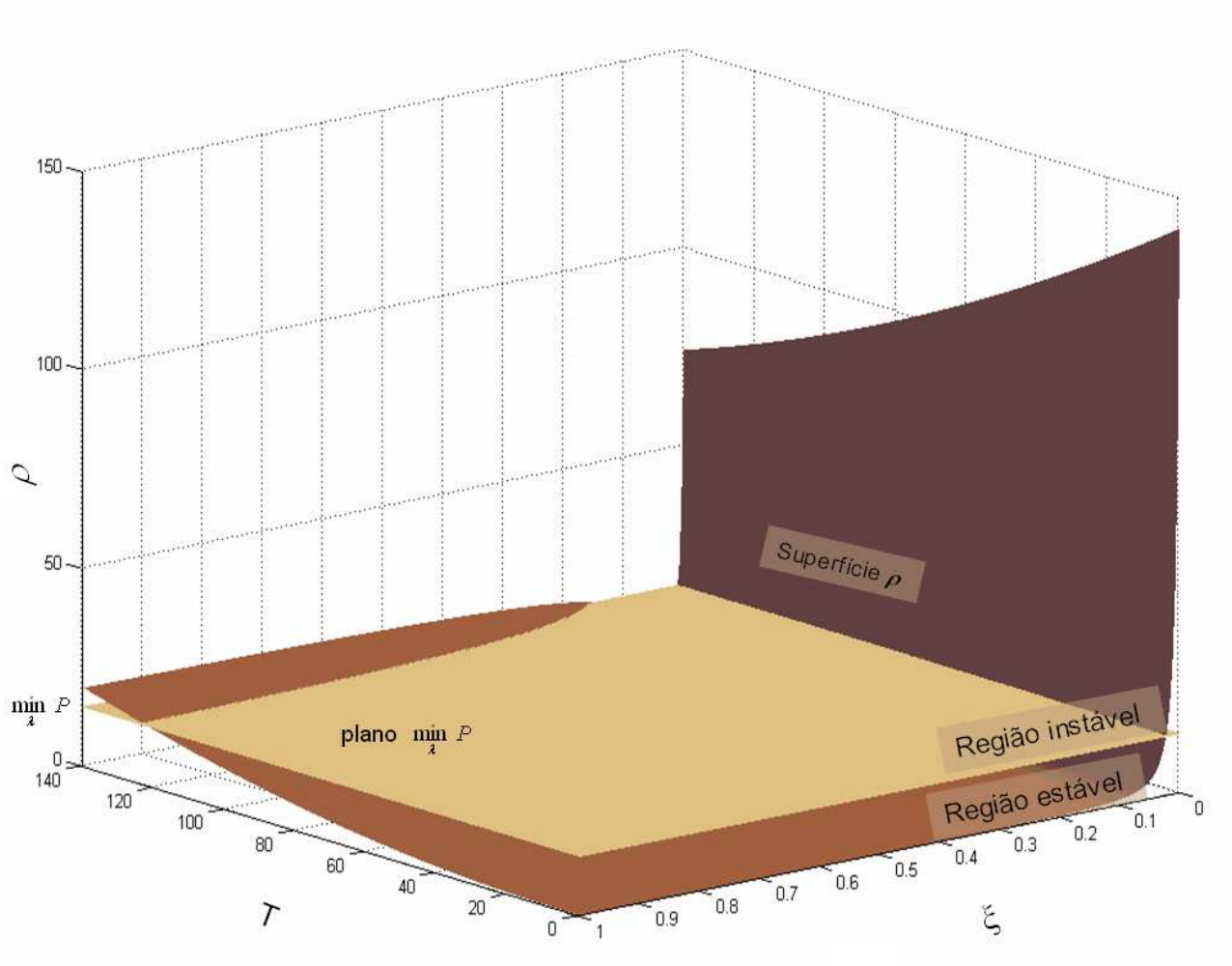


Figura 7.3: resultados numéricos do Teorema 7.2 apresentando regiões aproximadas de estabilidade/instabilidade como função do período de amostragem T e do parâmetro ξ.

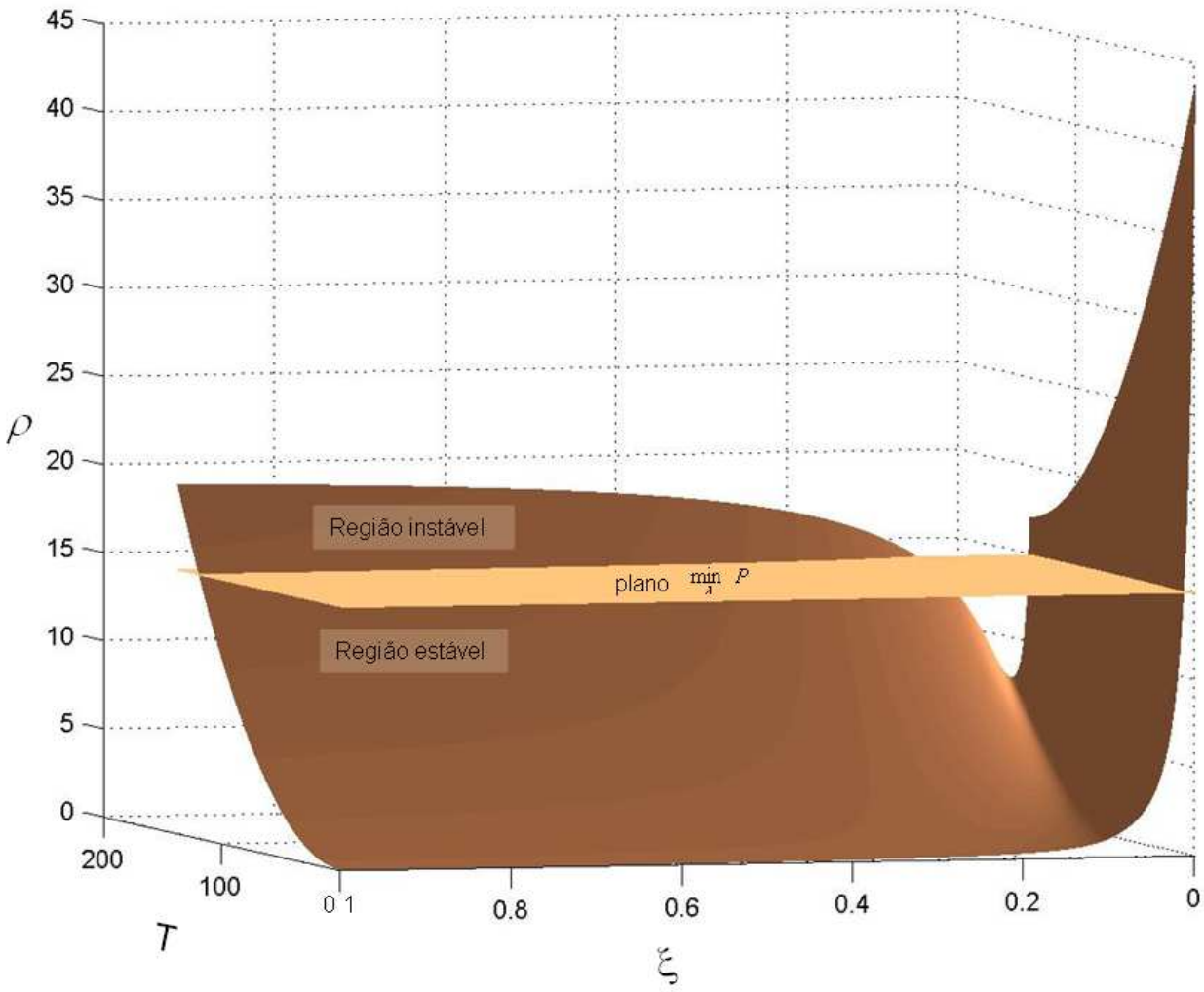


Figura 7.4: mesmo resultado numérico da figura anterior visualizado por ângulo diferente.

Podemos verificar na figura abaixo a vista superior do gráfico anterior mostrando com maior clareza as regiões de estabilidade e instabilidade.

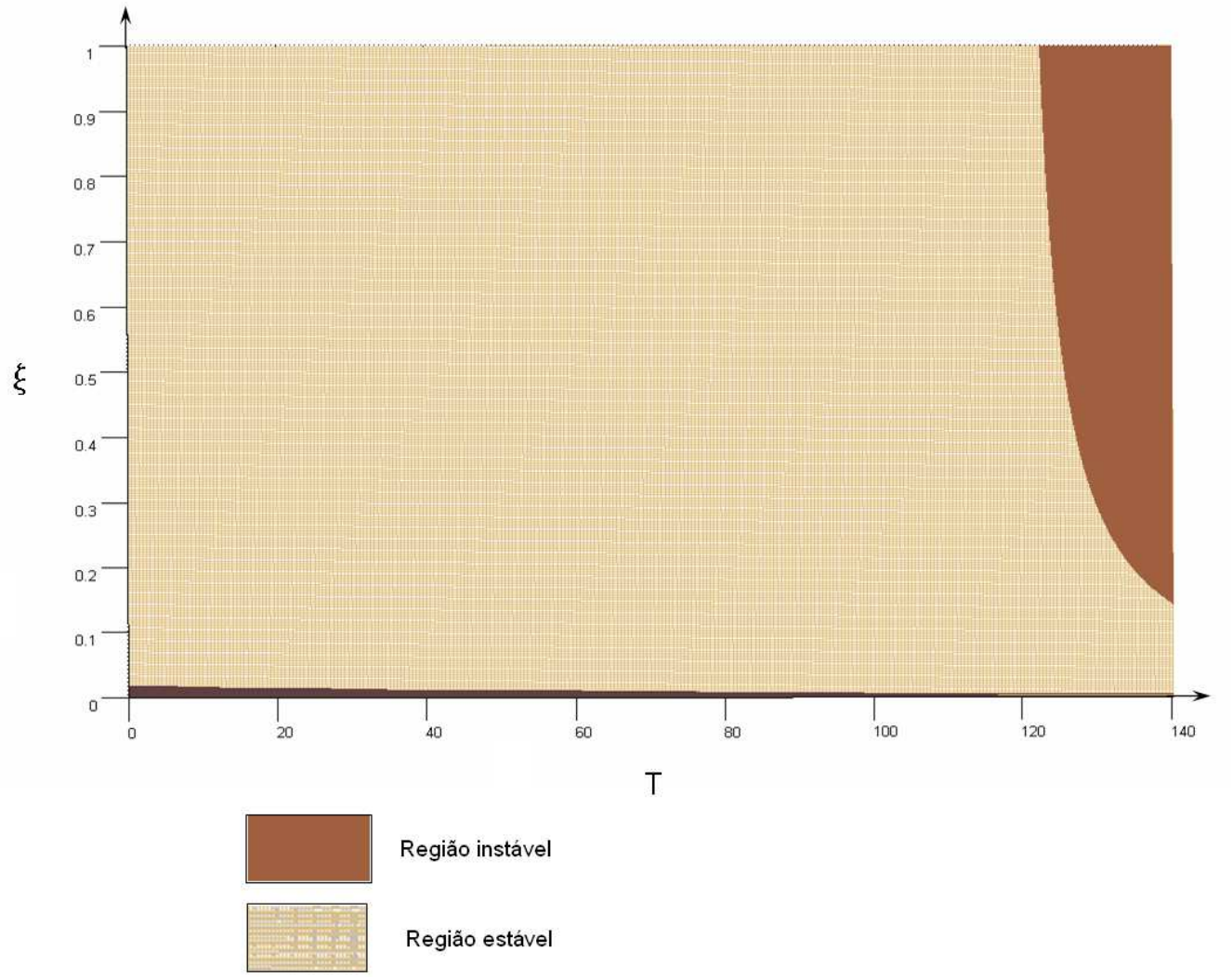


Figura 7.5: resultados numéricos do Teorema 7.2 apresentando regiões aproximadas de estabilidade/instabilidade como função do período de amostragem T e do parâmetro ξ (vista superior da figura anterior). O topo da figura representa a regra de Tustin (ξ=1) e a parte inferior da figura representa a regra "backward" (ξ=0).

Quanto ao resultado numérico para o Teorema 7.3 podemos verificar nas figuras abaixo o raio espectral ρ ("Rho") (ver Corolário 7.2) como função do período de amostragem T e dos parâmetros ξ_1 e ξ_2.

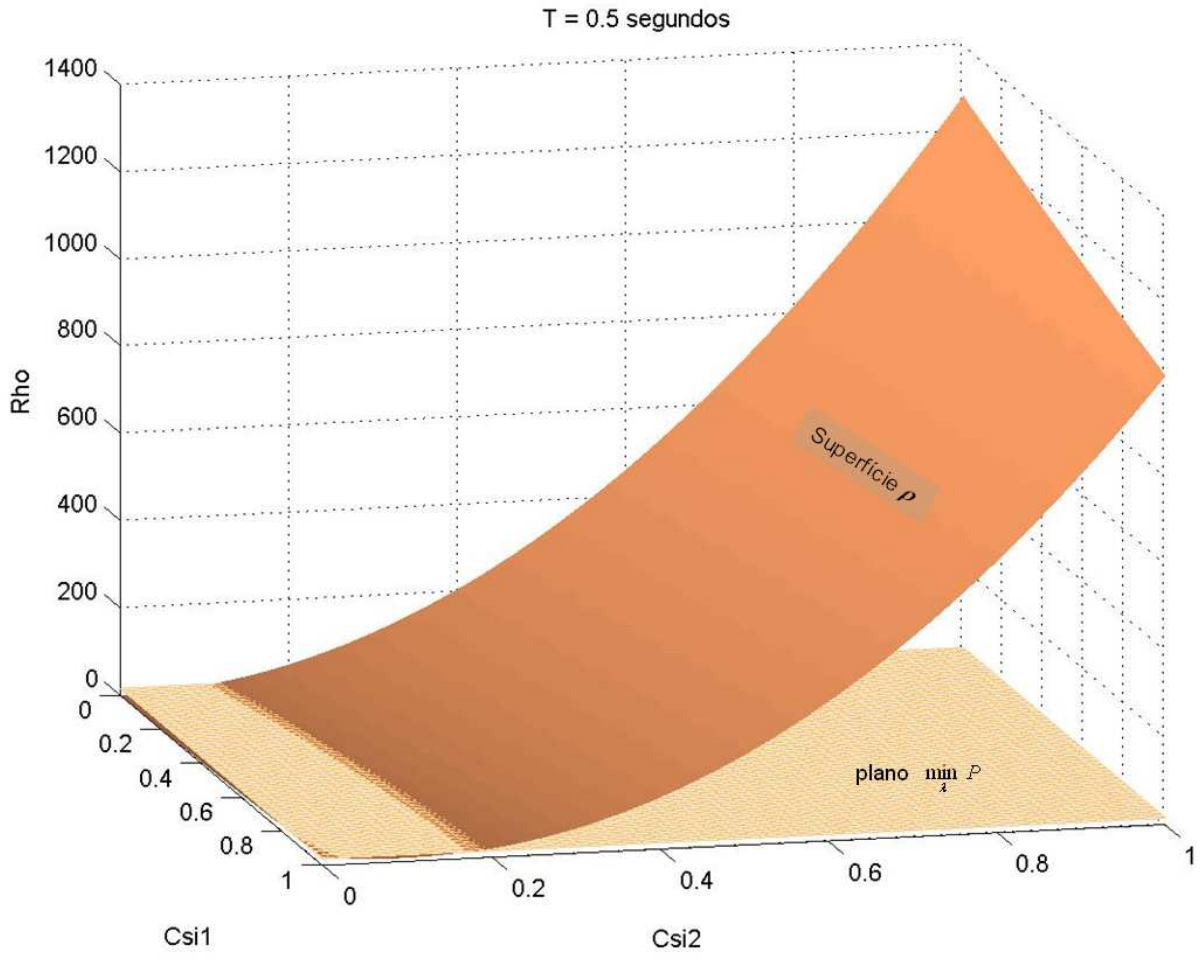


Figura 7.6: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 0.5 segundo em função dos parâmetros ξ_1 e ξ_2.

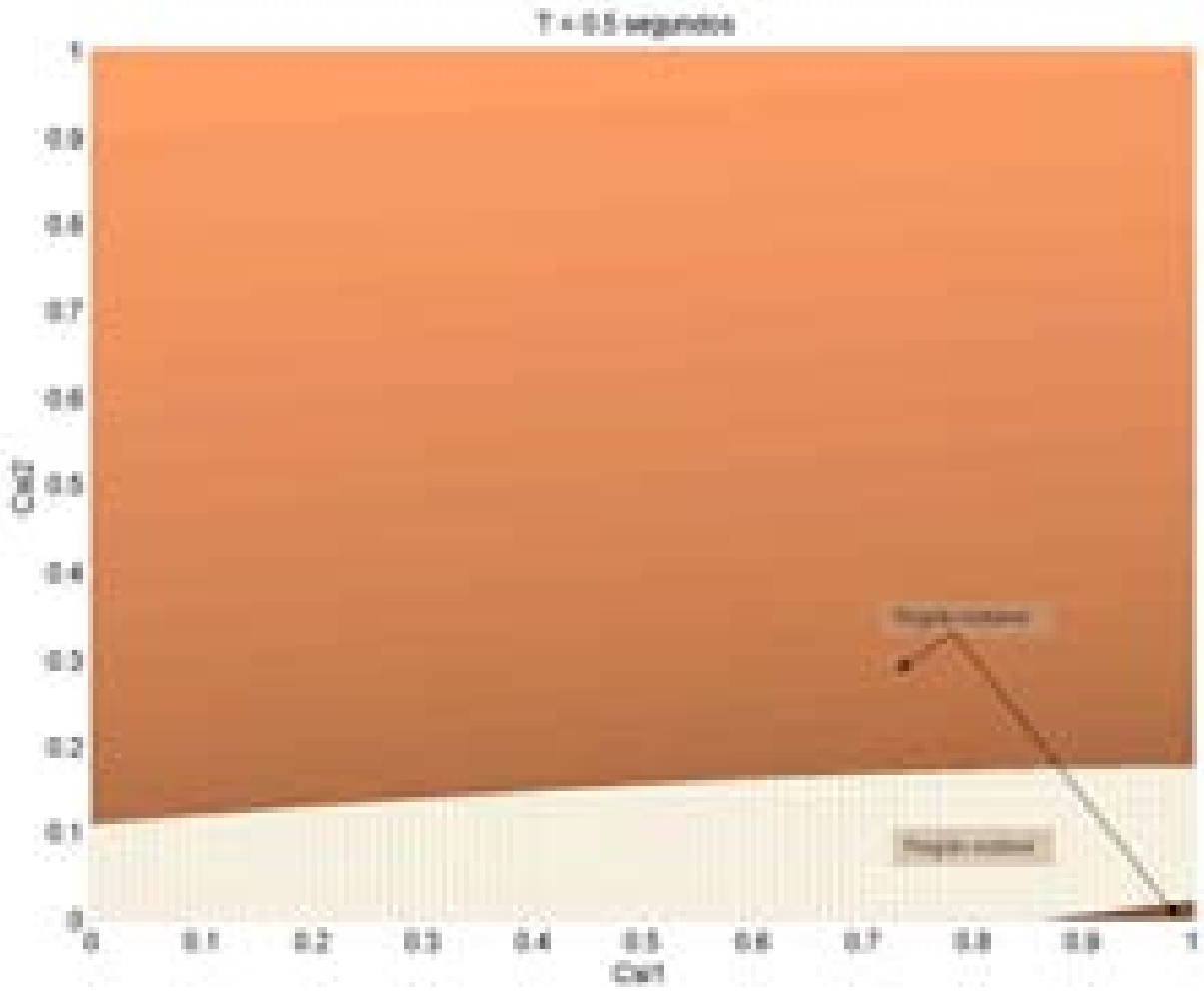


Figura 7.7: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 0.5 segundo em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

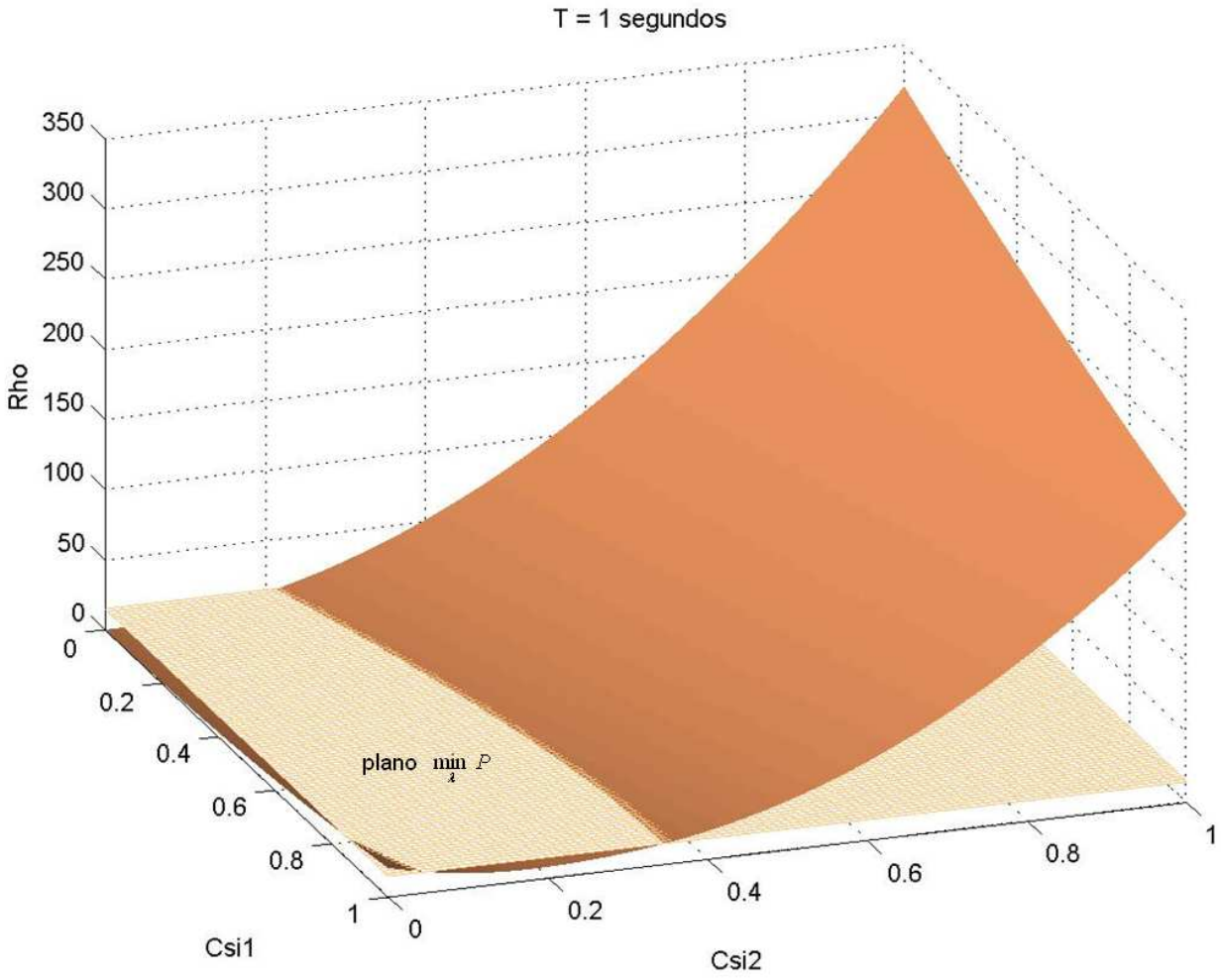


Figura 7.8: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1 segundo em função dos parâmetros ξ_1 e ξ_2.

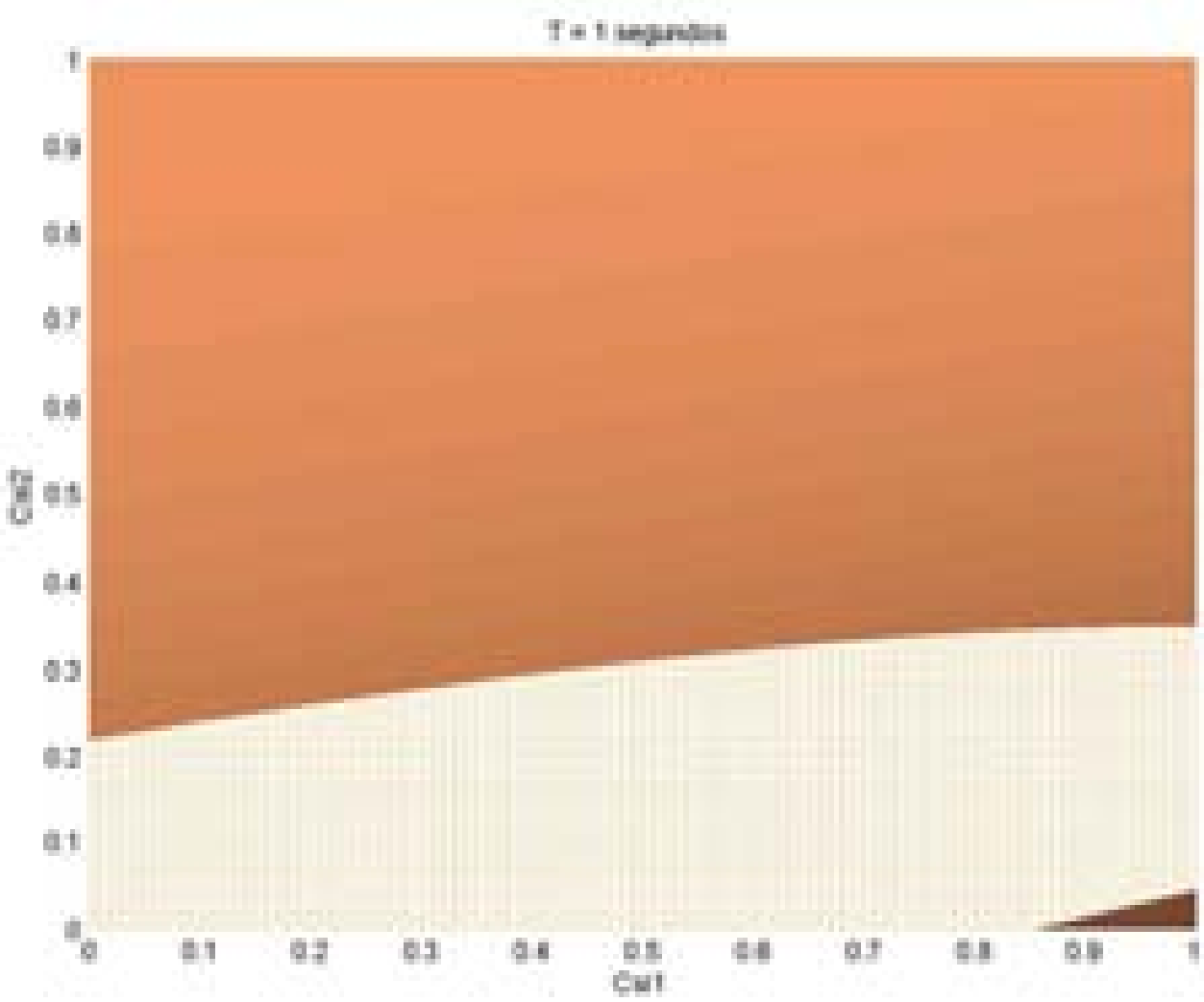


Figura 7.9: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1 segundo em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

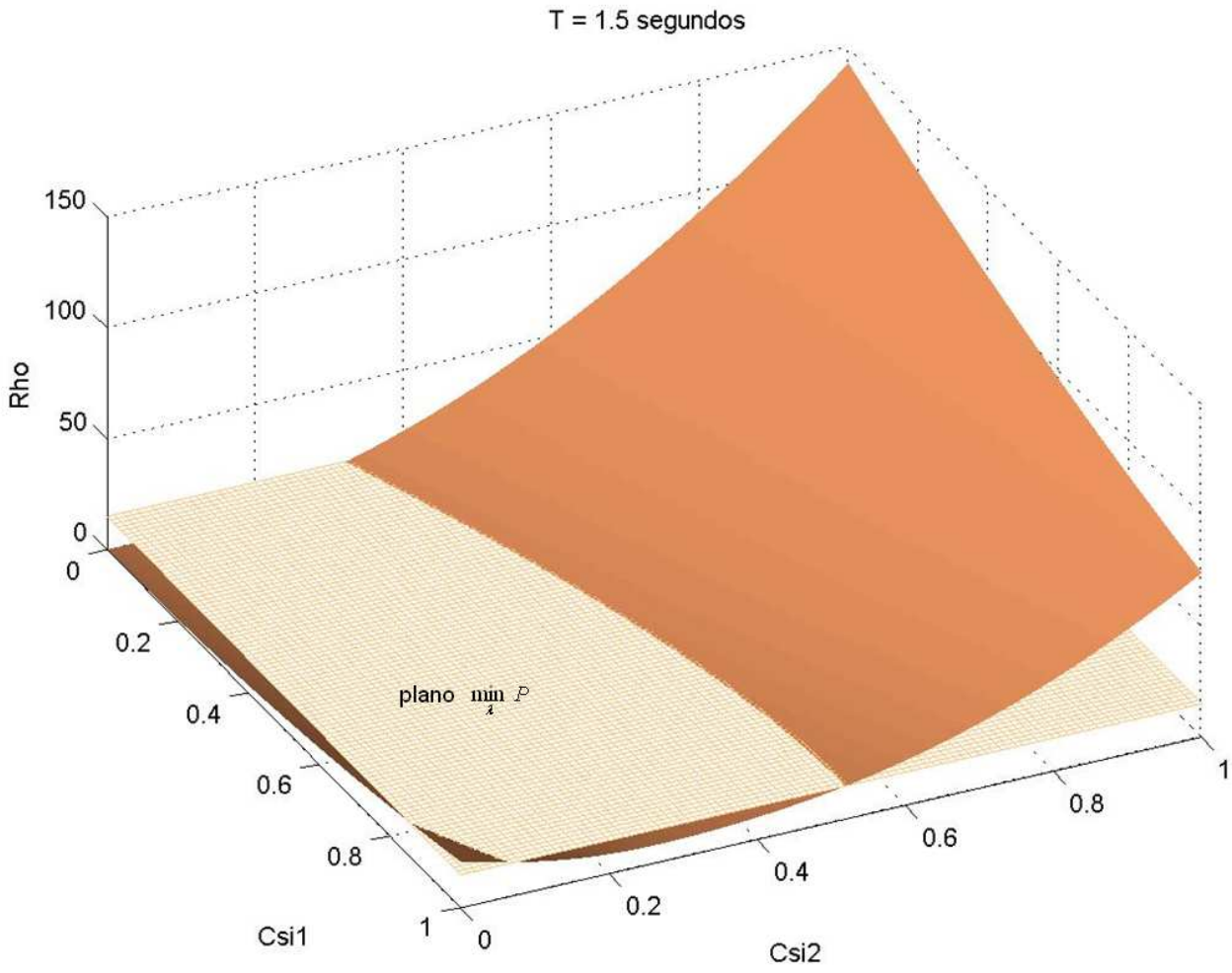


Figura 7.10: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1,5 segundos em função dos parâmetros ξ_1 e ξ_2.

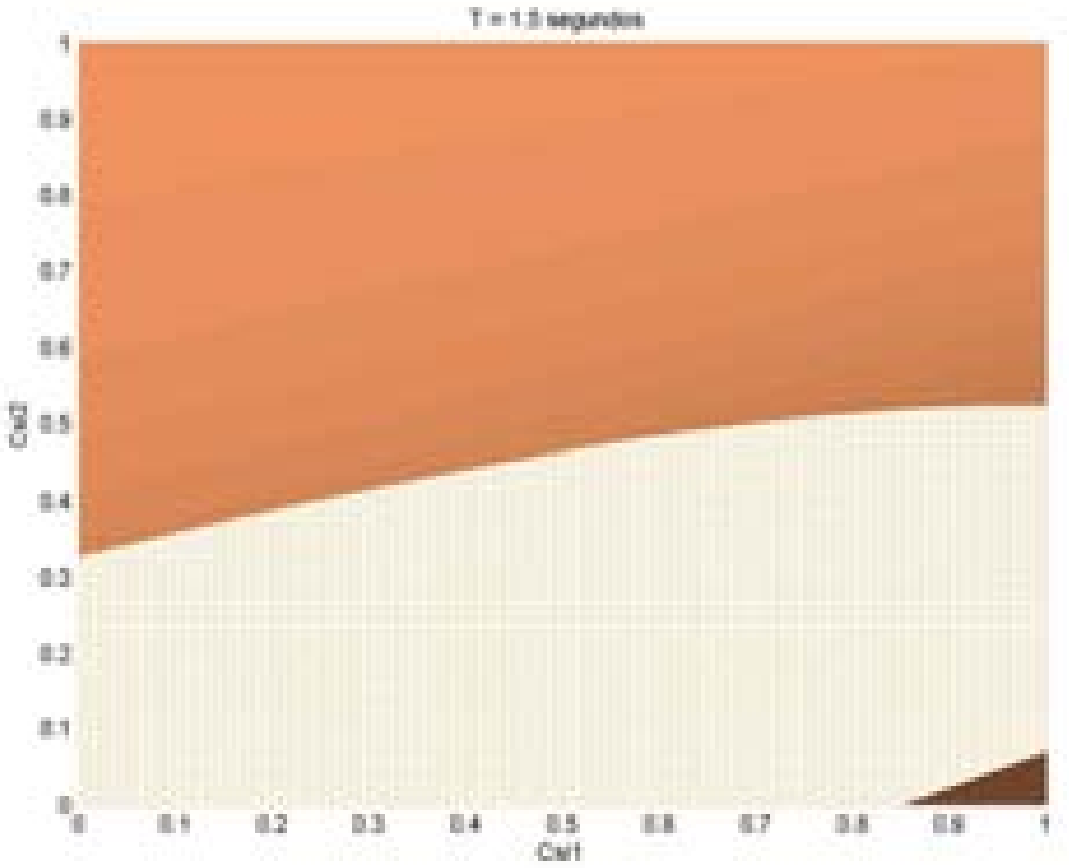


Figura 7.11: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1,5 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

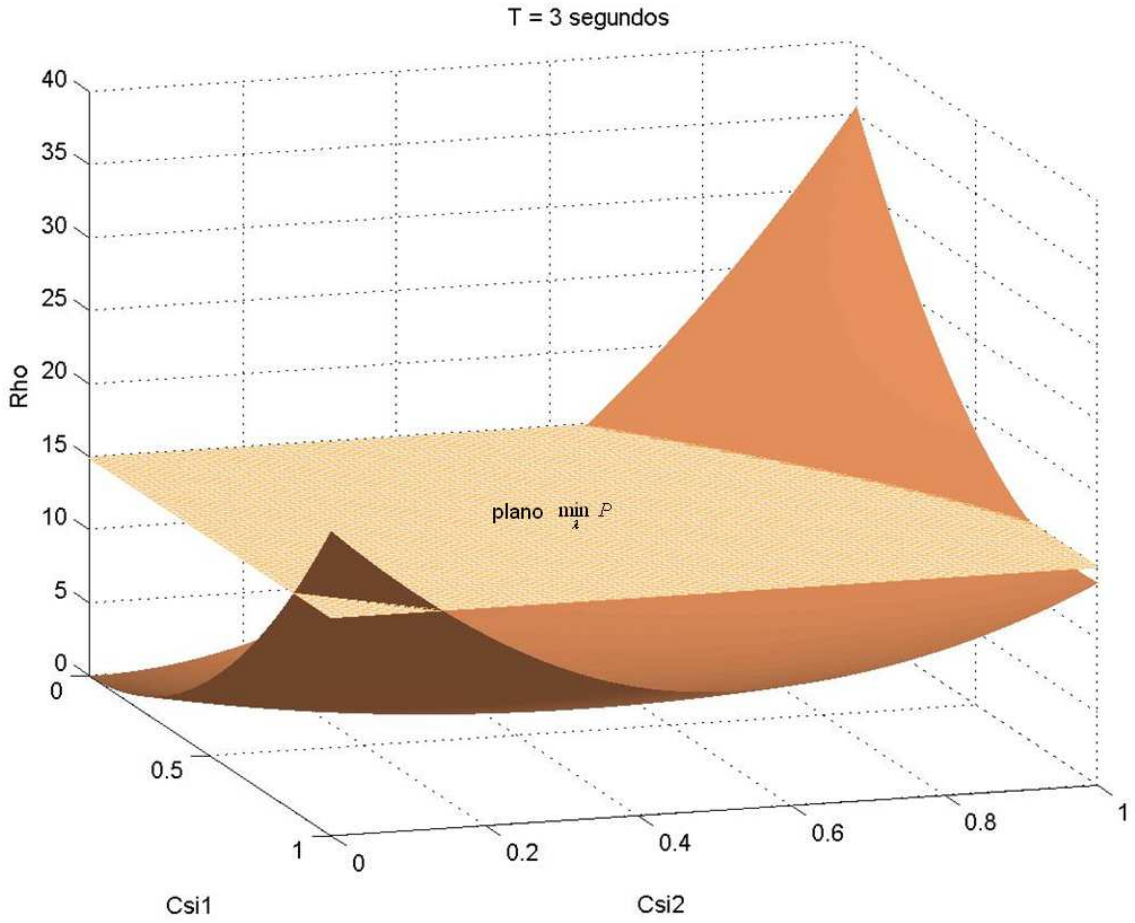


Figura 7.12: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 3 segundos em função dos parâmetros ξ_1 e ξ_2.

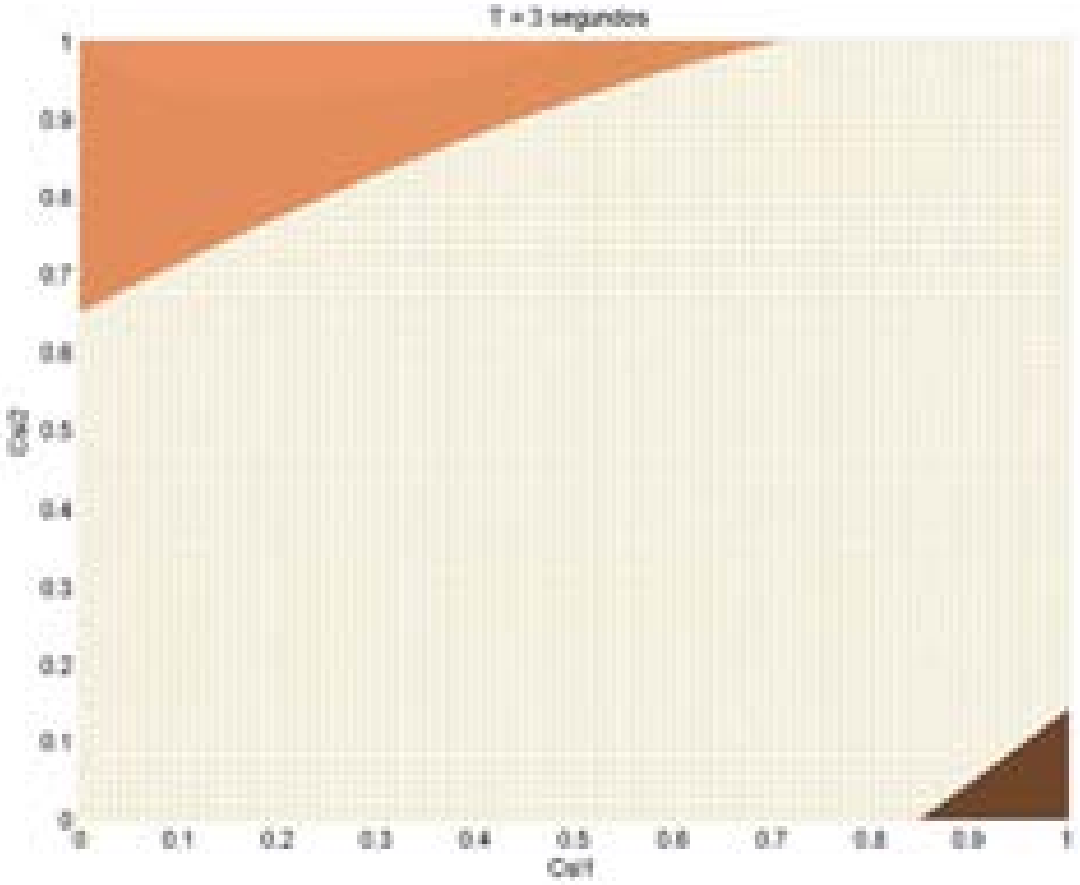


Figura 7.13: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 3 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

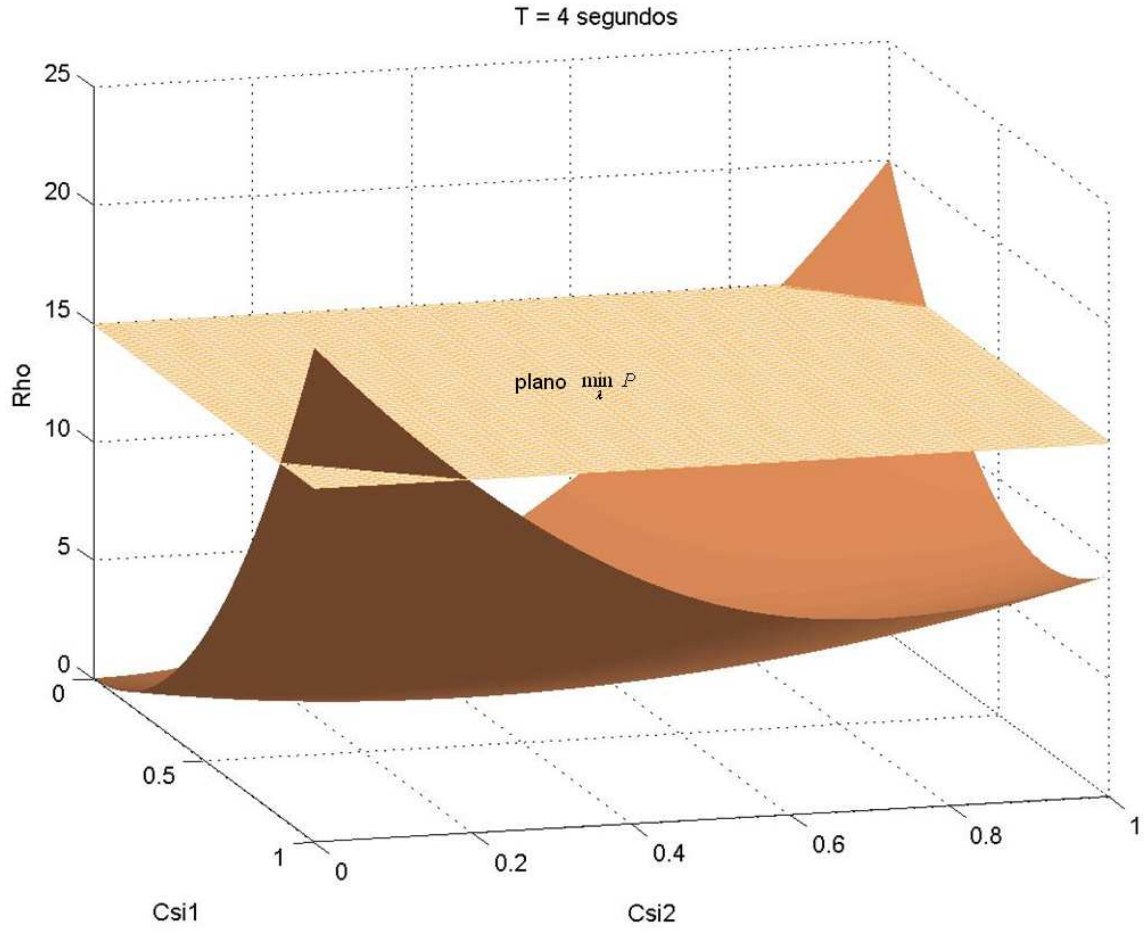


Figura 7.14: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 4 segundos em função dos parâmetros ξ_1 e ξ_2.

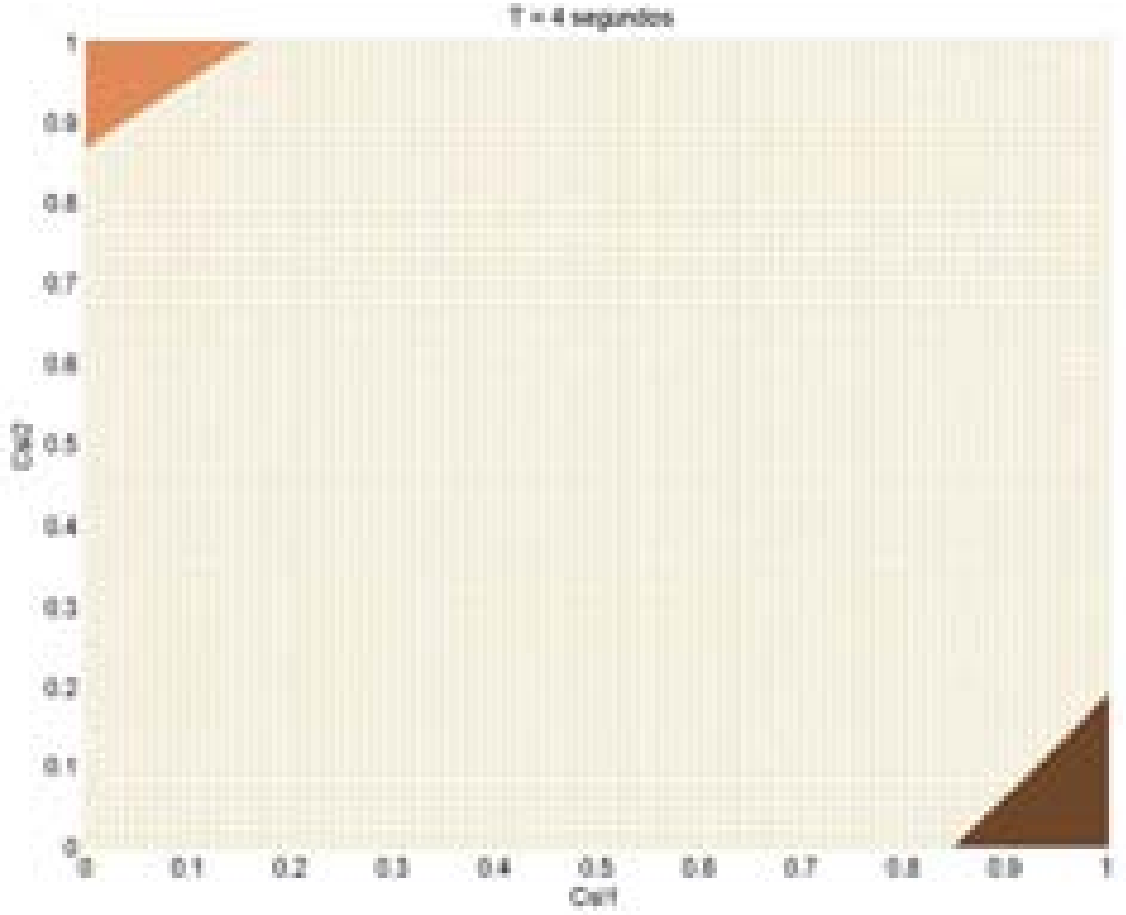


Figura 7.15: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 4 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

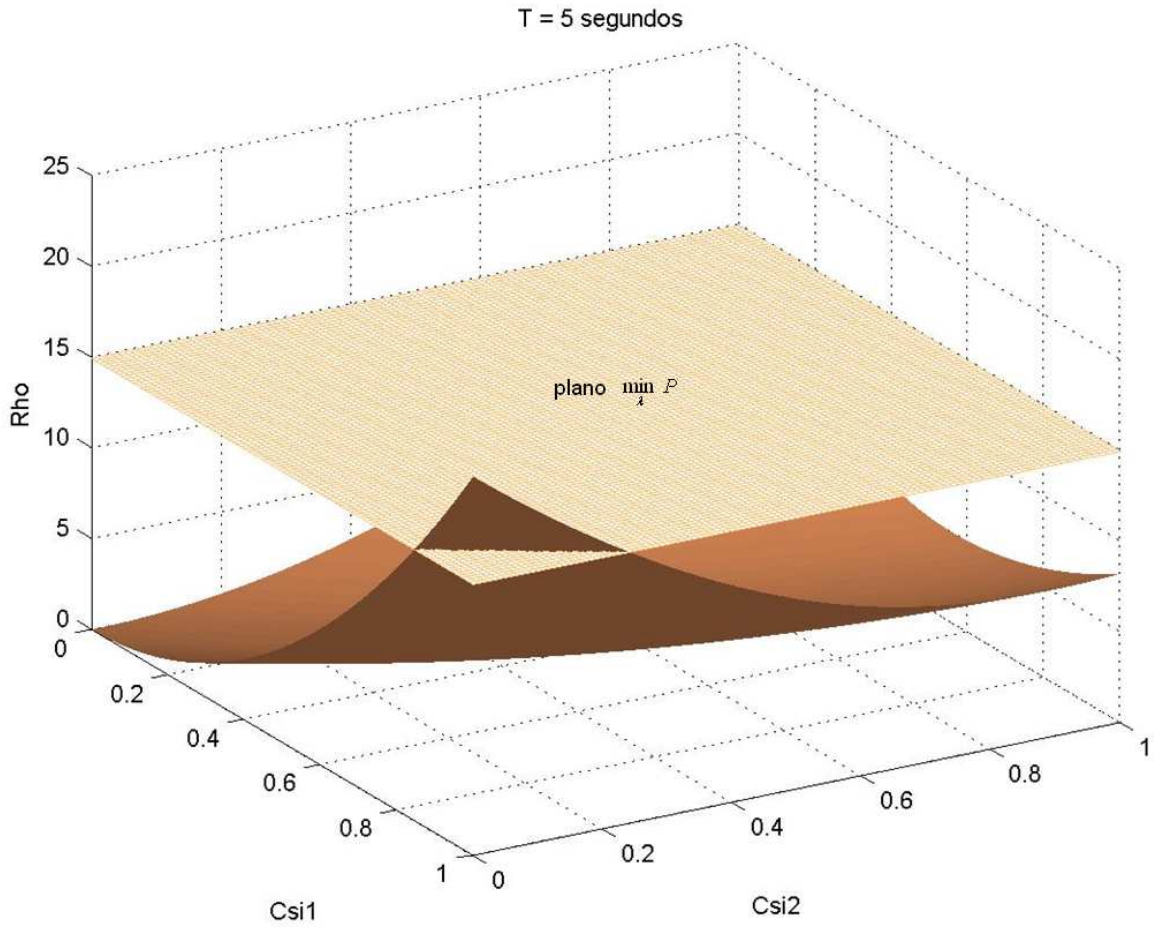


Figura 7.16: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 5 segundos em função dos parâmetros ξ_1 e ξ_2.

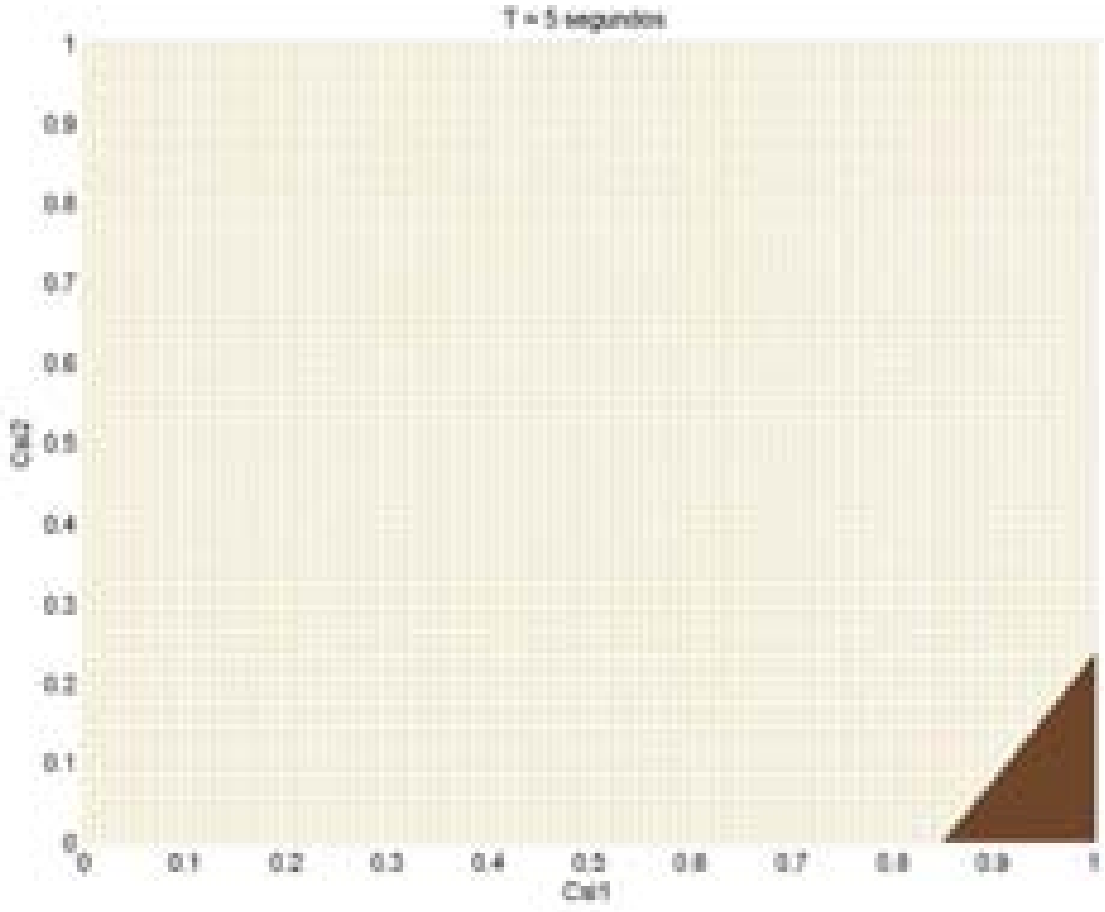


Figura 7.17: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 5 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

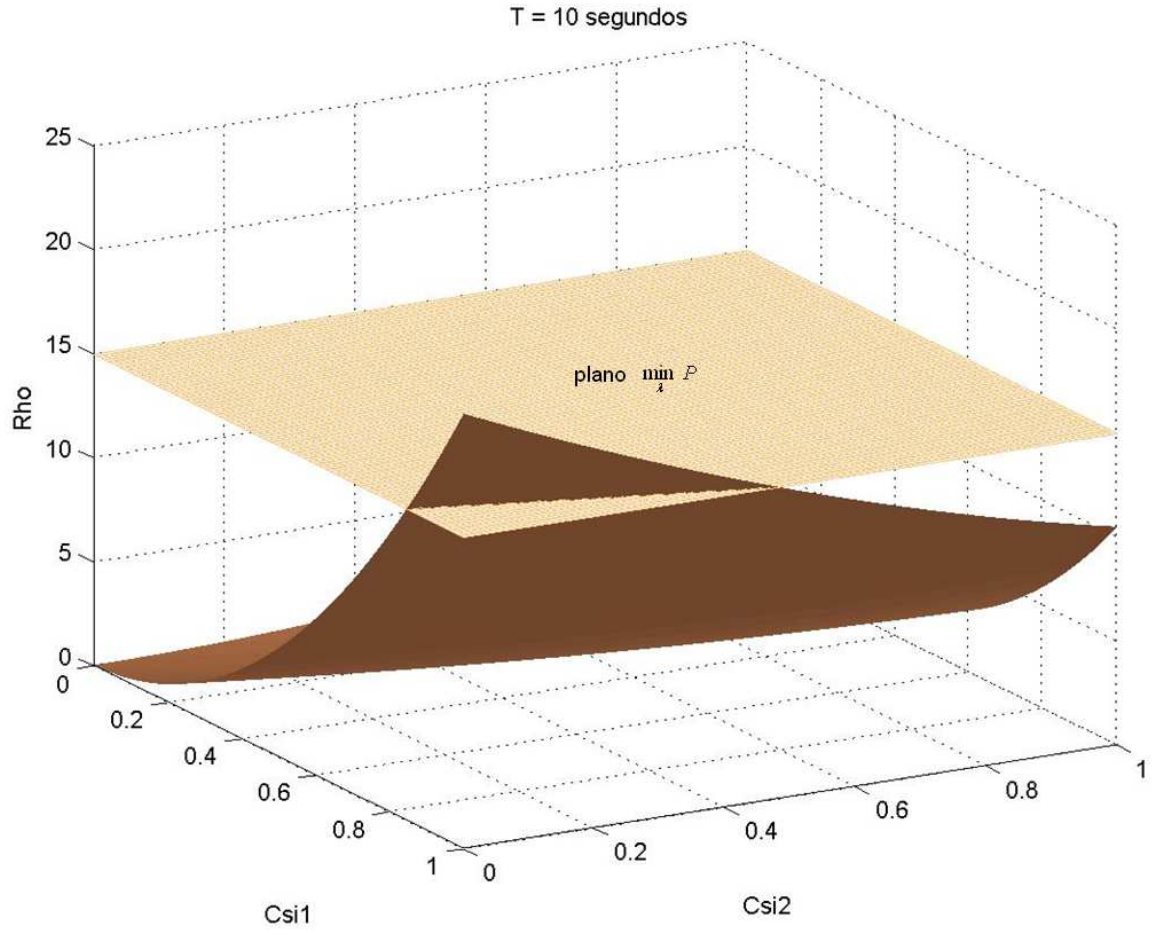


Figura 7.18: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10 segundos em função dos parâmetros ξ_1 e ξ_2.

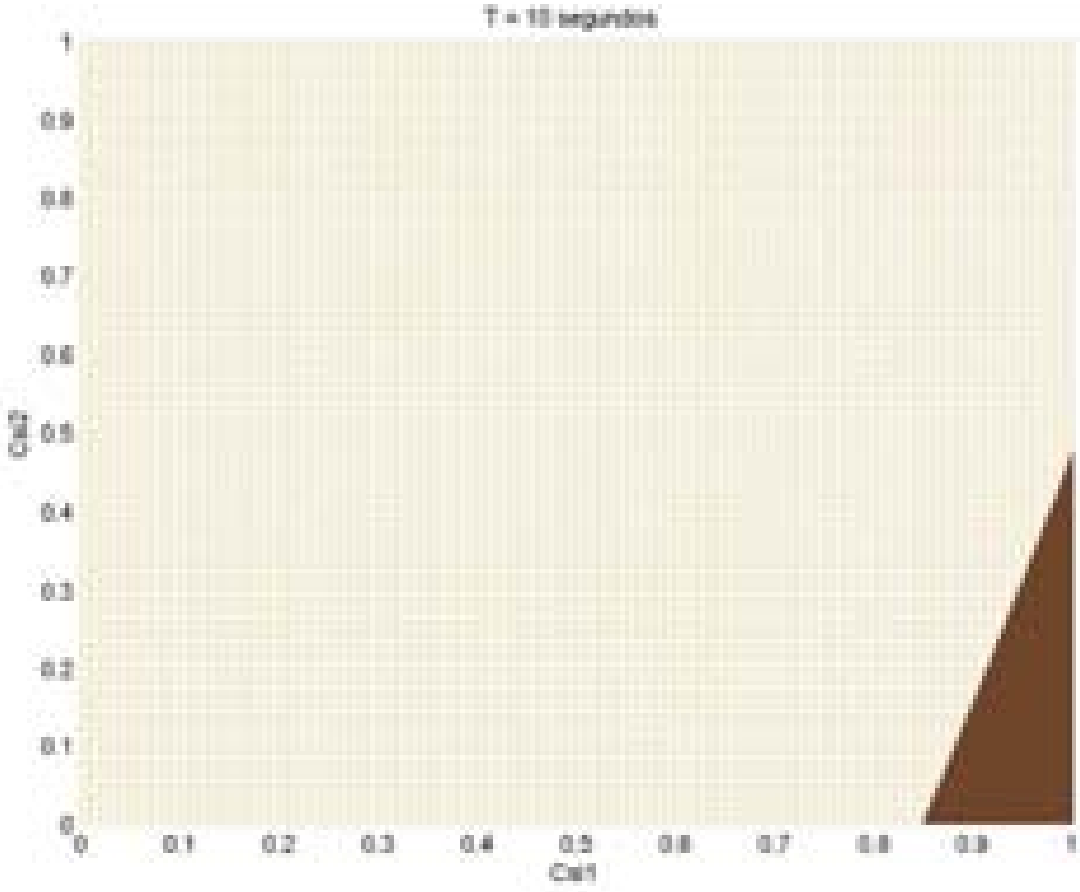


Figura 7.19: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

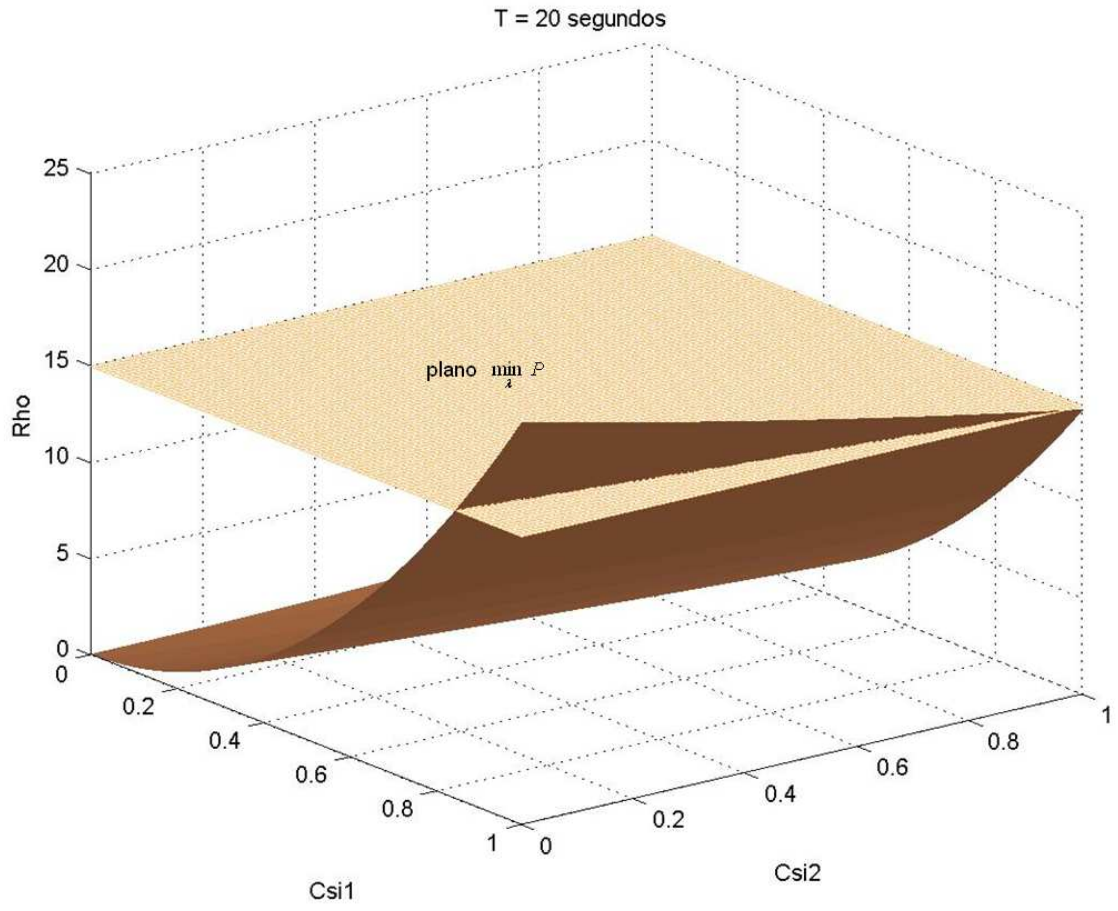


Figura 7.20: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 20 segundos em função dos parâmetros ξ_1 e ξ_2.

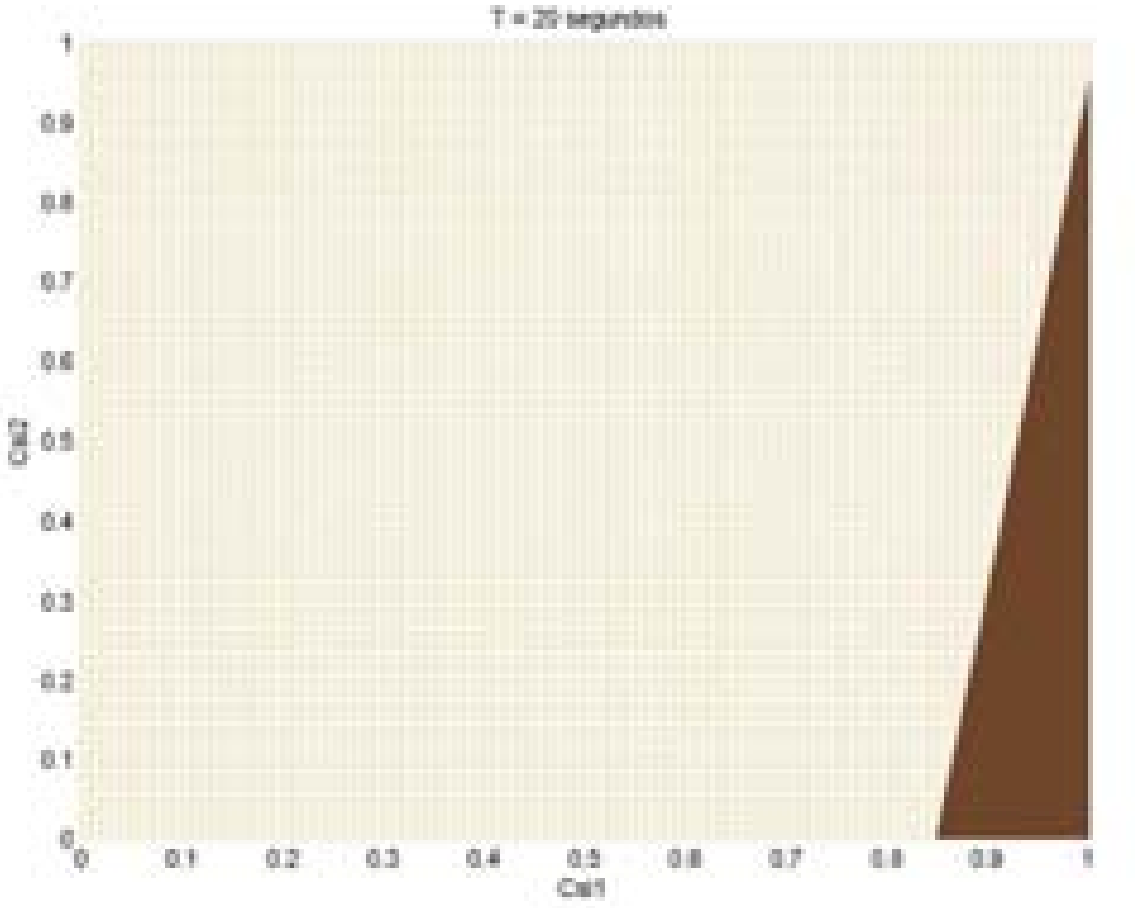


Figura 7.21: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 20 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

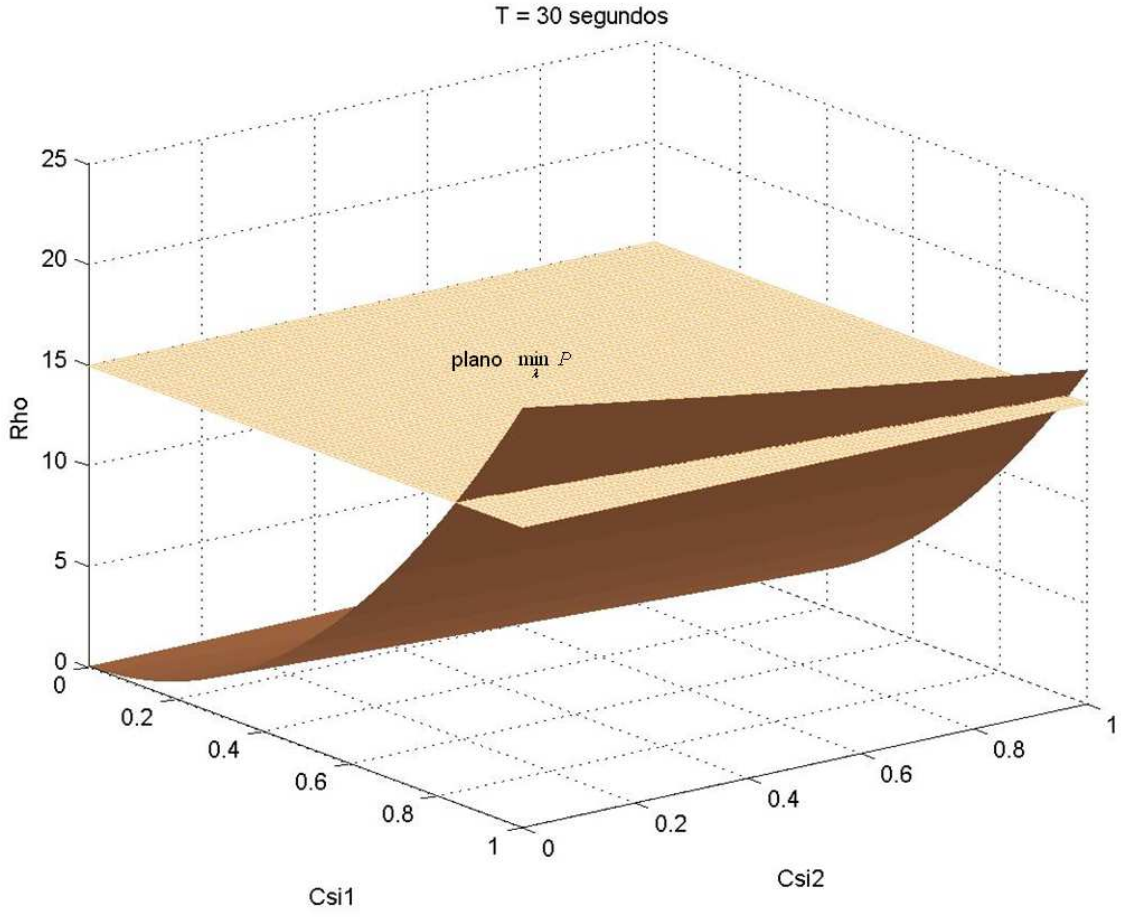


Figura 7.22: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 30 segundos em função dos parâmetros ξ_1 e ξ_2.

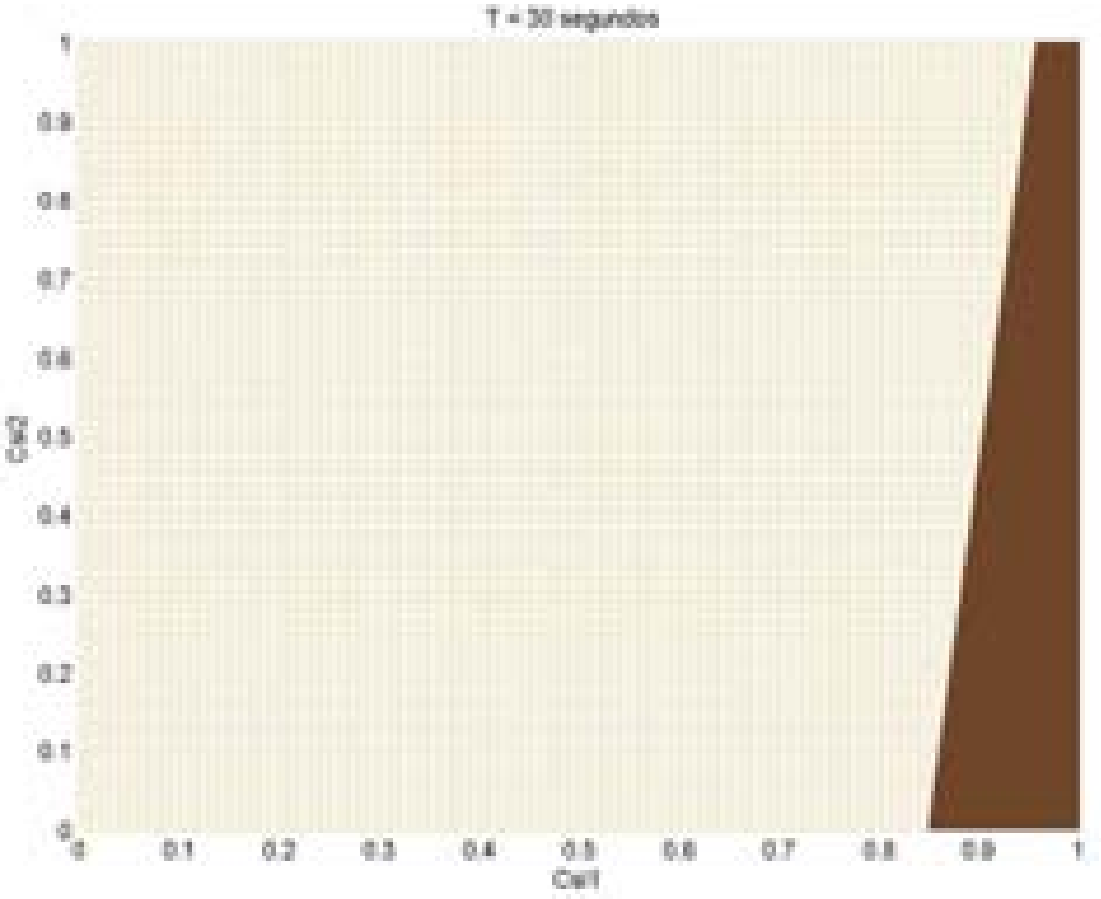


Figura 7.23: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 30 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

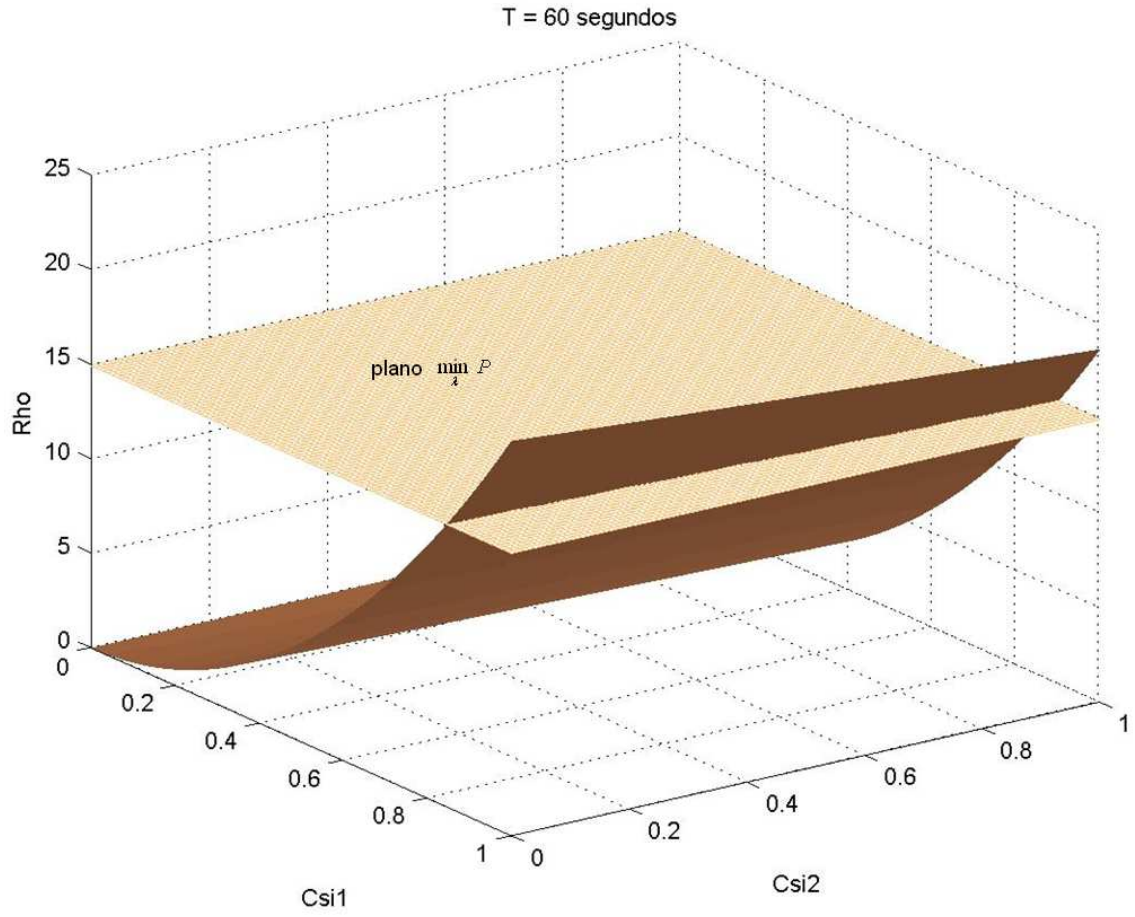


Figura 7.24: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 60 segundos em função dos parâmetros ξ_1 e ξ_2.

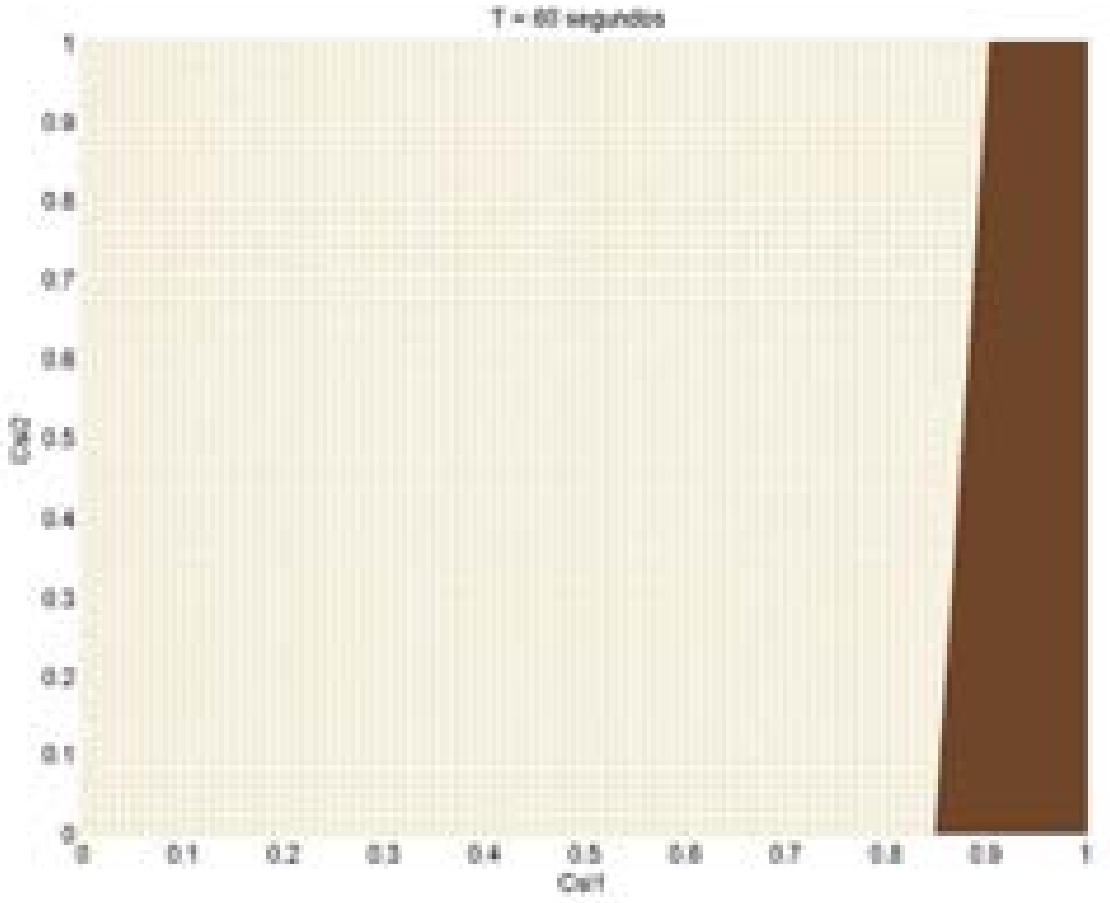


Figura 7.25: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 60 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

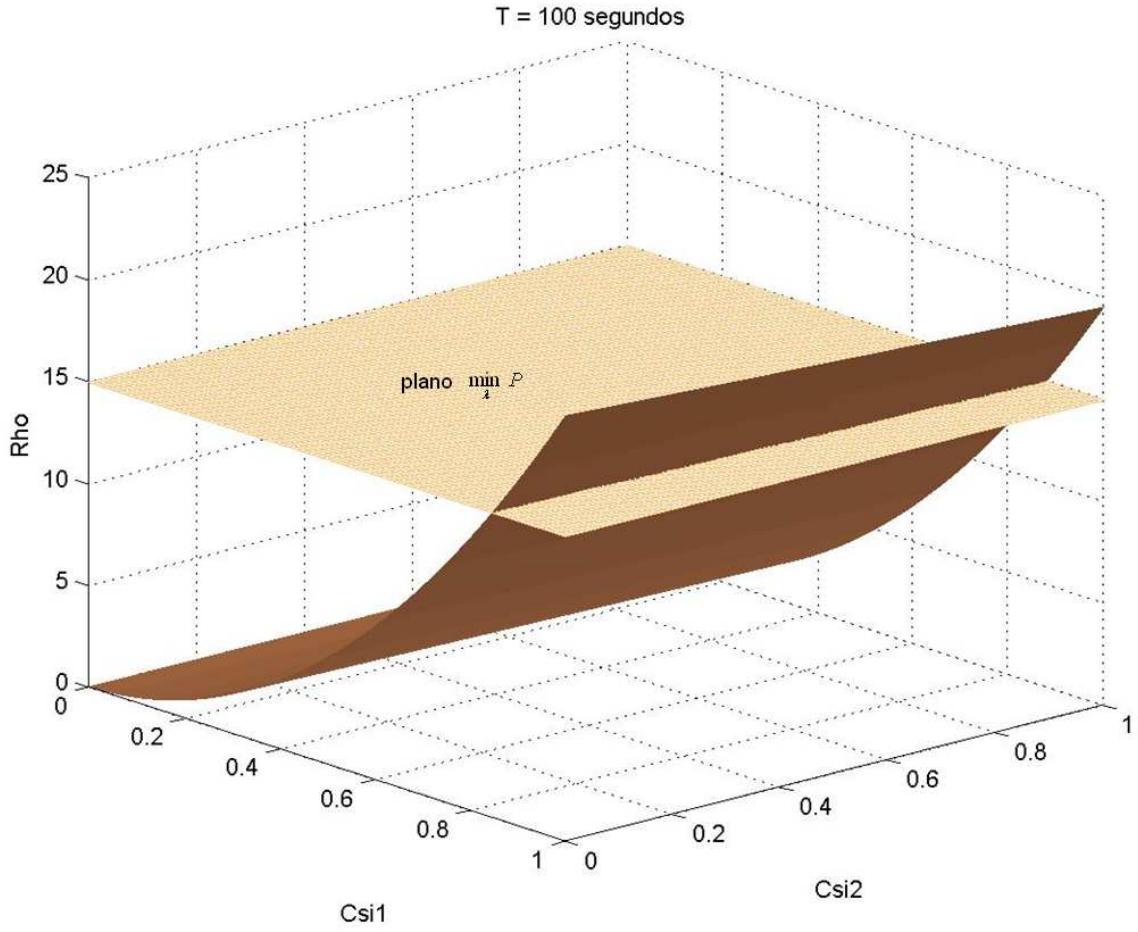


Figura 7.26: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 100 segundos em função dos parâmetros ξ_1 e ξ_2.

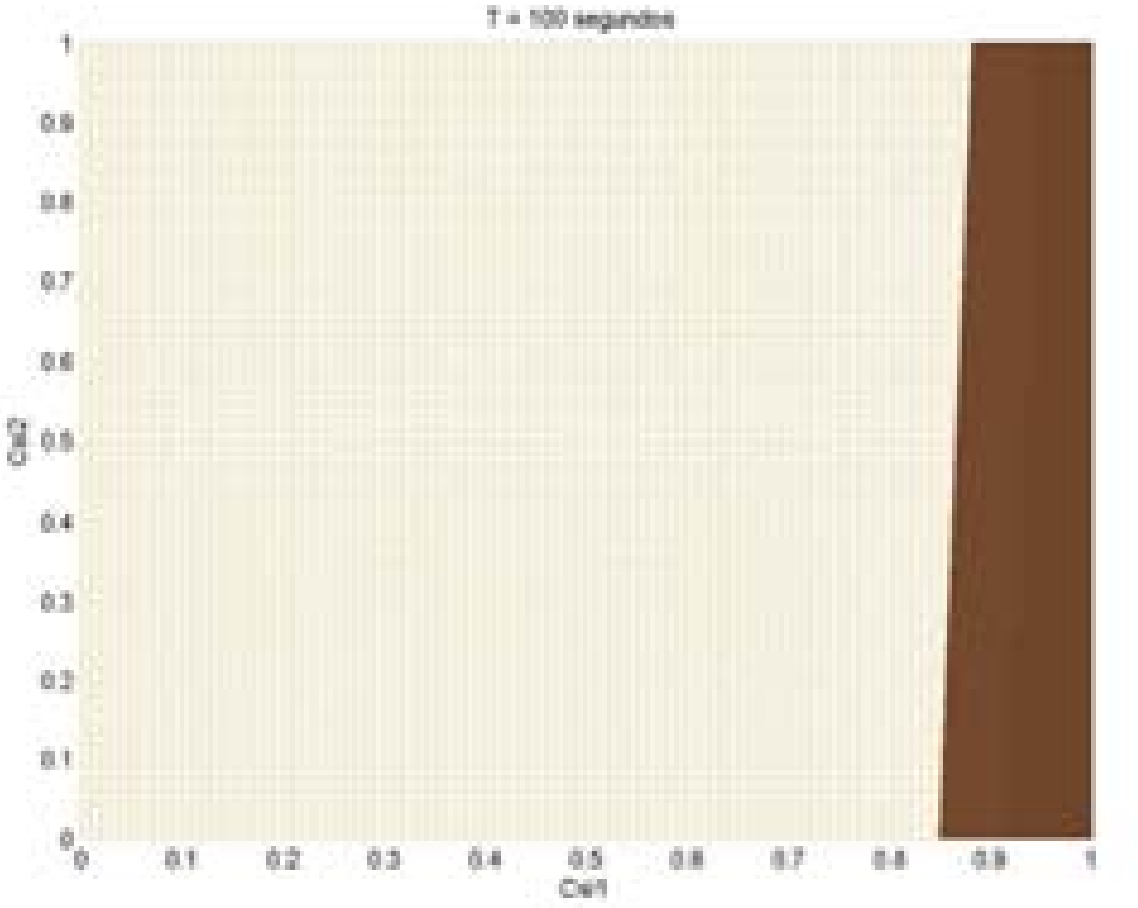


Figura 7.27: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 100 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

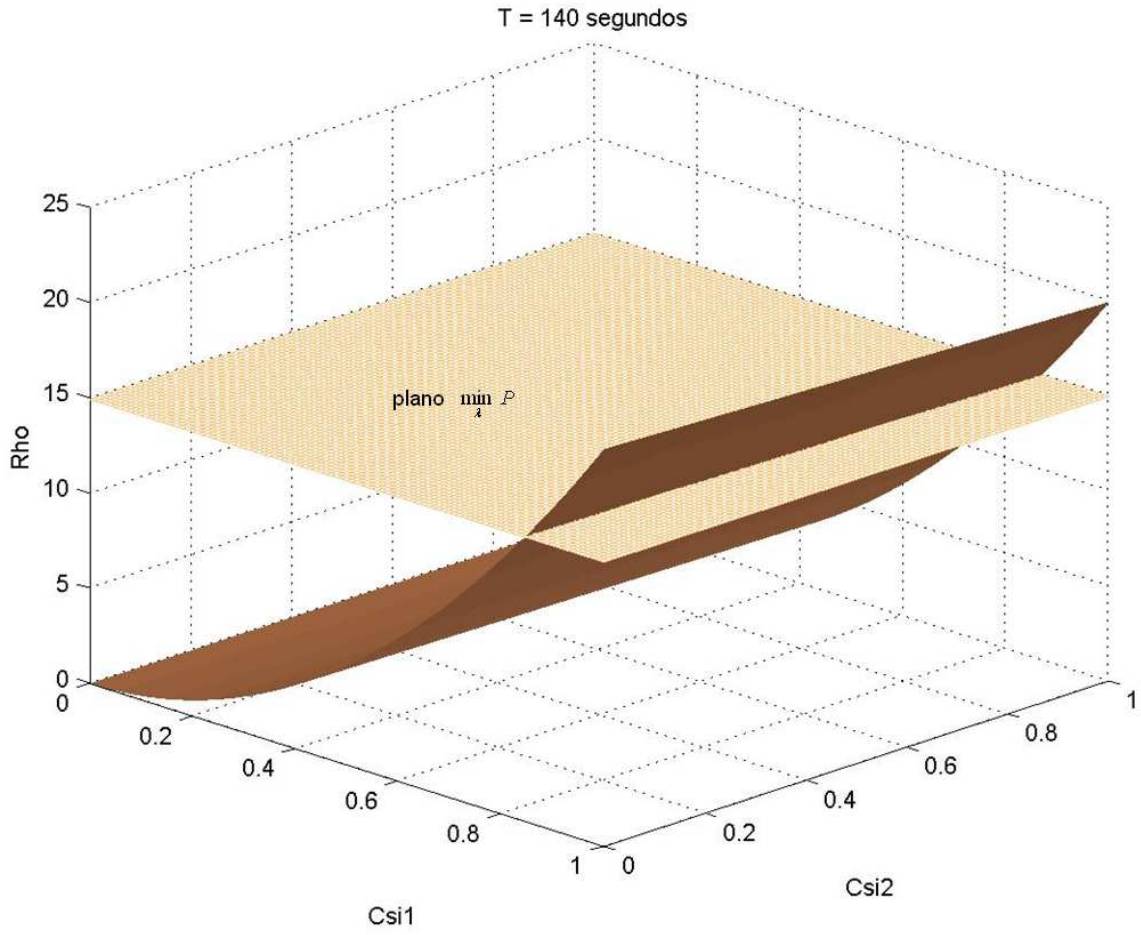


Figura 7.28: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 140 segundos em função dos parâmetros ξ_1 e ξ_2.

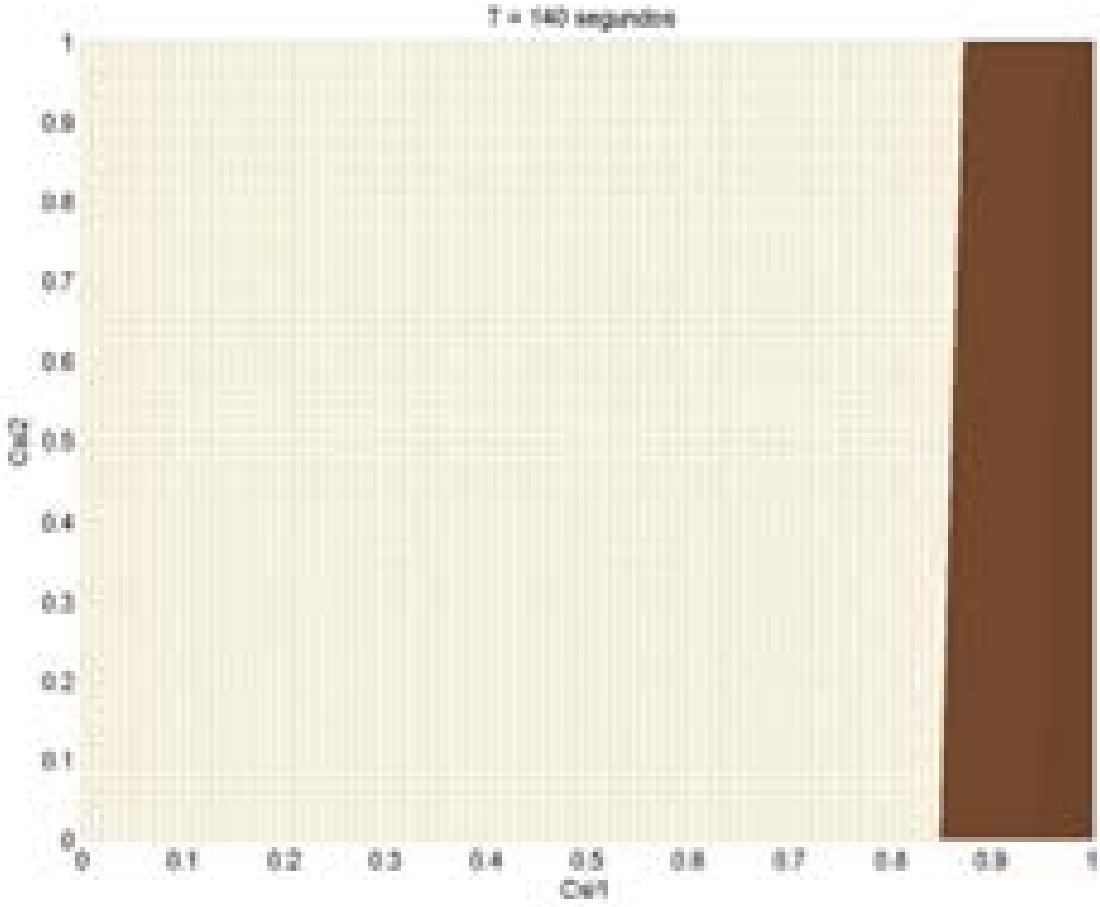


Figura 7.29: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 140 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

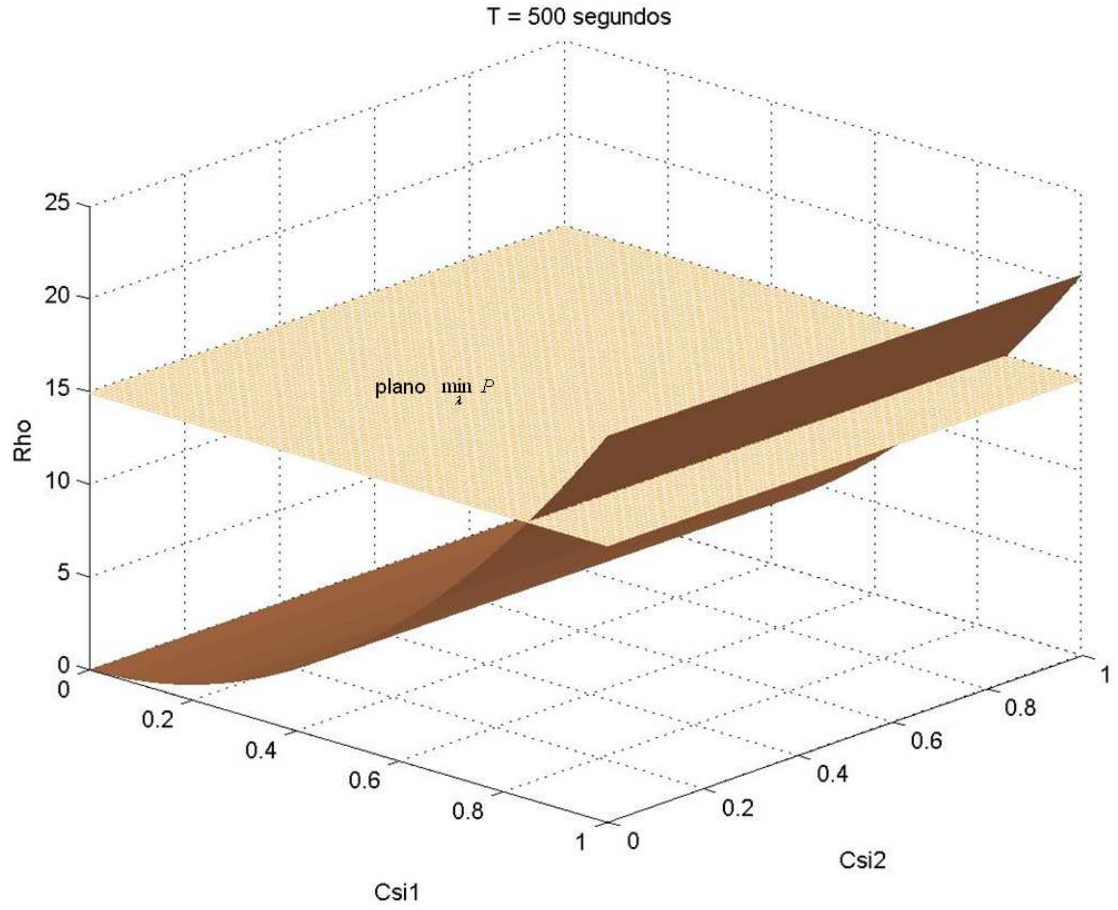


Figura 7.30: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 500 segundos em função dos parâmetros ξ_1 e ξ_2.

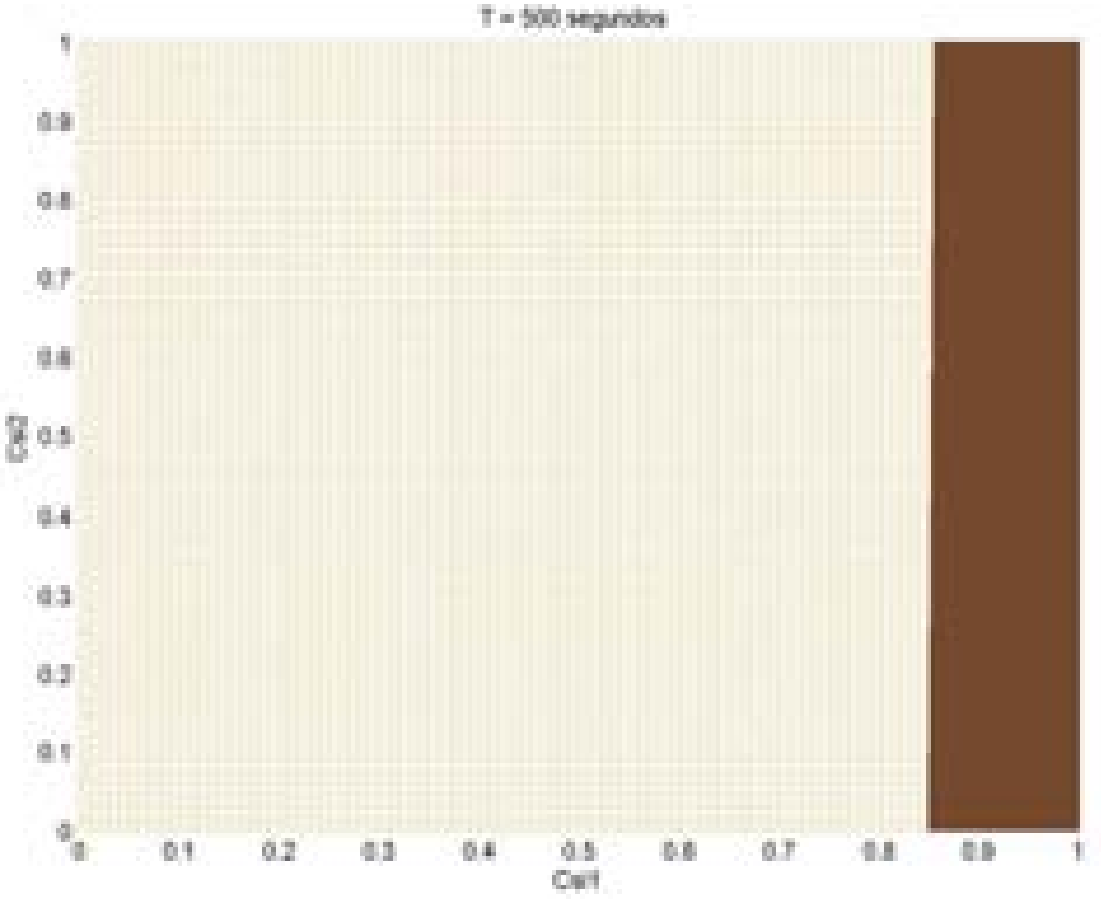


Figura 7.31: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 500 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

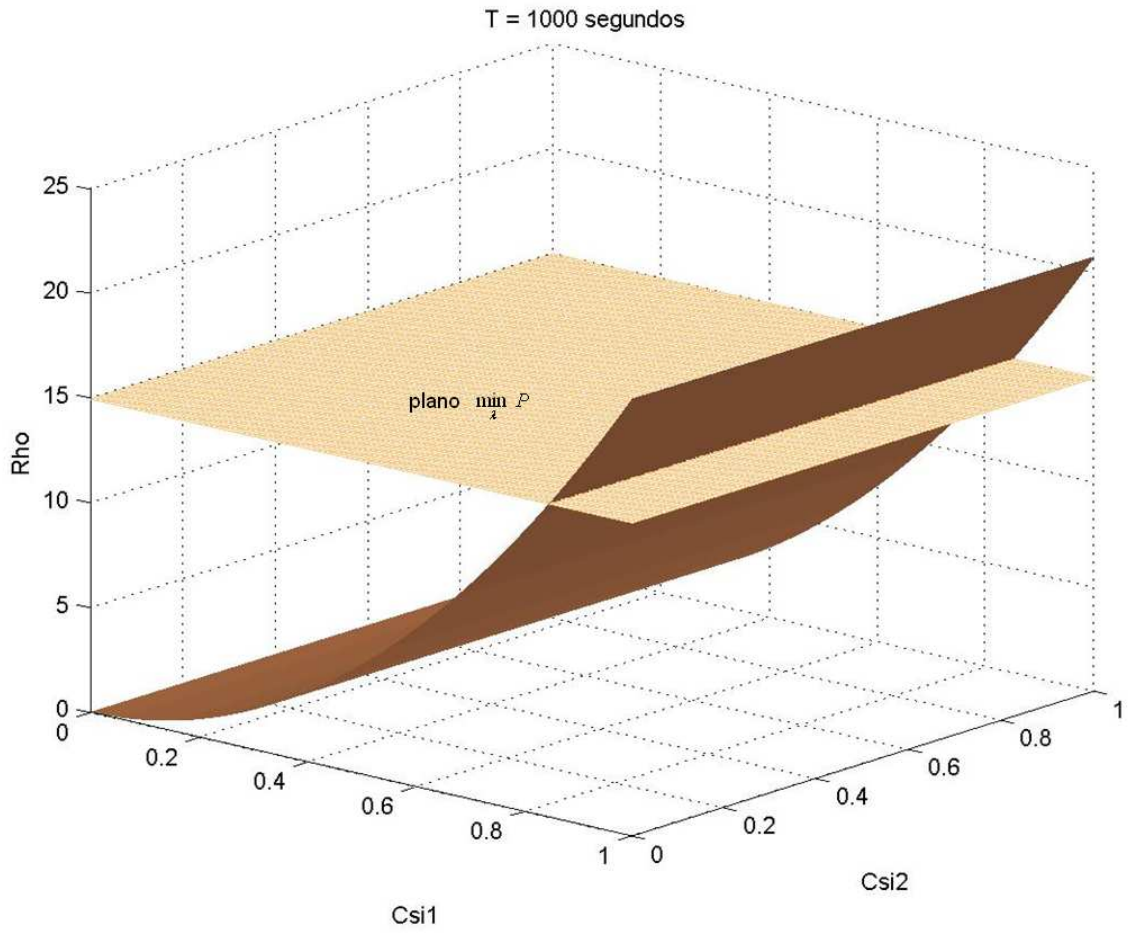


Figura 7.32: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1000 segundos em função dos parâmetros ξ_1 e ξ_2.

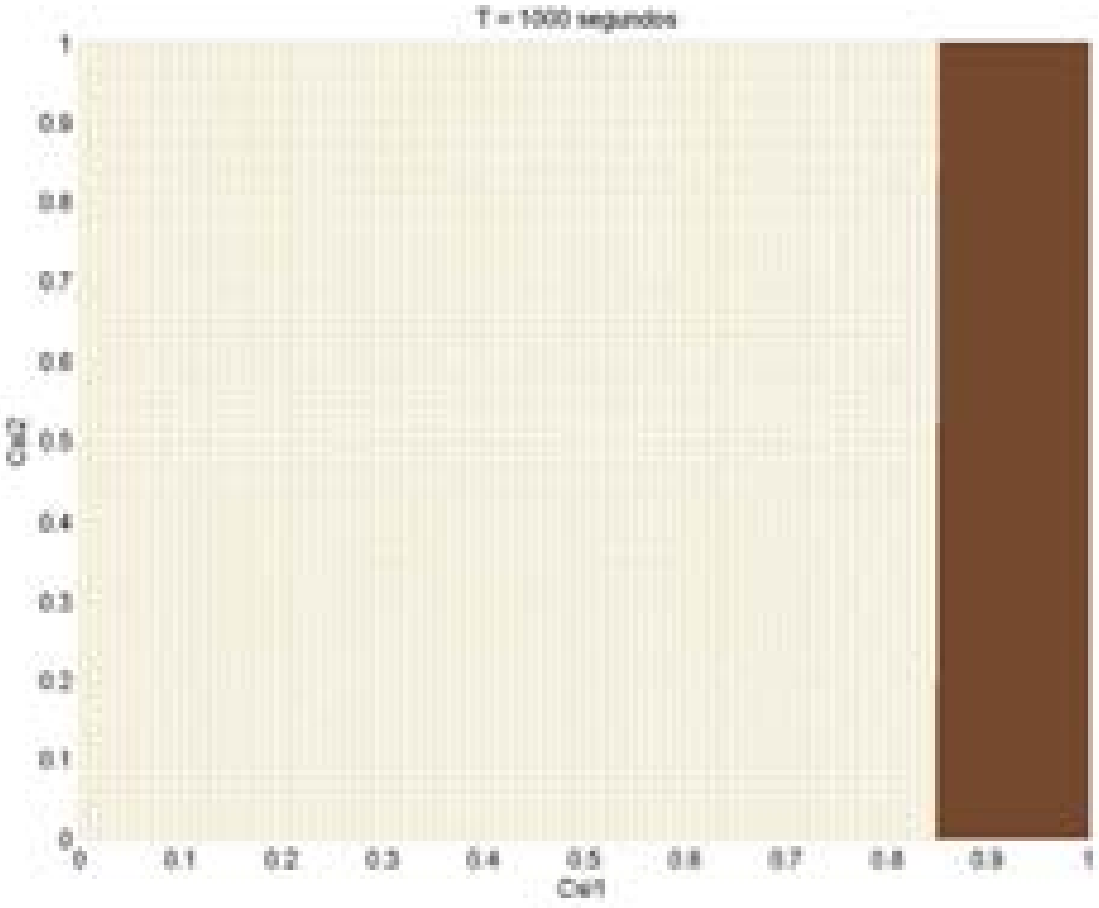


Figura 7.33: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 1000 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

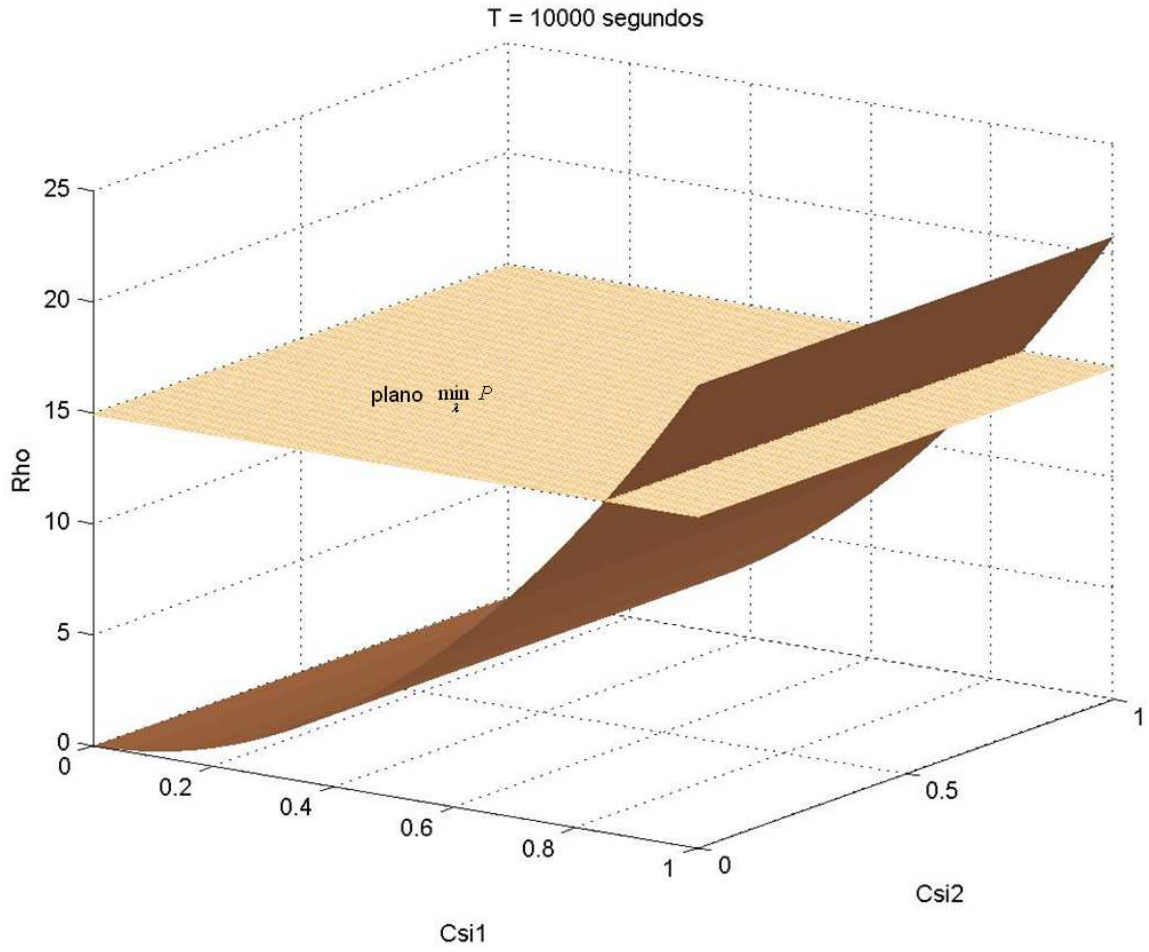


Figura 7.34: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10000 segundos em função dos parâmetros ξ_1 e ξ_2.

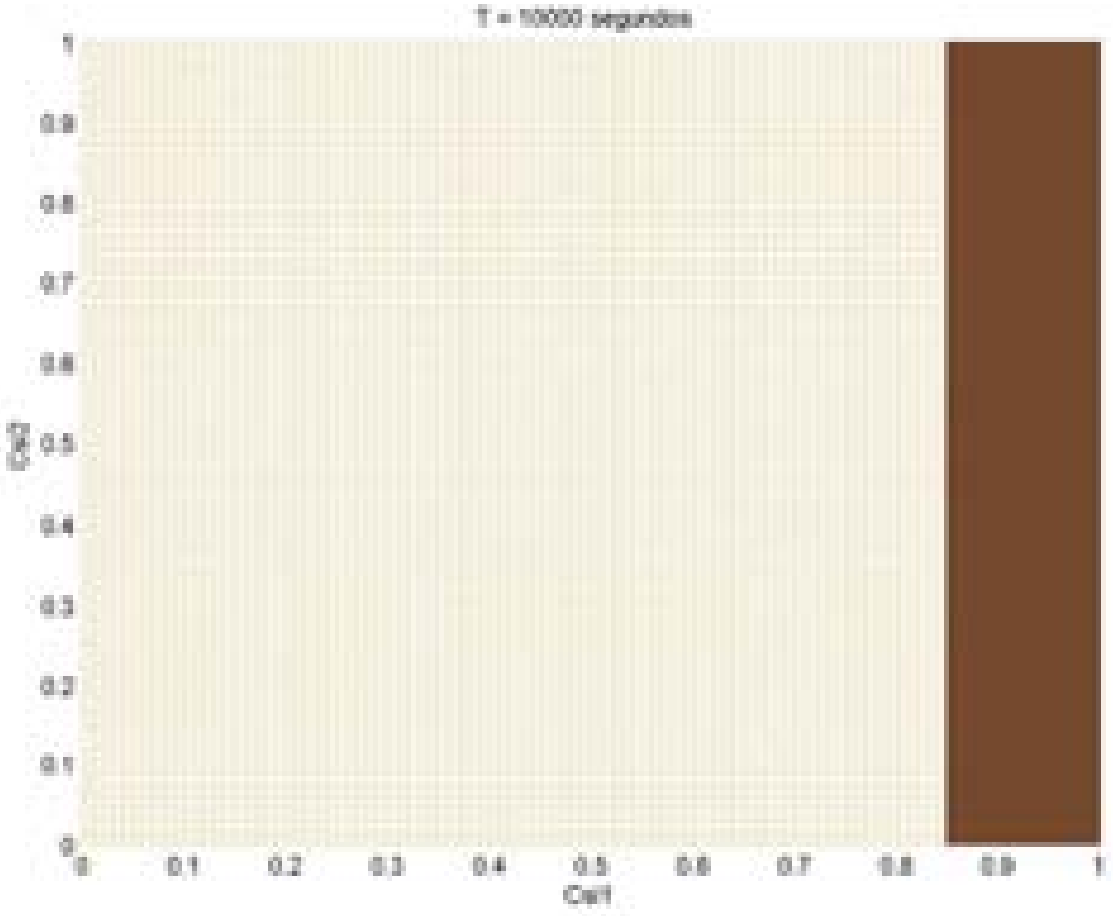


Figura 7.35: resultados numéricos do Teorema 7.3 apresentando regiões de estabilidade/instabilidade para um período de amostragem T = 10000 segundos em função dos parâmetros ξ_1 e ξ_2, tal figura consiste na vista superior da figura anterior.

Mediante os resultados acima apresentados verifica-se que a regra ST2 possui uma maior flexibilidade de projeto permitindo ajustar seus parâmetros de forma a poder ter um desempenho superior às regras ST1 e Tustin para determinados valores do período de amostragem T. Os resultados obtidos com a regra ST2 estão resumidos na Figura 7.34, onde as regiões claras correspondem ás regiões em que o sistema em malha-fechada apresentou estabilidade assintótica e nas regiões escuras, a instabilidade.

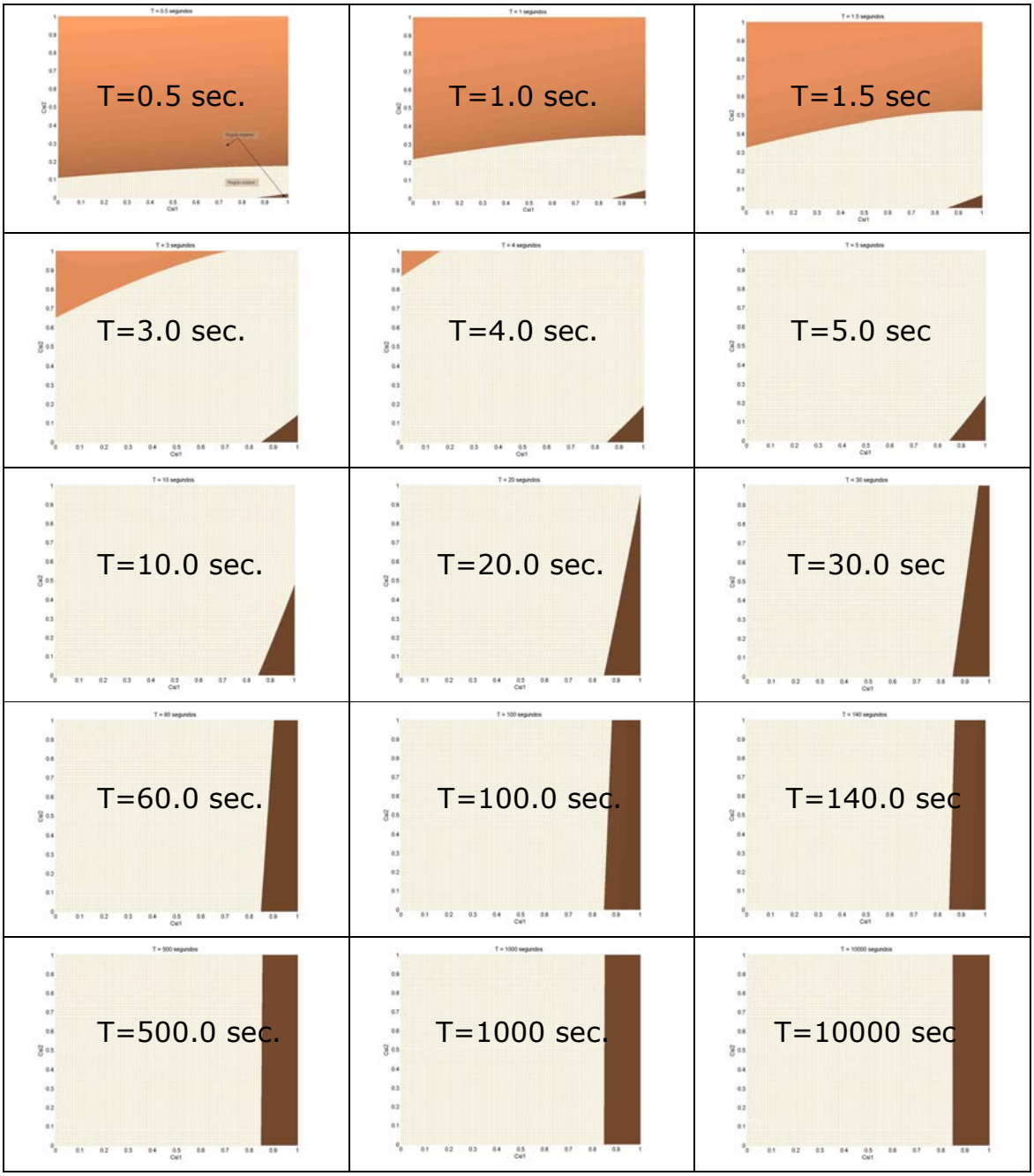


Figura 7.36: resumo dos resultados numéricos obtidos mediante o Teorema 7.3. Regiões claras correspondem às regiões em que o sistema em malha-fechada apresentou estabilidade assintótica e nas regiões escuras, a instabilidade, para ξ_1 no eixo das absissas e ξ_2 no eixo das ordenadas.

CAPÍTULO 8

CONCLUSÕES, SUGESTÕES E RECOMENDAÇÕES

Foram estudadas neste trabalho métodos alternativos de análise do desempenho / comportamento de sistemas de controle discreto de plantas analógicas. Como principais resultados podemos citar: a) é possível dimensionar a fronteira superior da primeira faixa assintoticamente estável em função do período de amostragem, i.e., o valor mais elevado de período de amostragem que ainda garante a estabilidade assintótica do sistema de controle; b) mediante aproximações numéricas precisas da função exponencial (por Padé de até décima-sexta ordem) foi provado não apenas a existência da fronteira de estabilidade da primeira faixa estável bem como o surgimento de faixas de estabilidade e instabilidade em função do período de amostragem; c) encontrou-se uma representação por multicaminhos em paralelo para um modelo matemático do falseamento ("aliasing"); d) foi proposta uma nova regra de mapeamento s-z com dois parâmetros de sintonia, mais geral que a apresentada em **Tredinnick (1999)** e e) foi realizada estensão do trabalho de **Balas (1982)** com a elaboração de importantes Teoremas e Corolários.

Sugere-se que: a) estudos futuros sejam feitos partindo dos resultados obtidos nesse trabalho, porém utilizando modelos matemáticos de outras plantas industriais complexas, possivelmente envolvendo sistemas rígido-flexíveis; b) estudos futuros que se basearem nos resultados desse trabalho considerem efeitos de quantização em amplitude bem como atrasos de processamento e palavras de comprimento finito e c) estudos futuros sejam desenvolvidos utilizando microcontroladores por sua fácil aquisição e programação/aplicabilidade.

Uma proposta para estudos futuros seria buscar reinterpretar um sistema em malha-fechada por meio de decomposição espectral (dispondo de uma análise geométrica), i.e., que a análise da estabilidade se desse em duas etapas: a) inicialmente calculam-se os autovetores tanto da planta quanto do controle (seja este analógico ou digital), b) feito isso se parte para o modelo do interfaceamento

entre a planta e o controle, isto é, que estrutura geométrica serve de interface entre esses dois. Supõe-se que talvez haja uma conexão entre as projeções desses autovetores (da planta e do controle) e alguma estrutura geométrica que seja função do sinal de referência a ser rastreado. Naturalmente que tal interpretação deve servir de instrumento para o projeto de controle digital. Outra proposta para estudos futuros é o desenvolvimento de um estudo a respeito do controle do caos ensinado em **Grebogi (1990, 1992a, 1992b, 1993, 1995)** aplicando os ensinamentos desta Tese. O controle do caos possui diversas aplicações interessantes, dentre as quais está a manipulação da trajetória de furacões, tal como mostrado em simulações numéricas nos estudos de Grebogi, podendo ser estendido para controle de arritmias cardíacas e certas anomalias mentais. Tal investigação poderia ser iniciado com o controle discreto do sistema caótico apresentado em 2.2.1 desta Tese. Outra investigação a ser feita seria a respeito do controle vibracional de íons dispostos em armadilhas iônicas, muito útil para preservar a informação quântica de Qubits num eventual futuro computador quântico (**Labaziewicz, 2008**; **Hirano, 2007**; **Negrevergne, 2006**; **Hughes, 2004**).

REFERÊNCIAS BIBLIOGRÁFICAS

BALAS, M.J. Direct velocity feedback control of large space structures. **J. Guidance, Control and Dynamics**. v.2, n.3, May/June. 1979.

BALAS, M.J. Discrete-time stability of continuous-time controller designs for large space structures. **J. Guidance, Control and Dynamics**. v.5, n.5, Sept./Oct. 1982.

BAMIEH, B.A. A general framework for linear periodic systems with applications to H^{∞} sampled-data control. **IEEE Transactions on Automatic Control**, v.37, n.4, p.418 a 434. April. 1992.

BARABANOV, A.T.; STAROZHILOV, Ye.F. Investigations of the stability of solutions of continuous discrete systems by lyapunov's second method. UDC 681.5.037.26; ISSN08882-570X/88/0006/0035. **Automatic Control Theory and its Use**. Scripta Technica, Inc. 1989.

BOSE, N. K. Comments on "s-plane to z-plane mapping using a simultaneous equation algorithm based on the bilinear transformation. **IEEE Transactions on Automatic Control**, v.33, n.11, p.1085-1086. Nov. 1988.

BOYKIN, W.H.; FRAZIER, B.D. Analysis of multiloop, multirate sampled-data systems. **AIAA Journal**, v.13, n.4, p.453-456. April. 1965.

BURDEN, R.L.; FAIRES, J.D. **Numerical analysis**. EUA, Boston: International Thomson Publishing (ITP). 1997.

CELLIER, F.E. **Lecture notes on digital control**. University of Arizona. EUA, julho de 2006. Disponível em: http://www.inf.ethz.ch/personal/fcellier/lect/dc/lect_dc_index.html . Acesso em: 10/09/2008

CHEN, T.; DING, F.; QIU, L. Reconstruction of continuous-time systems from their non-uniformly sampled discrete-time systems. **Automatica**, 2008. Doi: 10.1016/j.automatica.2008.08.007. *No prelo*.

CHEN, T.; RAFEE, N.; MALIK, O.P. A Technique for Optimal Digital Redesigns of Analog Controllers. **IEEE Trans. Control Systems Technology**, v.5, n.1, p.89 a 99. January. 1997.

DAHLIN, E.B. Designing and tuning digital controllers [Part 1]. **Instruments and Control Systems**, v. 41, p. 77 - 83. June. 1968 (a).

DAHLIN, E.B.; ZIEMER, R.L.; WICKSTROM, W.A.; HORNER, M.G. Designing and Tuning Digital Controllers [Part 2]. **Instruments and Control Systems**, v. 41, p. 87 - 91. July. 1968 (b).

D'AZZO, J.J.; HOUPIS, C.H. **Linear control system analysis and design**. EUA, New York: McGraw-Hill. 1981.

DEETS, D.A.; SZALAI, K.J. Design and flight experience with a digital fly-by-wire control system using Apollo guidance system hardware on an F-8 aircraft. **AIAA**, 72-881. California, EUA. 1972.

DIDUCH, C.P.; DORAISWAMI, R. Sample period effects in optimally designed digital control systems. **IEEE Transactions on Automatic Control.** v. AC-32, n.9, p.838-841. September. 1987.

FADEEV, D.K.; FADEEVA, V.N. **Computational methods of linear algebra**. EUA, New York: Dover. 1959.

FEINTUCH, A.; SAEKS, R. **System theory**: a Hilbert space approach. New York: Academic Press, 319 p. 1982.

FRANKLIN, G.F.; POWELL, J.D. **Digital control of dynamical systems**. California: Addison-Wesley. 1981.

FRANCIS, B.A.; CHEN, T. Input-output stability of sampled-data systems. **IEEE Transactions on Automatic Control.** v.36, n.1, p.50 a 58. Jan.. 1991.

FRANCIS, B.A.; GEORGIOU, T.T. Stability theory for linear time-invariant plants with periodic digital controllers. **IEEE Transactions on Automatic Control**. v.33, n.9, p.820-832. Sept. 1988.

GINTER, S.D. **Attitude control of large flexible spacecraft**. Boston: 231p. Dissertação (Mestrado em Ciências) - Massachusetts Institute of Technology, 1978.

GLASSON, D.P. Development and applications of multirate digital control. **Control Systems Magazine**, v. 3, n. 4, p. 2-8. Nov. 1983.

HAGIWARA, T.; ARAKI, M. On the preservation of strong stabilizability under sampling. **IEEE Transactions on Automatic Control**, v. 33, n.11. Nov. 1988.

HAHN, W. **Theory and application of Liapunov´s direct method** Englewood Cliffs. N.J: Prentice-Hall. 1963.

HANSEN, L.P.; SARGENT, T.J. The Dimensionality of the Aliasing Problem in Models with Rational Spectral Densities. **Econometrica**, v. 51, n.2, p. 377 a 387. Mar. 1983.

HIRANO, M. **Demonstration of coupling-controlled Qubits and future outlook**. Nippon Electric Company (NEC), Japan Science and Technology Agency (JST) and the Institute of Physical and Chemical Research (RIKEN). Japão, 07 de maio de 2007. Disponível em:

http://www.nec.co.jp/rd/en/innovative/E3/08.html . Acesso em 05/12/2008
http://www.nec.co.jp/press/en/0705/0702.html . Acesso em 05/12/2008

HORI, N.; MARKAZI, A.H.D. A new method with guaranteed stability for discretization of continuous-time control systems. In: AMERICAN CONTROL CONFERENCE ACC/TA2, 1992, Chicago. **Proceedings…** Chicago: [s.n], 1992.

HOU, L.; MICHEL, A.N. Unifying theory of continuous, discontinuous and discrete-time dynamical systems. **Elsevier, nonlinear analysis: hybrid systems**, v.1, p.154 a 172. 2007.

HOUPIS, C.H. AND LAMONT, G.B. **Digital control systems –** theory, hardware, software. New York: McGraw-Hill. 1985.

HUGHES, R.J.D.; JAMES, F.V.J.; GOMEZ, J.; GULLEY, M.S.; M. HOLZSCHEITER, H.; KWIAT, P.G.; LAMOREAUX, S.K.; PETERSON, C.G.; SANDBERG, V.D.; SCHAUER, M.M.; SIMMONS, C.M.; THORBURN, C.E.; TUPA, D.; WANG, P. Z.; WHITE, A.G. The los alamos trapped ion quantum computer experiment. **Quantum Computing**. P.23-55, Wiley-VCH Verlag GmbH. School of electronic engineering & computer systems, University of Wales, United Kingdom. 2004.

IEKO,T.; OCHI, Y.; KANAI. K. New design method for pulse-width modulation control systems via digital redesign. **Journal of guidance, control and dynamics**, v. 22, n.1. Jan./Feb. 1999.

INTEL. **MCS-4 microcomputer set**. Datasheet de Novembro de 1971.

JORDAN, C. **Calculus of finite differences**.. New York: Chelsea Publishing Company, 1965.

JUANG, Y.-T.; KUO, T.-S.; TUNG, S.-L. Stability Analysis of Contiuous and Discrete Interval Systems. **Control Theory and Advanced Technology**, v..6, n. 2, p.221 a 235. MITA Press, June, 1990.

Jury, E.I. **Theory and application of the Z-transform method**. New York: Robert Krieger Publishing Co., Inc., 1964.

JURY, E.I.; PAVLIDIS, T. Analysis of a New Class of Pulse-Frequency Modulates Feedback Systems. **IEEE trans. autom. control**, v. AC-10, n. 1, p. 35 a 43. January. 1965.

JURY, E.I. **Sampled-data control systems**. New-York: John Wiley. 1958.

KATZ, P. **Digital control using microprocessors**. Englewood Cliffs: Prentice-Hall, 293p. 1981.

KATZ, P. ; POWELL, J.D. Sample rate selection for aircraft digital control. **AIAA Journal**, v.13, n.8, p.975-979. Aug. 1975.

KONGUETSOF, L. **Cálculo diferencial e integral**. New York: McGraw-Hill. 885 p. 1974.

KUHFITTIG, P.K.E. **Introduction to the Laplace transform**. New York: Plenum Press, 205 p. 1978.

LABAZIEWICZ, J. **High fidelity quantum gates with ions in cryogenic microfabricated ion traps**. Boston: 164p. Tese de Doutorado - Massachusetts Institute of Technology, setembro de 2008.

LAM, H.Y-F. **Analog and digital filters –** design and realization. New Jersey: Englewood Cliffs. Prentice-Hall, 632 p. 1979.

LAX, P.; RICHTMYER, R.D. Survey of the Stability of Linear Finite Difference Equations. **Communications on pure and applied mathematics**. v. 9, p. 267 a 293. 1956.

LePAGE, W. R. **Complex variables and the Laplace transform for engineers**. New York: Dover. 1961.

LEVINE, W. S. **The control handbook**... New York: IEEE Press e CRC Press. 1996. ISBN 0849385709.

LI, Z.G. ; WEN, C.Y. AND SOH, Y.C. A unified approach for stability analysis of impulsive hybrid systems. In: IEEE CONFERENCE ON DECISION AND CONTROL, 38., 1999, Fenix. **Proceedings...** Fenix: IEEE, p. 4398-4403, v.5, 1999.

MACIEJOWSKI, J.M. **Multivariable feedback design**. New York: Addison Wesley, p. 424. 1989.

MOSTOFI, Y. **Theory of linear systems**. New Mexico: University of New Mexico. Electrical and Computer Engineering Department. ECE500, lecture 27. EUA, outono de 2008. Disponível em: www.ece.unm.edu/~agon/ece500. Acesso em 15/12/2008.

MUTH, E. J. **Transform methods with applications to engineering and operations research**. New Jewrsey: Englewood Cliffs. Prentice-Hall. 1977.

NAYLOR, A.W.; SELL, G.R. **Linear operator theory in engineering and science**. New York: Springer-Verlag, 624 p. 1982.

ODEN, J.T. **Applied functional analysis:** a first course for students of mechanics and engineering science. [S.l.]: Prentice-Hall, 426 p.1979.

OGATA, K. **Discrete-time control systems**. Englewood Cliffs: Prentice-Hall, 984p. 1987.

OGATA, K. **Engenharia de controle moderno**. Rio de Janeiro: Prentice-Hall, 773p. 1993.

OPPENHEIM, A.V.; SCHAFER, R.W. **Digital signal processing**. New Jersey: Englewood Cliffs. Prentice-Hall, 585 p. 1975.

ORTEGA, J.M. Stability of difference equations and convergence of iterative processes. **SIAM, J. Numer. Anal.**, v. 10, n.2, p. 268 a 282. Apr. 1973.

ORTEGA, J.M.; RHEINBOLDT, W.C. On discretization and differentiation of operators with application to newton's method. **SIAM, J. Numer. Anal.**, v. 3, n.1, p. 143 a 156. Mar. 1966.

O'SHEA, R.P.; YOUNIS, M.I. A frequency-time domains stability criterion for sampled-data systems. **IEEE Transactions on Automatic Control**, v. AC-12, n.6, p. 719 a 724. Dec. 1967.

PHILLIPS, C.L.; NAGLE, H. T. **Digital control system analysis and design**. Englewood Cliffs, New Jersey: Prentice Hall International. 1995.

SCHNEIDER, A.M.; ANUSKIEWICZ, J.A.; BARGHOUTI, I.S. Accuracy and Stability of Discrete-Time Filters Generated by High-Order s-to-z Mapping Functions. **IEEE Transactions on Automatic Control**, v.39, n.2, p.435-441. Feb. 1994.

SCHNEIDER, A. M.; GROUTAGE, F. D.; VOLFSON, L. B. S-plane to z-plane mapping using a simultaneous equation algorithm based on the bilinear transformation. **IEEE Transactions on Automatic Control**, v.AC-32, n.7, p.635-637. July. 1987.

SCHNEIDER, A. M.; KANESHIGE, J. T.; GROUTAGE, F. D. Higher order s-to-z mapping functions and their applications in digitizing continuous-time filters. **Proceedings of the IEEE**, v.79, n.11, p.1661-1674. Nov. 1991.

SILVA, A.R. **Estudo de controle de um satélite artificial durante a transferência orbital e apontamento**. São José dos Campos. 108p. (INPE-6397-TDI/613). Dissertação (Mestrado em Engenharia e Tecnologia Espaciais / Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais, 1997.

THOMPSON, P.M.; DAILEY, R.L.; DOYLE, J.C. New conic sector for sampled-data feedback systems. **Systems and Control Letters**, v 7, p. 395 a 404. Sept. 1986.

THOMPSON, P.M.; ATHANS, M.; STEIN, G. Conic sector for sampled-data feedback systems. **Systems and control letters**. v. 3, p. 77 a 82. July. 1983.

THOMPSON, P.M. **Conic sector analysis of hybrid control systems**. Boston: 266p. Tese de Doutorado, sob orientação de Michael Athans. Massachusetts Institute of Technology. EUA, Boston. 1982.

TRIVELATO, G.C. **Controle de rodas de reação através de técnicas digitais usando modelos de referência**. São José dos Campos-SP, Brasil. 209p. (INPE–4618-TDL/335). Dissertação (Mestrado em Engenharia e Tecnologia Espaciais / Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais. Fevereiro. 1988.

TREDINNICK, M.R.A.C. **Controle discreto da atitude de satélites artificiais com apêndices flexíveis**. São José dos Campos-SP, Brasil. 261p. (INPE–7179-TDI/680). Dissertação (Mestrado em Engenharia e Tecnologia Espaciais / Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais. 26 de fevereiro. 1999.

United States Patents and Trademarks Office - USPTO. Patente de invenção de Gilbert P. Hyatt n^{o}. 4942516, **Single chip integrated circuit computer architecture**. EUA, 17 de julho de 1990. Disponível em: http://patimg1.uspto.gov/.piw?Docid=04942516&homeurl=http%3A%2F%2Fpatft.uspto.gov%2Fnetacgi%2Fnph-Parser%3FSect1%3DPTO1%2526Sect2%3DHITOFF%2526d%3DPALL%2526p%3D1%2526u%3D%25252Fnetahtml%25252FPTO%25252Fsrchnum.htm%2526r%3D1%2526f%3DG%2526l%3D50%2526s1%3D4,942,516.PN.%2526OS%3DPN%2F4,942,516%2526RS%3DPN%2F4,942,516&PageNum=&Rtype=&SectionNum=&idkey=NONE&Input=View+first+page

VON NEWMANN, J.; RICHTMEYER, R.D. A method for the numerical calculation of hydrodynamic shocks. **Journal of applied physics**, v. 21, p. 232 a 237. Mar. 1950.

YAMAMOTO, Y. Preservation of stability and spectrum for a class of infinite dimensional systems. **Proceedings IFAC 15TH triennial world congress Barcelona**: 15, 2002.

YAMAMOTO, Y. ; NAGAHARA, M. A new design for sampled-rate converters. In: IEEE CONFERENCE ON DECISION AND CONTROL, 39., 2000, Sydney. **Proceedings…** Sydney: IEEE, 2000. p. 4296 - 4301.

YAMAMOTO, Y. New approach to sampled-data control systems – a function space method. In: IEEE CONFERENCE DECISION AND CONTROL, 29., 1999, Hawai. **Proceedings…** Hawai: IEEE, 1999 a.

YAMAMOTO, Y.; ANDERSON, B.D.O.; MADIEVSKI, A.G. Approximation of frequency response for sampled-data control systems. **Automatica**. v. 35, p.729 a 734. 1999 (b).

YAMAMOTO, Y.; ANDERSON, B.D.O.; MADIEVSKI, A.G. Computation and convergence of frequency response via fast sampling for sampled-data control systems. In: IEEE CONFERENCE DECISION & CONTROL, 36., 1997, San Diego. **Proceedings…**San Diego, California: IEEE, 1997. p. 2157 - 2162.

YAMAMOTO, Y. A retrospective view on sampled-data control systems. **CWI-Quarterly**, v. 9, n. 3, p.261 a 276, 1996.

YAMAMOTO, Y.; HARA, S.; FUJIOKA, H.; KHARGONEKAR, P.P. Computational aspects of gain-frequency response for sampled-data systems. In: IEEE CONFERENCE DECISION & CONTROL, 34., 1995, New Orleans, L.A. **Proceedings…** New Orleans: IEEE, 1995. p.1784 - 1789.

YAMAMOTO, Y. A Function space approach to sampled data control systems and tracking problems. **IEEE Tran. Aut. Control**, v.39, n.4, p.703 a 713. Apr. 1994.

YAMAMOTO, Y. New approach to sampled-data control systems – a function space method. In: IEEE CONFERENCE DECISION & CONTROL, 29., 1990, Honolulu. **Proceedings…** Honolulu, Hawai: IEEE, 1990. p. 1882 - 1887.

XIAO, Y. Stability, controllability and observability of 2-d continuous-discrete systems. **IEEE**, 0-7803-7761—3/03. 2003.

ZHAI, G.; HAI-LIN, A.N.M.; YASUDA, K. Stability analysis for switched systems with continuous time and discrete time subsystems. In: AMERICAN CONTROL CONFERENCE, 2004, Boston, Massaschsetts. **Proceedings…** Boston: IEEE, 2004. p.4555 - 4560.

BIBLIOGRAFIA COMPLEMENTAR

AKHMETOV, M.U.; ZAFER, A.; SEJILOVA, R.D. The control of boundary value problems for quasilinear impulsive integro-differential equations. **Nonlinear Analysis**. Pergamon, vol. 48, p. 271 a 286. 2002.

AKHMETOV, M.U.; ZAFER, A. Successive approximation method for quasilinear impulsive differential equations with control. **Applied Mathematical Letters**, vol. 13, p. 99 a 105. 2000.

AKHMETOV, M.U.; ZAFER, A. Stability of the zero solution of impulsive differential equations by the liapunov second method. **Journal of mathematical analysis and applications**, vol. 248, p. 69 a 82. 2000.

AKHMETOV, M.U.; PERESTYUK, N.A.; TLEUBERGENOVA, M.A. Control over linear pulse systems. Tradução para inglês: **Plenum publishing corporation**. 1996. Ukrainskii Matematicheskii Zhurnal, vol. 47, n.3, p. 307-314. March. 1995.

AKHMETOV, M.U. On the expansion of solutions to differential equations with discontinuous right-hand side in a series in initial data and parameters. Tradução para inglês: **Plenum publishing corporation**. 1994. Ukrainskii Matematicheskii Zhurnal, vol. 45, n.5, p. 715-717. May. 1993.

AKHMETOV, M.U.; PERESTYUK, N.A. Differential properties of solutions and integral surfaces for nonlinear pulse systems. Tradução para o inglês: **Differential equations**, 28, n.4, 445-453. 1992. Differentsial'nye Uravneiya, vol.28, n.4, p.555-566. April. 1992.

ARAKI, M.; ITO, Y.; HAGIWARA, T. Frequency response of sampled-data systems. **Automatica**, vol.32, n.4, p. 483 a 497. 1996.

ARVESON, W. **A Short Course on Spectral Theory**. Springer-Verlag, 2002. 135 p.

ASTRÖM, K.J.; HAGGLUND, T. **Automatic tuning of PID controllers**. ISA – Research Triangle Park, NC- USA. 1988.

BAKER, G.A.; GRAVES-MORRIS, P.; GARRUTHERS, P.A. **Padé approximants**. Addison-Wesley. 1981.

BALAKRISHNAN, A.V. **Applied functional analysis**. Springer-Verlag, 309 p. 1976.

BOOLE, G. **Calculus of Finite Differences**. Chelsea Publ. Company. New York. 1970.

CABADA, A.; POUSO, R. L. On first order discontinuous scalar differential equations. **Nonlinear studies**, vol.6, n.2, p. 161 a 170. 1999.

CERAGIOLI, F.M. **Discontinuous ordinary differential equations and stabilization**. Tese de Doutorado em Matemática. Consorzio delle Universitá di Cagliari. Itália, Firenze, Modena, Perugia e Siena. 1999.

CHANG, C.S. An analytical method for obtaining the root-locus with positive and negative gain. **IEEE Trans. Aut. Control**, vol.AC-10, p. 92 a 94. 1965.

CLARK, S.; LATUSHKIN, Y.; MONTGOMERY-SMITH, S.; RANDOLPH, T. Stability radius and internal versus external stability in banach spaces: an evolution semigroup approach. **SIAM, J. Control Optim.** Vol.38, n.6, p. 1757 a 1793. 2000.

CRAIG, R.R. **Structural dynamics – an introduction to computer methods**. John Wiley & Sons, 1981. 525 p.

CRUZ, J.J. **Controle robusto multivariável**. Ed. Universidade de São Paulo. 1996.

D'AMBROSIO, U. **Métodos da topologia: introdução e aplicações**. Livros Técnicos e Científicos, 119 p. Rio de Janeiro. 1977.

DOBRZYNSKI, L.; PUSZKARSKI, H. Eigenvectors of composite systems: general theory. **J. physics of condensed matter**. V. 1, p. 1239-1245. 1989.

FAN, H. Delta operator based efficient estability tests. **Proc. of the american control conference**, p. 2508 a 2512, Albuquerque, New Mexico. June.1997.

FILIPPOV, A.F. Differential equations with discontinuous right-hand side. Publicado pela **American Mathematical Society** (translations). Traduzido do russo para o inglês por R.N. Goss. Rússia, Moscou. 1964.

GOUZE, J-L. E SARI, T. **A Class of Piecewise Linear Differential Equations Arising in Biological Models**. Institut National de Recherche en Informatique et en Automatique (INRIA), Sophia Antipolis, França. June. 2001.

GREBOGI, C.; BARRETO, E. Multiparameter control of chaos. **Physical review E**, vol.52, n.4, p. 3553 a 3557. October. 1995.

GREBOGI, C.; SHINBROT, T.; OTT, E.; YORKE, J.A. Using small perturbations to control chaos. **Nature**, vol.363, n.3, p. 411 a 417. June. 1993.

GREBOGI, C.; OTT, E.; ROMEIRAS, F.J.; DAYAWANSA, W.P. Controlling Chaotic Dynamical Systems. **Physica D**, vol.58, p. 165 a 192. 1992 (a).

GREBOGI, C.; OTT, E.; YORKE, J.A.; AUERBACH, D. Controlling chaos in high dimensional systems. **Physical review letters**, vol.69, n.24, p. 3479 a 2482. December, 14. 1992 (b).

GREBOGI, C.; OTT, E.; YORKE, J.A. Controlling chaos. **Physical review letters**, vol.64, n.11, p. 1196 a 1199. March. 1990.

GRIMBLE, M.J. **Robust industrial control**. Prentice-Hall, New York. 1994.

HENRICI, P. **Discrete variable methods in ordinary differential equations**. John Wiley & Sons. New York. 1962.

HESS, R.K.; BASS, D.I; BACA, J.B. Flying an experimental supersonic aircraft requires rock-solid flight software. **IEEE spectrum**, p.34 a 39. September. 2004.

HISLOP, P.D.; SIGAL, I.M. **Introduction to spectral theory**: with applications to schrödinger operators. Springer-Verlag, 337 p. 1991.

IAF ASTRODYNAMICS COMMITTEE. Astrodynamics problems of the space station. **Acta Astronautica**, vol.17, n.5, p.491 a 494. 1988.

KABAMBA, P.T. Worst case analysis and design of sampled-data control systems. **IEEE trans. autom. control**, v.39, n.9, p.1337 a 1357. September. 1993.

KANTOROVICH, L.V.; AKILOV, G.P. **Functional analysis**. Pergamon Press, 589 p. 1982.

La SALLE, J.; LEFSCHETZ, S. **Stability by Lyapunov's direct method**. Academic Press. 1961.

LAX, P. **Functional analysis**. John Wiley and Sons, Inc., New York. 563 p. 2002.

LAX, P. **Linear algebra**. John Wiley and Sons, Inc., New York. 247 p. 1997.

LORCH, E.R. **Spectral theory**. Oxford University Press, 157 p. 1962.

MALATERRE, P.O. E RODELLAR, J.; **Design and comparison of multivariable optimal and predictive controllers on a 2-pool irrigation canal**. Technical report. Disponível em:

www.montpellier.cemagref.fr/~pom/papers/gpc1.pdf

MEIROVITCH, L.; ÖZ, H.; JOHNSON JR., C.R. Some problems associated with digital control of dynamical systems. **J. guidance and control**. Vol.3, n.6, p.523-528. Nov.-Dec. 1980 (a).

MEIROVITCH, L. **Computational methods in structural dynamics**. Sijthoff and Noordhoff, p. 439. 1980 (b).

MIRKIN, L.; ROTSTEIN, H.P.; PALMOR, Z.J. H^2 and H^∞ designs of sampled-data systems using lifting – part I: General Framework and Solutions. **SIAM, J. control optim.** V.38 , n.1, p.175 a 196. 1999.

MORARI, M.; ZAFIRIOU, E. Digital controllers for siso systems: a review and a new algorithm. **International journal of control**. V.42 , n.4, p.844-876. 1985.

MORARI, M.; ZAFIRIOU, E. Design of robust digital controllers and sampling-time selection for siso systems. **International journal of control**. V.44 , n.3, p.711-735. 1986.

MOURA, C.A. **Análise funcional para aplicações - posologia**. Editora Ciência Moderna, 217 p. 2002.

NEGREVERGNE, C.; MAHESH, T.S.; RYAN, C.A.; DITTY, M.; RACINE, F.C.; BOULANT, N. Benchmarking quantum control method on a 12-qubit system. **Physical review letters**. Vol.96, p.170501-1 a 170501-4. May. 2006.

OISHI, Y. Asymptotic properties of the best sampled-data control performance. **IEEE transactions on automatic control**. Vol.16, n.7. July. 2001.

OISHI, Y. The best achievable performance of sampled-data control systems with a small sampling period. **IEEE proceedings of the 36th conference on decision and control**, EUA, California, San Diego. December. 1997 (a).

OISHI, Y. A bound of conservativeness in sampled-data robust stabilization and its dependence on sampling periods. **Systems and control letters**, Elservier Science, 37, p.11-19. 1997 (b).

RASBAND, S.N. **Chaotic dynamics of nonlinear systems**. John Wiley and Sons, 230 p. 1990.

SESEKIN, A.N. Dynamic systems with nonlinear impulse structure. **Proceedings of the Steklov institute of mathematics**. Suppl. 2, p. S158-S172. 2000.

SESEKIN, A.N. On connectivity of the set of discontinuous solutions of a dynamic system with impulsive control. **Russian mathematics**, Iz. VUS, vol.40, n.11,p.85-93. 1996.

SESEKIN, A.N. Properties of the attainability set of a dynamic system with impulsive control. **Automation and remote control**. Vol.55, n.2, p.190-195, Plenum Publishing Corporation. 1994.

SEYRANIAN, A.P. **Bifurcation of eigenvalues and stability problems in mechanics**. Notas de aula: IMPA, pós-graduação. Rio de Janeiro, 2006.

SEYRANIAN, A.P.; MAILIBAEV, A.A. Multiparameter stability theory with mechanical applications. **series on stability, vibration and control of systems**, series a. World Scientific. V. 13, p. 401. 2003.

SEYRANIAN, A.P.; KLIEM, W. Bifurcations of eigenvalues of gyroscopic systems with parameters near stability boundaries. **Trans. ASME - Journal of applied mechanics**. V. 68, pp. 199-205. 2001.

SEYRANIAN, A.P. Collision of eigenvalues in linear oscillatory systems **J. appl. math. mech.**. V. 58, No 5, p. 805 a 813. 1994.

SOBOLEV, S.L. **Applications of functional analysis in mathematical physics**. American math. Society. 1963.

OUBRAHIM, R.; LEONARD, F.; **Unified structure of a PID control**. Technical report: ENIM – Laboratoire de Mécatronique Industrielle. France. 1994.

TOIVONEN, H. T. **Advanced control methods**. Technical report: Abo Akademi University. Finlândia. September. 1998.

TOIVONEN, H. T. **Mixed H_2/H_∞ control of sampled-data systems**. Technical report: Abo Akademi University. Finlândia. September. 1996 (a).

Toivonen, H. T.; The Sampled-Data H_∞ Problem: The Equivalence of Discretization-Based Methods and a Riccati Equation Solution. **IEEE Proceedings of the 35nd Conference on Decision and Control**. Kobe, Japan, dezembro 1996 (b).

TOIVONEN, H. T. **Mixed H_2/H_∞ control: synthesis equations for the discrete-time case**. Technical report: Abo Akademi University. Finlândia. June. 1996 (c).

TOIVONEN, H. T.; **A robust H_2 performance problem for discrete-time systems**. Technical report: Abo Akademi University. Finlândia. May. 1996 (d).

TOIVONEN, H. T.; Worst-case sampling for sampled-data H_∞ design. **IEEE proceedings of the 32nd conference on decision and control**. San Antonio, Texas. December. 1993.

WIE, B. **Space vehicle dynamics and control**. AIAA Education Series. Reston, VA. 1998.

ZARANTONELLO, E. H. **Projections on convex sets in hilbert space and spectral theory**. University of Wisconsin, 181 p. 1971.

APÊNDICE A

CONTRIBUIÇÃO CLÁSSICA PARA O ESTUDO DO PROJETO DE CONTROLADORES DISCRETOS

Tal como iniciado no Capítulo 3 dessa Tese, este Apêndice trata das aplicações das técnicas de mapeamento analógico-digital no projeto do controlador digital, com aplicações no controle da posição de osciladores harmônicos.

A.1. Controle Proporcional + Derivativo

Agora será apresentada a mesma política de controle PD, dada analogicamente por

$$D(s) = k_p + k_d.s \tag{A.1}$$

na ótica de diferentes métodos de integração numérica tais como: “forward”, “backward”, Tustin e Schneider. Na expressão (A.1) k_p e k_d são os ganhos de controle proporcional e derivativo, respectivamente.

a) PD por aproximação “forward”:

$$s \sim \frac{z-1}{T_S} \tag{A.2}$$

Substituindo a equação (A.2) na equação (A.1) vem,

$$D(z) = k_p + k_d.\frac{z-1}{T_S} \tag{A.3}$$

que dará,

$$D(z) = \frac{k_d}{T_S}.\left[z + \left(\frac{k_p}{k_d}.T_S - 1\right)\right] \tag{A.4}$$

b) PD por aproximação “backward”:

$$s \sim \frac{z-1}{T_S.z} \tag{A.5}$$

substituindo a equação (B.5) na equação (B.3) vem,

$$D(z) = k_p + k_d.\frac{z-1}{T_S.z} \tag{A.6}$$

que dará,

$$D(z) = \left(k_p + \frac{k_d}{T_S}\right).\frac{z - \dfrac{k_d}{k_p.T_S + k_d}}{z} \tag{A.7}$$

c) PD por aproximação “Tustin”:

$$s \sim \frac{2}{T_S}.\frac{z-1}{z+1} \tag{A.8}$$

Substituindo a equação (B.8) na equação (B.3) vem,

$$D(z) = k_p + k_d.\frac{2}{T_S}.\frac{z-1}{z+1} \tag{A.9}$$

que dará,

$$D(z) = \left(k_p + \frac{2.k_d}{T_S}\right).\frac{z + \left(\dfrac{k_p.T_S - 2.k_d}{k_p.T_S + 2.k_d}\right)}{z+1} \tag{A.10}$$

d) PD por equivalência “Schneider”:

$$D(z) = \frac{\left(5\cdot k_p + 12\cdot \frac{k_d}{T_S}\right).z^2 + \left(8\cdot k_p - 12\cdot \frac{k_d}{T_S}\right).z - k_p}{5.z^2 + 8.z - 1} \tag{A.11}$$

A.2. Oscilador harmônico simples, forçado e não amortecido

A dinâmica de um oscilador harmônico simples não amortecido e com termo forçante u(t) é descrita pela seguinte equação diferencial:

$$m.\ddot{x}(t) + K.x(t) = u(t) \tag{A.12}$$

$$\ddot{x}(t) + \frac{K}{m}.x(t) = \frac{1}{m}.u(t) \tag{A.13}$$

e possui a seguinte função de transferência analógica entre a saída x(t) e o sinal de controle u(t), após aplicar a transformada de Laplace à equação (A.13):

$$G(s) \stackrel{\Delta}{=} \frac{X(s)}{U(s)} = \frac{1}{K} \cdot \frac{\omega_n^{\,2}}{s^2 + \omega_n^{\,2}} \tag{A.14}$$

sendo que $\omega_n^{\,2} = K \diagup m$.

Conforme **Franklin (1981)** o equivalente “zero-order hold” da equação (A.14) é calculado como segue,

$$G_{H0}(z) = (1 - z^{-1}).Z\left\{ \left. \mathrm{L}^{-1}\left\{ \frac{G(s)}{s} \right\} \right|_{t=k.T_S} \right\} \tag{A.15}$$

Aplicando a equação (A.14) na equação (A.15) teremos, simplificando com K = 1,

$$G_{H0}(z) = (1 - z^{-1}).Z\left\{ \left. \mathrm{L}^{-1}\left\{ \frac{\omega_n^{\,2}}{s.(s^2 + \omega_n^{\,2})} \right\} \right|_{t=k.T_S} \right\} \tag{A.16}$$

$$G_{H0}(z) = (1 - z^{-1}).Z\left\{ \left. \mathrm{L}^{-1}\left\{ \frac{1}{s} - \frac{s}{s^2 + \omega_n^{\,2}} \right\} \right|_{t=k.T_S} \right\} \tag{A.17}$$

teremos finalmente, normalizando k=1:

$$G_{H0}(z) = [1 - \cos(\omega_n . T_S)] . \frac{z+1}{z^2 - (2.\cos(\omega_n . T_S)).z + 1} \quad \text{(A.18)}$$

A.3. Oscilador harmônico simples, forçado e amortecido

Se o oscilador harmônico simples possuir amortecimento,

$$m.\ddot{x}(t) + b.\dot{x}(t) + K.x(t) = u(t) \quad \text{(A.19)}$$

sua função de transferência analógica será dada por:

$$G(s) \stackrel{\Delta}{=} \frac{X(s)}{U(s)} = \frac{1}{K} . \frac{\omega_n^{\ 2}}{s^2 + 2.\zeta.\omega_n.s + \omega_n^{\ 2}} \quad \text{(A.20)}$$

analogamente ao caso anterior, o equivalente “zero-order hold” da equação (A.20) será dado por, normalizando K = 1:

$$G_{H0}(z) = \frac{z.\left[1 - 2e^{-\sigma T}.\cos(\omega_d T) - e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d T) - \cos(\omega_d T)\right)\right] + e^{-2\sigma T} + e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d T) - \cos(\omega_d T)\right)}{z^2 - [2.e^{-\sigma T}.\cos(\omega_d T)]z + e^{-2\sigma T}} \quad \text{(A.21)}$$

onde T é o período de amostragem, $\sigma = \zeta.\omega_n$, $\omega_d = \omega_n.\sqrt{1-\zeta^2}$ é a freqüência natural amortecida, ζ é a razão de amortecimento deste modo natural de vibração, e ω_n é a freqüência natural não amortecida deste modo. A expressão (A.21) pode ser reduzida à expressão (B.18) quando $\zeta = 0$.

A.4. PD ANALÓGICO + OSCILADOR HARMÔNICO ANALÓGICO

O controle de um oscilador harmônico é interpretado como um tipo de controle ativo vibracional. A idéia de estudar inicialmente um controle ativo vibracional de um oscilador harmônico para depois aplicá-lo no controle de atitude de um satélite com um apêndice flexível (por ex.: CBERS-1), como foi feito neste trabalho, consiste intuitivamente de que uma situação assintoticamente estável para o controle ativo vibracional poderia ser descrita de forma semelhante com o controle

de atitude do satélite, considerando as vibrações do apêndice flexível como principal elemento perturbador da atitude do satélite.

Conforme ensinado por **Ogata (1993)**, como a ação derivativa de controle possui um zero e nenhum pólo, não é possível realizá-la eletricamente apenas com elementos RLC passivos. Entretanto, usando-se amplificadores operacionais juntamente com resistores e capacitores tal ação de controle torna-se possível praticamente apenas padecendo do problema de ruídos. **Ogata (1993)** também ensina que não há nenhum problema se o controlador PD for aproximado pelo uso de elementos hidráulicos ou pneumáticos, e que o controle PD melhora a resposta transitória (por ex.: fornecendo pequeno sobre-sinal) pois pode ser considerado o paradigma do compensador em avanço de fase.

Para os estudos do controle ativo vibracional considerou-se um sistema de controle em malha fechada tal como o apresentado na Figura A.1 e usou-se um controle PD analógico segundo a equação (A.10) como o D(s). Os osciladores harmônicos livre, segundo a equação (B.18), e amortecido, segundo a equação (A.21), foram os dois casos de G(s) usados.

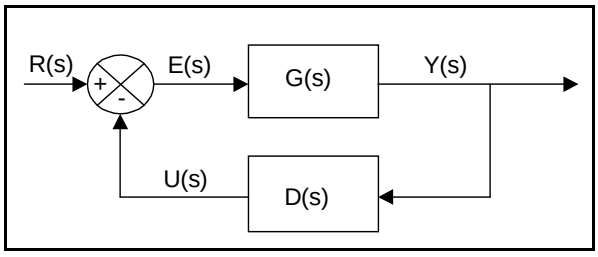


Figura A.1 - Diagrama de blocos do sistema analógico em malha fechada.

Para o caso totalmente analógico desejou-se obter respostas transitórias que obedecessem às seguintes especificações: $t_p \cong 0{,}6$ segundos; $t_s \cong 5$ segundos; $M_P \cong 0{,}15$ Nm. Sendo que t_p é o tempo de pico; t_s é o tempo de acomodação (“setling-time”) e M_P é o sobre-sinal máximo.

A.5. CONTROLE DISCRETO DE UM OSCILADOR HARMÔNICO E SUA ESTABILIDADE

A.5.1. PD + Oscilador Harmônico Livre

O esquema apresentado nesta seção para um sistema de controle PD discreto por aproximação “Tustin” (D(z) da equação (A.10)) de um oscilador harmônico simples livre de amortecimento ($G_{ho}(z)$ da equação (A.21)) possui a seguinte função de transferência de malha fechada (Figura A.2):

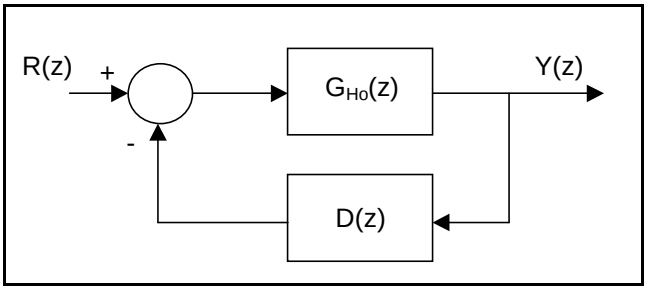


Figura A.2 - Diagrama de blocos do oscilador harmônico controlado por PD discreto no tempo.

A função de transferência em malha-fechada é dada por,

$$H(z)=\frac{Y(z)}{R(z)}=\frac{\dfrac{\left[1-\cos(\omega_n.T)\right].(z+1)}{z^2-2.\cos(\omega_n.T).z+1}}{1\ +\ \dfrac{\left[1-\cos(\omega_n.T)\right].(z+1)}{z^2-2.\cos(\omega_n.T).z+1}\ .\ \dfrac{\left(k_p+2.k_d/T\right).\left(z+\left(\dfrac{k_p.T-2.k_d}{k_p.T+2.k_d}\right)\right.}{(z+1)}} \qquad \text{(A.22)}$$

Sem cancelar pólo do D(z) com zero do $G_{h0}(z)$, para efeito de comparação com a expressão do H(z) com amortecimento, teremos:

$$H(z) = [1-\cos(\omega_n.T_S)].\frac{z^2+2.z+1}{z^3+\left[1-2.\cos(\omega_n.T)+\left(k_p+{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)]\right].z^2+ \ldots}$$

$$\ldots\ \overline{\left[1-2.\cos(\omega_n.T)+\left(k_p+{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)].\left(\frac{k_p.T-2.k_d}{k_p.T+2.k_d}+1\right)\right].z+1+ \ldots}$$

$$\ldots\ \overline{+\left(k_p+{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)].\left(\frac{k_p.T-2.k_d}{k_p.T+2.k_d}\right)} \qquad \text{(A.23)}$$

Se cancelar pólo do D(z) com zero do $G_{h0}(z)$ no denominador da equação (A.23) teremos:

$$H(z) = [1-\cos(\omega_n.T)].\frac{z+1}{z^2+\left[\left(k_p+{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)]-2.\cos(\omega_n.T)\right].z+1+\left(k_p+{}^{2k_d}\!/_{T}\right).} \ \ldots$$

$$\ldots\ \overline{.[1-\cos(\omega_n.T)].\left(\frac{k_p.T-2.k_d}{k_p.T+2.k_d}\right)} \qquad \text{(A.24)}$$

$$H(z) = [1-\cos(\omega_n.T)].\frac{z+1}{z^2+\left[\left(k_p+{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)]-2.\cos(\omega_n.T)\right].z+} \ \ldots$$

$$\ldots\ \overline{+1+\left(k_p-{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)]} \qquad \text{(A.25)}$$

com a seguinte equação característica:

$$1+\left(k_p+{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)].\frac{z+\left(\frac{k_p.T-2.k_d}{k_p.T+2.k_d}\right)}{z^2-[2.\cos(\omega_n.T)].z+1}=0 \qquad \text{(A.26)}$$

ou,

$$z^2+\left[\left(k_p+{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)]-2.\cos(\omega_n.T)\right].z+1+\left(k_p-{}^{2k_d}\!/_{T}\right).[1-\cos(\omega_n.T)]=0 \qquad \text{(A.27)}$$

$$z^2+\left[\left(\frac{k_p.T+2k_d}{T}\right).[1-\cos(\omega_n.T)]-2.\cos(\omega_n.T)\right].z+1+\left(\frac{k_p.T-2k_d}{T}\right).[1-\cos(\omega_n.T)]=0 \quad \text{(A.28)}$$

A.5.2. PD + Oscilador Harmônico Amortecido

O esquema apresentado nesta seção para um sistema de controle PD discreto por aproximação “Tustin” de um oscilador harmônico simples amortecido possui a seguinte função de transferência de malha fechada:

$$H(z)=\frac{z^2.\left[1-2e^{-\sigma T}.\cos(\omega_d.T)-e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d.T)-\cos(\omega_d.T)\right)\right]+z.\left[1-2e^{-\sigma T}.\cos(\omega_d.T)+e^{-2\sigma T}\right]+}{z^3+z^2.\left[1-2e^{-\sigma T}.\cos(\omega_d.T)+\left(k_p+\frac{2k_d}{T}\right).\left(1-2e^{-\sigma T}.\cos(\omega_d.T)-e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d.T)-\cos(\omega_d.T)\right)\right)\right]+} \ldots$$

$$\ldots\frac{+e^{-2\sigma T}+e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d.T)-\cos(\omega_d.T)\right)}{+z.\left[e^{-2\sigma T}+\left(k_p+\frac{2k_d}{T}\right).\left(1-2e^{-\sigma T}.\cos(\omega_d.T)-e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d.T)-\cos(\omega_d.T)\right)\right).\left(\frac{k_p.T-2k_d}{k_p.T+2k_d}\right)+\right.} \ldots$$

$$\ldots\frac{}{-2e^{-\sigma T}.\cos(\omega_d.T)+\left(e^{-2\sigma T}+e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d.T)-\cos(\omega_d.T)\right)\right).\left(k_p+\frac{2k_d}{T}\right)\left.\right]+e^{-2\sigma T}+\left(k_p+\frac{2k_d}{T}\right).} \ldots$$

$$\ldots\frac{}{\left[e^{-2\sigma T}+e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d.T)-\cos(\omega_d.T)\right)\right].\left(\frac{k_p.T-2k_d}{k_p.T+2k_d}\right).} \quad \text{(A.30)}$$

onde T é o período de amostragem, $\sigma=\zeta.\omega_n$, ω_d é a freqüência natural amortecida, sendo ζ o fator de amortecimento deste modo de vibração com freqüência natural não amortecida ω_n. Pode-se verificar facilmente que se $\zeta=0$, caso sem amortecimento, a equação (A.30) se reduz à equação (A.25).

A.5.3. Análise por “Root-Locus” e Resposta Transitória dos Sistemas Descritos

a) Oscilador harmônico simples e forçado: a Figura A.3 mostra o root locus no plano-s para um oscilador harmônico analógico sendo controlado analogicamente por um PD, com ganhos k_p = 3,2 e k_d = 4,8.

A planta G(s) e o controle D(s) são dados pelas equações (A.14) e (A.1), respectivamente. O “Root-locus” que gerou o a Figura A.3 utiliza a equação (A.31) que é a equação característica do sistema em malha-fechada:

$$1 + D(s).G(s) = 0 \qquad \text{(A.31)}$$

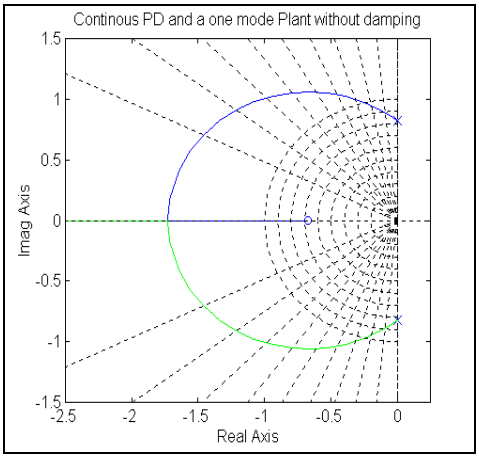


Figura A.3 – “Root-locus” no plano-s para o sistema totalmente analógico da Figura A.1.

A Figura A.4 mostra a respectiva resposta a um impulso unitário.

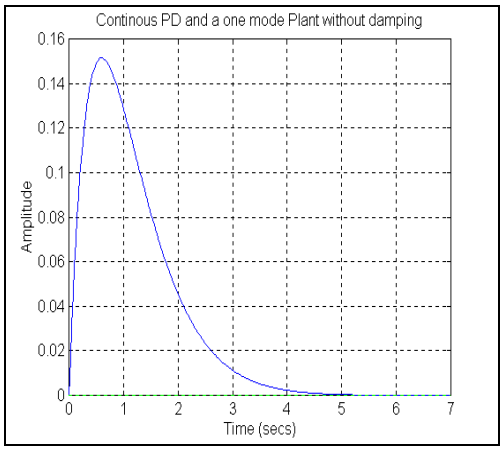


Figura A.4 - Resposta ao impulso unitário para o sistema totalmente analógico.

b) Oscilador harmônico simples sem amortecimento com controle PD projetado por aproximação “Tustin”:

Apliquemos agora um T = 0,1 segundos. Vejamos nas Figuras A.5 e A.6 o que acontece quando um controle PD projetado por aproximação “Tustin” é inserido com um período de amostragem de 0,1 segundos, e nas Figuras A.7 e A.8 para T = 1,6 segundos. Tal sistema discretizado no tempo está mostrado na Figura A.2, e as equações de D(z) e $G_{h0}(z)$ são as equações (A.7).e (A.18), respectivamente.

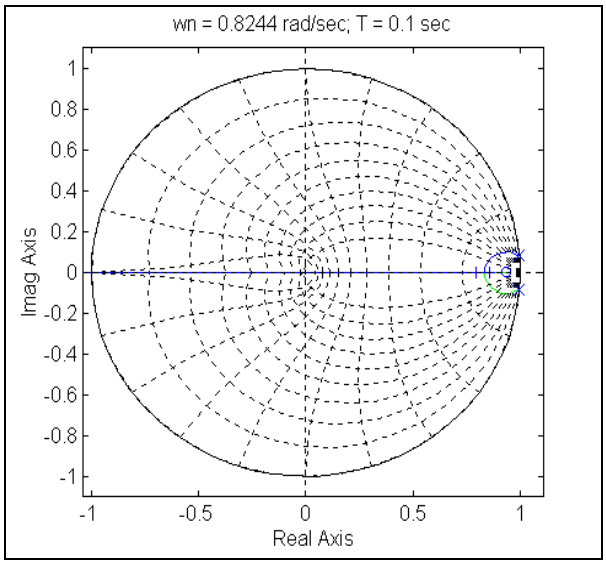


Figura A.5 – “Root-locus” no plano-z para T = 0,1 segundos.

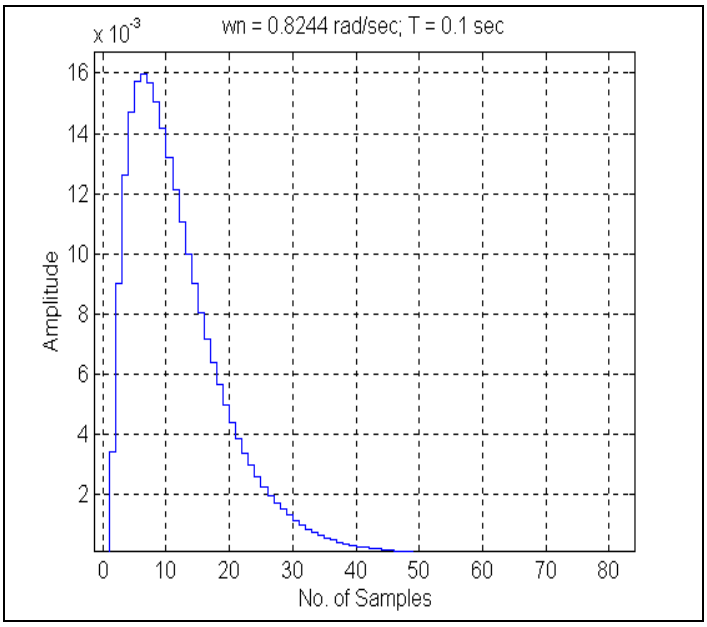


Figura A.6 - Resposta ao impulso unitário para T = 0,1 segundos: claramente temos a estabilidade para este caso.

Conforme pode-se notar mediante a Figura A.6, apesar de ter havido uma degradação do transitório o sistema manteve-se assintóticamente estável, como pode-se ver também pelo “root-locus” mostrado na Figura A.5, com todos os pólos alocados no interior do círculo unitário.

Agora vejamos o que acontece quando um controle PD discreto por equivalência “Tustin” é submetido a um período de amostragem de 1,6 segundos. O “root-locus” de tal sistema e a resposta ao pulso unitário estão mostrados nas nas Figuras A.7 e A.8, respectivamente.

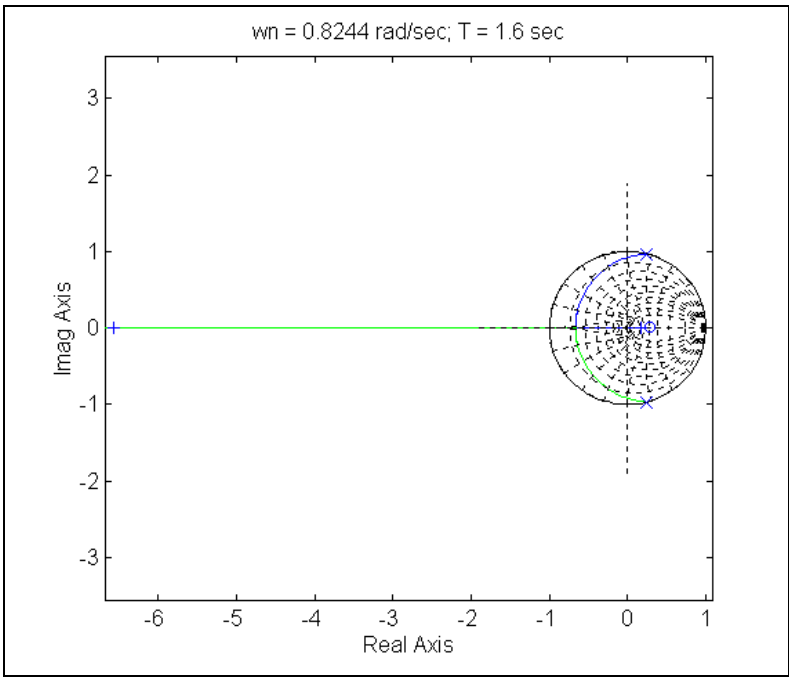


Figura A.7 – “Root-locus” no plano-z para o controle PD discreto por “Tustin”, com T = 1,6 segundos: observar o pólo fora do círculo unitário, representando um fator instabilizante.

Na Figura A.7 pode-se perceber uma cruz indicando a posição do pólo fora do círculo unitário, na extremidade esquerda da figura e sobre o eixo real, descrevendo assim uma situação instável.

De acordo com **The Mathworks Inc. (1995)** a função “rlocus” da ferramenta MatLab (versão 6.5) esboça o “root-locus” do sistema em malha-fechada segundo a seguinte expressão:

$$q = 1 + \kappa . \frac{num}{den} \tag{A.32}$$

onde $\frac{num}{den}$ é a função de transferência da malha-aberta do tal sistema em tela correspondendo, por exemplo, à convolução entre as expressões (A.7) com (A.18) e κ é o vetor de ganhos que é automaticamente determinado, resultando no traçado dos ramos (ou trajetória) do “root-locus”. Se κ = 1 temos a posição dos

pólos de malha fechada do sistema para o ganho inerente à expressão $\frac{num}{den}$ e então poderemos saber pelo “root-locus” se haverá estabilidade ou instabilidade no sistema apenas variando o período de amostragem.

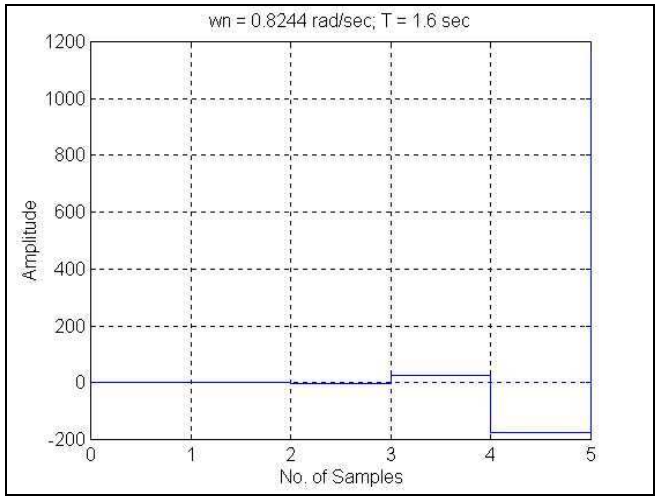


Figura A.8 - Resposta ao pulso unitário para T = 1,6 segundos.

Pode-se notar claramente pela Figura A.8 que o sistema tornou-se instável quando T aumentou de 0,1 para 1,6 segundos.

Podemos analisar o limiar da estabilidade pelo que segue nas Figuras A.9 a A.13.

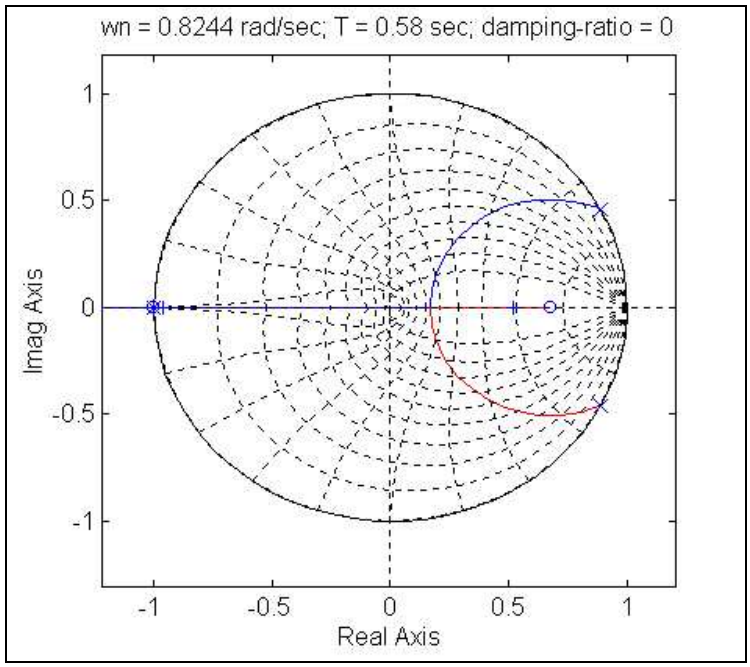


Figura A.9 – “Root-locus” do oscilador-harmônico para T = 0,58 segundos.

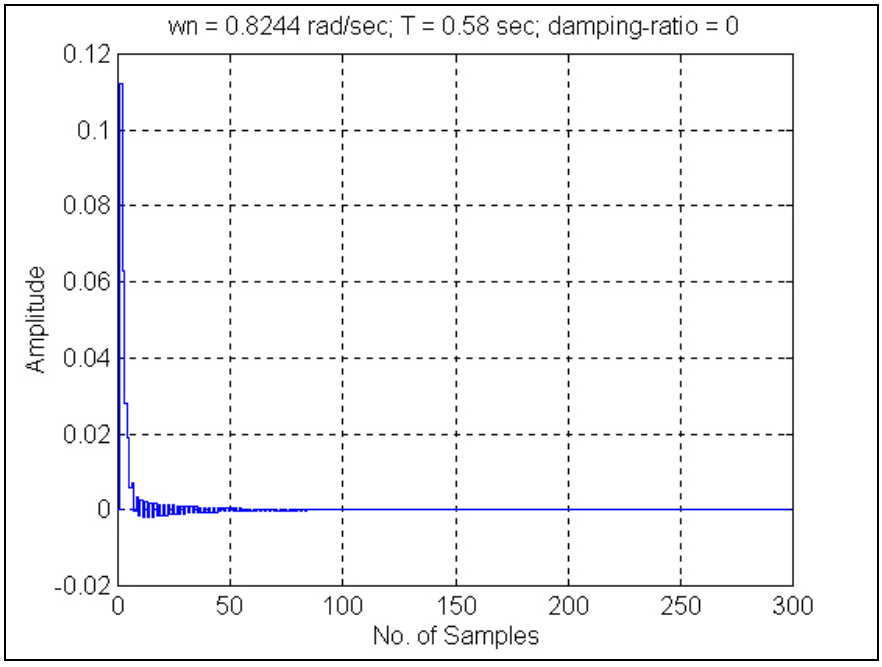


Figura A.10 - Resposta ao pulso unitário para a configuração de pólos do “root-locus” da Figura B.9.

Portanto, até T = 0,58 segundos tal sistema do oscilador-harmônico controlado por um PD projetado pela regra de Tustin é garantidamente estável. A partir desse valor o sistema torna-se instável, tal como pode-se perceber mediante as Figuras A.11 a A.13, onde foi usado um T = 0,59 segundos.

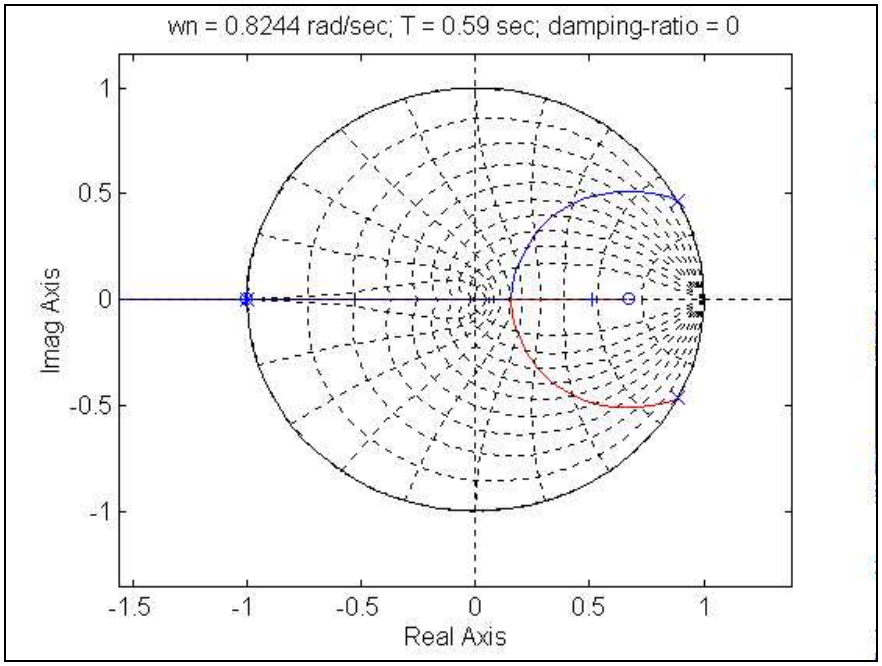


Figura A.11 – “Root-locus” do oscilador-harmônico para T_S = 0,59 segundos.

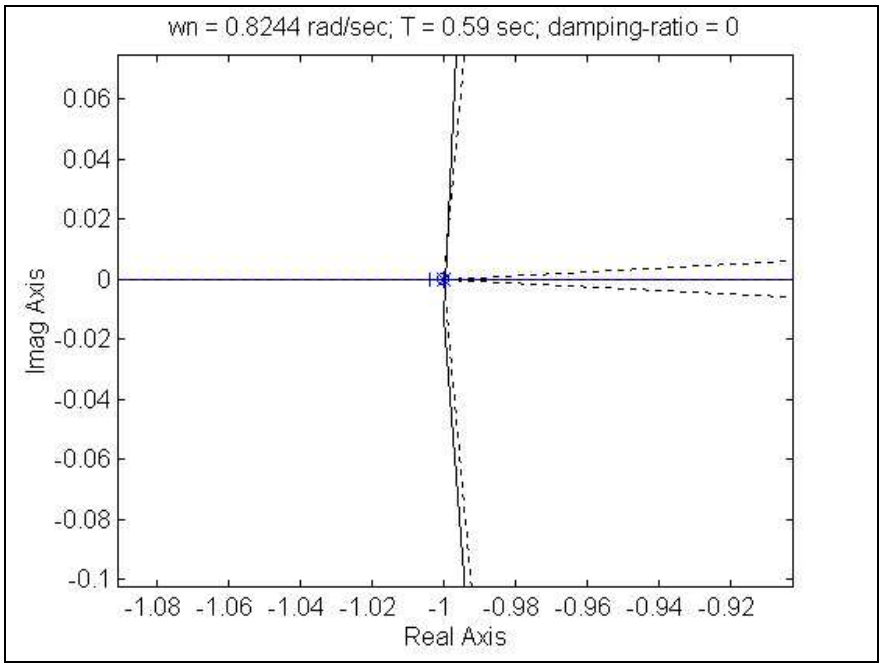


Figura A.12 - Detalhe instabilizador da Figura A.11.

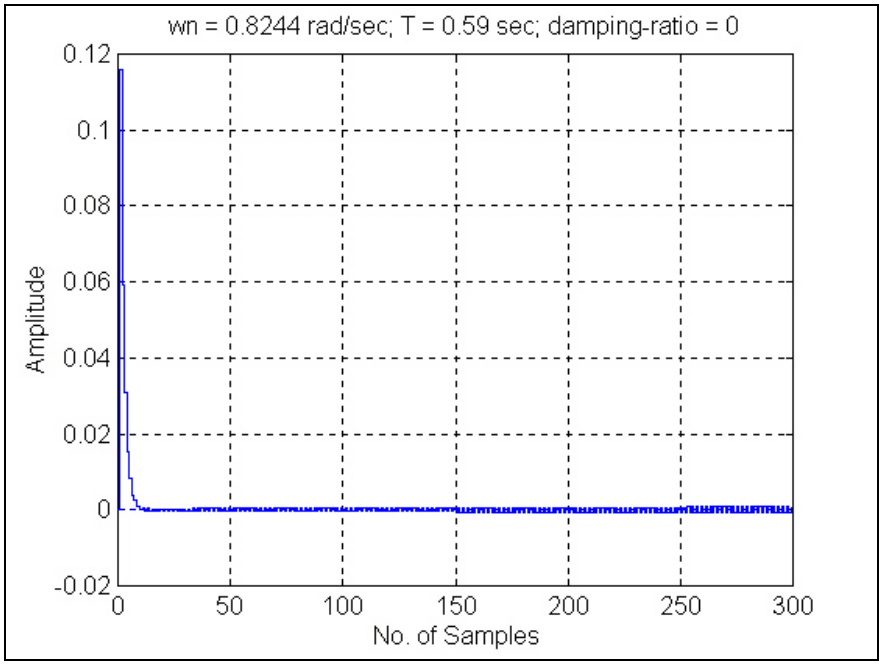


Figura A.13 - Resposta ao pulso unitário relativa aos dois últimos “root-loci”: Figuras A.11 e A.12.

c) Oscilador harmônico simples com amortecimento ζ = 0,1 com controle PD discreto por aproximação “Tustin”, conforme pode-se ver nas Figuras A.14 a A.21.

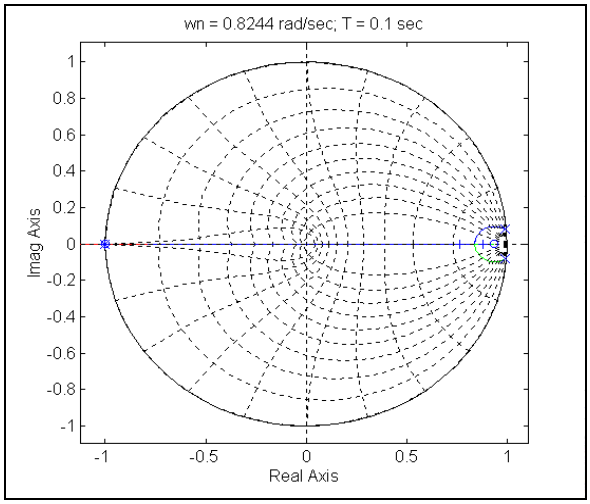


Figura A.14 – “Root-locus” no plano-z para T = 0,1 segundos.

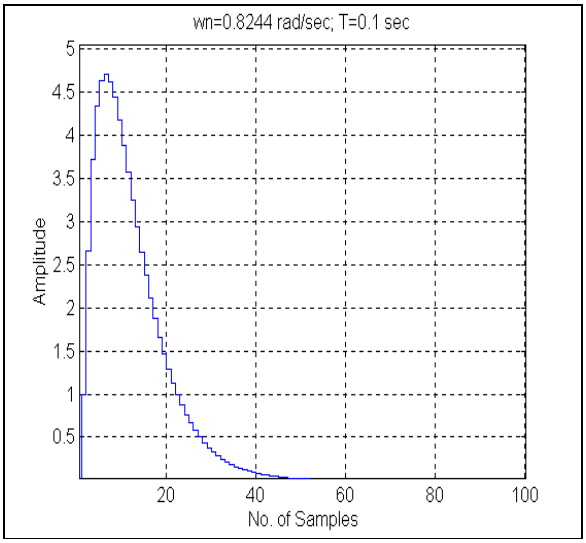


Figura A.15 - Resposta ao impulso unitário para T = 0,1 segundos.

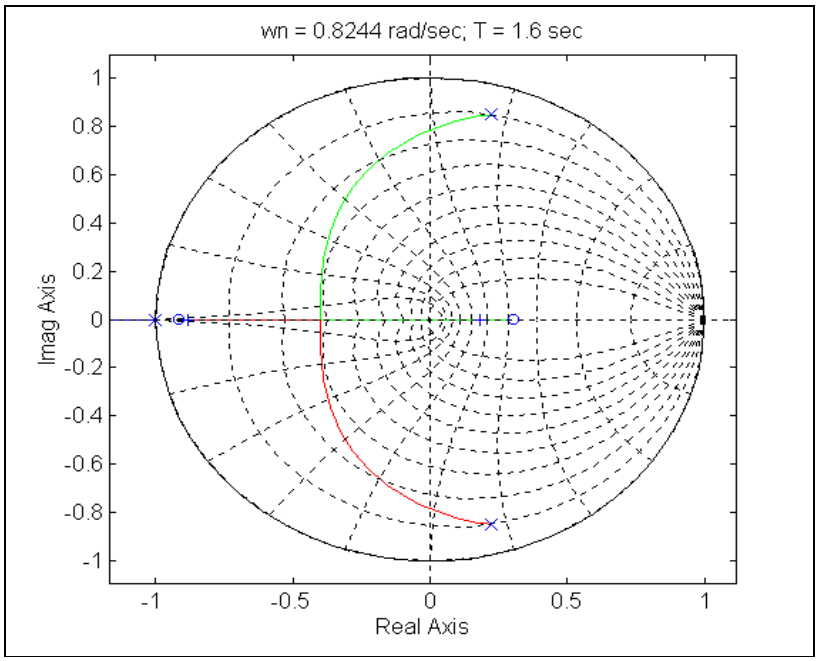


Figura A.16 – “Root-locus” no plano-z para T = 1,6 segundos.

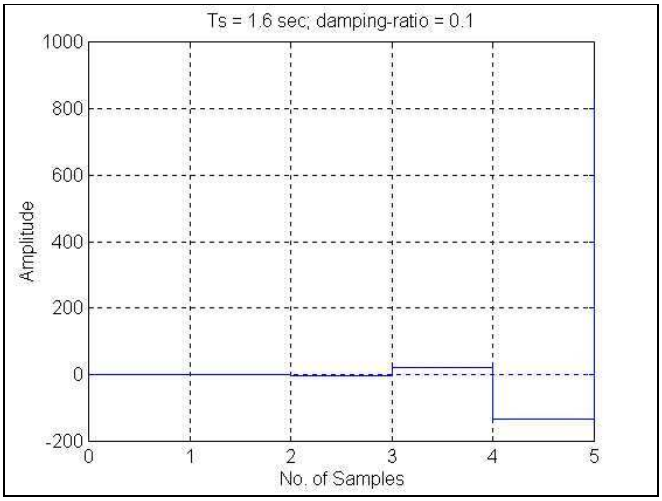


Figura A.17 - Resposta ao pulso unitário para T = 1,6 segundos.

Para o caso do limiar de estabilidade, ainda para o caso amortecido com ζ=0,1:

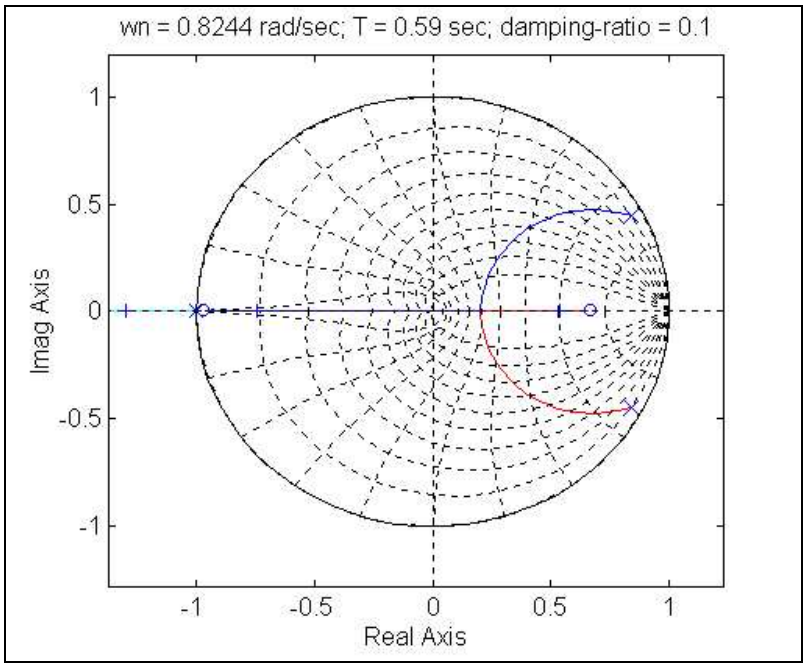


Figura A.18 – “Root-locus” para o caso instável do limiar de estabilidade.

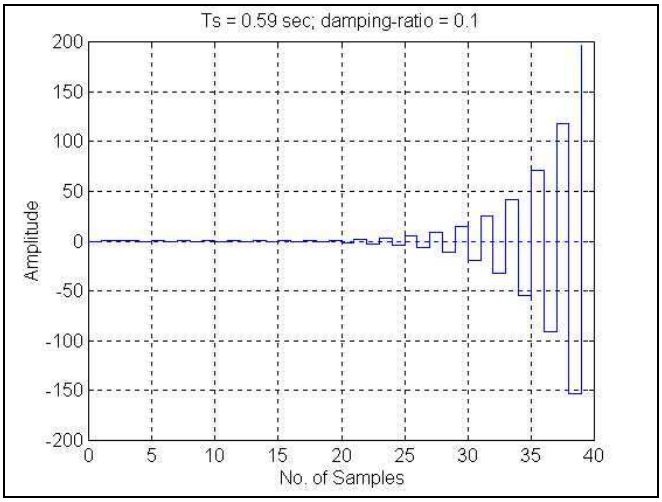


Figura A.19 - Resposta ao pulso unitário para o “root-locus” da Figura A.18.

A instabilidade persiste mesmo com um amortecimento elevado de $\zeta = 0,85$ conforme mostrado nas Figuras A.20 e A.21, para T = 0.59 segundos.

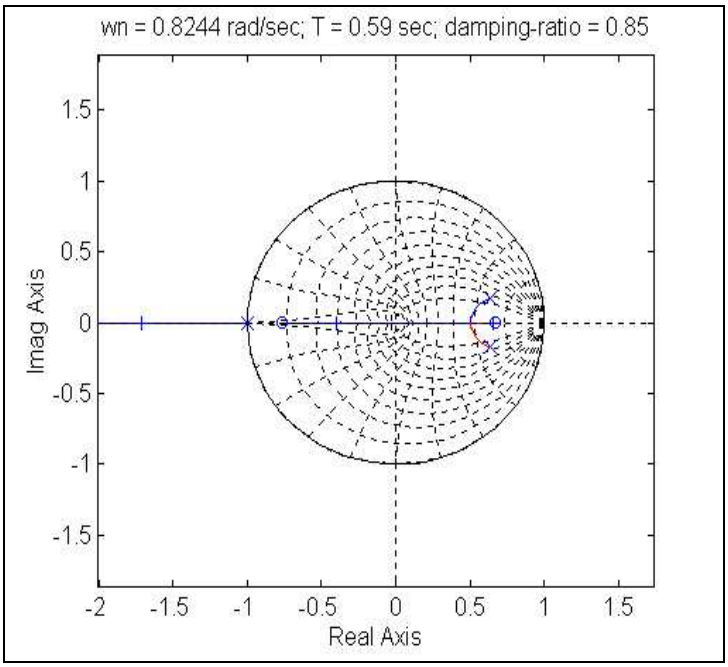


Figura A.20 – “Root-locus” do oscilador harmônico amortecido.

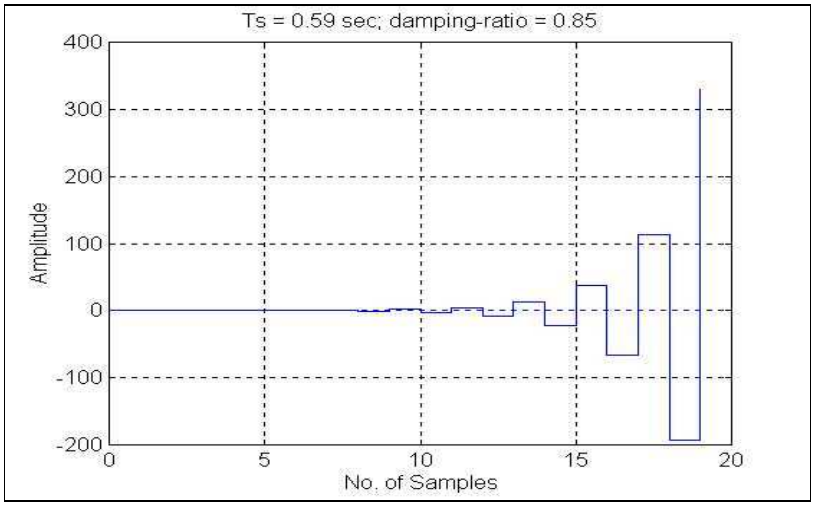


Figura A.21 - Resposta ao pulso unitário para o “root-locus” da Figura A.20.

d) Simulações visando buscar a estabilização mediante uma compensação de fase: poderemos contemplar três comparações entre as simulações com o “zero-order-hold” do oscilador harmônico sobre-amortecido (com razão de amortecimento $\zeta = 6$) controlado por um PD projetado pela regra de Tustin e as simulações com a mesma planta controlada por um PD projetado pela

nova-regra, ou “regra ST”. O objetivo aqui é tentar proporcionar uma compensação de fase no controlador PD tendo em vista o atraso de fase provocado pelo “zero-order hold”.

O atraso provocado pelo “zero-order hold” pode ser calculado pela seguinte maneira (para o primeiro modo do modelo do CBERS-1, usado neste trabalho, no eixo x; $\omega_n = 2.\pi.0{,}1312 = 0{,}8244 \ \text{rad/seg}$; e T = 1,6 segundos):

$$\Psi = \frac{180}{\pi}.\left(2.\pi.0{,}1312.\frac{T_S}{2}\right) = 37^{\circ}{,}78 \qquad \text{(A.33)}$$

que é o atraso de fase provocado pelo “hold” dentro do sistema.

O avanço de fase proporcionado pelo controle PD pode ser calculado como segue:

$$D(s) = k_p + k_d.s \rightarrow D(j\omega) = k_p + j.k_d.\omega \qquad \text{(A.34)}$$

Daí, o avanço de fase do PD será dado por:

$$\phi = arctg\left(\frac{k_d.\omega}{k_p}\right) \qquad \text{(A.35)}$$

Vamos agora nos deter em diversos casos simulados:

d.1) $\phi = 87^{\circ}{,}7777$, o que pode ser conseguido com k_d = 100; k_p = 3,2; $\omega_n = 2.\pi.0{,}1312 = 0{,}8244 \ \text{rad/seg}$; T = 1,6 segundos.

A Figura A.22 descreve uma indesejável situação instável para tal controle de vibração.

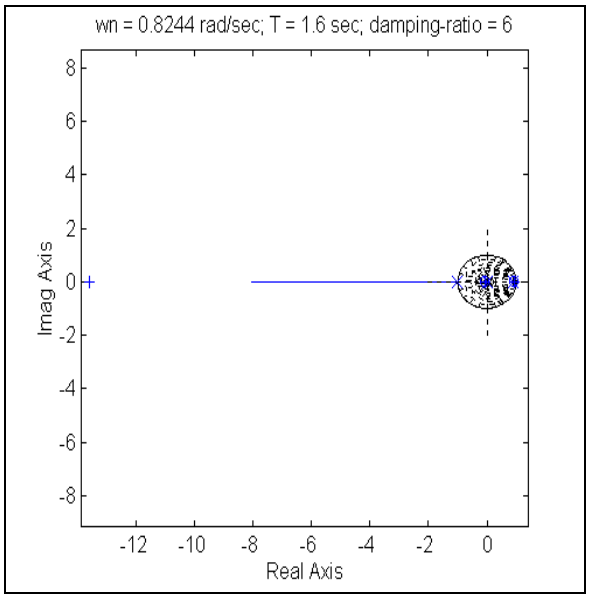


Figura A.22 – “Root-locus” com o Z.O.H. do oscilador harmônico sobre-amortecido controlado por PD projetado com a regra deTustin.

Apesar do avanço de fase mostrado o sistema não conseguiu garantir a estabilidade. Isso se deve ao fato de que a fase depende estritamente da razão entre os ganhos, e, em contrapartida, o ganho de malha aberta depende estritamente dos valores dos módulos dos ganhos individualmente tal como pode ser verificado mediante (A.35).

d.2) $\phi = 57^{o},0983$, o que pode ser conseguido com k_d = 6; k_p = 3,2; $\omega_n = 2.\pi.0,1312 = 0,8244$ rad/seg ; T = 1,6 segundos.

A Figura A.23 também descreve uma semelhante indesejável situação instável para o tal controle de vibração.

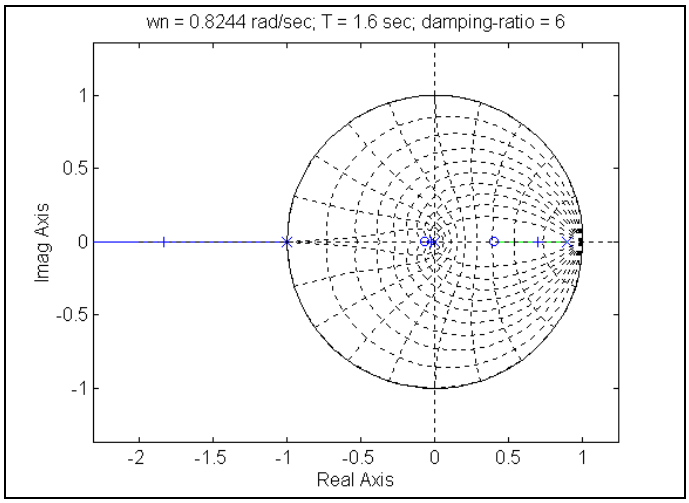


Figura A.23 – Root-locus para o Z.O.H. do oscilador harmônico sobre-amortecido controlado por PD projetado com a regra deTustin.

d.3) $\phi = 51^{o},0370$, o que pode ser conseguido com k_d = 4,8; k_p = 3,2; $\omega_n = 2.\pi.0,1312 = 0,8244$ rad/seg ; T = 1,6 segundos.

A Figura A.24 também descreve uma indesejável situação instável para tal controle de vibração.

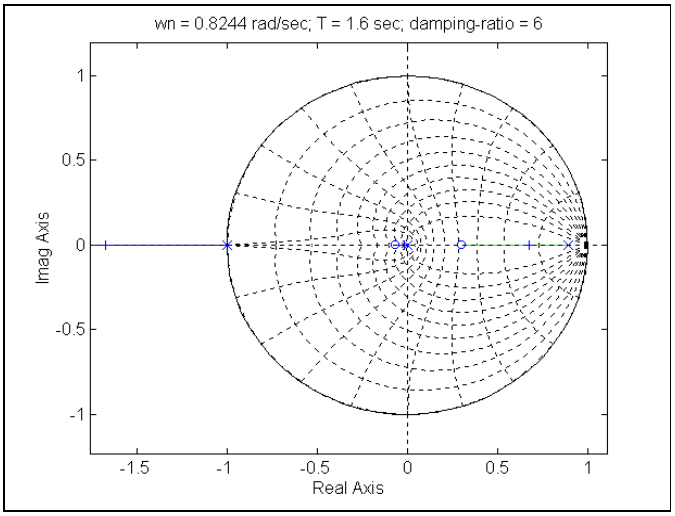


Figura A.24 – Root-locus para o Z.O.H. do oscilador harmônico sobre-amortecido controlado por PD projetado com a regra deTustin.

Uma explicação razoável de porquê o aumento de fase compensada piora o desempenho do sistema de controle em malha fechada) é que os ganhos k_p e k_d influenciam explicitamente no ganho de malha aberta do sistema, tal como pode ser verificado em (A.34) e lembrando que a equação característica é dada por $1+D(z).G_{H0}(z) = 0$, onde D(z) seria a discretização de (A.34).

e) Simulação com PD projetado com a regra de Schneider $s \sim \frac{12}{T_S}.\frac{z^2 - z}{5z^2 + 8z - 1}$ com planta $G_{H0}(z)$ de um oscilador harmônico simples amortecido (Figuras A.25 a A.30).

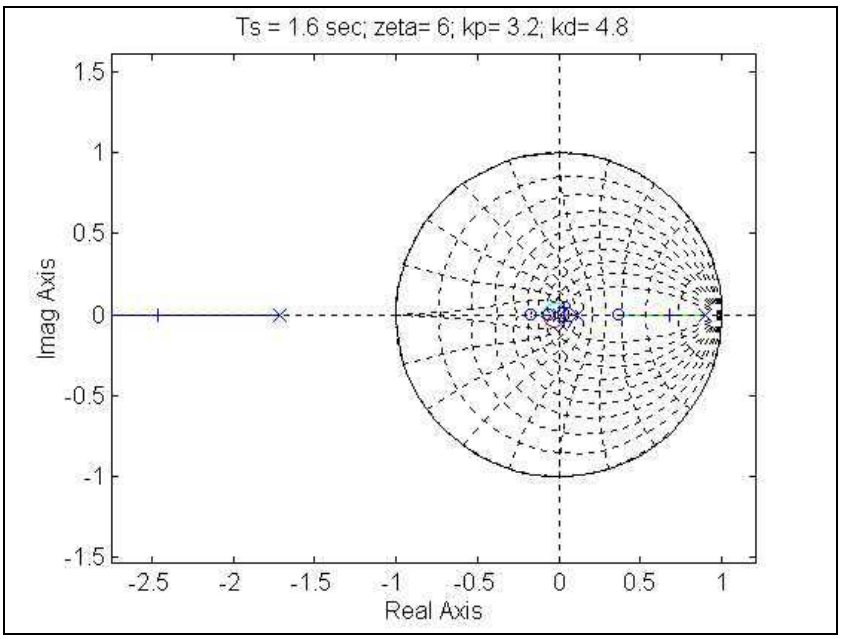


Figura A.25 – “Root-locus” do oscilador harmônico amortecido com PD projetado por Schneider.

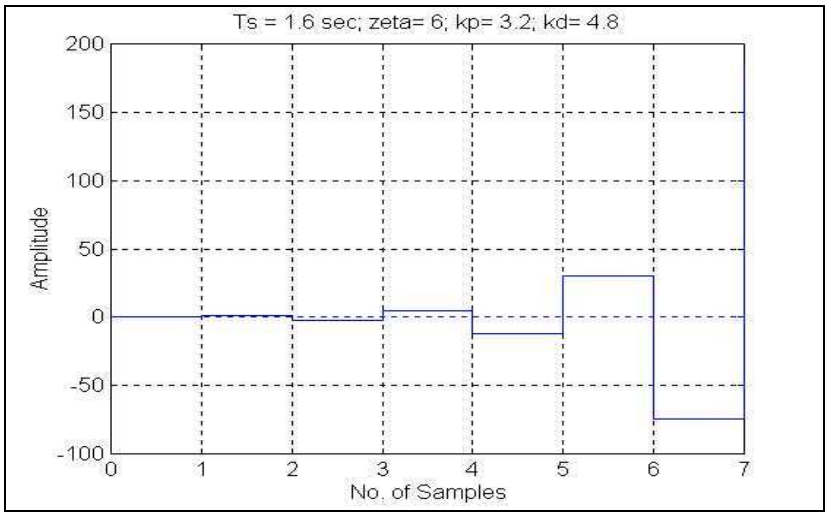


Figura A.26 – Resposta (instável) ao pulso unitário.

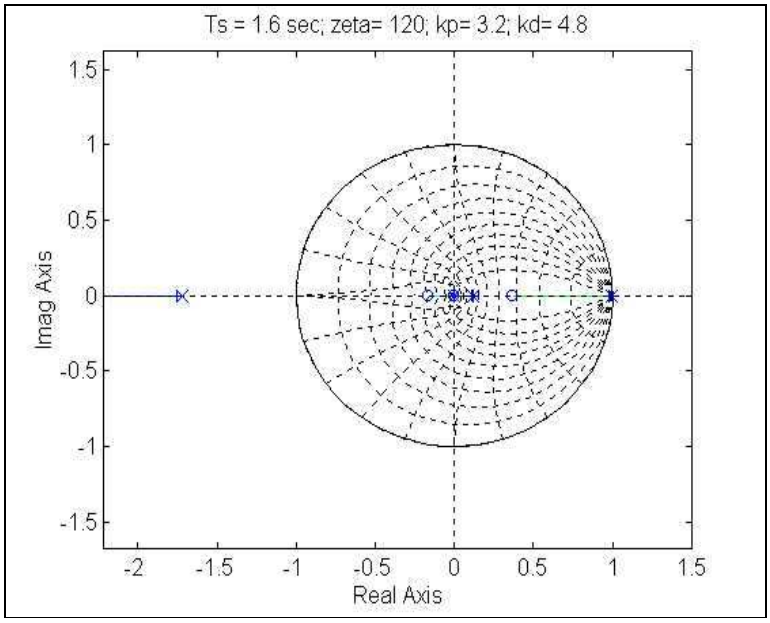


Figura A.27 – “Root-locus” do oscilador harmônico amortecido com PD por Schneider.

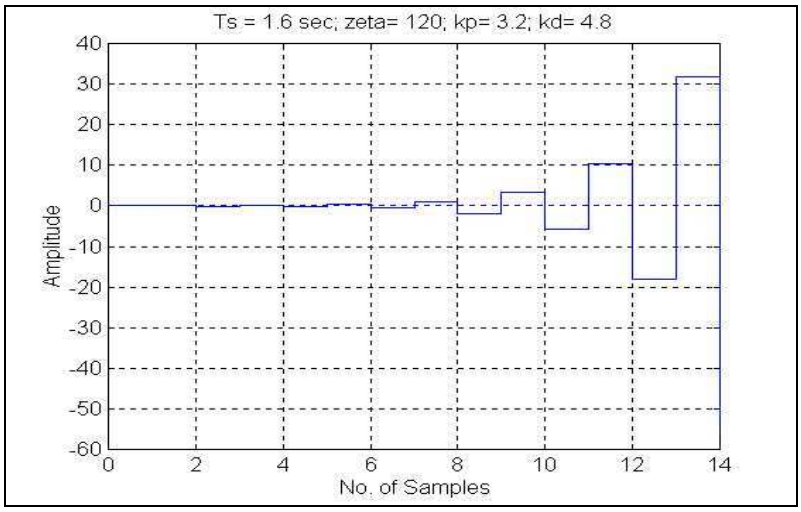


Figura A.28 – Resposta (instável) ao pulso unitário.

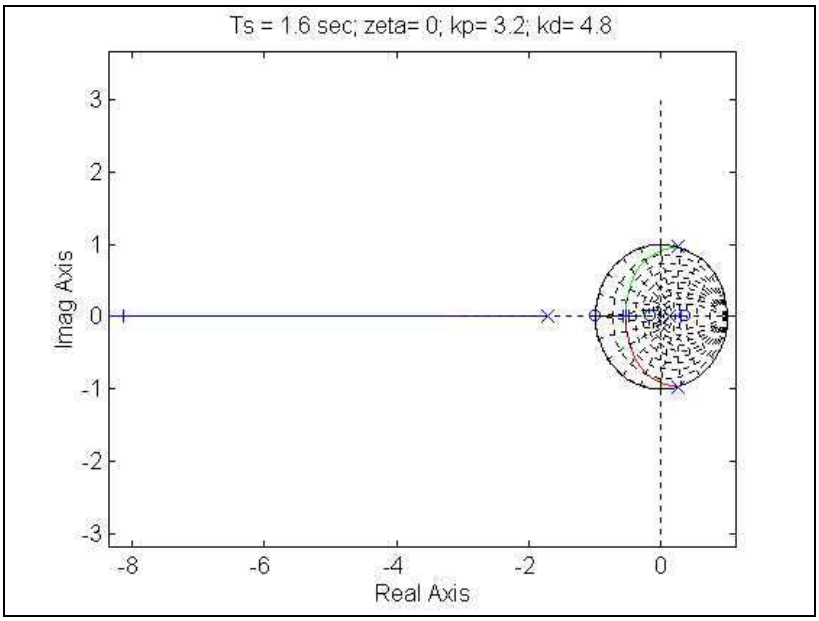


Figura A.29 – “Root-locus” do oscilador harmônico não-amortecido com PD por Schneider.

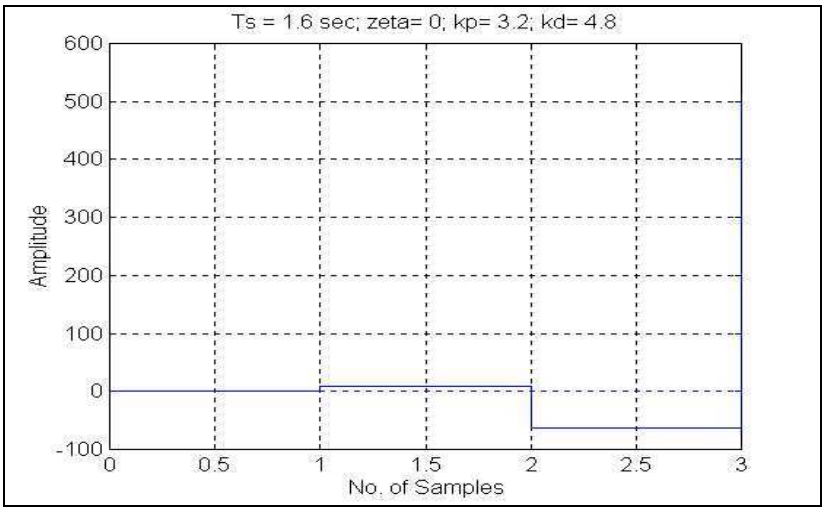


Figura A.30 – Resposta (instável) ao pulso unitário.

APÊNDICE B

Aplicação do Critério de Estabilidade de Jury ao Oscilador Harmônico Simples com Controle PD Mapeado pela Regra de Tustin

Este Apêndice mostra como o critério da estabilidade de Jury (**Ogata, 1987)** pode ser útil para identificar regiões de estabilidade e instabilidade de um sistema de controle discreto em malha-fechada como função do período de amostragem.

Segundo **Isermann (1989)** o critério de estabilidade em tempo-discreto equivalente ao critério de estabilidade de Routh-Hurwitz para sistemas analógicos é conhecido como critério de Schur-Cohn-Jury. Entretanto, **Jury (1964)** verificou que quando o critério de Schur-Cohn-Jury era aplicado a sistemas de ordem elevada uma enorme quantidade de cálculos era necessária, o que o levou a propor um "método de redução" que é conhecido como "critério de estabilidade de Jury". De acordo com **Isermann (1989)** e **Ogata (1987)**, aplicando o critério de estabilidade de Jury para uma dada equação característica em malha fechada P(z) = 0, nós podemos construir uma tabela cujos elementos estão baseados nos coeficientes de P(z). Assim poderemos buscar um método de análise que permita selecionar o período de amostragem T_S de forma a garantir a estabilidade assintótica em sistemas de controle em tempo-discreto. Se o polinômio característico em malha fechada assume a forma:

$$P(z) = a_0 z^n + a_1 z^{n-1} + ... + a_{n-1} z + a_n \qquad \text{(B.1)}$$

Onde $a_0 > 0$; então a tabela do critério de estabilidade de Jury pode ser montada conforme a Tabela B.1.

TABELA B.1 - TABELA DE JURY GENERALIZADA.

z^0	z^1	z^2	z^3	...	z^{n-2}	z^{n-1}	z^n
a_n	a_{n-1}	a_{n-2}	a_{n-3}	...	a_2	a_1	a_0
a_0	a_1	a_2	a_3	...	a_{n-2}	a_{n-1}	a_n
b_{n-1}	b_{n-2}	b_{n-3}	b_{n-4}	...	b_1	b_0	
b_0	b_1	b_2	b_3	...	b_{n-2}	b_{n-1}	
c_{n-2}	c_{n-3}	c_{n-4}	c_{n-5}	...	c_0		
c_0	c_1	c_2	c_3	...	c_{n-2}		
M	M	M					
p_3	p_2	p_1	p_0				
p_0	p_1	p_2	p_3				
q_2	q_1	q_0					

Onde,

$$b_k = \begin{vmatrix} a_n & a_{n-1-k} \\ a_0 & a_{k+1} \end{vmatrix} \qquad ; k = 0,1,2,\ldots,n-1 \tag{B.2}$$

$$c_k = \begin{vmatrix} b_{n-1} & b_{n-2-k} \\ b_0 & b_{k+1} \end{vmatrix} \qquad ; k = 0,1,2,\ldots,n-2 \tag{B.3}$$

M M

$$q_k = \begin{vmatrix} p_3 & p_{2-k} \\ p_0 & p_{k+1} \end{vmatrix} \qquad ; k = 0,1,2 \tag{B.4}$$

Finalmente, o sistema será assintoticamente estável se as seguintes condições forem todas satisfeitas:

1. $|a_n| < |a_0|$ (B.5)

2. $P(z)\big|_{z=1} > 0$ (B.6)

$$3.\ P(z)\big|_{z=-1}\begin{cases} >0 & para\ \ n\ \ par \\ <0 & para\ \ n\ \ ímpar \end{cases} \tag{B.7}$$

$$4.\ \begin{matrix} |b_{n-1}|>|b_0| \\ |c_{n-2}|>|c_0| \\ \mathbf{M} \\ |q_2|>|q_0| \end{matrix} \tag{B.8}$$

Quando usamos, por exemplo, um oscilador harmônico como planta flexível e um PD projetado pela aproximação “Tustin” como controle discreto nós teremos um sistema de segunda ordem cuja equação característica em malha fechada $\mathrm{P(z)=0}$ é:

$$a_0 z^2 + a_1 z + a_2 = 0 \tag{B.9}$$

$$P(z) = z^2 + z.\left[\left(\frac{k_p.T+2.k_d}{T}\right).(1-\cos(\omega_n.T)) - 2.\cos(\omega_n.T)\right] + \left(\frac{k_p.T-2.k_d}{T}\right).(1-\cos(\omega_n.T)) + 1 = 0 \tag{B.10}$$

Neste simples porém importante caso o critério de estabilidade de Jury torna-se:

$$1.\ |a_2|<|a_0|;\ \ 2.\ \ P(1)>0;\ \ 3.\ \ P(-1)>0;\qquad n=2=par \tag{B.11}$$

Da primeira desigualdade de (B.11), obtém-se:

$$i)\ -2<\left(\frac{k_p.T-2.k_d}{T}\right).(1-\cos(\omega_n.T))<0 \tag{B.12}$$

Da segunda desigualdade de (B.11), obtém-se:

$$ii)\ \ k_p>-1 \tag{B.13}$$

$$iii)\ \ \cos(\omega_n.T)<1 \tag{B.14}$$

que é um limitante superior.

Da terceira desigualdade de (C.11), obtém-se:

$$iii)\ \cos(\omega_n.T) > \frac{2.k_d - T}{2.k_d + T} \tag{B.15}$$

que é um limitante inferior.

Como podemos conferir com base na Seção B.5.3 para o caso sem amortecimento, todas situações instáveis apresentadas para o oscilador harmônico simples não amortecido violam o critério de estabilidade de Jury da seguinte forma:

a) instabilidade:

$$1)\ |a_2| < |a_0| \Rightarrow \left| \left(\frac{k_p T - 2k_d}{T} \right).(1 - \cos(\omega_n.T)) + 1 \right| < 1 \tag{B.16}$$

Em números teremos para os ganhos do controle PD projetado pela aproximação Tustin k_p = 3,2 e k_d = 4,8; a freqüência natural não amortecida $\omega_n = 2\pi.0{,}1312$ = 0,8244 (rad/seg) (equivalente ao 1º modo de vibração no eixo x calculado para o modelo do CBERS-1 (Silva, 1997) e um período de amostragem elevado: T = 1,6 segundos:

$$1{,}1025 > 1 \tag{B.17}$$

correspondendo claramente a uma violação do critério de estabilidade, daí vem o porquê da instabilidade.

$$2)\ P(1) > 0 \tag{B.18}$$

Em números,

$$6{,}3075 > 0 \tag{B.19}$$

$$3)\ P(-1) > 0 \tag{B.20}$$

Em números,

$$-6{,}5125 > 0 \tag{B.21}$$

que corresponde a mais uma violação do critério de estabilidade de Jury.

b) Estabilidade:

$$1)\ |a_2| < |a_0| \Rightarrow \left| \left(\frac{k_p T - 2k_d}{T} \right).(1 - \cos(\omega_n.T)) + 1 \right| < 1 \tag{B.22}$$

Em números para os mesmos k_p, k_d e ω_n que o caso anterior mas com um período de amostragem menor de T = 0,1 segundos não teremos nenhuma violação das condições impostas nas inequações (B.11), como podemos verificar mediante as expressões (B.23) a (B.27):

$$0{,}6848 < 1 \tag{B.23}$$

$$2)\ P(1) > 0 \tag{B.24}$$

Em números:

$$0{,}0285 > 0 \tag{B.25}$$

$$3)\ P(-1) > 0 \tag{B.26}$$

Em números,

$$3{,}3411 > 0 \tag{B.27}$$

Vejamos agora simulações numéricas das regiões de estabilidade do sistema de controle de atitude (equivalente ao eixo de “roll”, rolamento) para a seguintes funções $f_1(T_S)$, $f_2(T_S)$, $f_3(T_S)$ relativas às três condições de estabilidade impostas pelo critério de Jury, respectivamente, para um oscilador harmônico livre

controlado por PD discretizado por aproximação “Tustin” com ganhos kp = 3,2 e kd= 4,8. Da primeira das condições de estabilidade de Jury podemos extrair:

$$f_1(T)=\left(\frac{k_p.T-2.k_d}{T}\right).(1-\cos(\omega_n.T))+1 \qquad \text{(B.28)}$$

cuja condição de estabilidade é:

$$-1 < f_1(T) < 1 \qquad \text{(B.29)}$$

Da segunda das condições de estabilidade de Jury temos:

$$f_2(T) = P(1) = 2.(1+k_p).(1-\cos(\omega_n.T)) \qquad \text{(B.30)}$$

cuja condição de estabilidade é:

$$f_2(T) > 0 \qquad \text{(B.31)}$$

Da terceira das condições de estabilidade de Jury temos:

$$f_3(T) = P(-1) = 2.\left(\left(1-\frac{2.k_d}{T}\right)+\cos(\omega_n.T).\left(1+\frac{2.k_d}{T}\right)\right) \qquad \text{(B.32)}$$

cuja condição de estabilidade é:

$$f_3(T) > 0 \qquad \text{(B.33)}$$

As Figuras B.1 a B.6 mostram alguns ensaios realizados para as expressões (B.28), (B.30) e (B.32) em função do período de amostragem T, representando uma forma gráfica que é útil no processo de análise das regiões de estabilidade assintótica.

Observando a Figura B.1 podemos perceber duas marcas verticais que indicam os casos em que T = 0,1 segundos e T = 1,6 segundos. Pela Figura B.1 também pode-se notar que para T = 0,1 segundos estamos obedecendo a condição de

estabilidade assintótica imposta pela equação (B.29), o que não ocorre para T = 1,6 segundos.

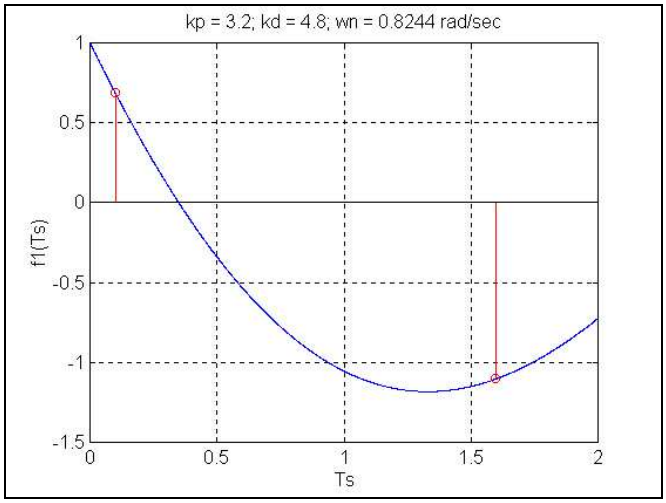


Figura B.1 – Gráfico de $f_1(T)$.

Na Figura B.2 nota-se que tanto T = 0,1 segundos quanto T = 1,6 segundos obedecem à condição de estabilidade assintótica imposta pela equação (B.31).

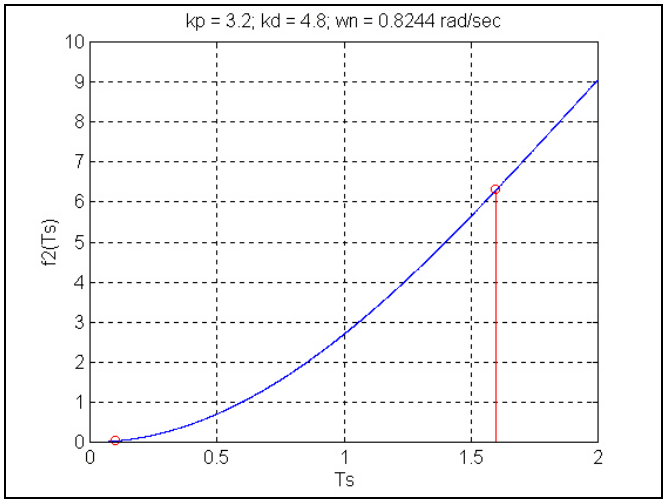


Figura B.2 – Gráfico de $f_2(T)$.

A Figura B.3 indica que para T = 0,1 segundos a estabilidade assintótica estará garantida, visto que a condição (B.33) foi obedecida, o que não ocorreu quando T= 1,6 segundos.

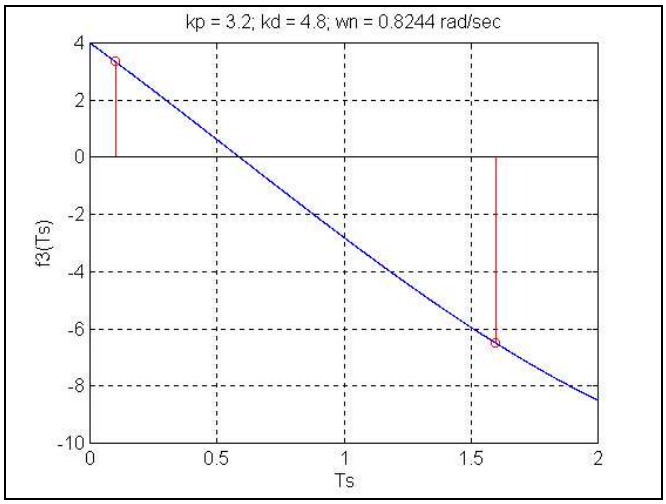


Figura B.3 – Gráfico de $f_3(T_S)$.

As figuras acima sugerem que o controlador digital em questão trabalhará em segurança se o período de amostragem pertencer, por exemplo, à faixa mostrada na inequação (B.34).

$$0 < T < 0{,}34 \text{ [segundos]} \qquad \text{(B.34)}$$

Agora serão apresentados gráficos considerando casos em que T possui valores muito grandes, sendo muito claro o porquê da instabilidade para esses casos, pois há violações na 1ª e 3ª condições de estabilidade (bastaria uma só violação):

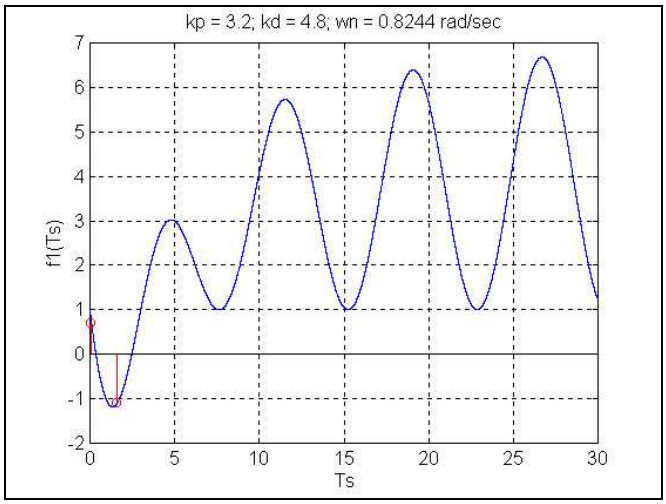


Figura B.4 – Gráfico de $f_1(T)$.

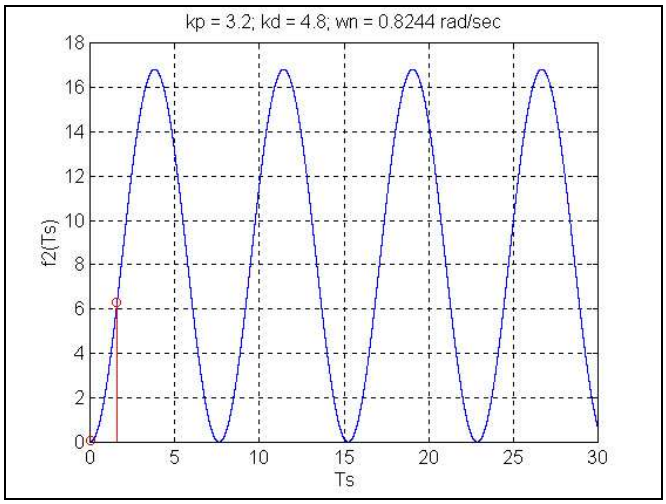


Figura B.5 – Gráfico de $f_2(T)$.

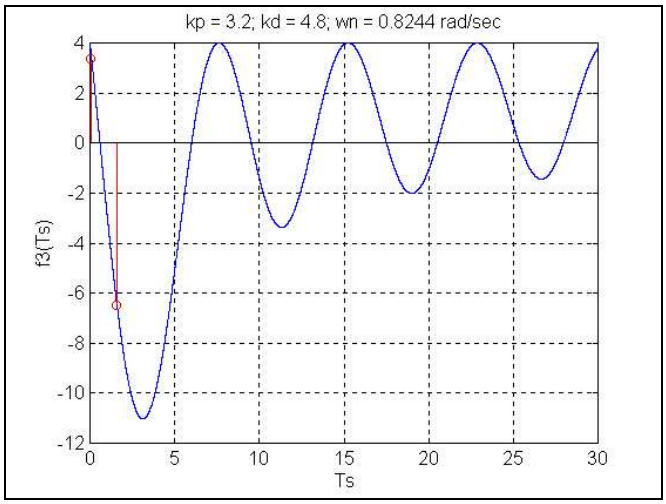


Figura B.6 – Gráfico de $f_3(T)$.

Pode-se prever o comportamento das três funções $f_1(T)$, $f_2(T)$ e $f_3(T)$ quando T tende a valores muito elevados aplicando-se nas equações (B.28), (B.30) e (B.32) o limite quando T tende para infinito. Como resultados temos: a) $f_1(T)$ oscilará entre 1 e $(1+2.k_p)$; b) $f_2(T)$ oscilará entre 0 e $4(1+k_p)$; c) $f_3(T)$ oscilará entre 0 e 4.

APÊNDICE C

SIMULAÇÕES NUMÉRICAS COM UM MODELO DO SATÉLITE CBERS-1

Este Apêndice apresenta resultados obtidos com simulações numéricas do controle digital da atitude de um modelo do satélite assimétrico CBERS-1 (“China-Brazil Earth Resources Satellite”). Mediante tais resultados pode-se verificar os esforços significativos para manter a estabilidade da atitude do satélite mediante as técnicas usuais ou clássicas de controle quando o satélite era obrigado a utilizar grandes valores de período de amostragem. Para aumentar a preocupação dos envolvidos com tais resultados numéricos em tal situação nenhuma das técnicas de controle usuais ou clássicas se mostrou eficaz, obrigando a uma saída drástica e até então não imaginada: a criação de uma nova regra de mapeamento analógico-digital. Para nossa felicidade tal nova regra, obtida experimentalmente, resolveu o problema da instabilidade. A posteriori, um tratamento teórico adequado foi realizado e uma extensão a dois parâmetros de sintonia da primeira nova regra a um parâmetro foi apresentada (ver Capítulo 3).

C.1. Um Modelo Analógico da Estrutura do Satélite CBERS-1

O modelo analógico da estrutura do satélite CBERS-1 usando o método dos modos assumidos foi elaborado por Silva (1997), em sua dissertação de mestrado, e é capaz de executar as simulações para até 5 modos de vibração, considerando as equações dinâmicas acopladas deste sistema MIMO.

A simulação usando um modelo do CBERS-1 (um satélite assimétrico) utilizou os seguintes dados estruturais:

- massa total: 1400 kg;
- massa do apêndice flexível: 49 kg;
- comprimento do apêndice flexível: 6,135 m;
- tensor de inércia do corpo rígido:

$$I_0 = \begin{bmatrix} 1983 & -12{,}8 & 21{,}5 \\ -12{,}8 & 1002 & 9{,}8 \\ 21{,}5 & 9{,}8 & 1831 \end{bmatrix} \qquad \text{(C.1)}$$

- tensor de inércia do apêndice flexível:

$$I_a = \begin{bmatrix} 720 & 0 & 0 \\ 0 & 22{,}744 & 0 \\ 0 & 0 & 698{,}06 \end{bmatrix} \qquad \text{(C.2)}$$

Pela Equação (C.2) pode-se notar que apesar dos cálculos terem sido feitos para o modelo de uma viga de Euler-Bernoulli delgada, foi considerada inércia ao redor do eixo y (eixo de "pitch" $I_{a22} = 22{,}744$) visando forçar a simulação a trabalhar como se tivesse uma placa, o que é interessante tendo em vista a complexidade dos cálculos de modelamento de placas.

A Figura C.1 mostra a relação entre os eixos X,Y e Z com os eixos de "roll" (rolamento), "pitch" (cabragem) e "yaw" (direção) que serão úteis na interpretação dos resultados apresentados neste apêndice.

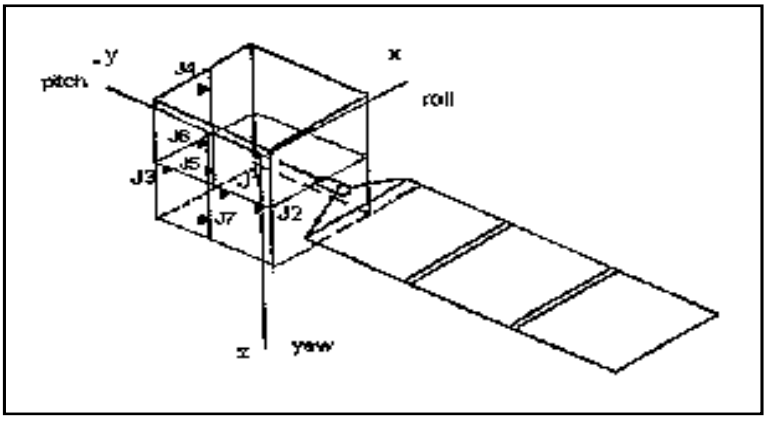


Figura C.1 – Eixos de atitude para a simulação com um modelo do satélite CBERS-1.

FONTE: Silva (1997, p.10).

Conforme Silva (1997), foram consideradas nas equações dinâmicas as perturbações devido ao arrasto atmosférico, à pressão de radiação solar e os

resíduos magnéticos. Tais perturbações afastam os eixos de atitude de suas condições iniciais consideradas nulas (0 radianos). Para corrigir o efeito provocado pelas perturbações o controle de atitude age. Tal controle é dado por torques e nas simulações deste trabalho não foram considerados limites para os mesmos. Entretanto, convém afirmar que uma simulação mais próxima da realidade poderia considerar limites para os sinais de controle, como apresentado por Silva (1997) que usou um torque de controle de $\pm 0{,}9$ N.m. Entretanto, o principal objetivo deste trabalho centra-se na análise da estabilidade do sistema de controle em tempo discreto, não importando, portanto, tais limites.

Aqui serão apresentadas várias simulações do controle de atitude para um modelo discreto do satélite CBERS-1, simulado com 5 modos de vibração para o apêndice flexível com razão de amortecimento igual a 0,02 para todos os modos. Essa convenção não está em consonância com a realidade visto que a razão de amortecimento cresce, segundo alguma lei, à medida que as freqüências dos modos crescem. Mas, compreende-se que se trata de uma simplificação aceitável em termos de simulação, apresentando, pois, situações mais arriscadas que o caso real, isto é, testando a política de controle para os cinco primeiros modos de vibração com a razão de amortecimento do primeiro modo (a mais tênue). Em todas as simulações, em ambiente MatLab 6.5 / Simulink 3.1c, foi usado um integrador Runge-Kutta 4-5 com passo fixo de 0,01 segundos (100 Hz) e os seguintes ganhos de controle PD: a) caso 1: roll: kp=0,0316; kd=0,0482; pitch: kp=0,107 ;kd=0,163; yaw: kp=0,300; kd=0,460; b) caso 2: roll: kp=3,16; kd=4,82; pitch: kp=10,7 ;kd=16,3; yaw: kp=30,0; kd=46,0. Todas as simulações foram feitas com o intuito de efetuar futuras comparações entre as mesmas.

A razão de utilizarmos passo de integração fixo de 0,01 segundos é simples: para que o computador consiga simular o modo de freqüência mais elevada é preciso que possua uma freqüência de integração no mínimo igual ao dobro da freqüência deste modo. Assim, podemos evitar os fenômenos do falseamento (mascaramento ou “aliasing”) e das oscilações escondidas (“hidden oscilations”) durante o processo de simulação. Como podemos verificar na Tabela C.1 abaixo, a maior freqüência de modo é cerca de 25,7241 Hz implicando que estaremos livres do

falseamento e das oscilações escondidas se no mínimo uma freqüência de 51.4482 Hz for utilizada como freqüência do integrador da simulação, com freqüência de Nyquist de 25,7241 Hz. Utilizando 100 Hz de freqüência do integrador (cuja frequência de Nyquist é 50 Hz) nos asseguramos de que todos os cinco modos, nos três eixos, serão simulados no computador sem problemas. Os inconvenientes são: o tempo para o término de cada simulação se estende em demasia; e o volume de dados armazenados torna-se grande. Na Tabela C.1 os eixos x, y, z correspondem aos eixos de “roll”, “pitch” e “yaw”, respectivamente, conforme estabelecido neste trabalho.

TABELA C.1 - MODOS DE VIBRAÇÃO PARA UM MODELO ASSIMÉTRICO DO CBERS-1

Modo	Eixo	Freqüência (Hz)	Razão de amortecimento
Rígido	x	0	0
	y	0	0
	z	0	0
1	x	0.1312	0,02
	y	0.1331	0,02
	z	0.4526	0,02
2	x	0.6874	0,02
	y	0.6883	0,02
	z	1.8999	0,02
3	x	1.9005	0,02
	y	2.8361	0,02
	z	3.7141	0,02
4	x	3.7146	0,02
	y	6.1346	0,02
	z	6.1350	0,02
5	x	7.9411	0,02
	y	15.5614	0,02
	z	25.7241	0,02

O sistema de simulação com controle discreto no tempo (Figura C.2) possui os blocos mostrados nas Figuras C.3 a C.10.

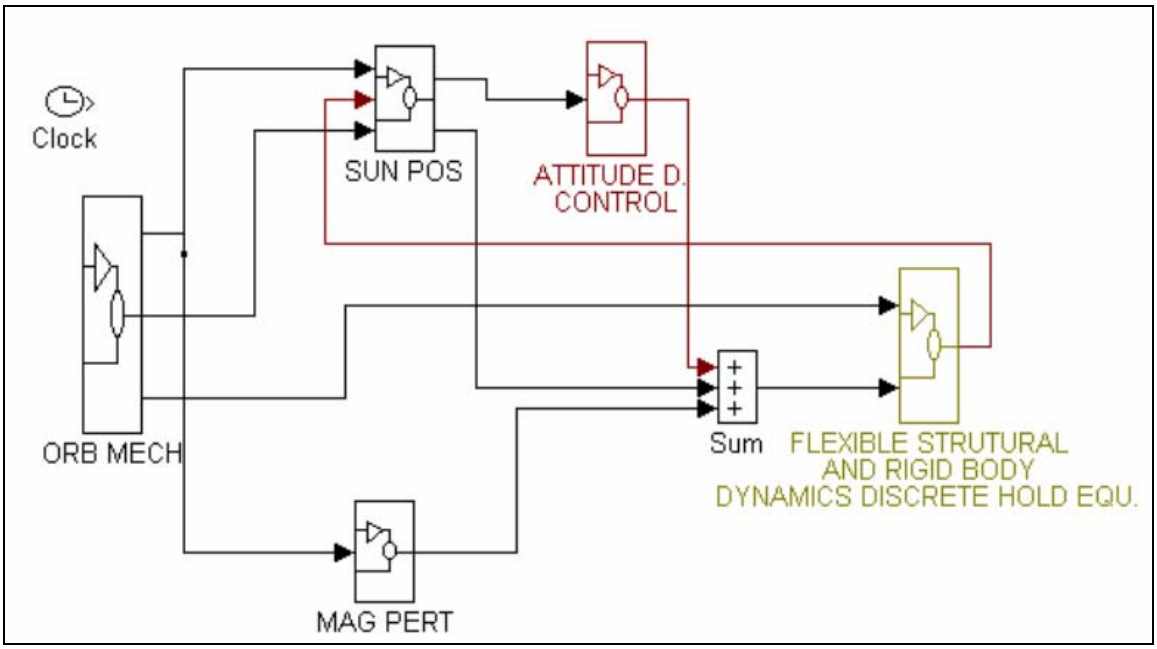


Figura C.2 – Sistema originalmente desenvolvido por Silva (1997) para o caso totalmente analógico (planta e controle), agora com controle discreto no tempo para atender às simulações de atitude.

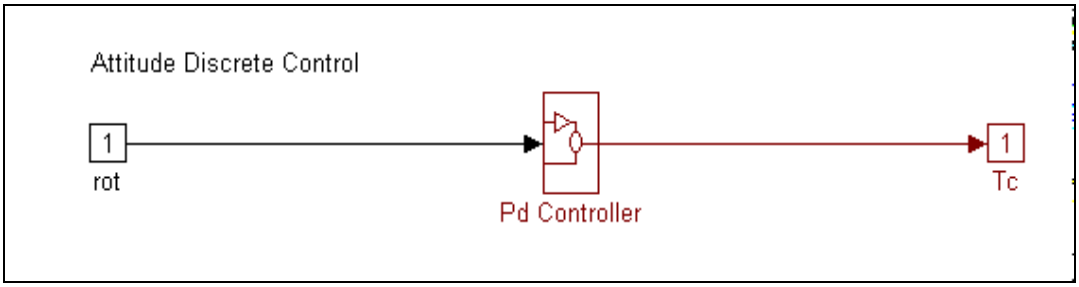


Figura C.3 – Bloco de controle nos três eixos.

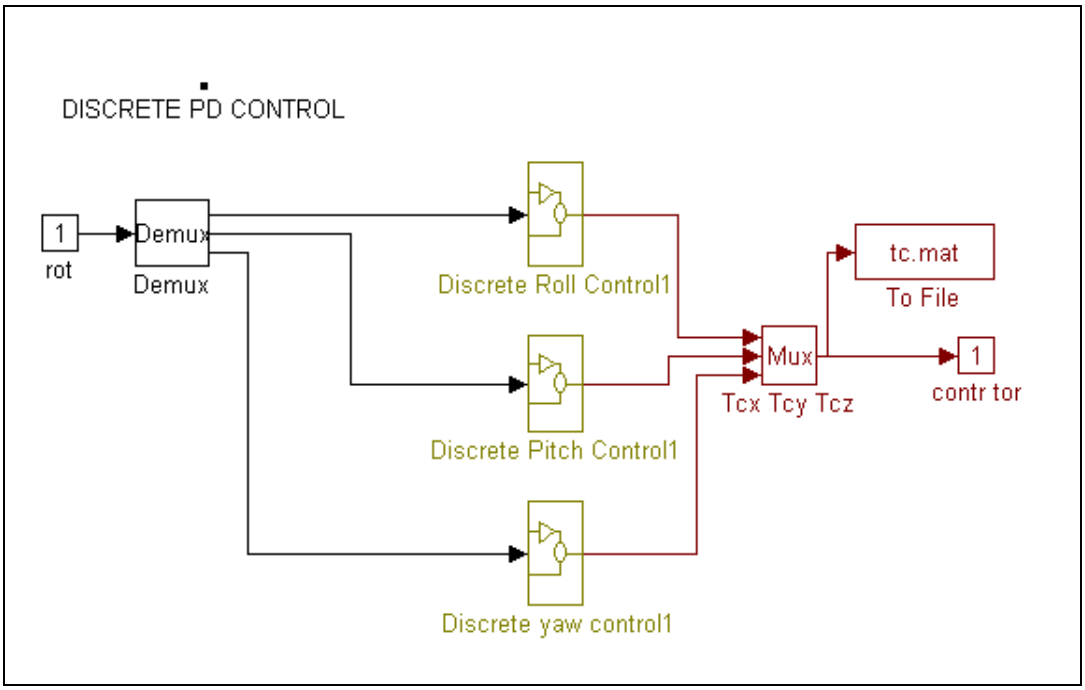


Figura C.4 – Estrutura interna do bloco de controle da Figura D.3.

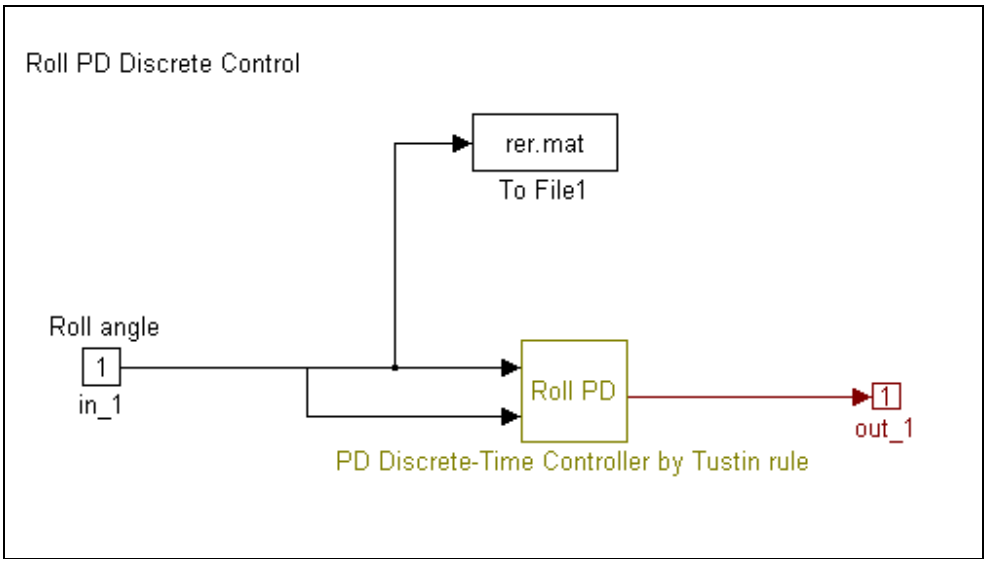


Figura C.5 – Detalhe para o controle de “roll”.

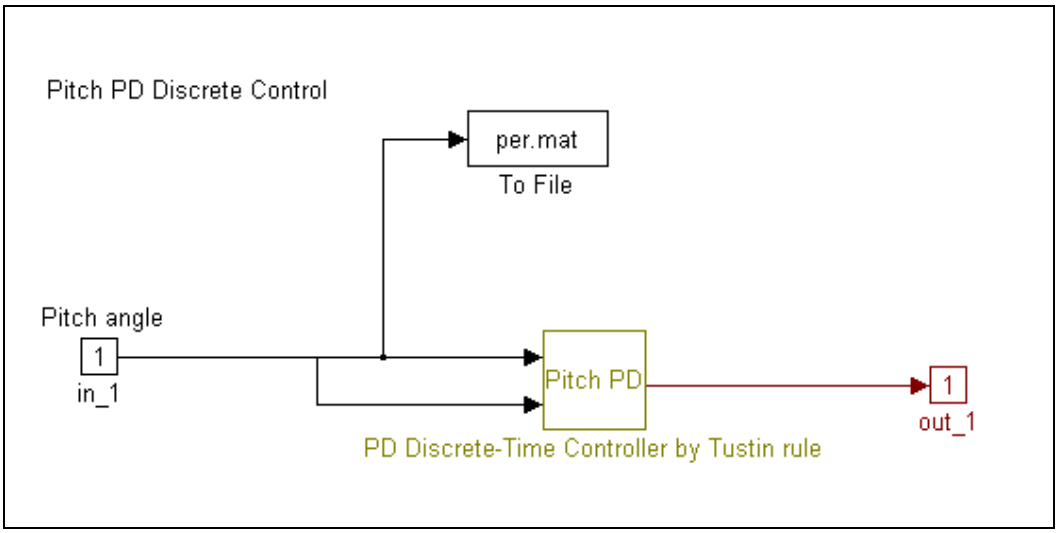


Figura C.6 – Detalhe do controle de “pitch”.

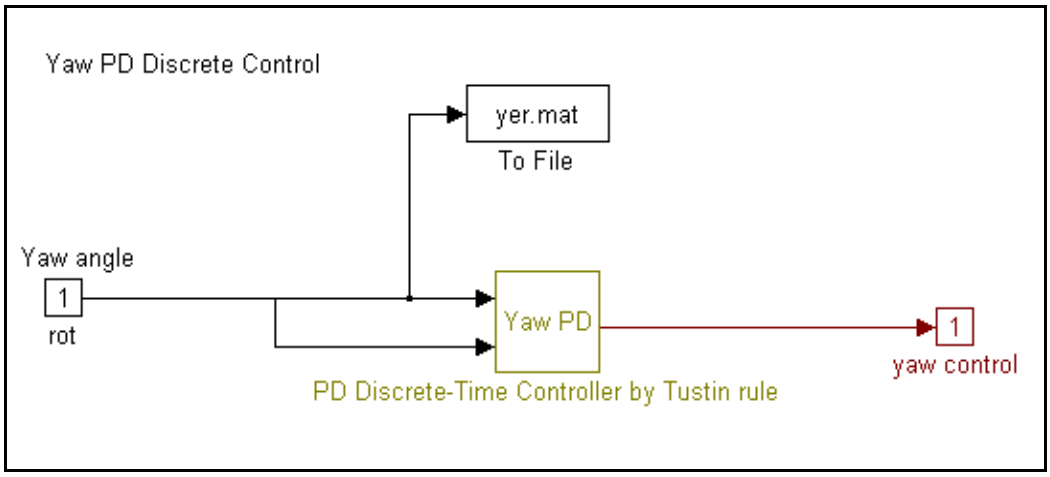


Figura C.7 – Detalhe do controle de “yaw”.

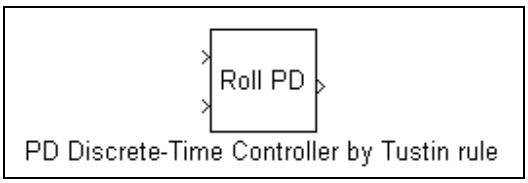


Figura C.8 – Detalhe de construção de uma “s-function” gráfica.

Definições da máscara do agrupamento de “s-function” (por exemplo, “roll PD”) da Figura C.8;

- “New block type”: “PD discrete time controller”;

- “Dialog strings” separadas por | : “PD discrete time controller by Tustin equivalence|Proportional Gain kp|Derivative Gain kd|Sample Time T (sec)”;

- “Initialization commands”*:* KP=@1;KD=@2;T=@3;

- “Drawing commands”: Roll PD

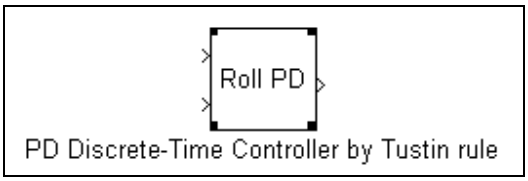


Figura C.9 – “S-function” gráfica de teste.

Tal bloco da Figura D.9 desagrupado fica conforme mostrado na Figura C.10.

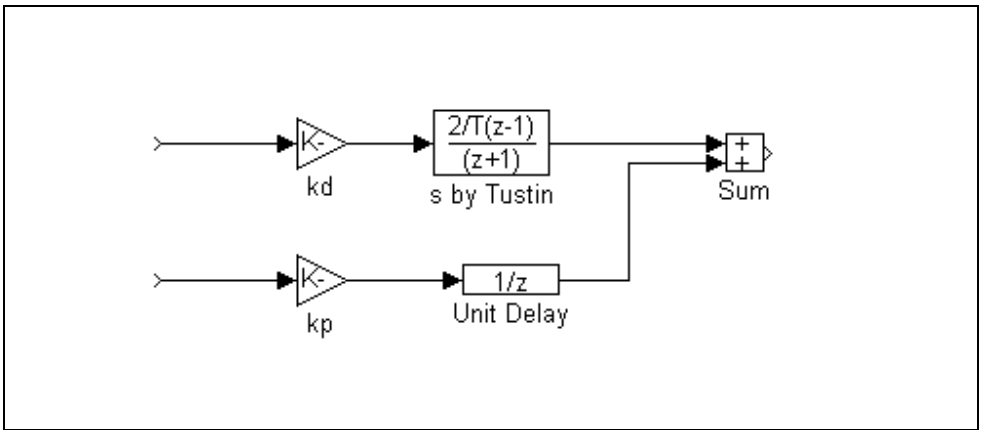


Figura C.10 – “S-funtion” desagrupada.

C.2. Planta Analógica e Controle PD Analógico

Nas simulações executadas para o caso totalmente analógico o desempenho do controle PD foi testado para os ganhos elevados do Caso 2, tal como vemos nas figuras a seguir.

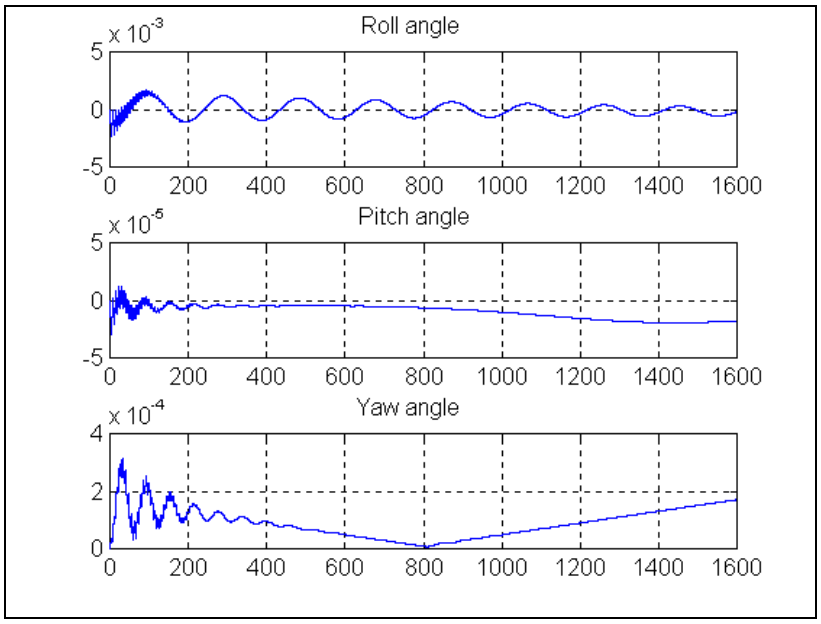


Figura C.11 – Ângulos de atitude para o controle PD analógico da planta analógica (CBERS-1).

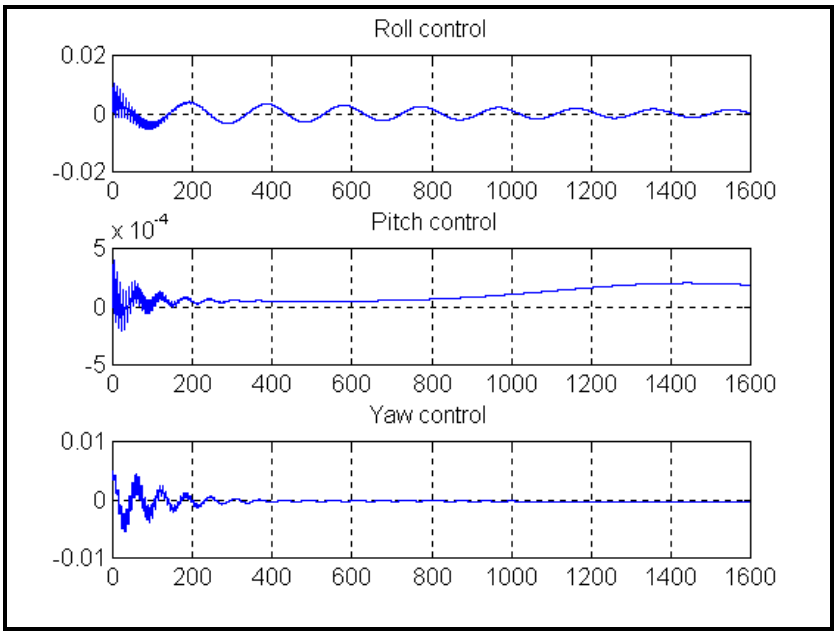


Figura C.12 – Sinais de controle de atitude.

C.3. Planta Analógica e Controle PD Discreto por Tustin

Podemos conferir a seguir os resultados das simulações feitas com o modelo analógico do satélite CBERS-1 com a atitude controlada por um PD em tempo-discreto projetado pela usual e clássica regra de “Tustin”. As simulações foram feitas usando T = 0,1 segundos e T = 1,6 segundos para os dois casos citados.

a) Caso 2, T = 0,1 segundos:

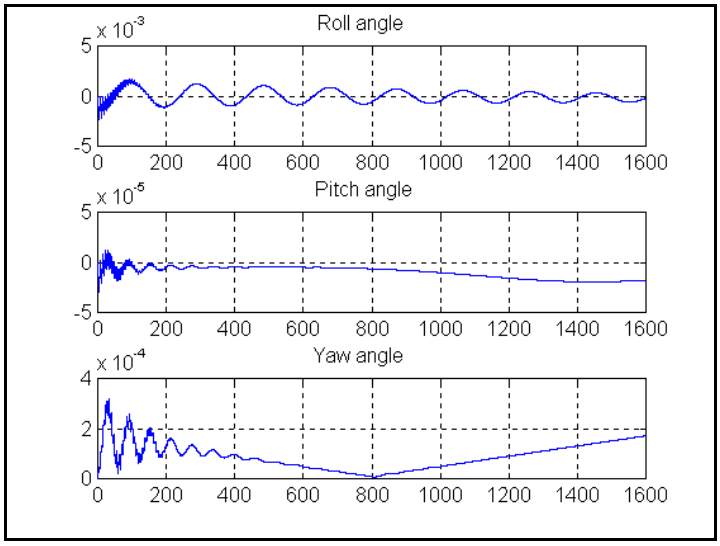


Figura C.13 – Ângulos de atitude.

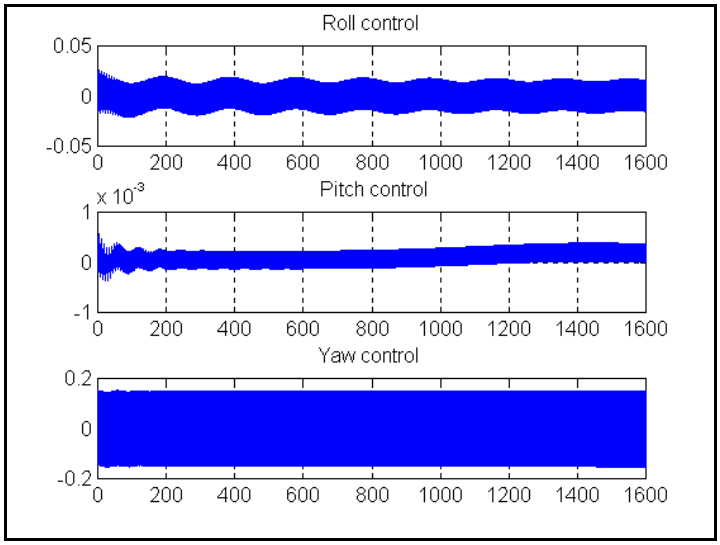


Figura C.14 – Sinais de controle de atitude.

Note na Figura C.14 os fortes sinais de controle para produzir uma estabilização dos ângulos de atitude. Isso ocorre devido ao chaveamento do amostrador na freqüência da taxa de amostragem (ω_S), possivelmente devido à presença de pólos complexos conjugados com parte real negativa no plano-Z. A explicação analítica desse efeito está muito bem documentada em **Isermann (1989)**. Entretanto, em **Franklin (1981)**, também pode-se encontrar alguma informação a respeito desse efeito, mas sem tantos detalhes quanto em **Isermann (1989)**.

b) Caso 1, T = 1,6 segundos:

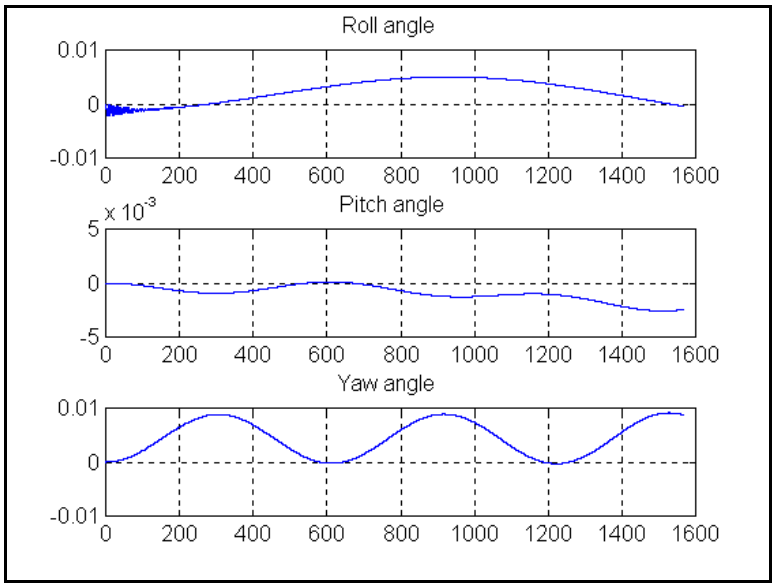


Figura C.15 – Ângulos de atitude.

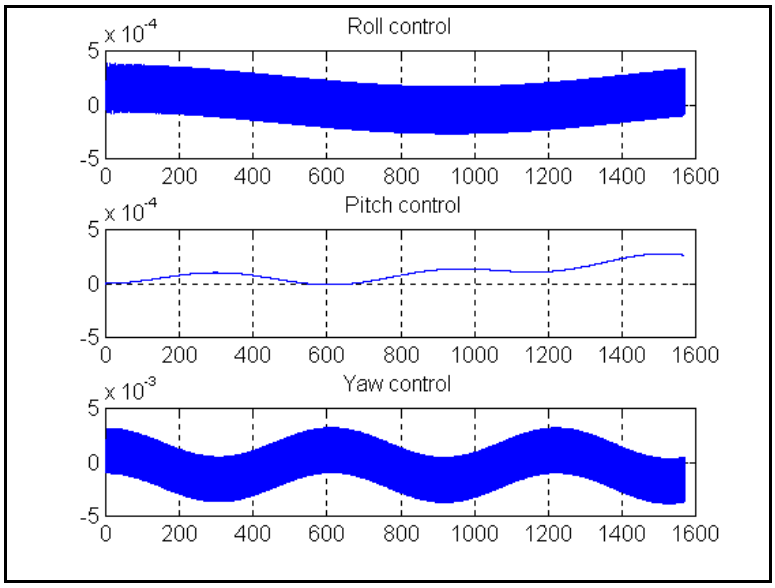


Figura C.16 – Sinais de controle de atitude.

c) Caso 2, T = 1,6 segundos:

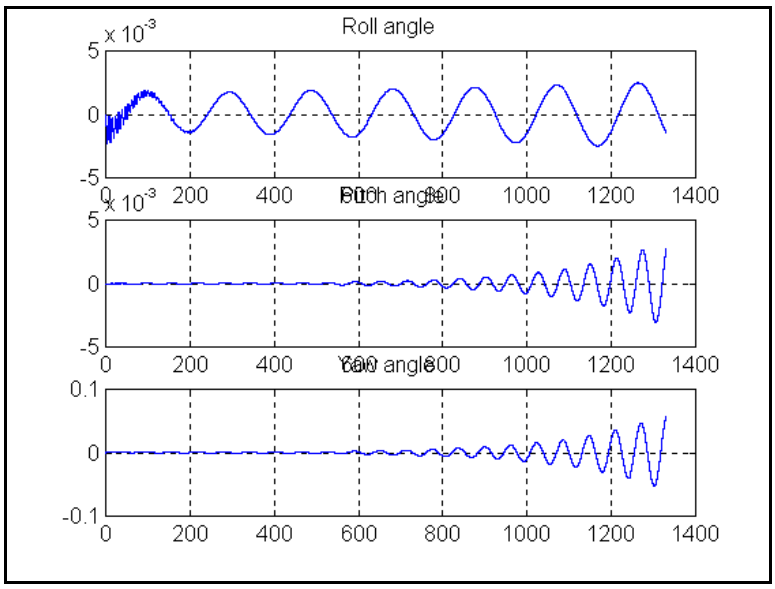


Figura C.17 – Ângulos de atitude.

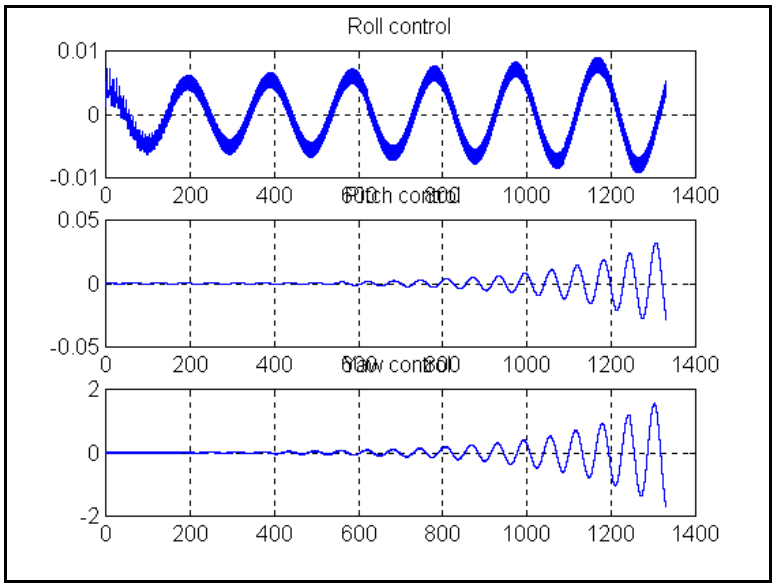


Figura C.18 – Sinais de controle de atitude.

C.4. Planta Analógica e Controle PD Discreto por Tustin, com Pré-Filtro "Anti-Aliasing"

Tentando corrigir o problema da instabilização que foi apresentado na Seção C.3, item b, quando os ganhos de controle e o período de amostragem T são elevados, foram inseridos filtros "anti-aliasing" nas saídas dos sensores. O filtro "anti-aliasing" é, basicamente, um filtro passa-baixas que tem por função atenuar fortemente a realimentação dos sinais dos modos de vibração que sejam superiores à freqüência de Nyquist buscando evitar/minimizar os efeitos deletérios do fenômeno do mascaramento ("aliasing"). Como pré-filtro "anti-aliasing" foram utilizados nas saídas dos sensores filtros Butterworth de quarta ordem.

a) caso 1, T = 1,6 segundos:

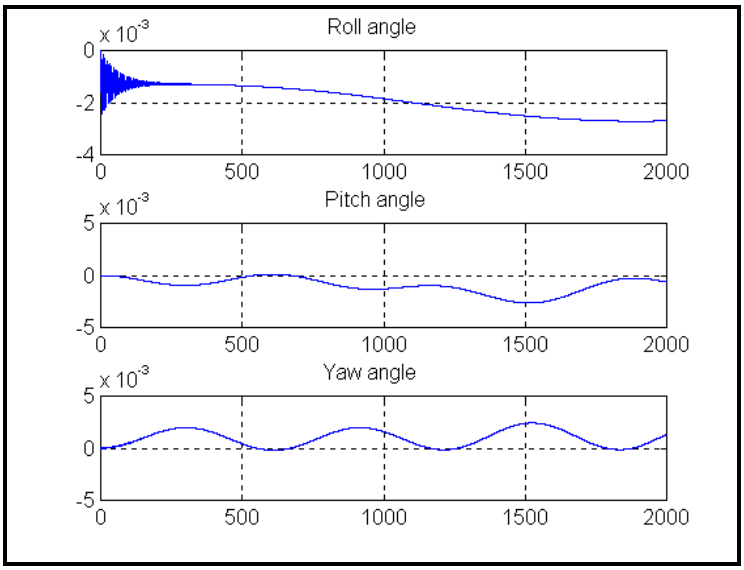


Figura C.19 – Ângulos de atitude.

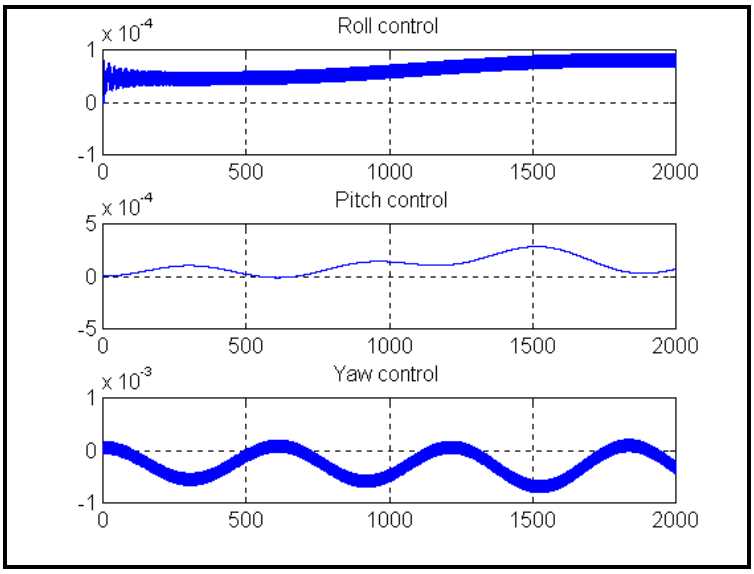


Figura C.20 – Sinais de controle de atitude.

A Figura CD.20 ao ser comparada com a Figura C.16 permite destacar uma vantagem da inserção do pré-filtro “anti-aliasing”: economia da energia de controle, visto que a Figura C.20 apresenta sinais de controle menos fortes que os sinais de controle apresentados na Figura C.16 evidenciando uma menor amplitude de oscilação do sinal elétrico de controle.

b) Caso 2, T = 1.6 segundos:

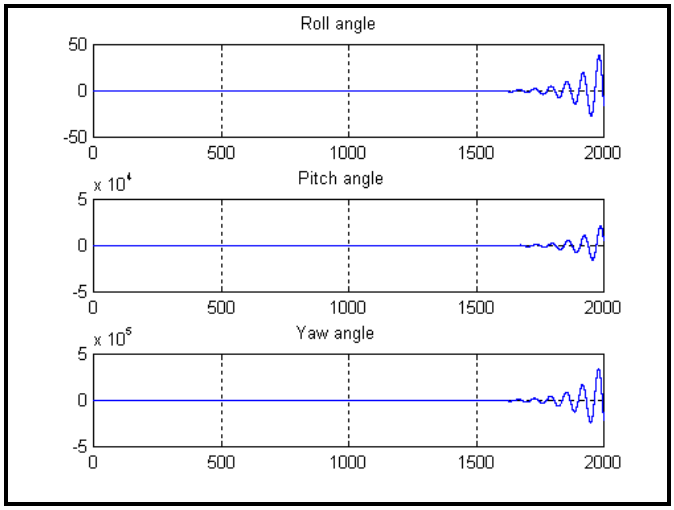


Figura C.21 – Ângulos de atitude.

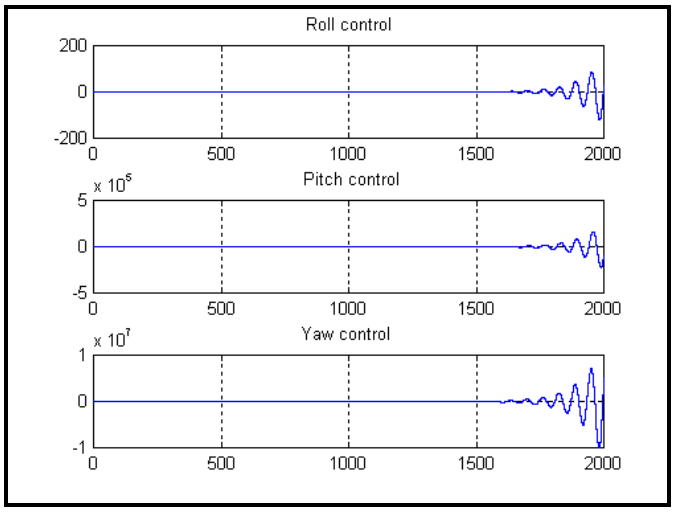


Figura C.22 – Sinais de controle de atitude.

Como podemos notar mediante as Figuras C.21 e C.22 a inclusão de um pré-filtro “anti-aliasing” não se mostrou satisfatória na tentativa de corrigir o problema da instabilidade.

C.5. Equivalente “Zero-Order Hold” da Planta Analógica e Controle PD Discreto por Tustin

a) Caso 1, T = 0.05 segundos. Serão inicialmente mostrados os resultados para até quatro segundos de simulação para que se torne clara a discretização no tempo.

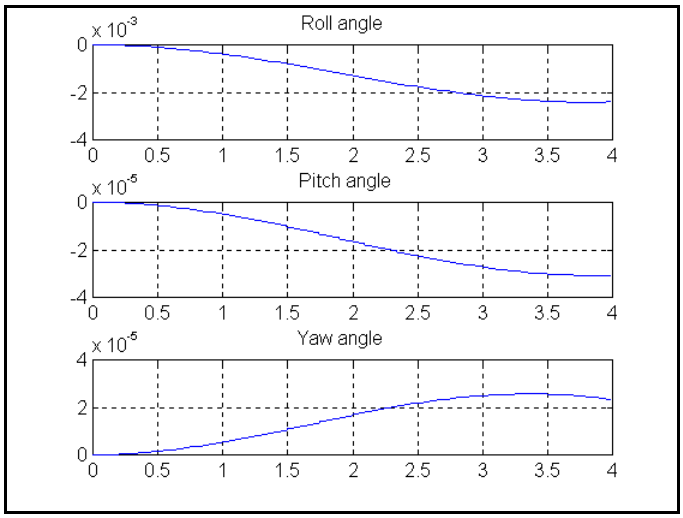


Figura C.23 – Ângulos de atitude.

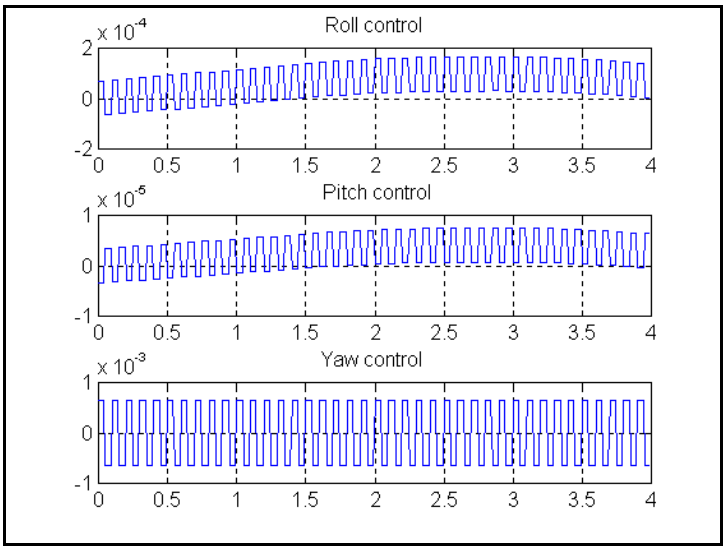


Figura C.24 – Sinais de controle de atitude.

Agora serão mostrados os resultados da mesma simulação executada com uma duração maior (400 segundos):

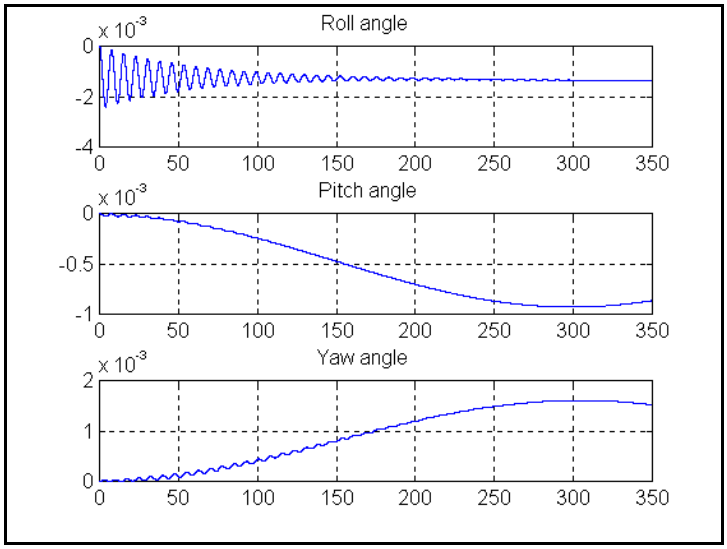


Figura C.25 – Ângulos de atitude.

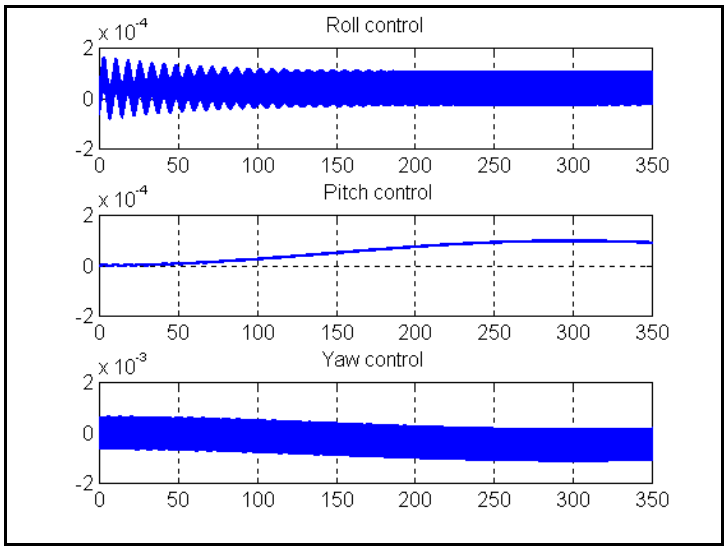


Figura C.26 – Sinais de controle da atitude.

b) Caso 1, T = 1,6 segundos:

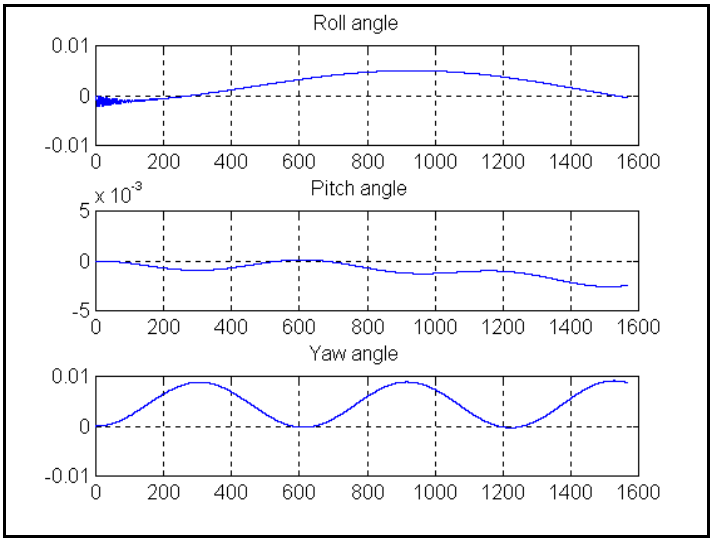


Figura C.27 – Ângulos de atitude.

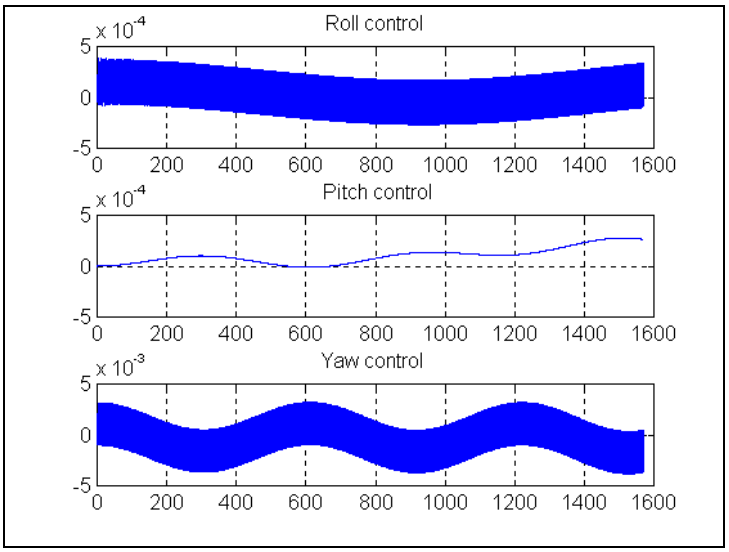


Figura C.28 – Sinais de controle de atitude.

c) Caso 2, T_S = 0,1 segundos

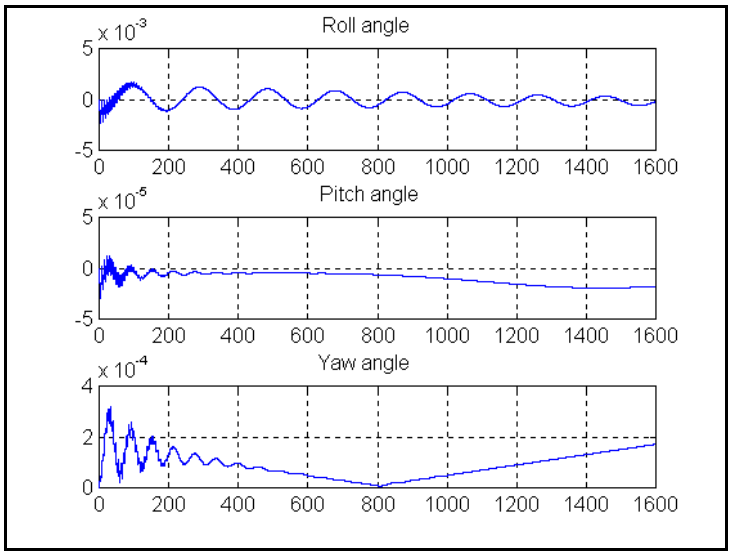


Figura C.29 – Ângulos de atitude.

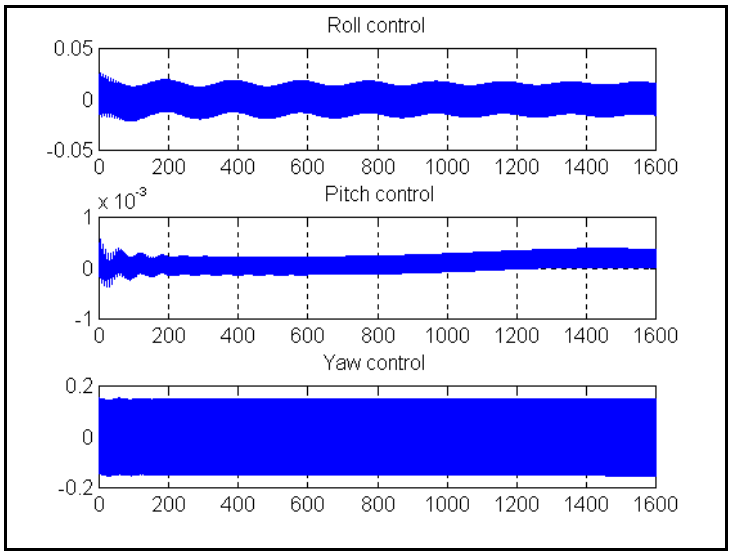


Figura C.30 – Sinais de controle de atitude.

d) Caso 2, T = 1,6 segundos:

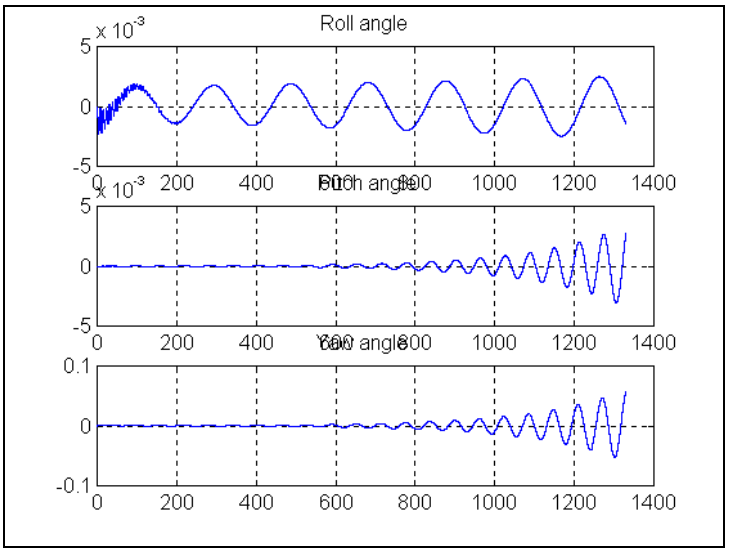


Figura C.31 – Ângulos de atitude.

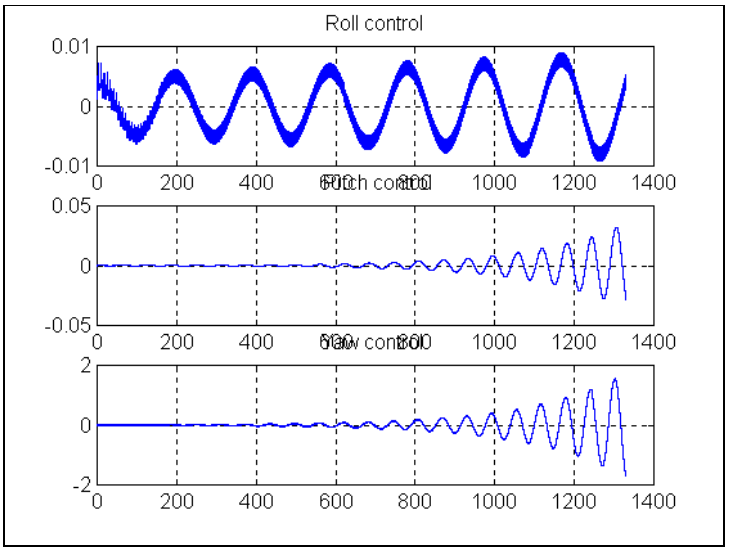


Figura C.32 – Sinais de controle de atitude.

C.6. Planta Analógica, Controle PD Discreto por Tustin, com Razão de Amortecimento Muito Elevada

Como a inclusão do pré-filtro “anti-aliasing” não se mostrou eficaz na restauração da estabilidade pensou-se em um outro método de estabilização: aumentar o amortecimento estrutural. Sabe-se que o aumento do amortecimento tem o poder de atrair os pólos para o interior do círculo unitário no plano-z que é a região assintoticamente estável o que, teoricamente, parecia ser um poderoso remédio

para resolver o problema em questão. Entretanto, infelizmente, tal método também não se mostrou eficiente diante do caso em que se tinham ganhos de controle e período de amostragem T elevados, conforme mostrado a seguir pode-se observar para o caso subamortecido, e para o caso sobreamortecido.

a) Caso 2. Razão de amortecimento = 0,6. T = 1,6 segundos. Caso subamortecido.

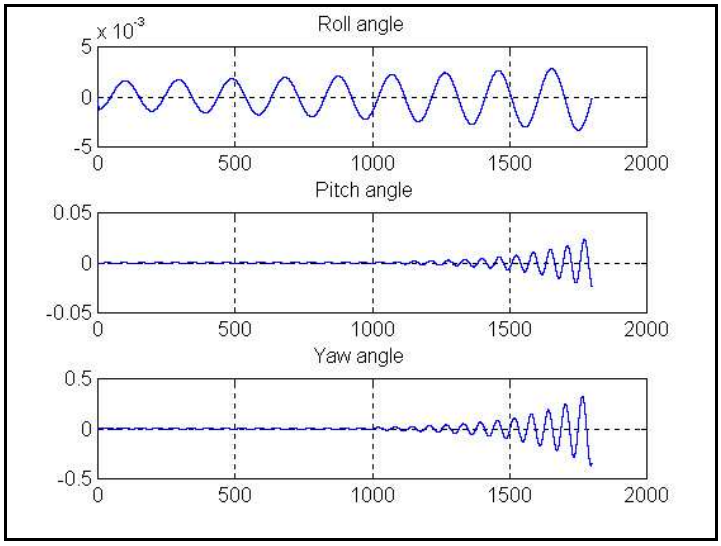


Figura C.33 – Ângulos de atitude.

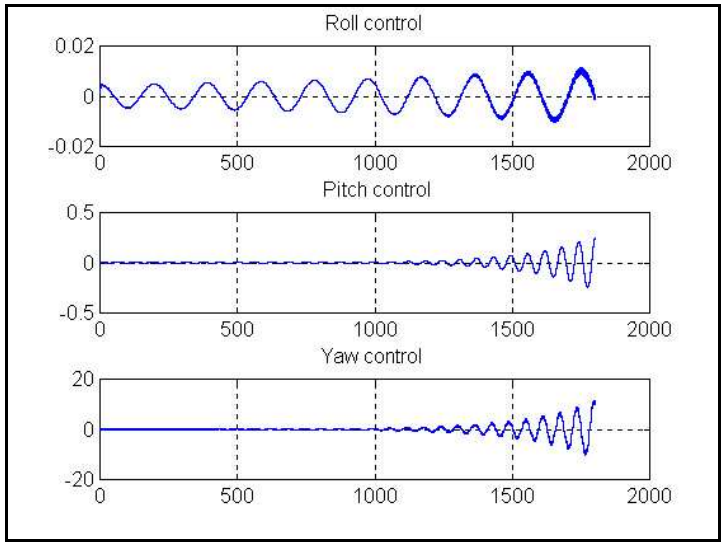


Figura C.34 – Sinais de controle de atitude.

b) Caso 2. T = 1,6 segundos. Razão de amortecimento = 2. Caso sobreamortecido.

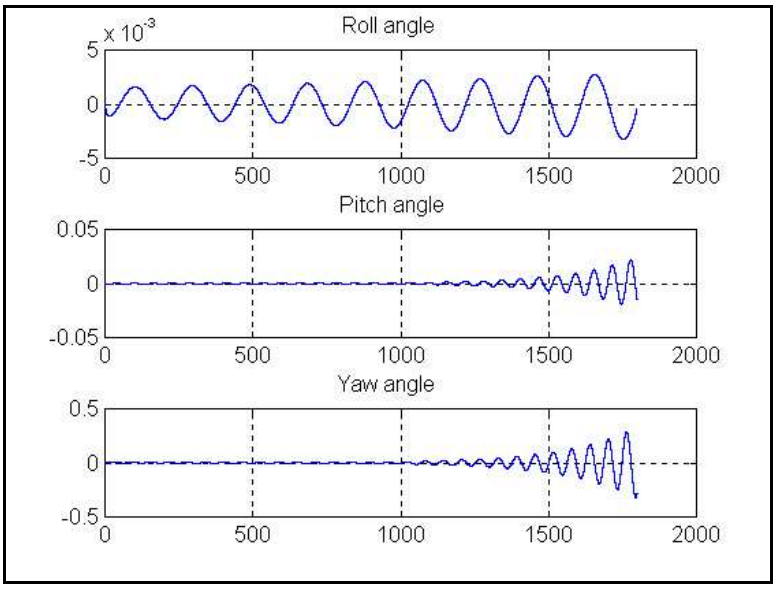


Figura C.35 – Ângulos de atitude.

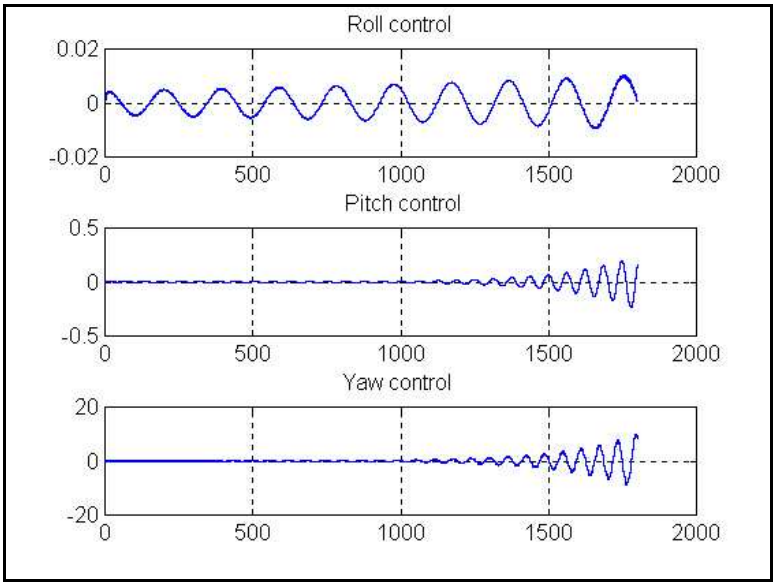


Figura C.36 – Sinais de controle de atitude.

c) Modelo analógico do CBERS-1 simulado para um único modo. Serão baixados os valores dos ganhos de controle para uma situação intermediária em relação aos Casos 1 e 2, de ganhos de controle muito baixos e elevados, respectivamente. A razão de amortecimento será elevado para $\zeta = 6$ (caso

sobreamortecido); T = 1,6 segundos; ganhos de controle: roll: k_p = 0.316; k_d = 0.482; pitch: k_p = 1.07; k_d = 1.63; yaw: k_p = 3.0; k_d = 4.6.

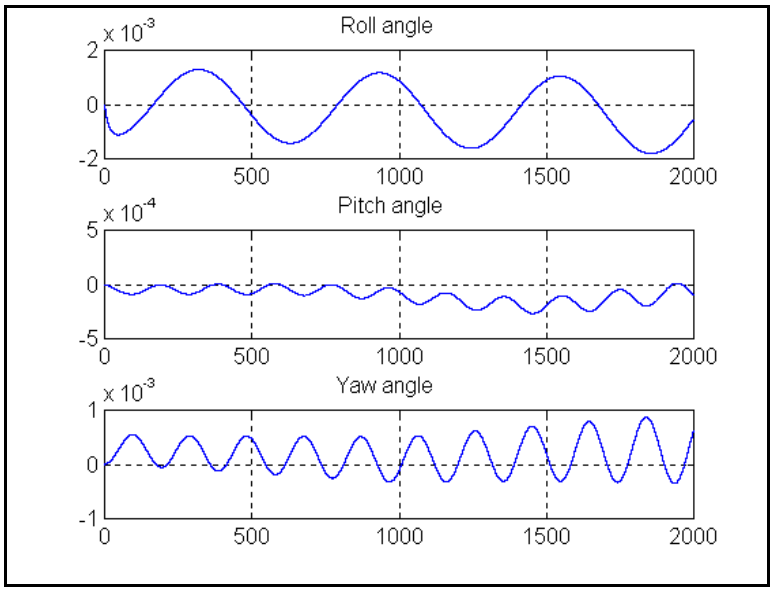


Figura C.37 – Ângulos de atitude.

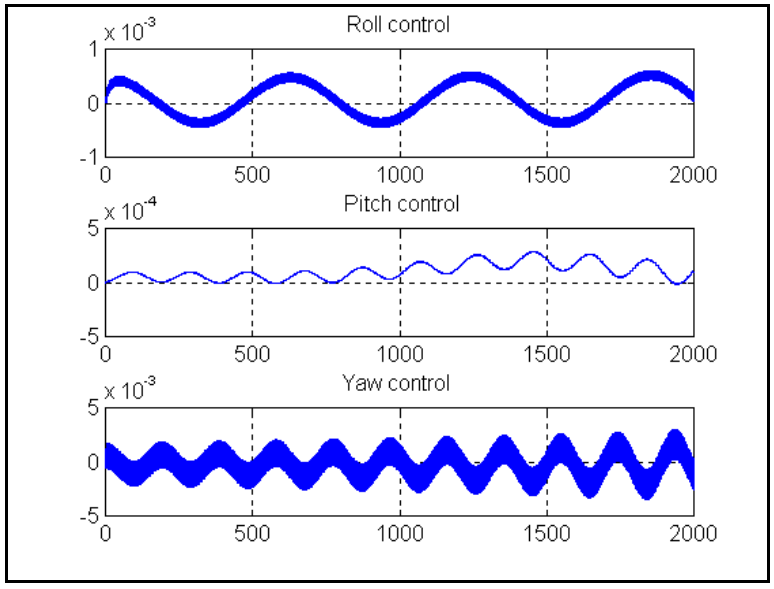


Figura C.38 – Sinais de controle de atitude.

Mediante as Figuras C.37 e C.38 podemos perceber nitidamente a presença da instabilidade nos eixos de pitch e yaw.

d) Modelo analógico do CBERS-1 simulado para dois modos. Razão de amortecimento ζ = 6 (caso sobreamortecido). T = 1,6 segundos. Caso intermediário: ganhos de controle: roll: k_p = 0.316; k_d = 0.482; pitch: k_p = 1.07; k_d = 1.63; yaw: k_p = 3.0; k_d = 4.6.

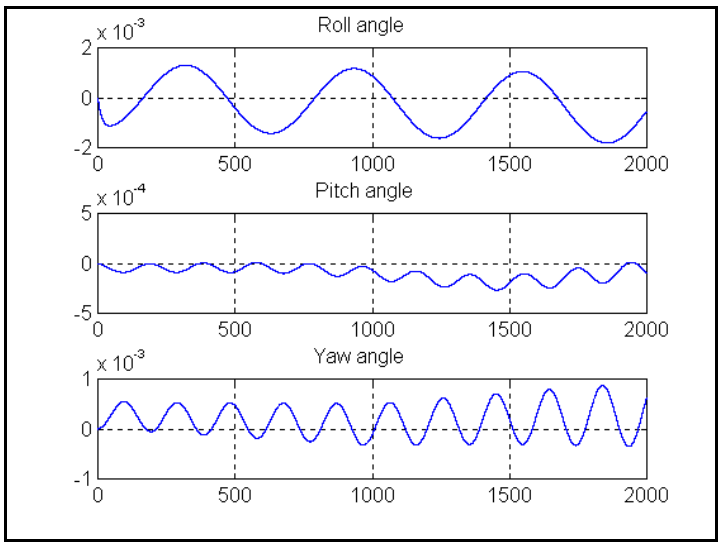


Figura C.39 – Ângulos de atitude.

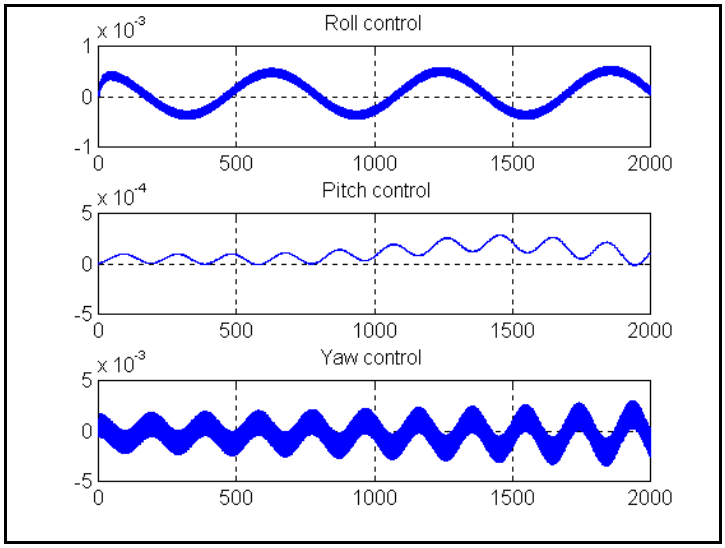


Figura C.40 – Sinais de controle de atitude.

Nota-se a indesejável presença da instabilidade nos eixos de “pitch” e “yaw” pelas Figuras C.39 e C.40.

Explicar-se-á agora porque um sistema em malha fechada com controle discreto no tempo atuando sobre uma planta sobreamortecida apresentou as oscilações mostradas: tal fenômeno ocorre em cada eixo devido à superposição das contribuições de: a) um par de pólos complexos conjugados em malha fechada originados do par de pólos em malha aberta s = 0 ou z = 1 do modo rígido; b) um par de pólos (reais ou complexos conjugados $s = \sigma \pm j.\omega$) em malha fechada originados do par de pólos (reais ou complexos conjugados $s = \sigma \pm j.\omega$) em malha aberta de cada modo flexível; c) e de um pólo real negativo $z \cong -1$ originado do pólo em z = -1 da regra de Tustin; o que é retratado no transitório como um chaveamento na freqüência da taxa de amostragem, descrevendo um sinal forte, e superposto à freqüência do modo, como podemos notar com base em **Isermann (1989, seção 3.5.1)** e **Franklin (1981, seção 2.4)**.

C.7. Planta Analógica Para Dois Modos Flexíveis, Razão de Amortecimento Muito Elevada, e Controle PD Discreto no Tempo por Schneider

Visto que os métodos do emprego do filtro “anti-aliasing” e do aumento do amortecimento estrutural não se mostraram eficazes na manutenção da estabilidade no sistema de controle em tempo-discreto projetado pela regra de “Tustin” para o caso em que os ganhos eram maiores que os do Caso 1, apresentado no início deste Capítulo, pensou-se numa outra alternativa: mudança de mapeamento s-z (mapeamento analógico digital). Intuitivamente, ocorreu a necessidade de buscar uma outra forma clássica de mapeamento diferente da regra de “Tustin” visto que esta não apresentou bons resultados quando os ganhos de controle e o período de amostragem eram elevados.

Foram então estudados os trabalhos de **Schneider (1991 e 1994)** onde apresentou-se uma regra de mapeamento baseada em métodos de integração numérica de ordem elevada. Em **Schneider (1991)** é apresentado o método de integração de Adams-Moulton de terceira ordem da equação (C.3), que pode ser reescrito como uma equação a diferenças finitas tal como mostra a equação (C.4).

Após a aplicação da transformada-Z sobre a equação (C.4) obtém-se, finalmente, a regra de mapeamento de Schneider conforme mostrado na equação (C.5):

$$u_k = u_{k-1} + \frac{T}{12}.(5.e_k + 8.e_{k-1} - e_{k-2}) \tag{C.3}$$

$$e_k = \frac{1}{5}.\left(\frac{12}{T}.\nabla u_k - 8.e_{k-1} + e_{k-2}\right) \tag{C.4}$$

$$\frac{E(s)}{U(s)} = s \sim \frac{E(z)}{U(z)} = \frac{12}{T}.\frac{z.(z-1)}{5z^2 + 8z - 1} \tag{C.5}$$

Tal novo mecanismo de ordem elevada para mapeamentos analógico-digitais chamou a atenção por ter se apresentado melhor que a regra de “Tustin” e que alguns outros métodos de mapeamento tal como mostrado em **Schneider (1994)** para o caso em que o mapeamento foi executado somente para ações integrais de controle. Era mesmo esperado que seus resultados se mostrassem melhores que o integrador Tustin visto que tratavam de integradores de mesma família (Adams-Moulton), porém sendo o de Schneider de ordem mais elevada. Os estudos de **Schneider (1991 e 1994)** motivaram as simulações que seguem abaixo. A dúvida que restava era como tal mapeamento poderia se comportar enquanto ação derivativa.

Vejamos a seguir os resultados de uma simulação que utilizou a regra de Schneider (Seção A.5.3.e) no projeto do controle PD em tempo-discreto. T = 1,6 segundos. Fator de amortecimento utilizado: ζ = 6. Ganhos do controle discreto no tempo: roll: k_p = 0.316; k_d = 0.482; pitch: k_p = 1.07; k_d = 1.63; yaw: k_p = 3.0; k_d = 4.6.

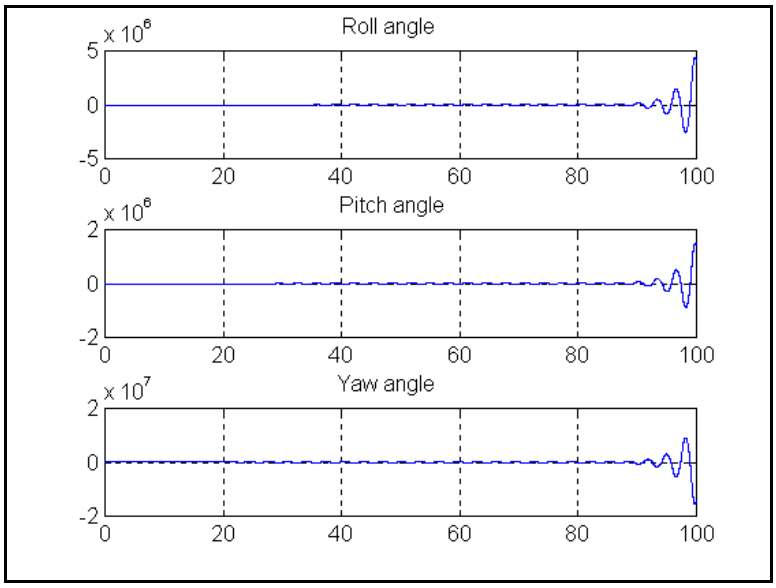


Figura C.41 – Ângulos de atitude.

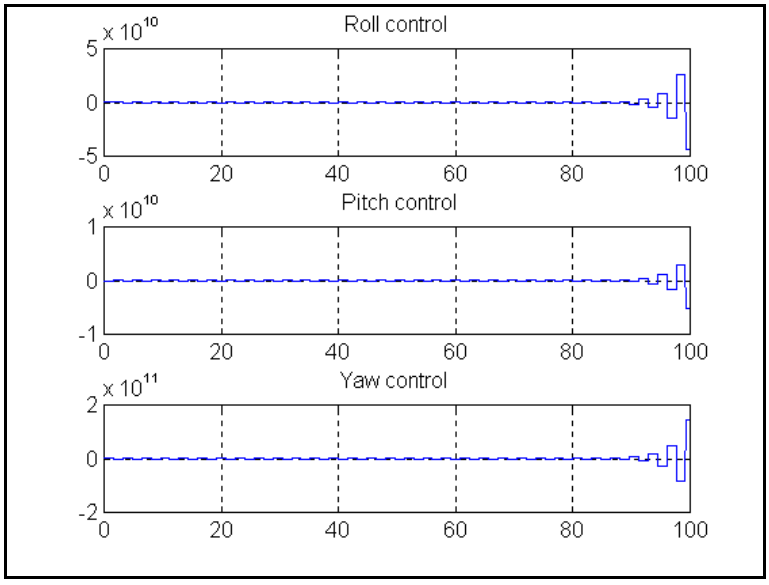


Figura C.42 – Sinais de controle de atitude.

Como podemos ver acima a regra de mapeamento s-z proposta por Schneider também não se mostrou eficiente pois a instabilidade tornou-se evidente.

A explicação do porquê a regra de Schneider mostrou-se ineficaz no projeto do controle PD pode ser contemplada na Seção A.5.3, item “e”.

C.8. Comparações

Comparamos as diversas situações em que o projeto do controle PD foi projetado pela usual regra de “Tustin” funciona e naquelas em que ele falha. Pudemos verificar também a ineficiência do projeto do PD por aproximação de Schneider devido à apresentação de um pólo fora do círculo unitário no plano-z como foi mostrado na Seção A.5.3, item “e”. Nenhum método conhecido até então resolveu o problema da instabilidade para o caso de elevado período de amostragem e elevados ganhos tal como dimensionados anteriormente.

C.9. Estabilização por Redução do Período de Amostragem

É o método mais simples e fácil de ser empregado, sendo o seu emprego recomendável sob este aspecto. Entretanto, a disponibilidade e o custo dos processadores, e conversores A/D e D/A, sobretudo de qualidade espacial, podem inviabilizar tal método, dependendo do orçamento disponível para um dado projeto. Vimos como o critério de estabilidade de Jury pode ser útil para fornecer as regiões ou intervalos onde o período de amostragem do controlador em tempo discreto garantirá a estabilidade no controle de um oscilador harmônico livre. Podemos verificar por meio das regiões de estabilidade e instabilidade que são fornecidas mediante a aplicação do critério de estabilidade de Jury para um oscilador harmônico livre controlado por um controle Proporcional+Derivativo projetado por aproximação Tustin.

C.10. Estabilização por Emprego de Filtro “Anti-Aliasing”

Embora amplamente usado, o filtro “anti-aliasing” não foi suficiente para estabilizar a planta como pôde ser observado nas simulações aqui apresentadas.

C.11. Estabilização por Redução dos Ganhos

A estabilização por redução dos ganhos do controle deve ser tentada mas possui o inconveniente de mudar as especificações do sistema em malha fechada (tempos de acomodação e subida, sobressinal, etc.).

C.12. Estabilização por Aumento do Amortecimento Estrutural

O aumento do amortecimento estrutural tem o poder de atrair os pólos de malha aberta para o interior do círculo unitário no plano-z, e nem sempre resolve o problema da instabilidade tal como podemos perceber. Também há o inconveniente de que o projeto estrutural do satélite seria alterado.

APÊNDICE D

COMPARAÇÃO DE DESEMPENHO ENTRE CONTROLADOR PID CLÁSSICO E PID ASTRÖM DA POSIÇÃO DE UM OSCILADOR HARMÔNICO MAPEADOS POR TUSTIN

Objetivando reduzir sobressinal máximo indesejável provocado por controladores PID classicamente projetados, um ajuste no modelo analógico de Ziegler-Nichols foi proposto por Aström, tal como ensinado em **Oubrahim (1994)**, contendo como parâmetros de ajuste (sintonização) os fatores de ponderação (ou de "peso") β e N e os ganhos Kp, Kd e Ki. G(s) é a função de transferência analógica da planta, no caso um oscilador harmônico simples (Figura D.1).

D.1. INTRODUÇÃO

Na Figura D.1 "r", "y", "e", "u" e "w" são respectivamente a entrada, a saída, o sinal de erro, o sinal de controle e a perturbação de ruído branco. A introdução de fator de ponderação β (onde $0 < \beta < 1$), oferece uma forma de ajustar os zeros da função de transferência em malha-fechada que afetam o valor do sobressinal máximo.

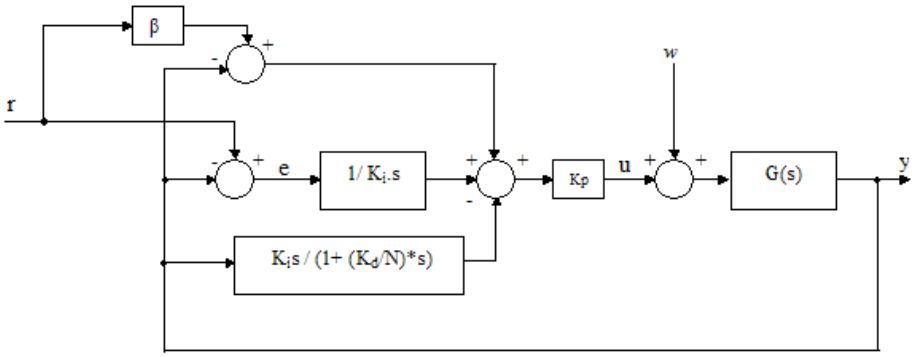


Figura D.1 – Controlador PID Aström analógico interferindo no comportamento da planta G(s), também analógica.

Os resultados obtidos apresentaram uma boa redução do sobressinal e assim mostra-se como uma técnica desejável e recomendável de controle PID.

Com essa principal vantagem da estrutura PID Aström, a redução de sobressinal, podemos evitar mal comportamento em muitas plantas industriais (transbordo em depósitos químicos ou sobrecarga elétrica em controle de sistemas de distribuição de energia elétrica) e controle de vibração. Não foi encontrado na literatura um equivalente digital para o modelo PID Aström original e mostrou-se relevante realizar tal tarefa nesse trabalho. Maiores detalhes sobre tais controladores PID analógicos podem ser contemplados na literatura: **Oubrahim (1990)** e **Aström (1988)**.

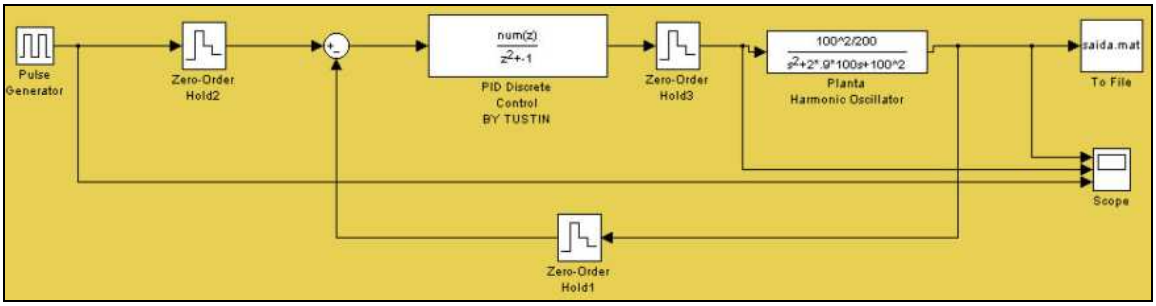


Figura D.2 – Implementação em Simulink da estrutura clássica do controlador PID digital interferindo no comportamento da planta analógica.

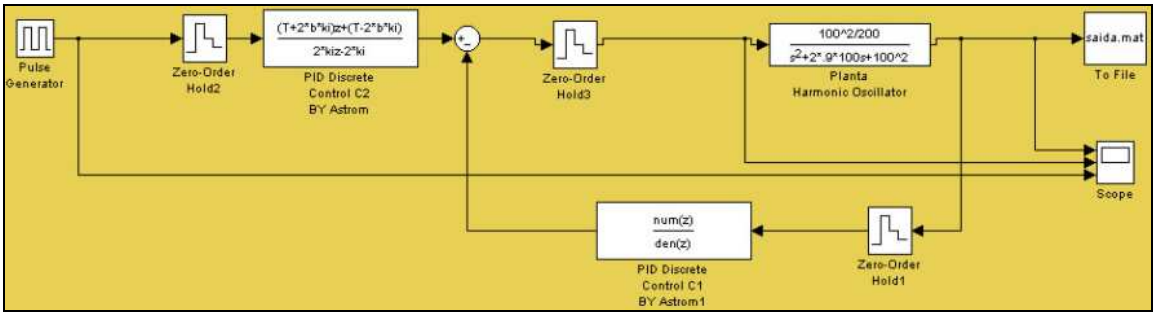


Figura D.3 – Implementação em Simulink da estrutura do controlador PID Aström digital interferindo no comportamento da planta analógica.

Nesse trabalho usamos dois controladores PID: clássico e Aström. Esses dois controladores foram discretizados usando as regras de Tustin e ST1. Foram usados: N = 50 e $\beta = 0$. Os ganhos PID usados foram: $k_i = 0{,}005$; $k_d = 0{,}02$ e $k_p = 0{,}003$. A razão de amortecimento do oscilador harmônico usado é $\zeta = 0.9$. Foram

realizadas simulações usando períodos de amostragem de 0,001 segundos a 3 segundos e os resultados serão discutidos a seguir. Os experimentos numéricos foram todos realizadas no mesmo ambiente computacional e usando um integrador Runge-Kutta de ordem 4 com passo-fixo de 0,0001 segundos com modo "single-tasking" ("stand alone").

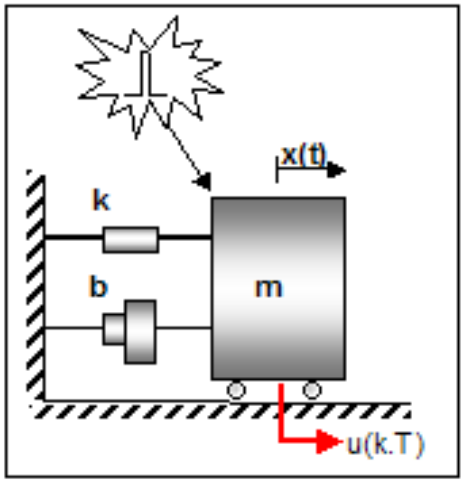


Figura D.4 – Oscilador harmônico excitado por um pulso de duração finita.

A saída é a posição x(t) da massa m medida em metros e que se desloca numa superfície sem atrito. A excitação é um "empurrão" (pulso) com intensidade unitária e duração de 0,01 segundos (Figura D.4).

D.2. Controle PID – Clássico discretizado pelo método de Tustin:

Do mapeamento Tustin, temos que a discretização Tustin do controlador PID clássico é dada por C(z):

$$s \cong \frac{2}{T} \cdot \frac{z-1}{z+1} \qquad \text{(D.1)}$$

$$C(s) = K_p + K_d + K_i \cdot \frac{1}{s} \qquad \text{(D.2)}$$

$$C(z) = \frac{K_p \cdot (z^2 - 1) + \left(\frac{2 \cdot K_d}{T}\right) \cdot (z^2 - 2 \cdot z + 1) + \left(\frac{2 \cdot K_i}{T}\right) \cdot (z^2 + 2 \cdot z + 1)}{(z^2 - 1)} \quad \text{(D.3)}$$

D.2. Controle PID Aström analógico segundo a configuração mostrada na Figura E.5.

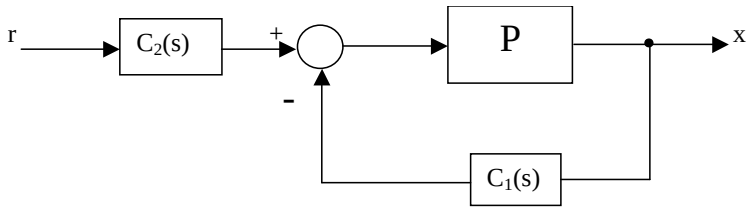


Figura D.5 – diagrama em blocos do controlador PID Aström analógico.

$$C_1(s) = K_p \cdot \left(\frac{1 + s \cdot \left(K_i + \frac{K_d}{N}\right) + s^2 \cdot K_i \cdot K_p \cdot \left(\frac{N+1}{N}\right)}{K_i \cdot s \cdot \left(1 + \frac{K_d}{N} \cdot s\right)} \right) \quad \text{(D.4)}$$

$$C_2(s) = K_p \left(\beta + \frac{1}{K_i \cdot s} \right) \quad \text{(D.5)}$$

D.3. Controle PID Aström discretizado por Tustin.

Detalhes dos calculos seguem abaixo: (obs: N$\in R_+^*$ e $\beta \in R_+$ são os parâmetros de ajuste)

$$s \cong \frac{2}{T} \cdot \frac{z-1}{z+1} \quad \text{(D.6)}$$

$$C_1(z) = K_p \cdot \left(\frac{1 + \frac{2}{T} \cdot \frac{z-1}{z+1} \cdot \left(K_i + \frac{K_d}{N} \right) + \left(\frac{2}{T} \right)^2 \cdot \left(\frac{z-1}{z+1} \right)^2 \cdot K_i \cdot K_p \cdot \left(\frac{N+1}{N} \right)}{K_i \cdot \frac{2}{T} \cdot \frac{z-1}{z+1} \cdot \left(1 + \frac{K_d}{N} \cdot \frac{2}{T} \cdot \frac{z-1}{z+1} \right)} \right) \quad \text{(D.7)}$$

$$C_1(z) = K_p \cdot \left(\frac{\left(z+1\right)^2 + \left(\frac{2}{T}\right) \cdot \left(z-1\right) \cdot (z+1) \cdot \left(K_i + \frac{K_d}{N}\right) + \left(\frac{4}{T^2}\right) \cdot \left(z-1\right)^2 \cdot K_i \cdot K_p \cdot \left(\frac{N+1}{N}\right)}{(z+1)^2} \right.$$
$$\left. \frac{2 \cdot K_i}{T} \cdot \frac{z-1}{z+1} \cdot \left(\frac{z+1+\left(2 \cdot \frac{K_d}{N \cdot T}\right) \cdot \left(z-1\right)}{(z+1)} \right) \right)$$

(D.8)

$$C_1(z) = K_p \cdot \left(\frac{\left(z+1\right)^2 + \left(\frac{2}{T}\right) \cdot \left(z^2-1\right) \cdot \left(K_i + \frac{K_d}{N}\right) + \left(\frac{4}{T^2}\right) \cdot K_i \cdot K_p \cdot \left(\frac{N+1}{N}\right) \cdot \left(z-1\right)^2}{\frac{2 \cdot K_i}{T} \cdot \left(z-1\right) \cdot \left(\left(1 + \frac{2 \cdot K_d}{N \cdot T}\right) \cdot z + \left(1 - \frac{2 \cdot K_d}{N \cdot T}\right)\right)} \right)$$

(D.9)

$$C_1(z) = K_p \cdot \left(\frac{z^2 + 2 \cdot z + 1 + \left(\frac{2}{T}\right) \cdot \left(K_i + \frac{K_d}{N}\right) \cdot z^2 - \left(\frac{2}{T}\right) \cdot \left(K_i + \frac{K_d}{N}\right) + \left(\frac{4}{T^2}\right) \cdot K_i \cdot K_d \cdot \left(\frac{N+1}{N}\right) \cdot z^2 - 2 \cdot \left(\frac{4}{T^2}\right) \cdot K_i \cdot K_d \cdot \left(\frac{N+1}{N}\right) \cdot z + 2 \cdot \left(\frac{4}{T^2}\right) \cdot K_i \cdot K_d \cdot \left(\frac{N+1}{N}\right)}{\frac{2 \cdot K_i}{T} \cdot \left(1 + \frac{2 \cdot K_d}{N \cdot T}\right) \cdot z^2 + \left(-\frac{2 \cdot K_i}{T} \cdot \left(1 + \frac{2 \cdot K_d}{N \cdot T}\right) + \frac{2 \cdot K_i}{T} \cdot \left(1 + \frac{2 \cdot K_d}{N \cdot T}\right)\right) \cdot z - \frac{2 \cdot K_i}{T} \cdot \left(1 - \frac{2 \cdot K_d}{N \cdot T}\right)} \right)$$

(D.10)

$$C_1(z) = K_p \cdot \left(\frac{\left[1 + \left(\frac{2}{T}\right) \cdot \left(K_i + \frac{K_d}{N}\right) + \left(\frac{4}{T^2}\right) \cdot K_i \cdot K_d \cdot \left(\frac{N+1}{N}\right)\right] \cdot z^2 + \left[-2 \cdot \left(\frac{4}{T^2}\right) \cdot K_i \cdot K_d \cdot \left(\frac{N+1}{N}\right)\right] \cdot z + \left(\frac{2}{T}\right) \cdot \left(K_i + \frac{K_d}{N}\right) + 2 \cdot \left(\frac{4}{T^2}\right) \cdot K_i \cdot K_d \cdot \left(\frac{N+1}{N}\right)}{\left[\frac{2 \cdot K_i}{T} \cdot \left(1 + \frac{2 \cdot K_d}{N \cdot T}\right)\right] \cdot z^2 + \left[\frac{K_i \cdot K_d}{N \cdot T^2}\right] \cdot z + \left[\frac{2 \cdot K_i}{T} \cdot \left(1 - \frac{2 \cdot K_d}{N \cdot T}\right)\right]} \right)$$

(D.11)

Agora vamos desenvolver a discretização de $C_2(s)$:

$$C_2(s) = K_p \cdot \left(\beta + \frac{1}{K_i \cdot s}\right) \tag{D.12}$$

$$C_2(s) = K_p \cdot \left(\beta + \frac{T \cdot (z+1)}{2 \cdot K_i \left((z-1)\right)}\right) = K_p \left(\frac{2 \cdot \beta \cdot K_i \cdot (z-1) \cdot T \cdot (z+1)}{2 \cdot K_i \cdot (z-1)}\right) \tag{D.13}$$

$$C_2(s) = K_p \cdot \left(\frac{\left(T + 2 \cdot \beta \cdot K_i\right) \cdot z + \left(T + 2 \cdot \beta \cdot K_i\right)}{2 \cdot K_i \cdot (z-1)} \right) \qquad \text{(D.14)}$$

D.4. Resultados das simulações numéricas.

a) PID Clássico com T = 0,001 segundo.

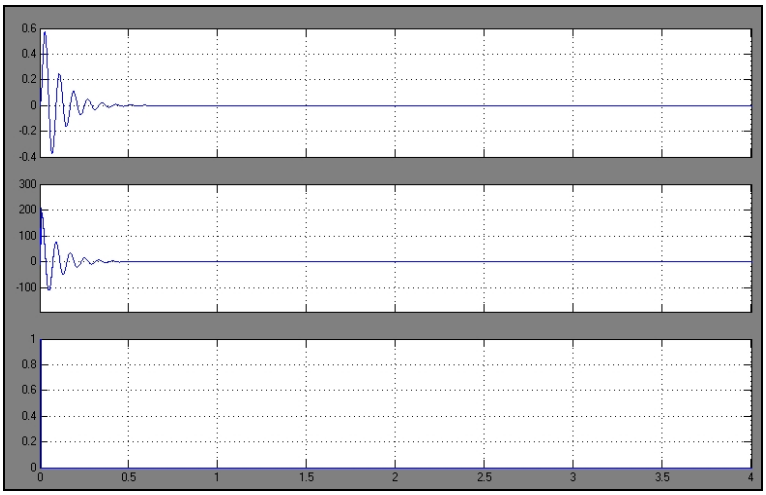


Figura D.6 – Resultados numéricos para a posição x(t) (setor superior da Figura), sinal de controle (setor intermediário da Figura acima) e pulso de duração 0,01 segundos (setor inferior da Figura acima).

b) PID Aström com T = 0,001 segundo.

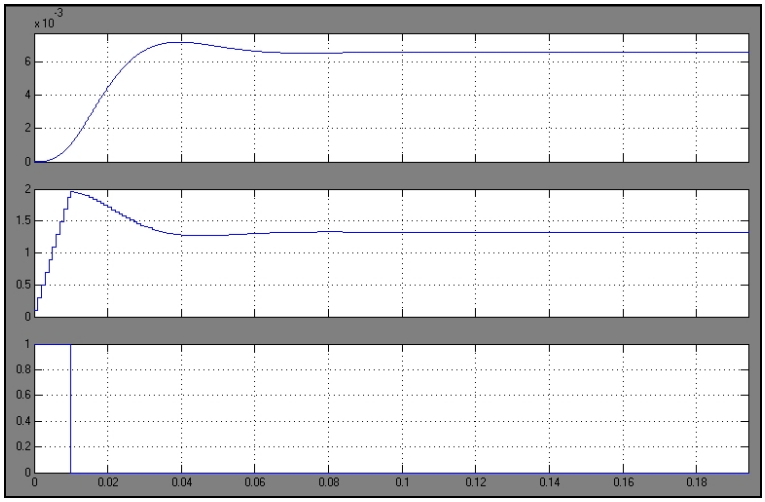


Figura D.7 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

c) PID Clássico com T = 0,05 segundo.

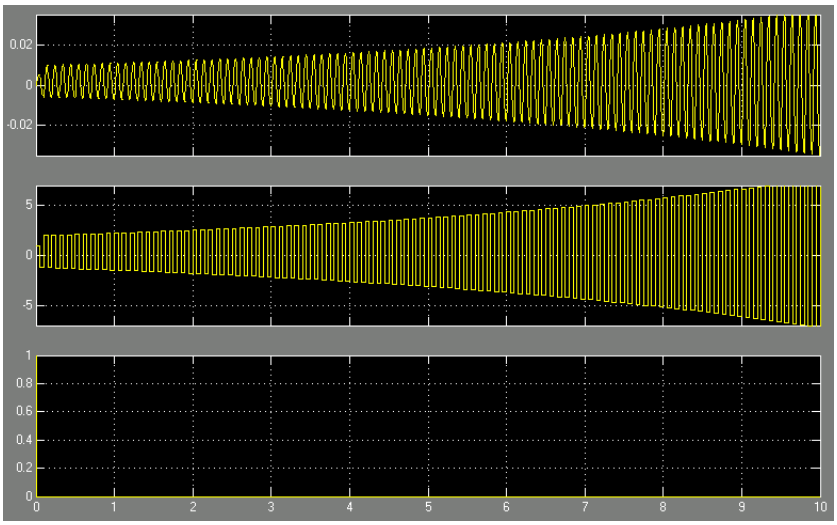


Figura D.8 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

d) PID Aström com T = 0,05 segundo.

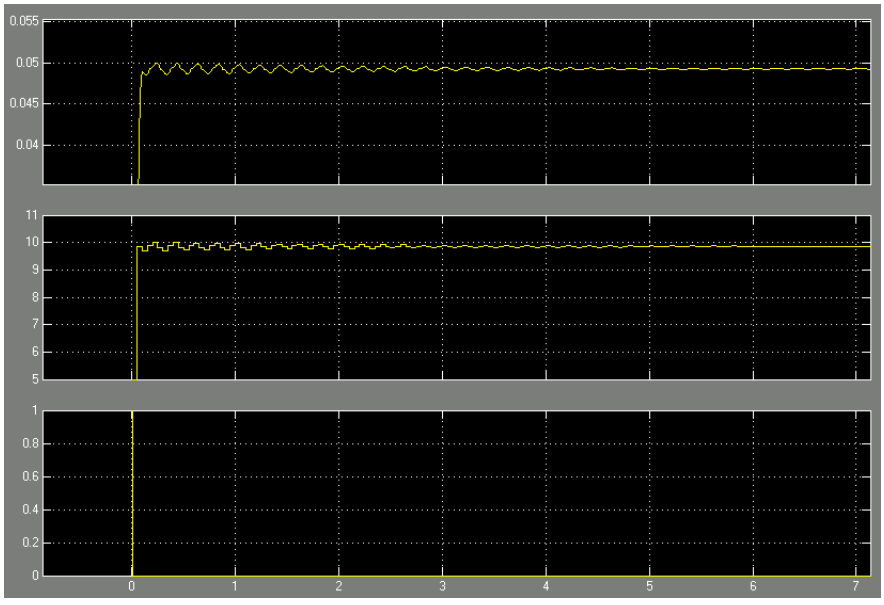


Figura D.9 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

e) PID Clássico com T = 0,1 segundo.

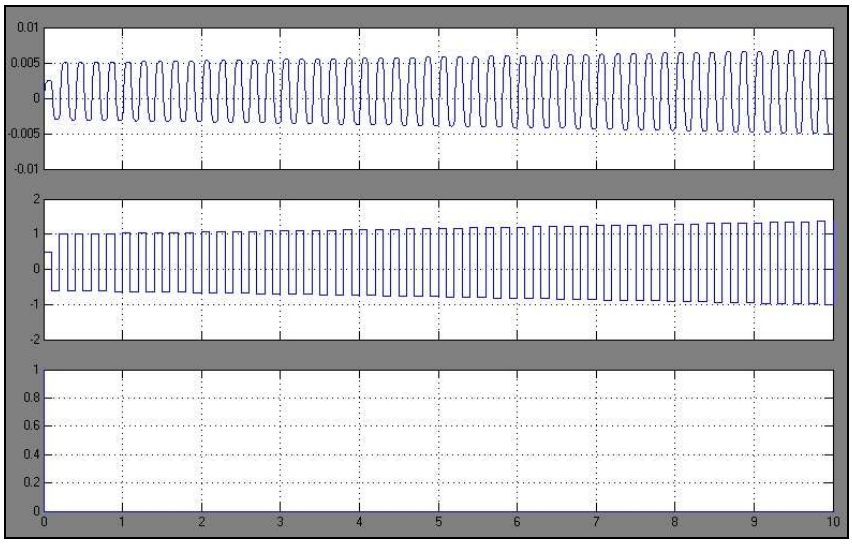


Figura D.10 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

f) PID Aström com T = 0,1 segundos:

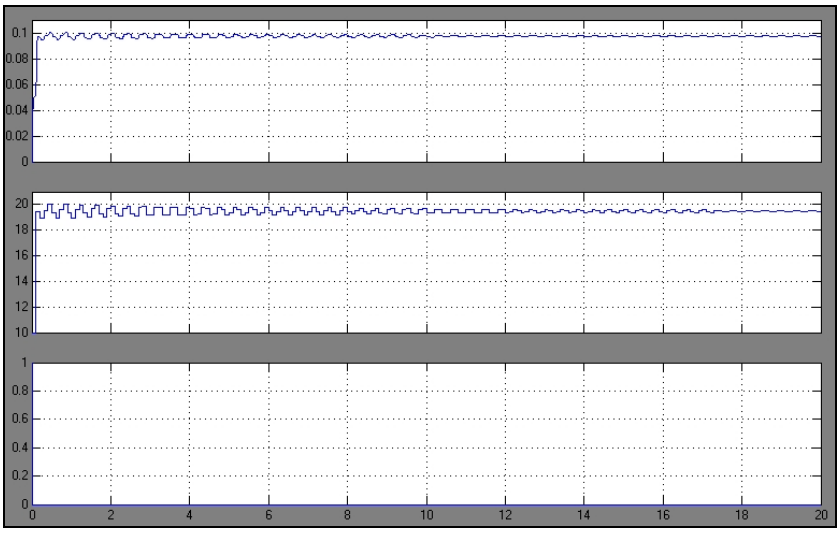


Figura D.11 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

g) PID Clássico com T = 0,5 segundo.

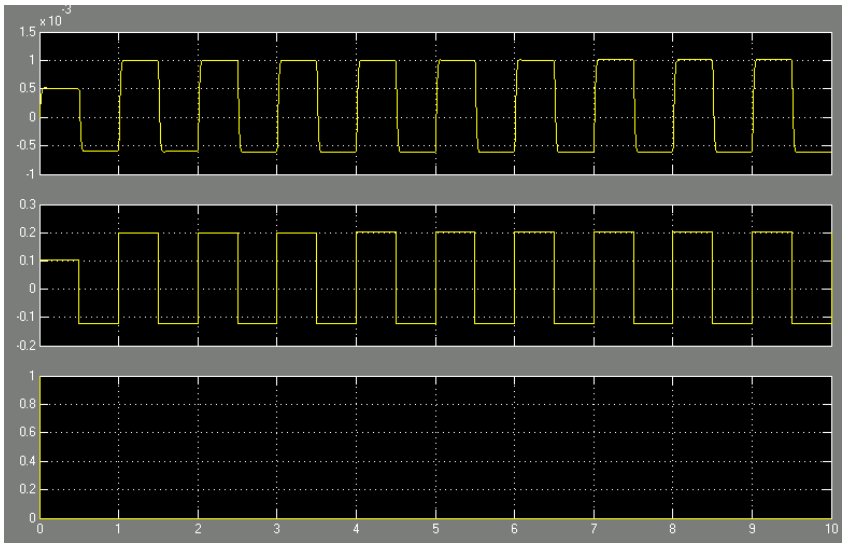


Figura D.12 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

h) PID Aström com T = 0,5 segundo.

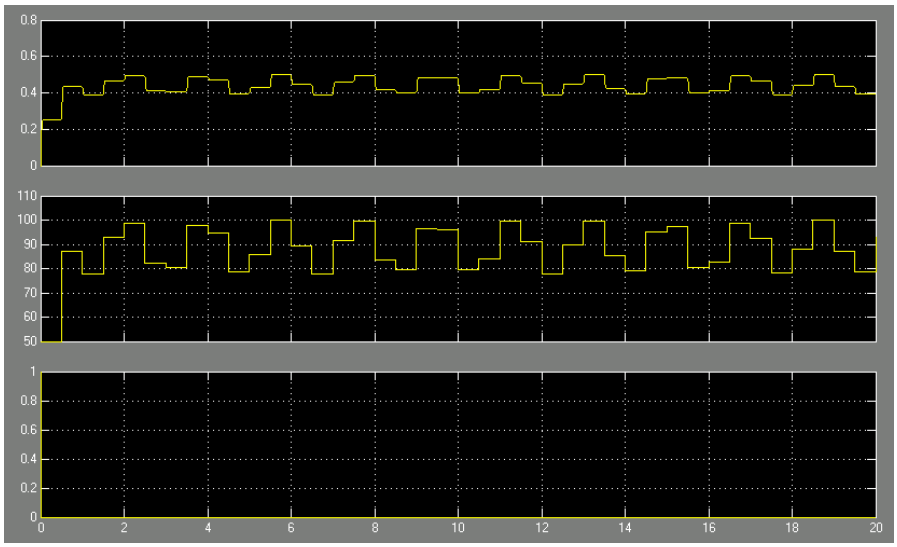


Figura D.13 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

i) PID Clássico com T = 1 segundo.

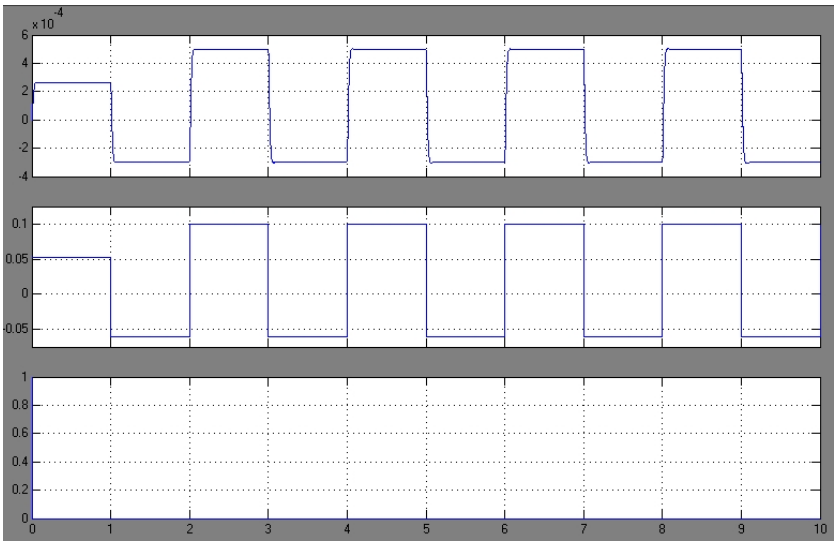


Figura D.14 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

j) PID Aström com T = 1 segundo.

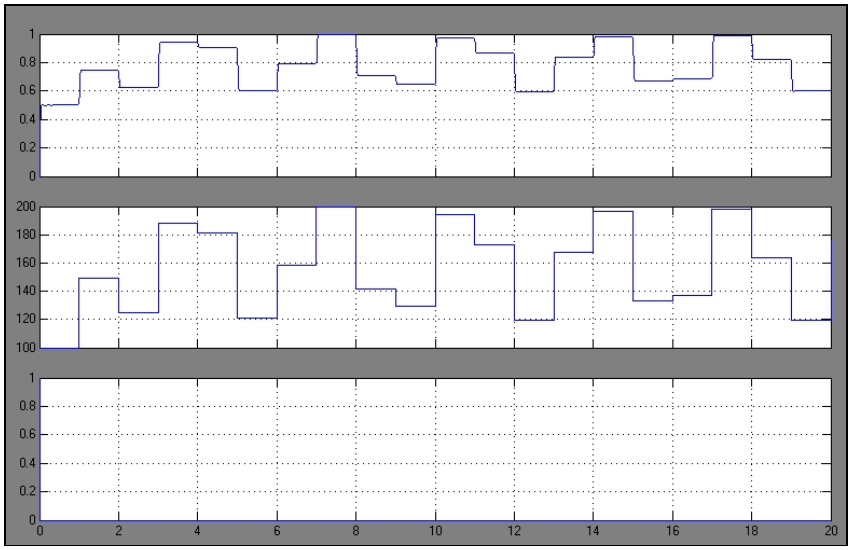


Figura D.15 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

k) PID Clássico com T = 1,5 segundos.

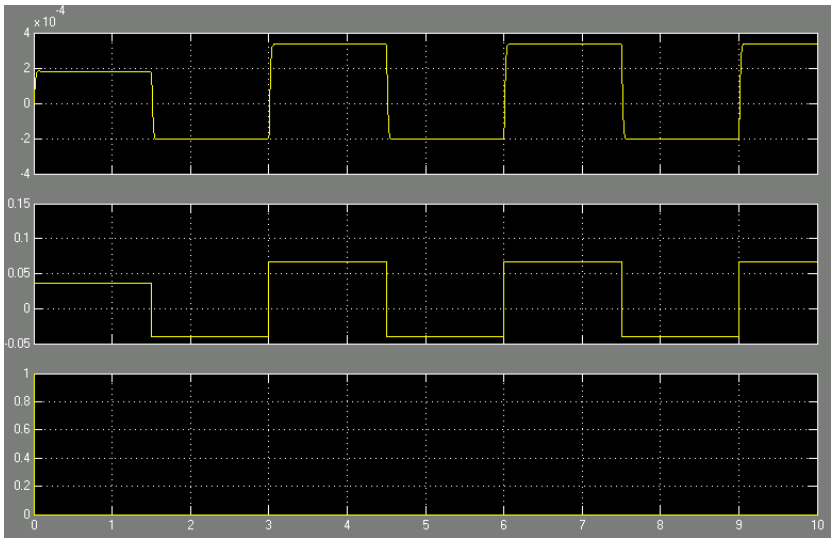


Figura D.16 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

l) PID Aström com T = 1,5 segundos.

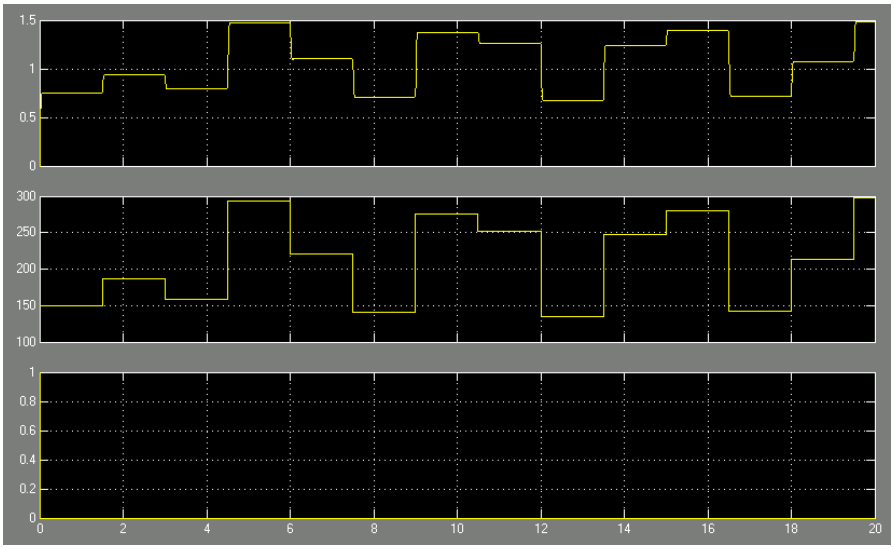


Figura D.17 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

m) PID Clássico com T = 2,0 segundos.

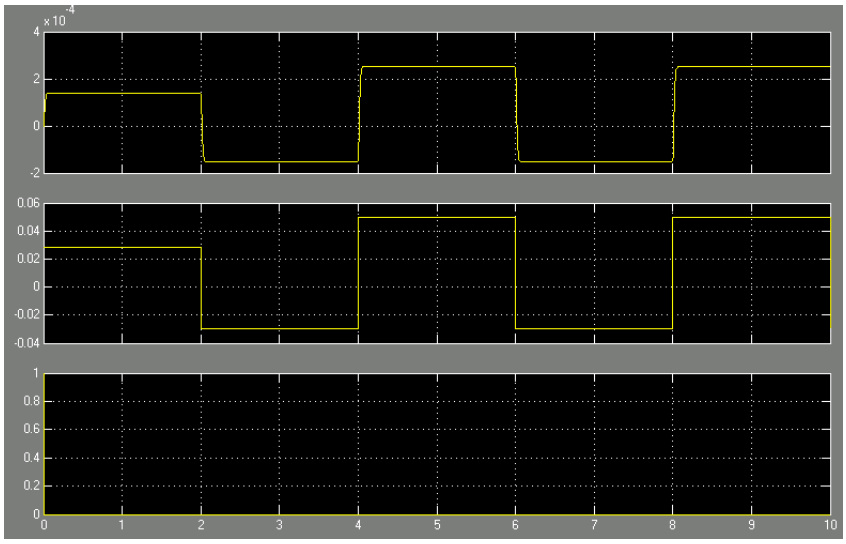


Figura D.18 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

n) PID Aström com T = 2,0 segundos.

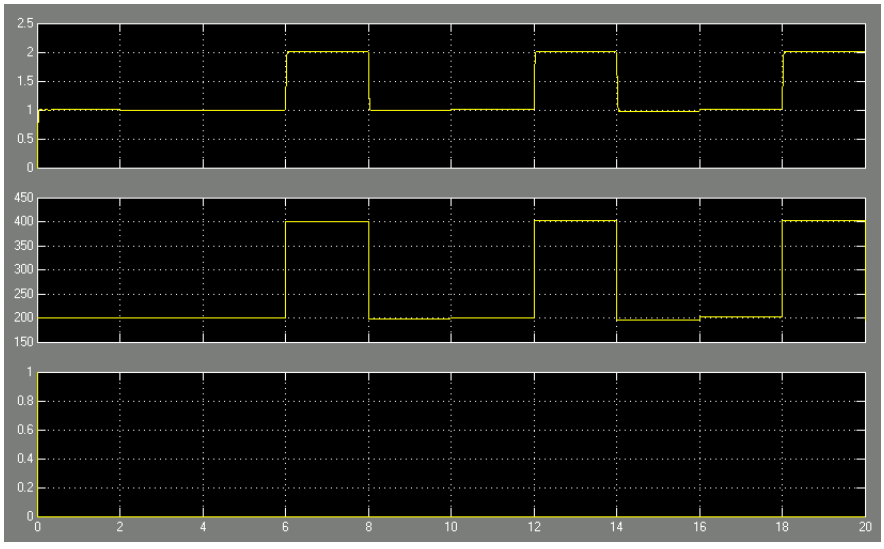


Figura D.19 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

o) PID Clássico com T = 2,5 segundos.

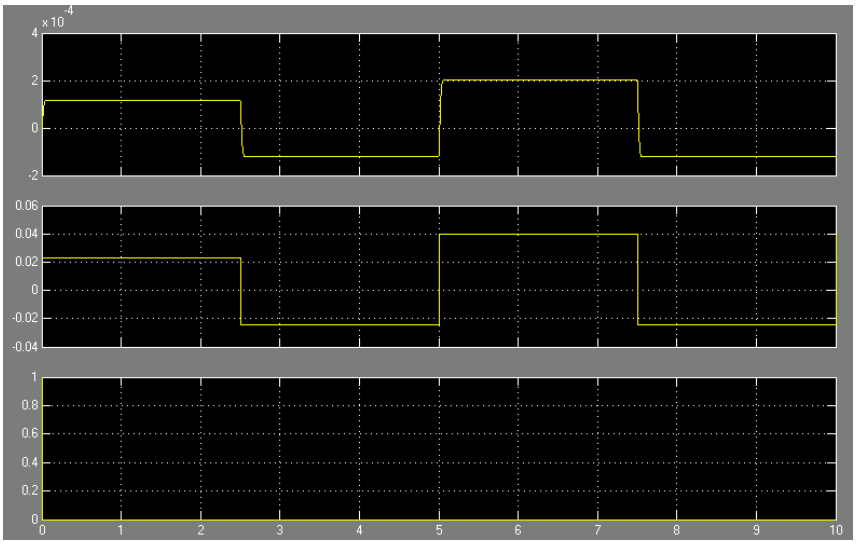


Figura D.20 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

p) PID Aström com T = 2,5 segundos:

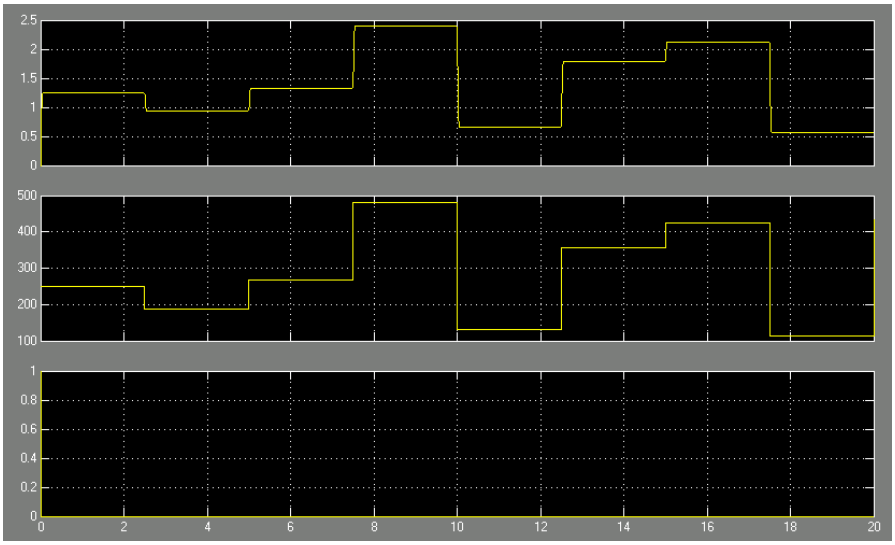


Figura D.21 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

q) PID Clássico com T = 3,0 segundos.

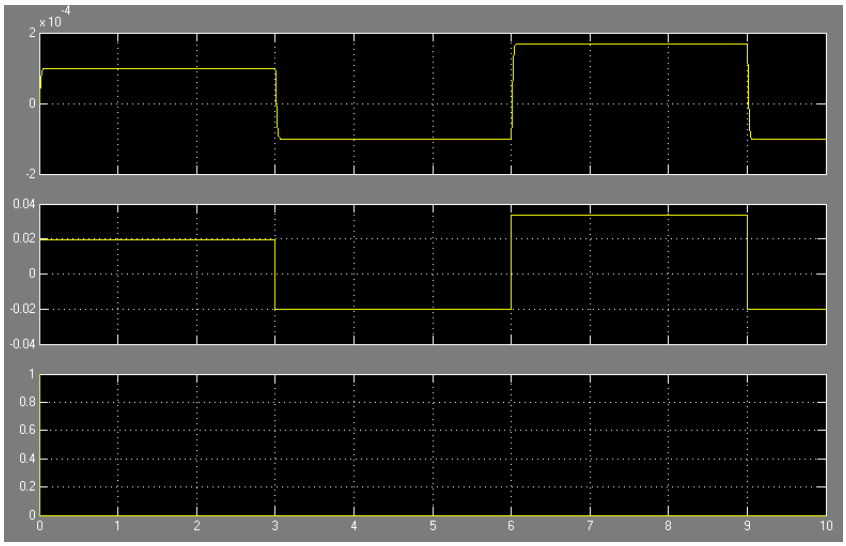


Figura D.22 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

r) PID Aström com T = 3,0 segundos.

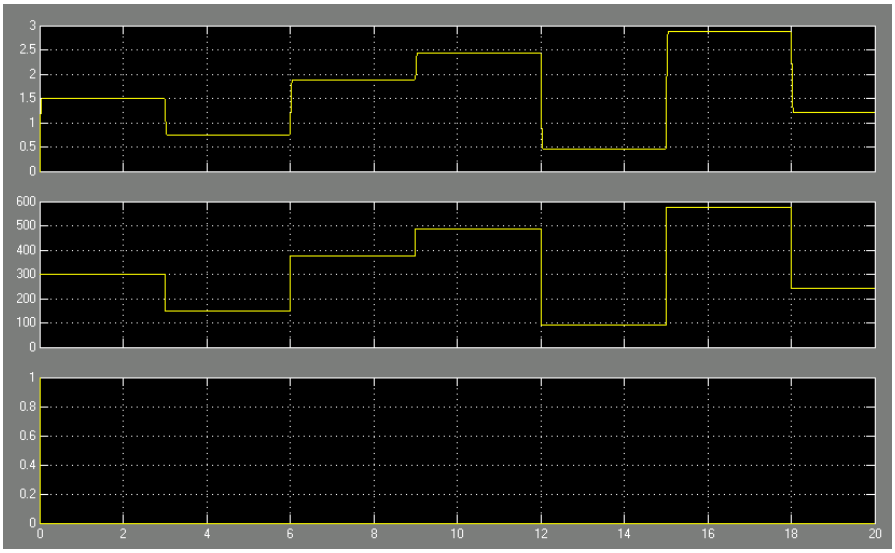


Figura D.23 – Resultados numéricos para a posição x(t), sinal de controle e pulso de duração 0,01 segundos.

Conforme podemos verificar a partir dos resultados da seção anterior, o PID de Aström de fato reduz o "overshoot" (sobressinal máximo), entretanto provoca o

surgimento de um nível não-nulo na posição x de um oscilador harmônico simples e subamortecido (um massa-mola com razão de amortecimento de ζ=0,9). Os diversos resultados obtidos foram reunidos na Tabela E.1.

Tabela D.1 - Análise da estabilidade em função do período de amostragem.

T [seg]	PID – Clássico	Máximo Sobre-sinal PID- Clássico	PID – Aström	Máximo Sobre-sinal PID – Aström
T=0.001	Assintoticamente estável	0,6	Assintoticamente estável	0,0073
T=0.05	Instável	-	Assintoticamente estável	0,05
T=0.1	Instável	-	Assintoticamente estável	0,1
T=0.5	Ciclo-limite	-	Caos	-
T=1	Ciclo-limite	-	Caos	-
T=1.5	Ciclo-limite	-	Caos	-
T=2.0	Ciclo-limite	-	Ciclo-limite	-
T=2.5	Ciclo-limite	-	Caos	-
T=3.0	Ciclo-limite	-	Caos	-

Nesta seção foi feito um estudo da análise comparativa de desempenho entre dois controladores digitais, o PID clássico digital e PID Aström digital, usando uma planta “padrão” de um oscilador-harmônico sub amortecido. Os dois controladores foram expostos a condições idênticas de simulação e obtivemos a confirmação do resultado de que o controlador PID Aström digital mostrou-se superior ao PID Clássico digital. Os benefícios principais do emprego do PID digital Aström foram a redução do sobressinal máximo (havendo o aparecimento de um nível pequeno, porém indesejável) e o aumento da robustez perante o aumento do período de amostragem. Fenômenos estranhos a projetos puramente analógicos também foram observados tais como o aparecimento de ciclos-limite e caos.

É muito importante considerar, que o modelo criado e estudado nesse Apêndice, representa um modelo linear, ou seja, não considera todas as nuances da estrutura real. Porém considera-se que seja válida a análise linearizada desde que o sistema amostrado não se afaste muito de seu ponto de operação inicialmente arbitrado. Portanto é conveniente que seja feito um estudo mais rigoroso para utilizar os resultados aqui obtidos em plantas muito mais complexas do que um

oscilador harmônico. Fica evidente o emprego do método científico aqui neste trabalho ao utilizarmos uma planta simples tal como o oscilador harmônico e a literatura mostra a eficiência de estudos iniciais como esse para depois servir de apoio para aplicações mais complexas.

APÊNDICE E

SIMULAÇÕES NUMÉRICAS COM AS REGRAS DE MAPEAMENTO – CONTROLE PD DA POSIÇÃO DE OSCILADOR HARMÔNICO E COMPARAÇÕES COM A REGRA DE TUSTIN, REGRA ST1 E REGRA ST2

Neste Apêndice as regras ST1 e ST2 apresentadas no Capítulo 3 são testadas com a regra clássica de Tustin em sistemas de controle discreto da posição de um oscilador harmônico e comparações de desempenho são realizadas.

E.1. INTRODUÇÃO

Foi analisado e simulado o controle da posição de um oscilador harmônico amortecido, supondo desprezível o atrito com o solo, dado por,

$$m.\ddot{x}(t) + b.\dot{x}(t) + k.x(t) = u(t) \tag{E.1}$$

Onde m é a massa do carrinho, k é a constante elástica da mola e b é o coeficiente de atrito viscoso. Aplicando a transformada de Laplace em (E.1) obtemos facilmente a seguinte função de transferência, supondo as condições iniciais nulas,

$$G(s) \stackrel{\Delta}{=} \frac{X(s)}{U(s)} = \frac{1}{k} \cdot \frac{\omega_n^{\,2}}{s^2 + 2.\zeta.\omega_n.s + \omega_n^{\,2}} \tag{E.2}$$

Onde $\omega_n = \sqrt{k/m}$ é a freqüência angular natural não amortecida deste modo de vibração e $\zeta = b/m$ é a razão de amortecimento.

Pelo método de extrapolação “step-invariance” (“zero-order hold” - ZOH), Franklin (1981) podemos obter:

$$G_{H0}(z) = (1 - z^{-1}).\ \mathcal{Z}\left\{ \mathcal{L}^{-1}\left\{ \frac{G(s)}{s} \right\}\Big|_{t=k.T} \right\} \tag{E.3}$$

Aplicando (E.2) em (E.3) e normalizando k=1 nós obtemos uma função de transferência de segunda ordem com o surgimento de um zero devido ao "hold" (segurador).

$$G_{H0}(z) = \frac{z\left[1 - 2e^{-\sigma T}.\cos(\omega_d T) - e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d T) - \cos(\omega_d T)\right)\right] + e^{-2\sigma T} + e^{-\sigma T}.\left(\frac{\sigma}{\omega_d}.\sin(\omega_d T) - \cos(\omega_d T)\right)}{z^2 - [2e^{-\sigma T}.\cos(\omega_d T)]z + e^{-2\sigma T}} \tag{E.4}$$

Convém lembrar que o controlador discreto estará "enxergando" a planta como $G_{HO}(z)$ e não como G(s). Observando (F.4) vê-se que existe uma dependência de $G_{HO}(z)$ do período de amostragem. Foi mostrado em **Tredinnick (1999)** que, dependendo do valor do período de amostragem T, o sistema de controle em malha-fechada pode se tornar instável, devido ao "aliasing" (falseamento).

Substituindo a expressão que governa o mapeamento Tustin num controle PD,

$$D(z) = k_p + k_d.s \tag{E.5}$$

teremos,

$$D(z) = \left(k_p + \frac{2.k_d}{T}\right).\frac{z + \left(\frac{k_p.T - 2.k_d}{k_p.T + 2.k_d}\right)}{z + 1} \tag{E.6}$$

Um controle PD projetado pela regra ST1 é dado por,

$$D(z) = PD_{control} = \left(k_p + \frac{2.k_d}{T}\right).\frac{z + \left(\frac{k_p.T.\xi - 2.k_d}{k_p.T + 2.k_d}\right)}{z + \xi} \tag{E.7}$$

Um controle PD projetado pela regra ST2, mais geral que a primeira, é dado por:

$$D(z) = PD_{control} = \left(k_p + \frac{2.k_d}{T}\right).\frac{z + \left(\frac{k_p.T.\xi_2 - 2.k_d.\xi_1}{k_p.T + 2.k_d}\right)}{z + \xi_2} \tag{E.8}$$

E.2. SIMULAÇÕES NUMÉRICAS

Foram usados nas simulações numéricas em MatLab os valores de constantes mostrados na Tabela E.1 (referenciados na Lista de Símbolos desta Tese).

Tabela E.1: constantes usadas nas simulações numéricas.

ζ	0.9
ω_N	0.3
k_p	3.0
k_d	4.8
ξ	0.1
ξ_1	0.8
ξ_2	0.1

Na Tabela E.2 podemos ver resultados de algumas simulações numéricas realizadas em MatLab. Nas Figuras E.1, E.2, E.3 e E.4 temos uma comparação gráfica usando diversos valores de período de amostragem T (medido em segundos) para as três regras: Tustin, ST1 e ST2.

Tabela E.2: resultados numéricos obtidos.

T	Regra de mapeamento	Estável	Instável
0.35	Tustin	✔	
	ST1	✔	
	ST2	✔	
0.7	Tustin		✔
	ST1	✔	
	ST2	✔	
1.0	Tustin		✔
	ST1	✔	
	ST2	✔	
1.2	Tustin		✔
	ST1	✔	
	ST2	✔	
1.6	Tustin		✔
	ST1	✔	
	ST2	✔	
3.0	Tustin		✔
	ST1	✔	
	ST2	✔	
4.0	Tustin		✔
	ST1	✔	
	ST2	✔	
4.5	Tustin		✔
	ST1		✔
	ST2		✔

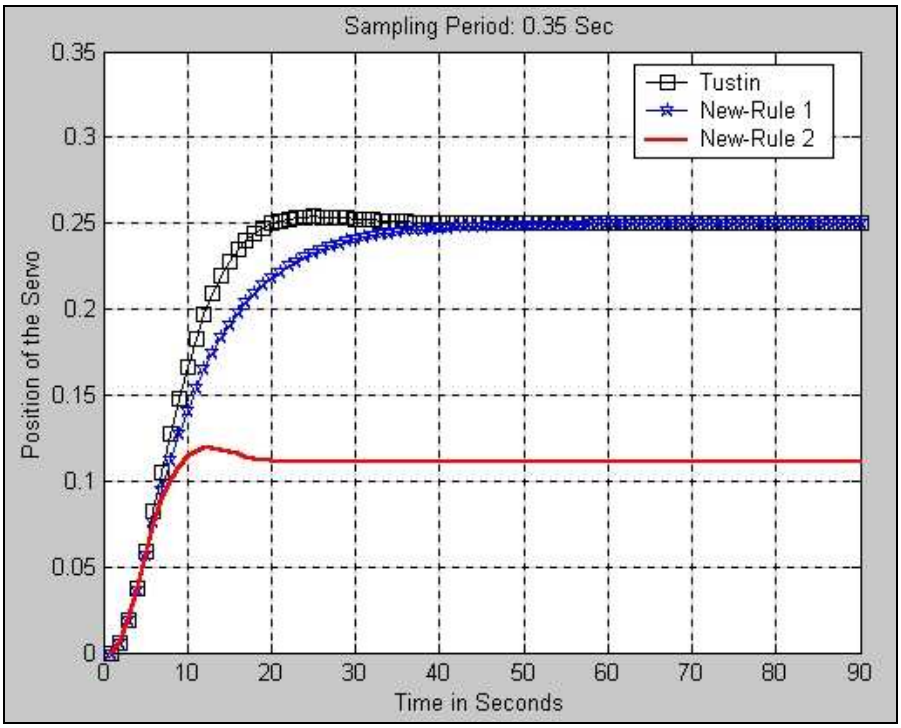


Figura E.1 – Resultados numéricos para T = 0,35 segundo.

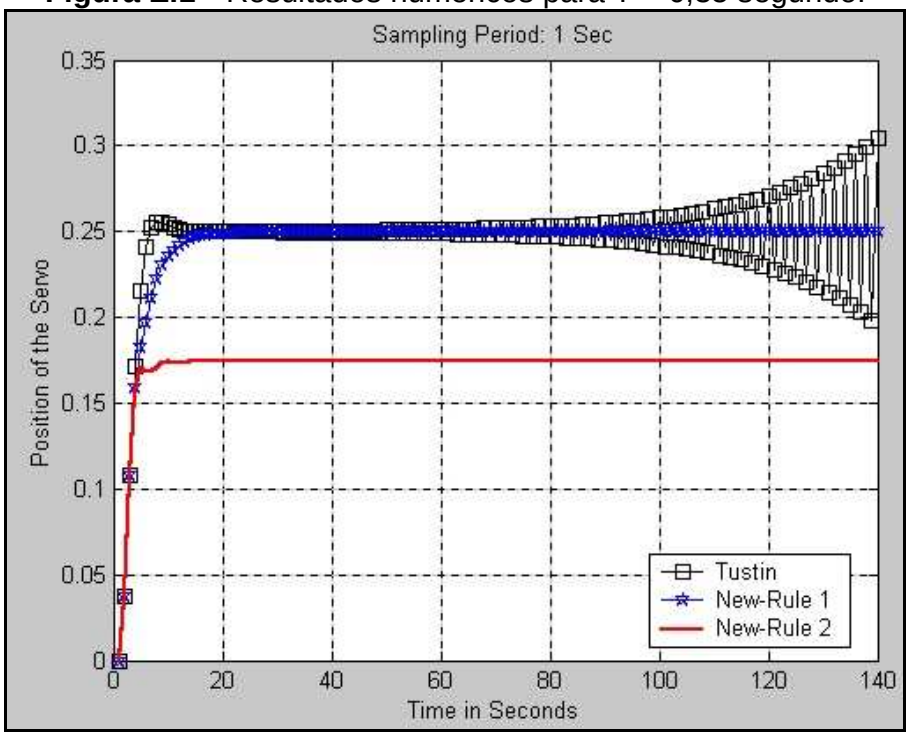


Figura E.2 – Resultados numéricos para T = 1 segundo.

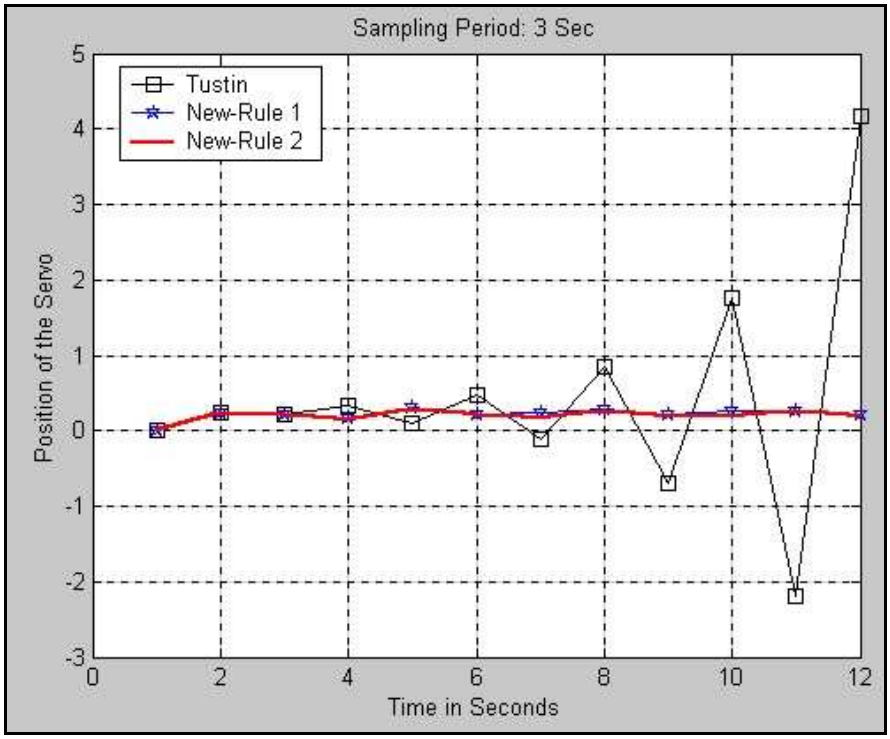


Figura E.3 – Resultados numéricos para T = 3 segundos.

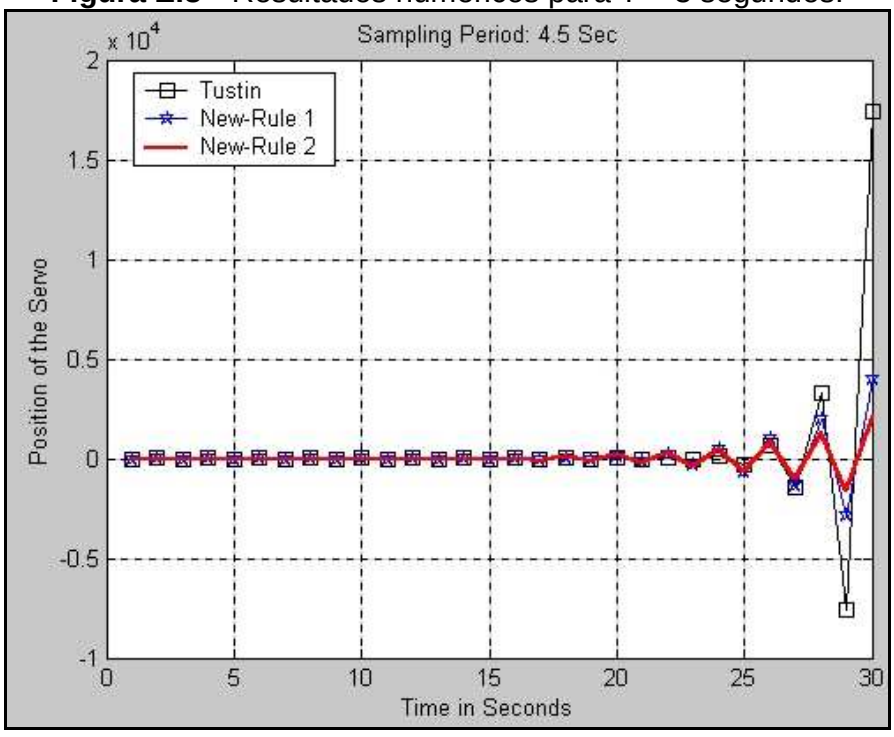


Figura E.4 – Resultados numéricos para T = 4,5 segundos.

Podemos concluir da Tabela E.2 e das Figuras E.1 a E.4 que as regras ST1 e ST2 são mais resistentes ao fenômeno do falseamento frente ao aumento do período

de amostragem T do que a regra clássica de Tustin. Em especial, a regra ST2 responde/rastreia a entrada em degrau unitário despendendo menos energia do que as demais usadas. A regra ST2 mostra-se assim melhor do que a regra ST1.

APÊNDICE F

APROXIMAÇÕES DA EXPONENCIAL DE MATRIZ POR McLAURIN-TAYLOR (MT) E PADÉ

Tal como utilizado no Capítulo 7 aqui neste Apêndice está ensinado como obter aproximações de Padé da função exponencial e também são apresentados exemplos de aproximações de Padé tais como [1/1], [3,2], [3,3], [6,6], [8,8]. [10,10], [12,12], [14,14] e [16,16].

F.1. INTRODUÇÃO

Neste Apêndice daremos uma simples explicação sobre as técnicas de MT e Padé iniciando com as séries de MT e posteriormente apresentando o algoritmo de Padé com exemplos.

Basicamente, uma função f(t) pode ser expressa por série de MT tal como em (F.1),

$$f(t) = \sum_{k=0}^{\infty} \frac{f^{[k]}(0)}{k!}.t^k = f(0) + \frac{df(0)}{dt}.t + \frac{d^2 f(0)}{dt^2}.\frac{t^2}{2} + ... \qquad \text{(F.1)}$$

A aproximação MT de enésima ordem de f(t) é claramente dada por,

$$f(t) \approx \sum_{k=0}^{N} \frac{f^{[k]}(0)}{k!}.t^k \qquad \text{(F.2)}$$

A aproximação racional de Padé é uma extensão natural das séries de McLaurin e Taylor fazendo uso dos coeficientes dessas expansões em séries em seus cálculos. O método de Padé usa polinômios racionais e assim se mostra mais preciso que o método MT.

Basicamente o algoritmo de Padé é resumido da seguinte forma,

$$\left[\sum_{k=0}^{N} \frac{f^{[k]}(0)}{k!}.t^k\right].\left(1 + q_1 t + q_2 t^2 + ... + q_M t^M\right) = p_0 + p_1 t + p_2 t^2 + ... + p_L t^L \qquad \text{(F.3)}$$

Onde o objetivo é encontrar um sistema linear que solucione os coeficientes q_i e p_i tendo N=L+M, e finalmente expressar a aproximação de Padé de grau [L/M] de f(t) conforme,

$$f(t)_{[L/M]} \approx \frac{p_0 + p_1 t + p_2 t^2 + ... + p_L t^L}{1 + q_1 t + q_2 t^2 + ... + q_M t^M} \tag{F.4}$$

Exemplo 1: seja $f(t) = e^{-t}$ calcule a aproximação de Padé para $f_{[3/2]}$.

A expansão MT será de ordem N=L+M=5,

$$\left[1 - t + \frac{t^2}{2} - \frac{t^3}{6} + \frac{t^4}{24} - \frac{t^5}{120}\right].\left(1 + q_1 t + q_2 t^2\right) = p_0 + p_1 t + p_2 t^2 + p_3 t^3 \tag{F.5}$$

o que dá,

$$.1 + q_1 t + q_2 t^2 - t - q_1 t^2 - q_2 t^3 + \frac{t^2}{2} + \frac{q_1 t^3}{2} + \frac{q_2 t^4}{2} - \frac{t^3}{6} - \frac{q_1 t^4}{6} - \frac{q_2 t^5}{6} + \frac{t^4}{24} + \frac{q_1 t^3}{24} + \frac{q_2 t^6}{24} +$$
$$- \frac{t^5}{120} - \frac{q_1 t^6}{120} - \frac{q_2 t^7}{120} = p_0 + p_1 t + p_2 t^2 + p_3 t^3 \tag{F.6}$$

isso resulta no seguinte sistema linear de N=5 equações:

$$\begin{cases} t^0: & p_0 = 1 \\ t^1: & p_1 = q_1 - 1 \\ t^2: & p_2 = q_2 - q_1 + \frac{1}{2} \\ t^3: & p_3 = -q_2 + \frac{q_1}{2} - \frac{1}{6} \\ t^4: & \frac{q_2}{2} - \frac{q_1}{6} + \frac{1}{24} = 0 \\ t^5: & -\frac{q_2}{6} + \frac{q_1}{24} - \frac{1}{120} = 0 \end{cases} \tag{F.7}$$

Que dá,

$$q_1 = \frac{2}{5}; q_2 = \frac{1}{20}; p_0 = 1; p_1 = \frac{-3}{5}; p_2 = \frac{3}{20}; p_3 = \frac{-1}{60}. \tag{F.8}$$

Finalmente, tal aproximação por Padé será:

$$e^{-t} \approx e^{-t}{}_{[3/2]} = \frac{1 - \frac{3}{5}t + \frac{3}{20}t^2 - \frac{1}{60}t^3}{1 + \frac{2}{5}t + \frac{1}{20}t^2} \qquad \text{(F.9)}$$

■

Alguns resultados comparatives das aproximações da exponencial e^{-t} por MT e Padé podem ser vistos na Figura F.1 para aproximação MT de quinta ordem e Padé [1/1],[3/2] e [3/3]. Os erros de aproximação relacionados com a Figura F.1 encontram-se na Figura G.2 reforçando a idéia de que o método de Padé é a melhor técnica de aproximação dentre as apresentadas.

Exemplo 2: dada $f(s) = e^{sT}$ mostre que $f_{[1/1]}$ corresponde ao mapeamento s-z Tustin (bilinear ou trapezoidal). Para $f_{[1/1]}$ L=M=1, N=2. O "T" aqui é o período de amostragem dos conversores A/D e D/A. A expansão MT de segunda ordem para tal caso é: $1 + Ts + \frac{T^2 s^2}{2}$. Aplicando isso no algoritmo de Padé vem,

$$\left[1 + Ts + \frac{T^2 s^2}{2}\right].(1 + q_1 s) = p_0 + p_1 s \qquad \text{(F.10)}$$

$$1 + Ts + \frac{T^2 s^2}{2} + q_1 s + q_1 T s^2 + q_1 \frac{T^2 s^3}{2} = p_0 + p_1 s \qquad \text{(F.11)}$$

que fornece o seguinte sistema linear,

$$\begin{cases} s^0 : p_0 = 1 \\ s^1 : p_1 = T + q_1 \\ s^2 : \frac{T^2}{2} + q_1 T = 0 \end{cases} \qquad \text{(F.12)}$$

que dá,

$$q_1 = -T/2;\ p_0 = 1;\ p_1 = T/2 \qquad \text{(F.13)}$$

Finalmente, chegamos à aproximação por Padé,

$$e^{sT} \approx e^{sT}{}_{[1/1]} = \frac{p_0 + p_1 s}{1 + q_1 s} = \frac{1 + {}^{T}\!/_{2}\, s}{1 - {}^{T}\!/_{2}\, s} \tag{F.14}$$

Chamando $z \coloneqq e^{sT}$ temos,

$$z \approx \frac{1 + {}^{T}\!/_{2}\, s}{1 - {}^{T}\!/_{2}\, s} \tag{F.15}$$

Que é o mapeamento Tustin. Para chegarmos a uma expressão mais familiar isola-se o “s” acima para obtermos,

$$s \approx \frac{2}{T}\frac{z-1}{z+1} \tag{F.16}$$

Que representa o mapeamento clássico Tustin encontrado na literatura clássica do assunto (**Tredinnick, 1999**; **Franklin, 1981**; **Katz, 1981**).

F.2. SIMULAÇÕES NUMÉRICAS DA FUNÇÃO EXPONENCIAL

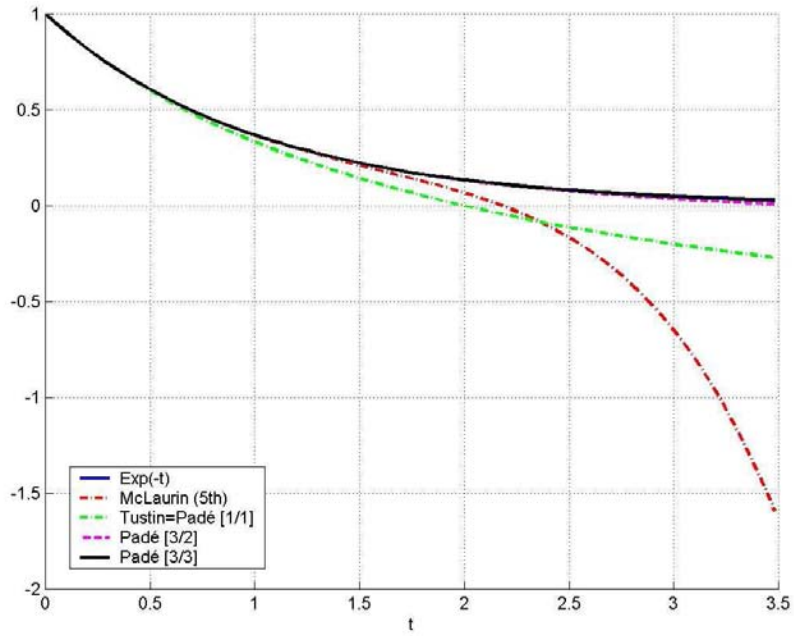


Figura F.1: aproximações para e^{-t}.

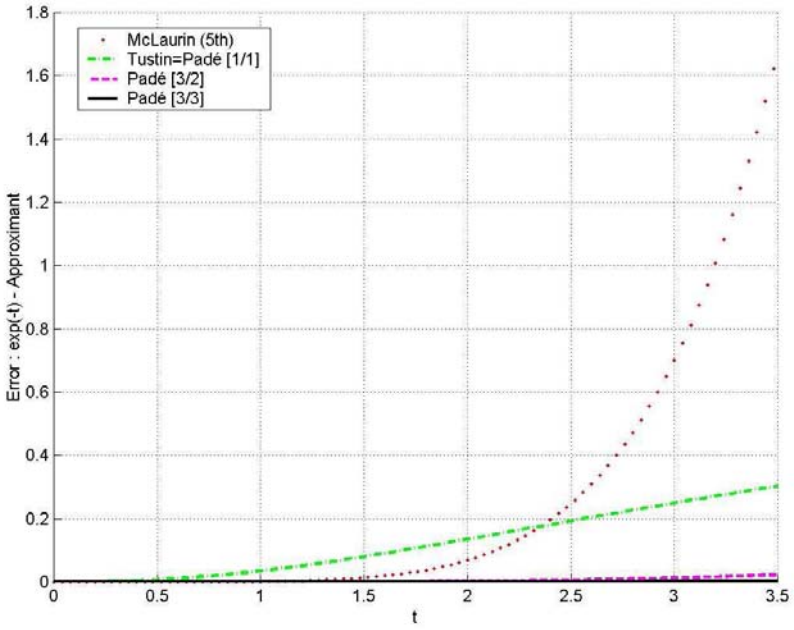


Figura F.2: erro na aproximação da exponencial.

A aproximação de Padé de terceira ordem da exponencial usada neste Apêndice (Figuras F.1 e F.2) bem como as aproximações de Padé da exponencial utilizadas para as simulações no Capítulo 7 são:

$$e^{AT}_{[3/3]} = (120.I\text{-}60TA + 12T^2A^2\text{-}T^3A^3)^{-1}(120.I + 60TA + 12T^2A^2 + T^3A^3) \tag{F.17}$$

$$e^{AT}_{[6/6]} = (\,665280.I\text{-}332640.TA + 75600.T^2A^2\text{-}10080.T^3A^3 + 840.T^4A^4 - 42.T^5A^5 + T^6A^6\,)^{-1}$$
$$(\,665280.I + 332640.TA + 75600.T^2A^2 + 10080.T^3A^3 + 840.T^4A^4 + 42.T^5A^5 + T^6A^6\,) \tag{F.18}$$

$$e^{AT}_{[8/8]} = (518918400.I\text{-}259459200.TA + 60540480.T^2A^2\text{-}8648640.T^3A^3 + 831600T^4A^4\text{-}55440T^5A^5 +$$
$$+ 2520T^6A^6\text{-}72T^7A^7 + T^8A^8)^{-1}(518918400.I + 259459200.TA + 60540480.T^2A^2 +$$
$$+ 8648640.T^3A^3 + 831600.T^4A^4 + 55440.T^5A^5 + 2520.T^6A^6 + 72.T^7A^7 + T^8A^8) \tag{F.19}$$

$$e^{AT}_{[10/10]} = (\,670442572800.I\text{-}335221286400.TA + 79394515200.T^2A^2\text{-}11762150400.T^3A^3 +$$
$$+ 1210809600.T^4A^4\text{-}90810720.T^5A^5 + 5045040.T^6A^6\text{-}205920.T^7A^7 + 5940.T^8A^8 - 110.T^9A^9 + T^{10}A10\,)^{-1}$$
$$(\,670442572800.I + 335221286400.TA + 79394515200.T^2A^2 + 11762150400.T^3A^3 +$$
$$+ 1210809600.T^4A^4 + 90810720.T^5A^5 + 5045040.T^6A^6 + 205920.T^7A^7 + 5940.T^8A^8 + 110.T^9A^9 + T^{10}A^{10}\,) \tag{F.20}$$

$$e^{AT}_{[12/12]} = (1.2952950506496.10^{15}.I\text{-}647647525324800.TA + 154872234316800.T^2A^2\text{-}23465490048000.T^3A^3 +$$
$$+ 2514159648000.T^4A^4\text{-}201132771840.T^5A^5 + 12350257920.T^6A^6\text{-}588107520.T^7A^7 + 21621600.T^8A^8 +$$
$$- 600600.T^9A^9 + 12012.T^{10}A^{10} - 156.T^{11}A^{11} + T^{12}A^{12})^{-1}$$
$$(1.2952950506496.10^{15}.I + 647647525324800.TA + 154872234316800.T^2A^2 + 23465490048000.T^3A^3 +$$
$$+ 2514159648000.T^4A^4 + 201132771840.T^5A^5 + 12350257920.T^6A^6 + 588107520.T^7A^7 + 21621600.T^8A^8 +$$
$$+ 600600.T^9A^9 + 12012.T^{10}A^{10} + 156.T^{11}A^{11} + T^{12}A^{12}) \tag{F.21}$$

$$e^{AT}_{[14/14]} = (\,3.49729663675392.10^{18}.I\text{-}1.74864831837696.10^{18}.TA + 4.2097089146112.10^{17}.T^2A^2\text{-}6.476475253248.10^{16}.T^3A^3 +$$
$$+ 7.1241227785728.10^{15}.T^4A^4\text{-}593676898214400.T^5A^5 + 3871805857920.T^6A^6\text{-}2011327718400.T^7A^7 + 83805321600.T^8A^8 +$$
$$- 2793510720.T^9A^9 + 73513440.T^{10}A^{10} - 1485120.T^{11}A^{11} + 21840.T^{12}A^{12} - 210.T^{13}A^{13} + T^{14}A^{14}\,)^{-1}$$
$$(\,3.49729663675392.10^{18}.I + 1.74864831837696.10^{18}.TA + 4.2097089146112.10^{17}.T^2A^2 + 6.476475253248.10^{16}.T^3A^3 +$$
$$+ 7.1241227785728.10^{15}.T^4A^4 + 593676898214400.T^5A^5 + 3871805857920.T^6A^6 + 2011327718400.T^7A^7 + 83805321600.T^8A^8 +$$
$$+ 2793510720.T^9A^9 + 73513440.T^{10}A^{10} + 1485120.T^{11}A^{11} + 21840.T^{12}A^{12} + 210.T^{13}A^{13} + T^{14}A^{14}\,) \tag{F.22}$$

$$e^{AT}{}_{[16/16]} = (1.25762787057671.10^{22}.I\text{-}6.28813935288355.10^{21}.TA + 1.52132403698796.10^{21}.T^2A^2\text{-}2.36650405753682.10^{20}.T^3A^3 +$$
$$+ 2.65211661620506.10^{19}.T^4A^4\text{-}.2.27324281389005.10^{18}.T^5A^5 + 1.54355993535744.10^{17}.T^6A^6\text{-}8.48109854592.10^{15}.T^7A^7 +$$
$$+ 381649434566400.T^8A^8 - 14135164243200.T^9A^9 + 430200650880.T^{10}A^{10} - 10666131840.T^{11}A^{11} + 211629600.T^{12}A^{12} +$$
$$- 3255840.T^{13}A^{13} + 36720.T^{14}A^{14} - 272.T^{15}A^{15} + T^{16}A^{16})^{-1}$$
$$(1.25762787057671.10^{22}.I + 6.28813935288355.10^{21}.TA + 1.52132403698796.10^{21}.T^2A^2 + 2.36650405753682.10^{20}.T^3A^3 +$$
$$+ 2.65211661620506.10^{19}.T^4A^4 + .2.27324281389005.10^{18}.T^5A^5 + 1.54355993535744.10^{17}.T^6A^6 + 8.48109854592.10^{15}.T^7A^7 +$$
$$+ 381649434566400.T^8A^8 + 14135164243200.T^9A^9 + 430200650880.T^{10}A^{10} + 10666131840.T^{11}A^{11} + 211629600.T^{12}A^{12} +$$
$$+ 3255840.T^{13}A^{13} + 36720.T^{14}A^{14} + 272.T^{15}A^{15} + T^{16}A^{16})$$

F.23)

APÊNDICE G

DEFINIÇÕES BÁSICAS DE ANÁLISE FUNCIONAL*

Muito útil e interessante no estudo avançado do mapeamento analógico-digital bem como no estudo da análise funcional de operadores a diferenças-finitas são algumas definições elementares que estão apresentadas neste Apêndice.

Definição G.1 – Função (*ou Mapeamento, ou Transformação, ou Operador*)

Uma função f de um conjunto A para um conjunto B é uma relação denotada por $f : A \to B$, onde:

i) $f \subseteq A \times B$

ii) $\forall x \in A \ \exists \ y \in B : (x, y) \in f$

iii) $\forall x \in A \ se \ y_1, y_2 \in B \, e \, (x, y_1) \in f \, e \, (x, y_2) \in f \, então \ y_1 = y_2$

O conjunto A é o **Domínio** de f, o conjunto B é chamado **Contradomínio** de f, o elemento $y \in B$ em $f(x) = y$ é chamado de **Imagem** de $x \in A$, o elemento $f^{-1}(y) = \{x : f(x) = y, x \in A, y \in B\}$ é a **pré-imagem** de y e o ***Range*** ou **Conjunto Imagem** é dado por:

$$ran(f) = \{f(a) : a \in A\} \qquad (G.1)$$

Definição G.2 – Grafo

É o conjunto dos pares ordenados (a,b) onde $a \in A$ e $b \in B$ relacionados pela função $f : A \to B$ e é denotado por,

$$grafo \ f = \{(a, f(a)) : a \in A\} \qquad (G.2)$$

* Lax (2002), Hislop (1991); Naylor (1982); Craig (1981); Oden (1979); D'Ambrosio (1977); Balakrishnan (1976); Zarantonello (1971); Lorch (1962); Sobolev (1963).

Classificação de funções. Na Figura G.1 estão mostrados os diversos tipos de funções que existem.

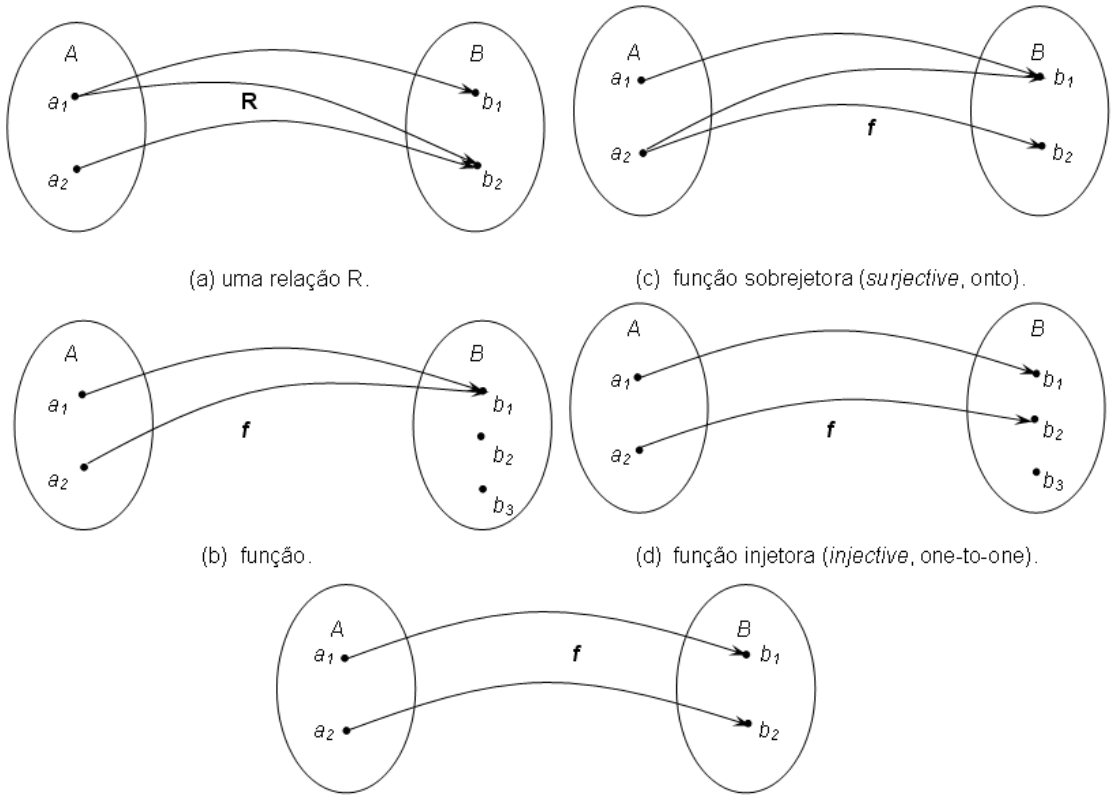


Figura G.1 - Classificação das funções $f : A \to B$.

Observação: dois conjuntos A e B são “semelhantes” (A ~ B) se há uma função biunívoca f que mapeia A em B (Oden (1979), pág.42).

Teorema G.1 (**Oden, 1979**):

Uma função $f : A \to B$ é inversível se e somente se ela for biunívoca.

Definição G.3 – Morfismos

Morfismo é um mapeamento entre dois objetos numa categoria abstrata que no caso de conjuntos pode ser:

Homeomorfismo: um morfismo geral entre dois conjuntos A e B.

Isomorfismo: é um morfismo biunívoco ("*onto*" e "*one-to-one*") entre dois conjuntos A e B.

Automorfismo: é um Isomorfismo de um conjunto nele mesmo.

Endomorfismo: é um morfismo sobrejetor de um conjunto nele mesmo.

Monomorfismo: é um morfismo $f : A \to B$ onde para quaisquer morfismos $u, v : C \to A$, $fu = fv$ implicar em *u = v.*

Epimorfismo: é um morfismo $f : A \to B$ onde para quaisquer morfismos $u, v : C \to A$, $uf = vf$ implicar em *u = v.*

<u>Definição G.4</u> – Grupóide

Grupóide é qualquer conjunto no qual uma operação fechada foi definida. Obs: por operação fechada entende-se qualquer operação binária (entre dois elementos de um conjunto) em que o resultado de tal operação também pertence a esse conjunto. <u>Exemplo</u>: o produto (operação binária de multiplicação: ".") entre dois números reais possui o resultado claramente pertencendo ao conjunto dos números reais.

<u>Definição G.5</u> – Semigrupo

É um Grupóide associativo, i.e., Grupóide que preserva a "lei associativa": $a \circ (b \circ c) = (a \circ b) \circ c$, sendo "$\circ$" uma operação binária qualquer. Um exemplo de semigrupo é a "matriz de transição" $\phi(t - t_0) = e^{A(t-t_0)}$.

<u>Definição G.6</u> – Monóide

É um Semigrupo que possua o elemento identidade.

<u>Definição G.7</u> – Grupo

É um Monóide que possua inversa.

<u>Definição G.8</u> – Grupo Abeliano

É um grupo que seja comutativo. Este termo foi cunhado por Niels Henrik Abel em 1823 e em lembrança a ele possui o nome "abeliano"

Definição G.9 – Anel

É um sistema abstrato $R = \{S,+,*\}$ que possui as operações binárias "+" e "*" e:

i) O sub-sistema $\{S,+\}$ é um grupo abeliano, i.e., com elemento-identidade "0".

ii) O sub-sistema $\{S,*\}$ é um semigrupo.

iii) Para todos $r_1, r_2 r_3 \in S$ a operação * é distributiva em relação a +:

$$r_1 * (r_2 + r_3) = r_1 * r_2 + r_1 * r_3$$

e

$$(r_2 + r_3) * r_1 = r_2 * r_1 + r_3 * r_1$$

Definição G.10 – Campo

É um sistema abstrato $\Im = \{S,+,*\}$ que consiste de um conjunto contendo pelo menos dois elementos nos quais duas operações binárias + e * foram definidas considerando que:

i) o sub-sistema $\{S,+\}$ é um grupo abeliano, i.e., com elemento-identidade "0".

ii) o sub-sistema $\{S,*\}$ é um grupo abeliano, i.e., com elemento-identidade "e".

iii) a operação * é distributiva em relação a +.

Definição G.11 – Espaço Vetorial

Seja $\Im$ um campo. Um espaço vetorial V sobre $\Im$ é um conjunto com duas operações fechadas $+ : V \times V \to V$ e $* : \Im \times V \to V$, tal que são válidas as considerações:

i) a operação + é fechada: $\underline{u} + \underline{v} \in V \ \forall \underline{u}, \underline{v} \in V$

ii) lei associativa (operação +): $(\underline{u} + \underline{v}) + \underline{w} = \underline{u} + (\underline{v} + \underline{w}) \quad \forall \underline{u}, \underline{v}, \underline{w} \in V$

iii) elemento identidade (operação +): $\exists \underline{0} \in V : \underline{u} + \underline{0} = \underline{u} \ \forall u \in V$

iv) elemento inverso (operação +): $\forall \underline{u} \in V \ \exists \underline{v} \in V : \underline{u} + \underline{v} = \underline{0}$

v) lei comutativa (operação +): $\underline{u} + \underline{v} = \underline{v} + \underline{u} \quad \forall \underline{u}, \underline{v} \in V$

vi) a operação * é fechada: $\alpha . \underline{u} \in V \ \forall \underline{u} \in V \text{ e } \forall \alpha \in \Im$

vii) lei associativa (operação *): $(\alpha.\beta).\underline{u} = \alpha.(\beta.\underline{u}) \quad \forall \underline{u} \in V \text{ e } \forall \alpha, \beta \in \Im$

viii) elemento identidade (operação *): $\exists e \in \Im : e.\underline{u} = \underline{u} \quad \forall u \in V$

ix) lei distributiva (operação *): $\left.\begin{array}{l}(\alpha+\beta).\underline{u} = \alpha.\underline{u} + \beta.\underline{u} \\ \underline{u}.(\alpha+\beta) = \underline{u}.\alpha + \underline{u}.\beta\end{array}\right| \quad \forall \underline{u} \in V \text{ e } \forall \alpha, \beta \in \Im$

<u>Definição G.12</u> – Funcional Linear

a) Um funcional linear $S = L(\underline{v})$ é uma função (ou mapeamento) de um espaço linear vetorial real V (onde o vetor <u>*v*</u> está contido) num campo numérico $\Im$ que na maioria dos casos práticos é o conjunto dos números reais *R*, e $S \in R$.

b) (Lorch, 1962, pág.10) Se X é um espaço de Banach (real ou complexo), um funcional sobre X é um mapeamento, isto é, uma função, de X para os escalares (reais ou complexos). Por exemplo, a função norma $\mathrm{f} \to \|\mathrm{f}\|$ é um funcional.

Particularmente, um funcional L é dito ser linear se o princípio da superposição e a condição de proporcionalidade são satisfeitos, respectivamente, sendo f e g funções contínuas: (i) $\mathrm{L}(\mathrm{f}+\mathrm{g}) = \mathrm{L}(\mathrm{f}) + \mathrm{L}(\mathrm{g})$ e (ii) $\mathrm{L}(\alpha \mathrm{f}) = \alpha.\mathrm{L}(\mathrm{f})$.

<u>Sugestão de definição</u> (M. Tredinnick): um Funcional L é um mapeamento de um espaço de Hilbert H, ou de um espaço de Banach B, para o conjunto dos números complexos C ou dos reais R. Tal Funcional L é dito ser um Funcional Linear se o princípio da superposição e a condição de proporcionalidade forem satisfeitos, i.e., $\mathrm{L}(\alpha_1 \mathrm{x}_1 + \alpha_2 \mathrm{x}_2) = \alpha_1 \mathrm{L}(\mathrm{x}_1) + \alpha_2 \mathrm{L}(\mathrm{x}_2)$, para todos $\alpha_1, \alpha_2 \in \mathrm{C}$ e $\mathrm{x}_1, \mathrm{x}_2 \in \mathrm{H}$, ou $\mathrm{x}_1, \mathrm{x}_2 \in \mathrm{B}$.

<u>Definição G.13</u> – Espaço Dual Algébrico

Algébrico Dual V* é o espaço de todos os Funcionais Lineares sobre V.

<u>Definição G.14</u> – Vizinhança

Seja $a \in A$. A vizinhança de a é o conjunto de todos os elementos $x \in A : |x - a| < \delta$, sendo δ um número real positivo.

Definição G.15 – Pontos interiores e Interior de um Conjunto

$a \in A$ é um ponto interior se existe uma vizinhança $|x-a| < \delta \quad \forall x \in A$. O *Interior* de A é o conjunto de todos os seus pontos interiores. Na figura 2 podemos notar que os elementos a, x_1 e x_2 são pontos interiores de A, porém x_3 não é.

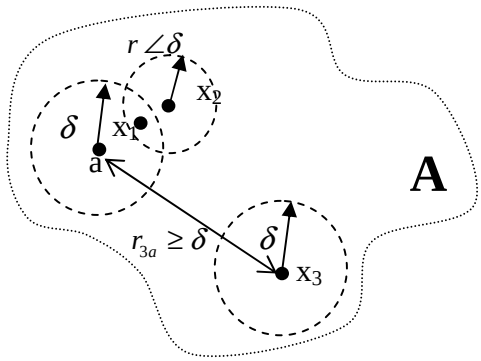


Figura G.2 - Pontos interiores de A.

Definição G.16 – Pontos Limite ou Pontos de Acumulação

Um elemento x não necessariamente em A é *ponto de acumulação* de A se e somente se a vizinhança de x contiver ao menos um elemento de A distinto de x. O conjunto dos pontos de acumulação de A é denotado por $\hat{A}$.

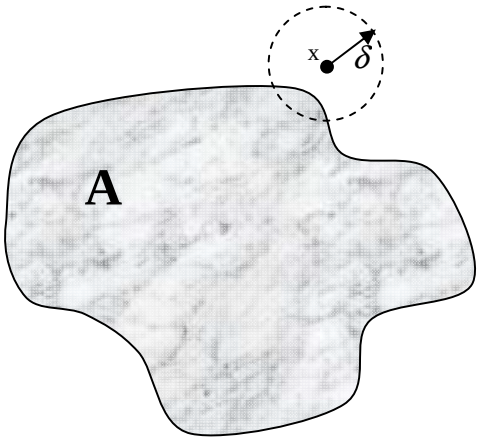


Figura G.3 - Ponto de Acumulação x.

Definição G.17 – Conjunto Aberto

Um conjunto A é aberto se todo $x \in A$ é um ponto interior de A.

Definição G.18 – Fecho (“*Closure*”)

O fecho de um conjunto é a união de $\hat{A}$ com o próprio A. O fecho é $\overline{A} = A \bigcup \hat{A}$.

Definição G.19 – Conjunto Fechado

Um conjunto é fechado se seu complemento é aberto.

Definição G.20 – Cobertura de um Conjunto

Uma classe $\wp$ (conjunto no qual seus elementos também são conjuntos) é dita cobrir o conjunto A se $A \subset \bigcup \{B : B \in \wp\}$. $\wp$ é a Cobertura do conjunto A. Se os conjuntos em $\wp$ são abertos então $\wp$ é uma cobertura aberta.

Definição G.21 – Subcobertura de um Conjunto

Se a classe $\wp$ já está cobrindo o conjunto A e se a subclasse $\aleph \subset \wp$ também cobre A então $\aleph$ é uma subcobertura de A.

Definição G.22 – Compacto

Um conjunto A é dito ser compacto se toda cobertura aberta de A contiver uma subcobertura finita. Serguei Levovich Sobolev (“Applications of Functional Analysis in Mathematical Physics”. American Math. Society, 1963, pg. 28): “um conjunto A é chamado compacto se para cada subconjunto infinitesimal podemos encontrar uma seqüência convergente”.

Teorema G.2 (**Heine-Borel**): um conjunto $A \subset R$ é compacto se e somente se ele for fechado e limitado.

Definição G.23 – Topologia

Seja A um conjunto não vazio. Uma classe $\wp$ dos subconjuntos de A é dita ser uma Topologia em A se e somente se:

i) A e Ø pertencem a $\wp$,

ii) A união de qualquer número de membros de $\wp$ pertence a $\wp$,

iii) A interseção de quaisquer membros de $\wp$ pertence a $\wp$.

Definição G.24 – Espaço Topológico

Seja A um conjunto não vazio e Γ uma topologia em A. Então o par (A,Γ) é um espaço topológico.

Definição G.25 – Denso

Um conjunto $B \subset A$ é dito ser Denso em A se $\overline{B} \supset A$ (ver Figura H.3). Se $\overline{B} = A$ então B é dito ser Denso por toda parte.

No caso de um operador linear L dizemos que D(L) (domínio de L) é denso se o operador (**Balakrishnan, 1976**, pág.64),

$$\|\mathrm{L}\| = \sup_{\mathrm{x}\in \mathrm{D(L)}} \frac{\|\mathrm{Lx}\|}{\|\mathrm{x}\|} < \infty \quad , \mathrm{x} \neq 0$$

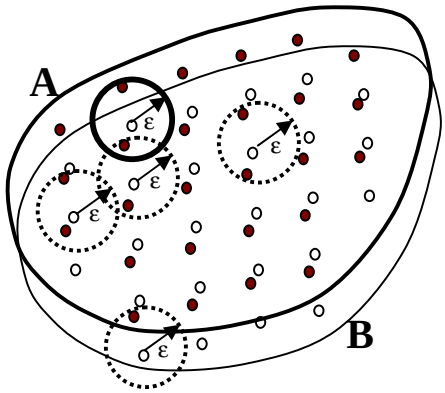


Figura G.4 **– I**nterpretação geométrica de conjunto denso.

Definição G.26 – Seqüência de Cauchy

Uma seqüência $\{a_n\}$ é dita ser uma seqüência de Cauchy se $\lim_{m,n\to\infty} (a_m - a_n) = 0$, isto é, tal seqüência é convergente.

Definição G.27 – Produto Interno

Seja X um espaço linear complexo. Um produto interno sobre X é um mapeamento (função) que associa cada par-ordenado de vetores x e y a um escalar, denotado (x,y), que satisfaz as seguintes propriedades:

(i) Associatividade: (x + y,z) = (x,y) + (y,z)

(ii) Homogeneidade: (a x, y) = a (x,y) ; $\forall a \in C$

(iii) Simetria: (x,y) = $\overline{(x, y)}$; a barra denota conjugado complexo

(iv) "*Positive Definiteness*" (Posividade): $(x, x) > 0$, quando $x \neq 0$

Definição G.28 – Conjunto Completo

Um conjunto A é completo se e somente se toda seqüência de Cauchy de pontos em A convergir para um ponto em A.

Definição G.29 – Norma

Uma função real $\|x\|$ definida sobre um espaço linear X, onde $x \in X$ é dita ser uma norma em X se são satisfeitas as seguintes condições,

(i) Positividade: $\|x\|$ >0,

(ii) Desigualdade triangular: $\|x + y\| \leq \|x\| + \|y\|$,

(iii) Homogeneidade: $\|\alpha x\| = |\alpha| \|x\|$, para $\alpha \in C$,

(iv) "*Positive Definiteness*": $\|x\|$ =0 se e somente se x=0.

Definição G.30 – Espaço Linear Normado

Um espaço linear normado é um par $(X, \|\cdot\|)$ onde X é um espaço linear e $\|\cdot\|$ é uma norma definida sobre X.

Definição G.31 – Espaço de Banach

É todo espaço linear normado completo. Espaço linear normado é todo espaço que admite o cálculo de normas $\|\cdot\|$.

Definição G.32 – Espaço de Produtos Internos

Um espaço de produtos internos é definido como um espaço linear X junto com um produto interno definido em X.

Definição G.33 – Espaço de Hilbert

Um espaço de Hilbert é um espaço de produtos internos completo.

Outra forma menos precisa de definir um espaço de Hilbert é dizer que se trata de um espaço vetorial com produto escalar $(\underline{u}, \underline{v})$ e norma.

Definição G.34 – Espaço de Sobolev

É o espaço das equações diferenciais. Para $d \geq 1$, Ω um subconjunto de R^d $p \in [1, \infty)$ e $s \in N$, o espaço de Sobolev $W^{s,p}(\Omega)$ é definido por,

$$W^{s,p}(\Omega) = \left\{ f \in L^p(\Omega) : \forall |\alpha| \leq s, \partial_x^\alpha f \in L^p(\Omega) \right\} \quad \text{(G.3)}$$

Definição H.35 – Raio Espectral

O raio espectral de uma matriz A com autovalores λ_i é definido por:

$$\rho(A) = \max_i |\lambda_i| \quad \text{(G.4)}$$

Numa simulação numérica (**Craig, 1981**) o raio espectral assume um importante papel na medida da performance numérica de um sistema de simulação. Se r<1 o haverá estabilidade e se r > 1 teremos a instabilidade numérica. (ver Figura G.5).

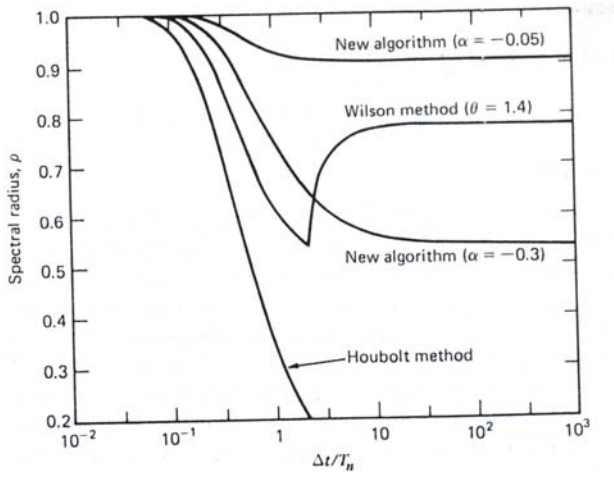


FIGURA G.5 - Raio espectral em função do período de amostragem numa simulação numérica. Para a notação adotada neste trabalho troca-se o ρ da figura por r.
FONTE: Craig (1981, p.460)

Definição G.36 – Conjunto Finito

Um conjunto A é finito se e somente se existe um $n \in N$, onde N é o conjunto dos números naturais, tal que A ~ {1 , 2 , 3 , ... , n}. O símbolo "~" se refere à semelhança (vide definição neste apêndice).

Nota: N = {1 , 2 , 3 , ...} e N_0 = {0 , 1 , 2 , 3 , ...}.

Definição G.37 – Conjunto Infinito

Um conjunto A é infinito se ele não é finito.

Definição G.38 – Conjunto Enumerável ("*denumerable set*")

Um conjunto A é enumerável se e somente se A ~ N.

Definição G.39 – Conjunto Contável

Um conjunto A é contável se e somente se A é finito ou enumerável.

APÊNDICE H

PROGRAMAS EM MATLAB

H.1. SCRIPT MATLAB PARA ESCANEAR DIAGRAMAS DE BODE IDENTIFICANDO COINCIDÊNCIA ENTRE OS CRUZAMENTOS DE MÓDULO E FASE (“GAIN GROSSOVER” E “PHASE CROSSOVER”) DE DIAGRAMAS DE BODE

```
%*********************************************************************************************
%*********************************************************************************************
% TESE DE DOUTORADO NO INPE/DMC (CURSO ETE) – MARCELO
% RICARDO ALVES DA COSTA TREDINNICK
% ORIENTADOR: DR. MARCELO LOPES DE OLIVEIRA E SOUZA
%*********************************************************************************************
%*********************************************************************************************
%    Escaneamento do diagrama de Bode buscando encontrar a coincidencia de
%                                  |H*|=1 e FASE(H*)=180 graus
%*********************************************************************************************
%*********************************************************************************************
% INICIALIZAÇAO DE CONSTANTES ITERATIVAS PARA A GRAVAÇAO DE
% VALORES DE H*(s)
%*********************************************************************************************
clc
clear all
a=1;        %constante existente na planta de segunda ordem G(s)=a/[s(s+a)]
kp = 1;     %ganho do controle Proporcional
kd = 1;      %ganho do controle Derivativo
ki = 1;      %ganho do controle Integral
klead = 1;  %ganho do controle REDE LEAD
klag = 1;   %ganho do controle REDE LAG
%NOTA: o ganho do controle REDE LEAD*LAG sera o produto klead*klag
csi = 0.3;  %parametro da regra ST1
```

```
csi1 = 0.2; %parametro da regra ST2 (zero)
csi2 = 0.3; %parametro da regra ST2 (polo)
z1 = 0.2;    %zero da rede LEAD
p1 = 1;      %polo da rede LEAD
z2 = 1;      %zero da rede LAG
p2 = 0.1;    %polo da rede LAG

%**************************************
% PLANTA DE PRIMEIRA ORDEM:
%**************************************

k1=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Proporcional
k2=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Derivativo Mapeado por Forward
k3=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Derivativo Mapeado por Backward
k4=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Derivativo Mapeado por Tustin
k5=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Derivativo Mapeado por ST1
k6=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Derivativo Mapeado por ST2
k7=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Derivativo Mapeado por Schneider
k8=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Integral Mapeado por Forward
k9=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Integral Mapeado por Backward
k10=0;     %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %Integral Mapeado por Tustin
```

```
k11=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %Integral Mapeado por ST1
k12=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %Integral Mapeado por ST2
k13=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %Integral Mapeado por Schneider
k14=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PD Mapeado por Forward
k15=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PD Mapeado por Backward
k16=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PD Mapeado por Tustin
k17=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PD Mapeado por ST1
k18=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PD Mapeado por ST2
k19=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PD Mapeado por Schneider
k20=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PID Mapeado por Forward
k21=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PID Mapeado por Backward
k22=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PID Mapeado por Tustin
k23=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PID Mapeado por ST1
k24=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PID Mapeado por ST2
k25=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
          %PID Mapeado por Schneider
k26=0;    %constante iterativa para o caso da Planta de 1a.Ordem com controle
```

```
        %REDE "LEAD" Mapeado por Forward
k27=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD" Mapeado por Backward
k28=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD" Mapeado por Tustin
k29=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD" Mapeado por ST1
k30=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD" Mapeado por ST2
k31=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD" Mapeado por Schneider

k32=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LAG" Mapeado por Forward
k33=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LAG" Mapeado por Backward
k34=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LAG" Mapeado por Tustin
k35=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LAG" Mapeado por ST1
k36=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LAG" Mapeado por ST2
k37=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LAG" Mapeado por Schneider
k38=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD"+"LAG" Mapeado por Forward
k39=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD"+"LAG" Mapeado por Backward
k40=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
        %REDE "LEAD"+"LAG" Mapeado por Tustin
k41=0;  %constante iterativa para o caso da Planta de 1a.Ordem com controle
```

```
            %REDE "LEAD"+"LAG" Mapeado por ST1
k42=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
            %REDE "LEAD"+"LAG" Mapeado por ST2
k43=0;      %constante iterativa para o caso da Planta de 1a.Ordem com controle
           %REDE "LEAD"+"LAG" Mapeado por Schneider

%*************************************
%PLANTA DE SEGUNDA ORDEM:
%*************************************

k44=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Proporcional
k45=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Derivativo Mapeado por Forward
k46=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Derivativo Mapeado por Backward
k47=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Derivativo Mapeado por Tustin
k48=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Derivativo Mapeado por ST1
k49=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Derivativo Mapeado por ST2
k50=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Derivativo Mapeado por Schneider
k51=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Integral Mapeado por Forward
k52=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Integral Mapeado por Backward
k53=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %Integral Mapeado por Tustin
k54=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
```

```
                %Integral Mapeado por ST1
k55=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %Integral Mapeado por ST2
k56=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %Integral Mapeado por Schneider
k57=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PD Mapeado por Forward
k58=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PD Mapeado por Backward
k59=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PD Mapeado por Tustin
k60=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PD Mapeado por ST1
k61=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PD Mapeado por ST2
k62=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PD Mapeado por Schneider
k63=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PID Mapeado por Forward
k64=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PID Mapeado por Backward
k65=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PID Mapeado por Tustin
k66=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PID Mapeado por ST1
k67=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PID Mapeado por ST2
k68=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %PID Mapeado por Schneider
k69=0;          %constante iterativa para o caso da Planta de 2a.Ordem com controle
                %REDE "LEAD" Mapeado por Forward
```

```
k70=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD" Mapeado por Backward
k71=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD" Mapeado por Tustin
k72=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD" Mapeado por ST1
k73=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD" Mapeado por ST2
k74=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD" Mapeado por Schneider
k75=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LAG" Mapeado por Forward
k76=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LAG" Mapeado por Backward
k77=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LAG" Mapeado por Tustin
k78=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LAG" Mapeado por ST1
k79=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LAG" Mapeado por ST2
k80=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LAG" Mapeado por Schneider
k81=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD"+"LAG" Mapeado por Forward
k82=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD"+"LAG" Mapeado por Backward
k83=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD"+"LAG" Mapeado por Tustin
k84=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
          %REDE "LEAD"+"LAG" Mapeado por ST1
k85=0;    %constante iterativa para o caso da Planta de 2a.Ordem com controle
```

```
            %REDE "LEAD"+"LAG" Mapeado por ST2
k86=0;      %constante iterativa para o caso da Planta de 2a.Ordem com controle
            %REDE "LEAD"+"LAG" Mapeado por Schneider

%*************************************************************************
%              ESCANEAMENTO DE DIAGRAMAS DE BODE
% Para cada valor de T (segundos) varre uma extensa faixa de w (rad/s)
%*************************************************************************

for T=.05:0.01:7;     %varredura do sampling period
      disp(' ')
      disp(' ')
disp('######################################################################## ')
disp(['      SAMPLING PERIOD  T = ' num2str(T) ' seg'])
disp('######################################################################## ')

      for w=1e-3:.001:(2*pi/T) %faixa de frequencia a ser varrida em rad/seg
                               % (somente de 0 a 2*pi radianos = "determinaçao
                               %principal")
      w  %fornece na tela o valor de w

      %************************************************************************
      % PLANTA DE PRIMEIRA ORDEM JA COM ZERO ORDER HOLD:
      % (Franklin (1981), equ. 4.39)
      % G(s) = a/(s+a)
      % (1-exp(-sT)).(G(s)/s)* = 0.5/(exp(sT)-0.5)
      % convençao usada no livro Franklin (1981): exp(-aT)=0.5
      %************************************************************************

            holdGfirst =  0.5 ./ [exp(i*w*T)-0.5];
```

```
%************************************************************************
% PLANTA DE SEGUNDA ORDEM: (Franklin (1981), TABELA B2-EQ.13)
% G(s)=a/[s(s+a)]
% (G/s)* = (a/[s^2(s+a)])* = {exp(sT).[(aT-1+exp(-aT)).exp(sT) + 1-exp(-aT)-
% aTexp(-aT)]} / {a.(exp(sT)-1)^2.(exp(sT)-exp(-aT)}
% Com o HOLD de ordem zero:
% (1-exp(-sT).(G/s)*
%************************************************************************
        Gsecond = exp(i*w*T).*[(a*T-1+exp(-a*T)).*exp(i*w*T) + ( 1-exp(-a*T)-
a*T*exp(-a*T) )] ./ ( a*((exp(i*w*T) -1).^2) .*(exp(i*w*T)-exp(-a*T) ));
        holdGsecond = (1-exp(-i*w*T)) .* Gsecond;

%************************************************************************
%       CONTROLES
%************************************************************************

        %************************************************************************
        % CONTROLE PROPORCIONAL
        %************************************************************************
            Dproporcional =  kp;

        %************************************************************************
        % CONTROLE DERIVATIVO (mapeamento Forward)
        %************************************************************************
            Dderivativoforward = kd * (1/T) * (exp(i*w*T) - 1);
        %************************************************************************
        % CONTROLE DERIVATIVO (mapeamento Backward)
        %************************************************************************
            Dderivativobackward = kd * (exp(i*w*T) - 1)./(T*exp(i*w*T));
        %************************************************************************
        % CONTROLE DERIVATIVO (mapeamento Tustin)
```

```
%*************************************************************************
        Dderivativotustin = kd * (2/T) * (exp(i*w*T) - 1)./(exp(i*w*T) + 1);
%*************************************************************************
% CONTROLE DERIVATIVO (mapeamento ST1)
%*************************************************************************
        Dderivativost1 = kd * (2/T) * (exp(i*w*T) - 1)./(exp(i*w*T) + csi);
%*************************************************************************
% CONTROLE DERIVATIVO (mapeamento ST2)
%*************************************************************************
        Dderivativost2 = kd * (2/T) * (exp(i*w*T) - csi1)./(exp(i*w*T) +
csi2);
%*************************************************************************
% CONTROLE DERIVATIVO (mapeamento Schneider)
%*************************************************************************
        Dderivativoschneider = kd * (12/T) * (exp(2*i*w*T) -
exp(i*w*T))./(5*exp(2*i*w*T) + 8*exp(i*w*T) -1);

%*************************************************************************
% CONTROLE INTEGRAL (mapeamento Forward)
%*************************************************************************
        Dintegralforward = ki * (T) ./ (exp(i*w*T) - 1);
%*************************************************************************
% CONTROLE INTEGRAL (mapeamento Backward)
%*************************************************************************
        Dintegralbackward = ki * (T*exp(i*w*T)) ./ (exp(i*w*T) - 1);
%*************************************************************************
% CONTROLE INTEGRAL (mapeamento Tustin)
%*************************************************************************
        Dintegraltustin = ki * (T/2) * (exp(i*w*T) + 1) ./ (exp(i*w*T) - 1);
%*************************************************************************
% CONTROLE INTEGRAL (mapeamento ST1)
```

```
%*************************************************************************
        Dintegralst1 = ki * (T/2) * (exp(i*w*T) + csi) ./ (exp(i*w*T) - 1);
%*************************************************************************
% CONTROLE INTEGRAL (mapeamento ST2)
%*************************************************************************
        Dintegralst2 = ki * (T/2) * (exp(i*w*T) + csi2) ./ (exp(i*w*T) -
csi1);
%*************************************************************************
% CONTROLE INTEGRAL (mapeamento Schneider)
%*************************************************************************
        Dintegralschneider = ki * (T/12) * (5*exp(2*i*w*T) +
8*exp(i*w*T) -1) ./ (exp(2*i*w*T) - exp(i*w*T));


%*************************************************************************
% CONTROLE PD (mapeamento Forward)
%*************************************************************************
        Dpdforward = Dproporcional + Dderivativoforward;
%*************************************************************************
% CONTROLE PD (mapeamento Backward)
%*************************************************************************
        Dpdbackward = Dproporcional + Dderivativobackward;
%*************************************************************************
% CONTROLE PD (mapeamento Tustin)
%*************************************************************************
        Dpdtustin = Dproporcional + Dderivativotustin;
%*************************************************************************
% CONTROLE PD (mapeamento ST1)
%*************************************************************************
        Dpdst1 = Dproporcional + Dderivativost1;
%*************************************************************************
```

```
% CONTROLE PD (mapeamento ST2)
%************************************************************************
        Dpdst2 = Dproporcional + Dderivativost2;
%************************************************************************
% CONTROLE PD (mapeamento Schneider)
%************************************************************************
        Dpdschneider = Dproporcional + Dderivativoschneider;

%************************************************************************
% CONTROLE PID (mapeamento Forward)
%************************************************************************
        Dpidforward = Dproporcional + Dderivativoforward +
Dintegralforward;
%************************************************************************
% CONTROLE PID (mapeamento Backward)
%************************************************************************
        Dpidbackward = Dproporcional + Dderivativobackward +
Dintegralbackward;
%************************************************************************
% CONTROLE PID (mapeamento Tustin)
%************************************************************************
        Dpidtustin = Dproporcional + Dderivativotustin + Dintegraltustin;
%************************************************************************
% CONTROLE PID (mapeamento ST1)
%************************************************************************
        Dpidst1 = Dproporcional + Dderivativost1 + Dintegralst1;
%************************************************************************
% CONTROLE PID (mapeamento ST2)
%************************************************************************
        Dpidst2 = Dproporcional + Dderivativost2 + Dintegralst2;
%************************************************************************
```

```
% CONTROLE PID (mapeamento Schneider)
%*************************************************************************
    Dpidschneider = Dproporcional + Dderivativoschneider +
Dintegralschneider;
%*************************************************************************
% CONTROLE REDE LEAD (mapeamento Forward)
%*************************************************************************
    Dleadforward = klead * (exp(i*w*T) + T*z1 - 1) ./ (exp(i*w*T) +
T*p1 - 1);
%*************************************************************************
% CONTROLE REDE LEAD (mapeamento Backward)
%*************************************************************************
    Dleadbackward = klead * [(1+T*z1)*exp(i*w*T)-1] ./
[(1+T*p1)*exp(i*w*T)-1];
%*************************************************************************
% CONTROLE REDE LEAD (mapeamento Tustin)
%*************************************************************************
    Dleadtustin = klead * [(2+T*z1)*exp(i*w*T)+T*z1-2] ./
[(2+T*p1)*exp(i*w*T)+T*p1-2];
%*************************************************************************
% CONTROLE REDE LEAD (mapeamento ST1)
%*************************************************************************
    Dleadst1 = klead * [(2+T*z1)*exp(i*w*T)+T*z1*csi-2] ./
[(2+T*p1)*exp(i*w*T)+T*p1*csi-2];
%*************************************************************************
% CONTROLE REDE LEAD (mapeamento ST2)
%*************************************************************************
    Dleadst2 = klead * [(2+T*z1)*exp(i*w*T)+T*z1*csi2-2*csi1] ./
[(2+T*p1)*exp(i*w*T)+T*p1*csi2-2*csi1];
%*************************************************************************
% CONTROLE REDE LEAD (mapeamento Schneider)
```

```
%**************************************************************************
        Dleadschneider = klead * [(1 + 5*z1*T/12)*exp(2*i*w*T) +
(2*z1*T/3 -1)*exp(i*w*T) -z1*T/12] ./ [(1 + 5*p1*T/12)*exp(2*i*w*T) + (2*p1*T/3 -
1)*exp(i*w*T) -p1*T/12];
%**************************************************************************
% CONTROLE REDE LAG (mapeamento Forward)
%**************************************************************************
        Dlagforward = klag * (exp(i*w*T) + T*z2 - 1) ./ (exp(i*w*T) +
T*p2 - 1);
%**************************************************************************
% CONTROLE REDE LAG (mapeamento Backward)
%**************************************************************************
        Dlagbackward = klag * [(1+T*z2)*exp(i*w*T)-1] ./
[(1+T*p2)*exp(i*w*T)-1];
%**************************************************************************
% CONTROLE REDE LAG (mapeamento Tustin)
%**************************************************************************
        Dlagtustin = klag * [(2+T*z2)*exp(i*w*T)+T*z2-2] ./
[(2+T*p2)*exp(i*w*T)+T*p2-2];
%**************************************************************************
% CONTROLE REDE LAG (mapeamento ST1)
%**************************************************************************
        Dlagst1 = klag * [(2+T*z2)*exp(i*w*T)+T*z2*csi-2] ./
[(2+T*p2)*exp(i*w*T)+T*p2*csi-2];
%**************************************************************************
% CONTROLE REDE LAG (mapeamento ST2)
%**************************************************************************
        Dlagst2 = klag * [(2+T*z2)*exp(i*w*T)+T*z2*csi2-2*csi1] ./
[(2+T*p2)*exp(i*w*T)+T*p2*csi2-2*csi1];
%**************************************************************************
% CONTROLE REDE LAG (mapeamento Schneider)
```

```
%**************************************************************************
        Dlagschneider = klag * [(1 + 5*z2*T/12)*exp(2*i*w*T) +
(2*z2*T/3 -1)*exp(i*w*T) -z2*T/12] ./ [(1 + 5*p2*T/12)*exp(2*i*w*T) + (2*p2*T/3 -
1)*exp(i*w*T) -p2*T/12];
%**************************************************************************
% CONTROLE REDE LEAD*LAG (mapeamento Forward); o ganho
% da LEAD*LAG e igual ao produto dos ganhos da LEAD e
% da LAG
%**************************************************************************
        Dleadlagforward = Dleadforward .* Dlagforward;
%**************************************************************************
% CONTROLE REDE LEAD*LAG (mapeamento Backward)
%**************************************************************************
        Dleadlagbackward = Dleadbackward .* Dlagbackward;
%**************************************************************************
% CONTROLE REDE LEAD*LAG (mapeamento Tustin)
%**************************************************************************
        Dleadlagtustin = Dleadtustin .* Dlagtustin;
%**************************************************************************
% CONTROLE REDE LEAD*LAG (mapeamento ST1)
%**************************************************************************
        Dleadlagst1 = Dleadst1 .* Dlagst1;
%**************************************************************************
% CONTROLE REDE LEAD*LAG (mapeamento ST2)
%**************************************************************************
        Dleadlagst2 = Dleadst2 .* Dlagst2;
%**************************************************************************
% CONTROLE REDE LEAD*LAG (mapeamento Schneider)
%**************************************************************************
        Dleadlagschneider = Dleadschneider .* Dlagschneider;
```

```
%************************************************************************
%************************************************************************
% FUNÇAO DE TRANSFERENCIA DE MALHA-ABERTA H*(S) = (ZERO
% ORDER HOLD).D*.(G/s)*
% H*(S) = (1-exp(-sT)) . D* . (G/s)*
% Ja calculei o HOLD junto com (G/s)* nas linhas acima desse script.
%************************************************************************
%************************************************************************

%*********************************************
%*********************************************
%
%       PLANTA DE PRIMEIRA ORDEM
%
%*********************************************
%*********************************************

        %*****************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle Proporcional
        %*****************************************************************************
                Hfirstproporcional = holdGfirst .* Dproporcional;
                MagDBfirstproporcional=20*log10(abs(Hfirstproporcional));
                Phaseradfirstproporcional=angle(Hfirstproporcional);
                Phasefirstproporcional=Phaseradfirstproporcional*180/pi;

        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
        % (Forward)
        %************************************************************************
                Hfirstderivativoforward = holdGfirst .* Dderivativoforward;
```

```
MagDBfirstderivativoforward=20*log10(abs(Hfirstderivativoforward));
        Phaseradfirstderivativoforward=angle(Hfirstderivativoforward);

Phasefirstderivativoforward=Phaseradfirstderivativoforward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% (Backward)
%************************************************************************
        Hfirstderivativobackward = holdGfirst .* Dderivativobackward;

MagDBfirstderivativobackward=20*log10(abs(Hfirstderivativobackward));

Phaseradfirstderivativobackward=angle(Hfirstderivativobackward);

Phasefirstderivativobackward=Phaseradfirstderivativobackward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% (Tustin)
%************************************************************************
        Hfirstderivativotustin = holdGfirst .* Dderivativotustin;

MagDBfirstderivativotustin=20*log10(abs(Hfirstderivativotustin));
        Phaseradfirstderivativotustin=angle(Hfirstderivativotustin);
        Phasefirstderivativotustin=Phaseradfirstderivativotustin*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% (ST1)
%************************************************************************
        Hfirstderivativost1 = holdGfirst .* Dderivativost1;
        MagDBfirstderivativost1=20*log10(abs(Hfirstderivativost1));
```

```
        Phaseradfirstderivativost1=angle(Hfirstderivativost1);
        Phasefirstderivativost1=Phaseradfirstderivativost1*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% (ST2)
%************************************************************************
        Hfirstderivativost2 = holdGfirst .* Dderivativost2;
        MagDBfirstderivativost2=20*log10(abs(Hfirstderivativost2));
        Phaseradfirstderivativost2=angle(Hfirstderivativost2);
        Phasefirstderivativost2=Phaseradfirstderivativost2*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% (Schneider)
%************************************************************************
        Hfirstderivativoschneider = holdGfirst .* Dderivativoschneider;
MagDBfirstderivativoschneider=20*log10(abs(Hfirstderivativoschneider));
    Phaseradfirstderivativoschneider=angle(Hfirstderivativoschneider);
Phasefirstderivativoschneider=Phaseradfirstderivativoschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% (Forward)
%************************************************************************
        Hfirstintegralforward = holdGfirst .* Dintegralforward;
       MagDBfirstintegralforward=20*log10(abs(Hfirstintegralforward));
        Phaseradfirstintegralforward=angle(Hfirstintegralforward);
        Phasefirstintegralforward=Phaseradfirstintegralforward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% (Backward)
%************************************************************************
```

```
        Hfirstintegralbackward = holdGfirst .* Dintegralbackward;
   MagDBfirstintegralbackward=20*log10(abs(Hfirstintegralbackward));
        Phaseradfirstintegralbackward=angle(Hfirstintegralbackward);
     Phasefirstintegralbackward=Phaseradfirstintegralbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% (Tustin)
%************************************************************************
        Hfirstintegraltustin = holdGfirst .* Dintegraltustin;
        MagDBfirstintegraltustin=20*log10(abs(Hfirstintegraltustin));
        Phaseradfirstintegraltustin=angle(Hfirstintegraltustin);
        Phasefirstintegraltustin=Phaseradfirstintegraltustin*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% (ST1)
%************************************************************************
        Hfirstintegralst1 = holdGfirst .* Dintegralst1;
        MagDBfirstintegralst1=20*log10(abs(Hfirstintegralst1));
        Phaseradfirstintegralst1=angle(Hfirstintegralst1);
        Phasefirstintegralst1=Phaseradfirstintegralst1*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% (ST2)
%************************************************************************
        Hfirstintegralst2 = holdGfirst .* Dintegralst2;
        MagDBfirstintegralst2=20*log10(abs(Hfirstintegralst2));
        Phaseradfirstintegralst2=angle(Hfirstintegralst2);
        Phasefirstintegralst2=Phaseradfirstintegralst2*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% (Schneider)
```

```
%************************************************************************
        Hfirstintegralschneider = holdGfirst .* Dintegralschneider;
  MagDBfirstintegralschneider=20*log10(abs(Hfirstintegralschneider));
        Phaseradfirstintegralschneider=angle(Hfirstintegralschneider);

    Phasefirstintegralschneider=Phaseradfirstintegralschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD (Forward)
%************************************************************************
        Hfirstpdforward = holdGfirst .* Dpdforward;
        MagDBfirstpdforward=20*log10(abs(Hfirstpdforward));
        Phaseradfirstpdforward=angle(Hfirstpdforward);
        Phasefirstpdforward=Phaseradfirstpdforward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD
% (Backward)
%************************************************************************
        Hfirstpdbackward = holdGfirst .* Dpdbackward;
        MagDBfirstpdbackward=20*log10(abs(Hfirstpdbackward));
        Phaseradfirstpdbackward=angle(Hfirstpdbackward);
        Phasefirstpdbackward=Phaseradfirstpdbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD (Tustin)
%************************************************************************
        Hfirstpdtustin = holdGfirst .* Dpdtustin;
        MagDBfirstpdtustin=20*log10(abs(Hfirstpdtustin));
        Phaseradfirstpdtustin=angle(Hfirstpdtustin);
        Phasefirstpdtustin=Phaseradfirstpdtustin*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD (ST1)
```

```
%*************************************************************************
        Hfirstpdst1 = holdGfirst .* Dpdst1;
        MagDBfirstpdst1=20*log10(abs(Hfirstpdst1));
        Phaseradfirstpdst1=angle(Hfirstpdst1);
        Phasefirstpdst1=Phaseradfirstpdst1*180/pi;
%*************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD (ST2)
%*************************************************************************
        Hfirstpdst2 = holdGfirst .* Dpdst2;
        MagDBfirstpdst2=20*log10(abs(Hfirstpdst2));
        Phaseradfirstpdst2=angle(Hfirstpdst2);
        Phasefirstpdst2=Phaseradfirstpdst2*180/pi;
%*************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD
% (Schneider)
%*************************************************************************
        Hfirstpdschneider = holdGfirst .* Dpdschneider;
        MagDBfirstpdschneider=20*log10(abs(Hfirstpdschneider));
        Phaseradfirstpdschneider=angle(Hfirstpdschneider);
        Phasefirstpdschneider=Phaseradfirstpdschneider*180/pi;


%*************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PID
% (Forward)
%*************************************************************************
        Hfirstpidforward = holdGfirst .* Dpidforward;
        MagDBfirstpidforward=20*log10(abs(Hfirstpidforward));
        Phaseradfirstpidforward=angle(Hfirstpidforward);
        Phasefirstpidforward=Phaseradfirstpidforward*180/pi;
%*************************************************************************
```

```
% H*(s) para Planta de 1a. Ordem (com hold) e controle PID
% (Backward)
%************************************************************************
        Hfirstpidbackward = holdGfirst .* Dpidbackward;
        MagDBfirstpidbackward=20*log10(abs(Hfirstpidbackward));
        Phaseradfirstpidbackward=angle(Hfirstpidbackward);
        Phasefirstpidbackward=Phaseradfirstpidbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PID (Tustin)
%************************************************************************
        Hfirstpidtustin = holdGfirst .* Dpidtustin;
        MagDBfirstpidtustin=20*log10(abs(Hfirstpidtustin));
        Phaseradfirstpidtustin=angle(Hfirstpidtustin);
        Phasefirstpidtustin=Phaseradfirstpidtustin*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PID (ST1)
%************************************************************************
        Hfirstpidst1 = holdGfirst .* Dpidst1;
        MagDBfirstpidst1=20*log10(abs(Hfirstpidst1));
        Phaseradfirstpidst1=angle(Hfirstpidst1);
        Phasefirstpidst1=Phaseradfirstpidst1*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PID (ST2)
%************************************************************************
        Hfirstpidst2 = holdGfirst .* Dpidst2;
        MagDBfirstpidst2=20*log10(abs(Hfirstpidst2));
        Phaseradfirstpidst2=angle(Hfirstpidst2);
        Phasefirstpidst2=Phaseradfirstpidst2*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PID
% (Schneider)
```

```
%************************************************************************
        Hfirstpidschneider = holdGfirst .* Dpidschneider;
        MagDBfirstpidschneider=20*log10(abs(Hfirstpidschneider));
        Phaseradfirstpidschneider=angle(Hfirstpidschneider);
        Phasefirstpidschneider=Phaseradfirstpidschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LEAD
% (Forward)
%************************************************************************
        Hfirstleadforward = holdGfirst .* Dleadforward;
        MagDBfirstleadforward=20*log10(abs(Hfirstleadforward));
        Phaseradfirstleadforward=angle(Hfirstleadforward);
        Phasefirstleadforward=Phaseradfirstleadforward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LEAD
% (Backward)
%************************************************************************
        Hfirstleadbackward = holdGfirst .* Dleadbackward;
        MagDBfirstleadbackward=20*log10(abs(Hfirstleadbackward));
        Phaseradfirstleadbackward=angle(Hfirstleadbackward);
        Phasefirstleadbackward=Phaseradfirstleadbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LEAD
% (Tustin)
%************************************************************************
        Hfirstleadtustin = holdGfirst .* Dleadtustin;
        MagDBfirstleadtustin=20*log10(abs(Hfirstleadtustin));
        Phaseradfirstleadtustin=angle(Hfirstleadtustin);
        Phasefirstleadtustin=Phaseradfirstleadtustin*180/pi;
%************************************************************************
```

```
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LEAD
% (ST1)
%************************************************************************
        Hfirstleadst1 = holdGfirst .* Dleadst1;
        MagDBfirstleadst1=20*log10(abs(Hfirstleadst1));
        Phaseradfirstleadst1=angle(Hfirstleadst1);
        Phasefirstleadst1=Phaseradfirstleadst1*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LEAD
% (ST2)
%************************************************************************
        Hfirstleadst2 = holdGfirst .* Dleadst2;
        MagDBfirstleadst2=20*log10(abs(Hfirstleadst2));
        Phaseradfirstleadst2=angle(Hfirstleadst2);
        Phasefirstleadst2=Phaseradfirstleadst2*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LEAD
% (Schneider)
%************************************************************************
        Hfirstleadschneider = holdGfirst .* Dleadschneider;
        MagDBfirstleadschneider=20*log10(abs(Hfirstleadschneider));
        Phaseradfirstleadschneider=angle(Hfirstleadschneider);
        Phasefirstleadschneider=Phaseradfirstleadschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LAG
% (Forward)
%************************************************************************
        Hfirstlagforward = holdGfirst .* Dlagforward;
        MagDBfirstlagforward=20*log10(abs(Hfirstlagforward));
        Phaseradfirstlagforward=angle(Hfirstlagforward);
```

```
        Phasefirstlagforward=Phaseradfirstlagforward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LAG
% (Backward)
%************************************************************************
        Hfirstlagbackward = holdGfirst .* Dlagbackward;
        MagDBfirstlagbackward=20*log10(abs(Hfirstlagbackward));
        Phaseradfirstlagbackward=angle(Hfirstlagbackward);
        Phasefirstlagbackward=Phaseradfirstlagbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LAG
% (Tustin)
%************************************************************************
        Hfirstlagtustin = holdGfirst .* Dlagtustin;
        MagDBfirstlagtustin=20*log10(abs(Hfirstlagtustin));
        Phaseradfirstlagtustin=angle(Hfirstlagtustin);
        Phasefirstlagtustin=Phaseradfirstlagtustin*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LAG
% (ST1)
%************************************************************************
        Hfirstlagst1 = holdGfirst .* Dlagst1;
        MagDBfirstlagst1=20*log10(abs(Hfirstlagst1));
        Phaseradfirstlagst1=angle(Hfirstlagst1);
        Phasefirstlagst1=Phaseradfirstlagst1*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LAG
% (ST2)
%************************************************************************
        Hfirstlagst2 = holdGfirst .* Dlagst2;
        MagDBfirstlagst2=20*log10(abs(Hfirstlagst2));
```

```
        Phaseradfirstlagst2=angle(Hfirstlagst2);
        Phasefirstlagst2=Phaseradfirstlagst2*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE LAG
% (Schneider)
%************************************************************************
        Hfirstlagschneider = holdGfirst .* Dlagschneider;
        MagDBfirstlagschneider=20*log10(abs(Hfirstlagschneider));
        Phaseradfirstlagschneider=angle(Hfirstlagschneider);
        Phasefirstlagschneider=Phaseradfirstlagschneider*180/pi;


%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE
% LEAD*LAG (Forward)
%************************************************************************
        Hfirstleadlagforward = holdGfirst .* Dleadlagforward;
       MagDBfirstleadlagforward=20*log10(abs(Hfirstleadlagforward));
        Phaseradfirstleadlagforward=angle(Hfirstleadlagforward);
        Phasefirstleadlagforward=Phaseradfirstleadlagforward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE
% LEAD*LAG (Backward)
%************************************************************************
        Hfirstleadlagbackward = holdGfirst .* Dleadlagbackward;
   MagDBfirstleadlagbackward=20*log10(abs(Hfirstleadlagbackward));
        Phaseradfirstleadlagbackward=angle(Hfirstleadlagbackward);
     Phasefirstleadlagbackward=Phaseradfirstleadlagbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE
% LEAD*LAG (Tustin)
```

```
%**************************************************************************
        Hfirstleadlagtustin = holdGfirst .* Dleadlagtustin;
        MagDBfirstleadlagtustin=20*log10(abs(Hfirstleadlagtustin));
        Phaseradfirstleadlagtustin=angle(Hfirstleadlagtustin);
        Phasefirstleadlagtustin=Phaseradfirstleadlagtustin*180/pi;
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE
% LEAD*LAG (ST1)
%**************************************************************************
        Hfirstleadlagst1 = holdGfirst .* Dleadlagst1;
        MagDBfirstleadlagst1=20*log10(abs(Hfirstleadlagst1));
        Phaseradfirstleadlagst1=angle(Hfirstleadlagst1);
        Phasefirstleadlagst1=Phaseradfirstleadlagst1*180/pi;
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE
% LEAD*LAG (ST2)
%**************************************************************************
        Hfirstleadlagst2 = holdGfirst .* Dleadlagst2;
        MagDBfirstleadlagst2=20*log10(abs(Hfirstleadlagst2));
        Phaseradfirstleadlagst2=angle(Hfirstleadlagst2);
        Phasefirstleadlagst2=Phaseradfirstleadlagst2*180/pi;
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle REDE
% LEAD*LAG (Schneider)
%**************************************************************************
        Hfirstleadlagschneider = holdGfirst .* Dleadlagschneider;
   MagDBfirstleadlagschneider=20*log10(abs(Hfirstleadlagschneider));
        Phaseradfirstleadlagschneider=angle(Hfirstleadlagschneider);
     Phasefirstleadlagschneider=Phaseradfirstleadlagschneider*180/pi;
```

```
%*********************************************
%*********************************************
%
%       PLANTA DE SEGUNDA ORDEM
%
%*********************************************
%*********************************************

        %*************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle Proporcional
        %*************************************************************************
            Hsecondproporcional = holdGsecond .* Dproporcional;
          MagDBsecondproporcional=20*log10(abs(Hsecondproporcional));
            Phaseradsecondproporcional=angle(Hsecondproporcional);
           Phasesecondproporcional=Phaseradsecondproporcional*180/pi;

        %*************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
        % (Forward)
        %*************************************************************************
            Hsecondderivativoforward = holdGsecond .*
Dderivativoforward;

    MagDBsecondderivativoforward=20*log10(abs(Hsecondderivativoforward));
        Phaseradsecondderivativoforward=angle(Hsecondderivativoforward);
      Phasesecondderivativoforward=Phaseradsecondderivativoforward*180/pi;
        %*************************************************************************
```

```
% H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
% (Backward)
%************************************************************************
        Hsecondderivativobackward = holdGsecond .*
Dderivativobackward;

MagDBsecondderivativobackward=20*log10(abs(Hsecondderivativobackward));
    Phaseradsecondderivativobackward=angle(Hsecondderivativobackward);
  Phasesecondderivativobackward=Phaseradsecondderivativobackward*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
% (Tustin)
%************************************************************************
        Hsecondderivativotustin = holdGsecond .* Dderivativotustin;

MagDBsecondderivativotustin=20*log10(abs(Hsecondderivativotustin));
    Phaseradsecondderivativotustin=angle(Hsecondderivativotustin);
  Phasesecondderivativotustin=Phaseradsecondderivativotustin*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
% (ST1)
%************************************************************************
        Hsecondderivativost1 = holdGsecond .* Dderivativost1;

    MagDBsecondderivativost1=20*log10(abs(Hsecondderivativost1));
        Phaseradsecondderivativost1=angle(Hsecondderivativost1);
    Phasesecondderivativost1=Phaseradsecondderivativost1*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
% (ST2)
```

```
%************************************************************************
        Hsecondderivativost2 = holdGsecond .* Dderivativost2;

    MagDBsecondderivativost2=20*log10(abs(Hsecondderivativost2));
        Phaseradsecondderivativost2=angle(Hsecondderivativost2);
      Phasesecondderivativost2=Phaseradsecondderivativost2*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
% (Schneider)
%************************************************************************
        Hsecondderivativoschneider = holdGsecond .*
Dderivativoschneider;

 MagDBsecondderivativoschneider=20*log10(abs(Hsecondderivativoschneider));
     Phaseradsecondderivativoschneider=angle(Hsecondderivativoschneider);
  Phasesecondderivativoschneider=Phaseradsecondderivativoschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
% (Forward)
%************************************************************************
        Hsecondintegralforward = holdGsecond .* Dintegralforward;

   MagDBsecondintegralforward=20*log10(abs(Hsecondintegralforward));
        Phaseradsecondintegralforward=angle(Hsecondintegralforward);
    Phasesecondintegralforward=Phaseradsecondintegralforward*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
% (Backward)
%************************************************************************
```

```
        Hsecondintegralbackward = holdGsecond .*
Dintegralbackward;

    MagDBsecondintegralbackward=20*log10(abs(Hsecondintegralbackward));
      Phaseradsecondintegralbackward=angle(Hsecondintegralbackward);
    Phasesecondintegralbackward=Phaseradsecondintegralbackward*180/pi;

    %************************************************************************
    % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
    % (Tustin)
    %************************************************************************
        Hsecondintegraltustin = holdGsecond .* Dintegraltustin;

      MagDBsecondintegraltustin=20*log10(abs(Hsecondintegraltustin));
        Phaseradsecondintegraltustin=angle(Hsecondintegraltustin);
     Phasesecondintegraltustin=Phaseradsecondintegraltustin*180/pi;

    %************************************************************************
    % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
    % (ST1)
    %************************************************************************
        Hsecondintegralst1 = holdGsecond .* Dintegralst1;
        MagDBsecondintegralst1=20*log10(abs(Hsecondintegralst1));
        Phaseradsecondintegralst1=angle(Hsecondintegralst1);
        Phasesecondintegralst1=Phaseradsecondintegralst1*180/pi;
    %************************************************************************
    % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
    % (ST2)
    %************************************************************************
        Hsecondintegralst2 = holdGsecond .* Dintegralst2;
        MagDBsecondintegralst2=20*log10(abs(Hsecondintegralst2));
```

```
        Phaseradsecondintegralst2=angle(Hsecondintegralst2);
        Phasesecondintegralst2=Phaseradsecondintegralst2*180/pi;
    %************************************************************************
    % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
    % (Schneider)
    %************************************************************************
        Hsecondintegralschneider = holdGsecond .*
Dintegralschneider;

MagDBsecondintegralschneider=20*log10(abs(Hsecondintegralschneider));
 Phaseradsecondintegralschneider=angle(Hsecondintegralschneider);
Phasesecondintegralschneider=Phaseradsecondintegralschneider*180/pi;

    %************************************************************************
    % H*(s) para Planta de 2a. Ordem (com hold) e controle PD (Forward)
    %************************************************************************
        Hsecondpdforward = holdGsecond .* Dpdforward;
        MagDBsecondpdforward=20*log10(abs(Hsecondpdforward));
        Phaseradsecondpdforward=angle(Hsecondpdforward);
        Phasesecondpdforward=Phaseradsecondpdforward*180/pi;
    %************************************************************************
    % H*(s) para Planta de 2a. Ordem (com hold) e controle PD
    % (Backward)
    %************************************************************************
        Hsecondpdbackward = holdGsecond .* Dpdbackward;

      MagDBsecondpdbackward=20*log10(abs(Hsecondpdbackward));
        Phaseradsecondpdbackward=angle(Hsecondpdbackward);
       Phasesecondpdbackward=Phaseradsecondpdbackward*180/pi;
    %************************************************************************
    % H*(s) para Planta de 2a. Ordem (com hold) e controle PD (Tustin)
```

```
%************************************************************************
        Hsecondpdtustin = holdGsecond .* Dpdtustin;
        MagDBsecondpdtustin=20*log10(abs(Hsecondpdtustin));
        Phaseradsecondpdtustin=angle(Hsecondpdtustin);
        Phasesecondpdtustin=Phaseradsecondpdtustin*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PD (ST1)
%************************************************************************
        Hsecondpdst1 = holdGsecond .* Dpdst1;
        MagDBsecondpdst1=20*log10(abs(Hsecondpdst1));
        Phaseradsecondpdst1=angle(Hsecondpdst1);
        Phasesecondpdst1=Phaseradsecondpdst1*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PD (ST2)
%************************************************************************
        Hsecondpdst2 = holdGsecond .* Dpdst2;
        MagDBsecondpdst2=20*log10(abs(Hsecondpdst2));
        Phaseradsecondpdst2=angle(Hsecondpdst2);
        Phasesecondpdst2=Phaseradsecondpdst2*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PD
% (Schneider)
%************************************************************************
        Hsecondpdschneider = holdGsecond .* Dpdschneider;
     MagDBsecondpdschneider=20*log10(abs(Hsecondpdschneider));
        Phaseradsecondpdschneider=angle(Hsecondpdschneider);
       Phasesecondpdschneider=Phaseradsecondpdschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID
% (Forward)
```

```
%*************************************************************************
        Hsecondpidforward = holdGsecond .* Dpidforward;
        MagDBsecondpidforward=20*log10(abs(Hsecondpidforward));
        Phaseradsecondpidforward=angle(Hsecondpidforward);
        Phasesecondpidforward=Phaseradsecondpidforward*180/pi;
%*************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID
% (Backward)
%*************************************************************************
        Hsecondpidbackward = holdGsecond .* Dpidbackward;
    MagDBsecondpidbackward=20*log10(abs(Hsecondpidbackward));
        Phaseradsecondpidbackward=angle(Hsecondpidbackward);
      Phasesecondpidbackward=Phaseradsecondpidbackward*180/pi;
%*************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID (Tustin)
%*************************************************************************
        Hsecondpidtustin = holdGsecond .* Dpidtustin;
        MagDBsecondpidtustin=20*log10(abs(Hsecondpidtustin));
        Phaseradsecondpidtustin=angle(Hsecondpidtustin);
        Phasesecondpidtustin=Phaseradsecondpidtustin*180/pi;
%*************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID (ST1)
%*************************************************************************
        Hsecondpidst1 = holdGsecond .* Dpidst1;
        MagDBsecondpidst1=20*log10(abs(Hsecondpidst1));
        Phaseradsecondpidst1=angle(Hsecondpidst1);
        Phasesecondpidst1=Phaseradsecondpidst1*180/pi;
%*************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID (ST2)
%*************************************************************************
        Hsecondpidst2 = holdGsecond .* Dpidst2;
```

```
        MagDBsecondpidst2=20*log10(abs(Hsecondpidst2));
        Phaseradsecondpidst2=angle(Hsecondpidst2);
        Phasesecondpidst2=Phaseradsecondpidst2*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID
% (Schneider)
%************************************************************************
        Hsecondpidschneider = holdGsecond .* Dpidschneider;
    MagDBsecondpidschneider=20*log10(abs(Hsecondpidschneider));
        Phaseradsecondpidschneider=angle(Hsecondpidschneider);
      Phasesecondpidschneider=Phaseradsecondpidschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LEAD
% (Forward)
%************************************************************************
        Hsecondleadforward = holdGsecond .* Dleadforward;
       MagDBsecondleadforward=20*log10(abs(Hsecondleadforward));
        Phaseradsecondleadforward=angle(Hsecondleadforward);
        Phasesecondleadforward=Phaseradsecondleadforward*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LEAD
% (Backward)
%************************************************************************
        Hsecondleadbackward = holdGsecond .* Dleadbackward;
  MagDBsecondleadbackward=20*log10(abs(Hsecondleadbackward));
        Phaseradsecondleadbackward=angle(Hsecondleadbackward);
   Phasesecondleadbackward=Phaseradsecondleadbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LEAD
% (Tustin)
```

```
%************************************************************************
        Hsecondleadtustin = holdGsecond .* Dleadtustin;
        MagDBsecondleadtustin=20*log10(abs(Hsecondleadtustin));
        Phaseradsecondleadtustin=angle(Hsecondleadtustin);
        Phasesecondleadtustin=Phaseradsecondleadtustin*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LEAD
% (ST1)
%************************************************************************
        Hsecondleadst1 = holdGsecond .* Dleadst1;
        MagDBsecondleadst1=20*log10(abs(Hsecondleadst1));
        Phaseradsecondleadst1=angle(Hsecondleadst1);
        Phasesecondleadst1=Phaseradsecondleadst1*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LEAD
% (ST2)
%************************************************************************
        Hsecondleadst2 = holdGsecond .* Dleadst2;
        MagDBsecondleadst2=20*log10(abs(Hsecondleadst2));
        Phaseradsecondleadst2=angle(Hsecondleadst2);
        Phasesecondleadst2=Phaseradsecondleadst2*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LEAD
% (Schneider)
%************************************************************************
        Hsecondleadschneider = holdGsecond .* Dleadschneider;
  MagDBsecondleadschneider=20*log10(abs(Hsecondleadschneider));
        Phaseradsecondleadschneider=angle(Hsecondleadschneider);
    Phasesecondleadschneider=Phaseradsecondleadschneider*180/pi;
```

```
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LAG
% (Forward)
%************************************************************************
        Hsecondlagforward = holdGsecond .* Dlagforward;
        MagDBsecondlagforward=20*log10(abs(Hsecondlagforward));
        Phaseradsecondlagforward=angle(Hsecondlagforward);
        Phasesecondlagforward=Phaseradsecondlagforward*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LAG
% (Backward)
%************************************************************************
        Hsecondlagbackward = holdGsecond .* Dlagbackward;
    MagDBsecondlagbackward=20*log10(abs(Hsecondlagbackward));
        Phaseradsecondlagbackward=angle(Hsecondlagbackward);
      Phasesecondlagbackward=Phaseradsecondlagbackward*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LAG
% (Tustin)
%************************************************************************
        Hsecondlagtustin = holdGsecond .* Dlagtustin;
        MagDBsecondlagtustin=20*log10(abs(Hsecondlagtustin));
        Phaseradsecondlagtustin=angle(Hsecondlagtustin);
        Phasesecondlagtustin=Phaseradsecondlagtustin*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LAG
% (ST1)
%************************************************************************
        Hsecondlagst1 = holdGsecond .* Dlagst1;
```

```
        MagDBsecondlagst1=20*log10(abs(Hsecondlagst1));
        Phaseradsecondlagst1=angle(Hsecondlagst1);
        Phasesecondlagst1=Phaseradsecondlagst1*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LAG
% (ST2)
%************************************************************************
        Hsecondlagst2 = holdGsecond .* Dlagst2;
        MagDBsecondlagst2=20*log10(abs(Hsecondlagst2));
        Phaseradsecondlagst2=angle(Hsecondlagst2);
        Phasesecondlagst2=Phaseradsecondlagst2*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE LAG
% (Schneider)
%************************************************************************
        Hsecondlagschneider = holdGsecond .* Dlagschneider;
    MagDBsecondlagschneider=20*log10(abs(Hsecondlagschneider));
        Phaseradsecondlagschneider=angle(Hsecondlagschneider);
      Phasesecondlagschneider=Phaseradsecondlagschneider*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE
% LEAD*LAG (Forward)
%************************************************************************
        Hsecondleadlagforward = holdGsecond .* Dleadlagforward;
MagDBsecondleadlagforward=20*log10(abs(Hsecondleadlagforward));
       Phaseradsecondleadlagforward=angle(Hsecondleadlagforward);
 Phasesecondleadlagforward=Phaseradsecondleadlagforward*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE
% LEAD*LAG (Backward)
```

```
%************************************************************************
        Hsecondleadlagbackward = holdGsecond .*
Dleadlagbackward;

MagDBsecondleadlagbackward=20*log10(abs(Hsecondleadlagbackward));
  Phaseradsecondleadlagbackward=angle(Hsecondleadlagbackward);
Phasesecondleadlagbackward=Phaseradsecondleadlagbackward*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE
% LEAD*LAG (Tustin)
%************************************************************************
        Hsecondleadlagtustin = holdGsecond .* Dleadlagtustin;

    MagDBsecondleadlagtustin=20*log10(abs(Hsecondleadlagtustin));
      Phaseradsecondleadlagtustin=angle(Hsecondleadlagtustin);
   Phasesecondleadlagtustin=Phaseradsecondleadlagtustin*180/pi;

%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE
% LEAD*LAG (ST1)
%************************************************************************
        Hsecondleadlagst1 = holdGsecond .* Dleadlagst1;
        MagDBsecondleadlagst1=20*log10(abs(Hsecondleadlagst1));
        Phaseradsecondleadlagst1=angle(Hsecondleadlagst1);
        Phasesecondleadlagst1=Phaseradsecondleadlagst1*180/pi;
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle REDE
% LEAD*LAG (ST2)
%************************************************************************
        Hsecondleadlagst2 = holdGsecond .* Dleadlagst2;
```

```
                MagDBsecondleadlagst2=20*log10(abs(Hsecondleadlagst2));
                Phaseradsecondleadlagst2=angle(Hsecondleadlagst2);
                Phasesecondleadlagst2=Phaseradsecondleadlagst2*180/pi;
        %************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle REDE
        % LEAD*LAG (Schneider)
        %************************************************************************
                Hsecondleadlagschneider = holdGsecond .*
Dleadlagschneider;

    MagDBsecondleadlagschneider=20*log10(abs(Hsecondleadlagschneider));
        Phaseradsecondleadlagschneider=angle(Hsecondleadlagschneider);
      Phasesecondleadlagschneider=Phaseradsecondleadlagschneider*180/pi;


%************************************************************************
%************************************************************************
%
%   LOOPS CONDICIONAIS VISANDO FILTRAR VALORES DE 20.LOG|H*|
%   PROXIMO DE 0,01 E FASE(H*) PROXIMO DE 180 GRAUS (OU
%   MULTIPLOS IMPARES)
%   DAI GRAVA OS VALORES DO PERIODO DE AMOSTRAGEM T,
%   FREQUENCIA w,
%
%************************************************************************
%************************************************************************

%*******************************************
%*******************************************
%       PLANTA DE PRIMEIRA ORDEM
%*******************************************
```

```
%*****************************************

        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle Proporcional
        %************************************************************************

                if (abs(MagDBfirstproporcional)<0.01) &   (
((abs(Phasefirstproporcional)> 179.9) & (abs(Phasefirstproporcional)< 180.1)) |
((abs(Phasefirstproporcional)> 539) & (abs(Phasefirstproporcional)< 541)) |
((abs(Phasefirstproporcional)> 899) & (abs(Phasefirstproporcional)< 901)) |
((abs(Phasefirstproporcional)> 1259) & (abs(Phasefirstproporcional)< 1261)) |
((abs(Phasefirstproporcional)> 1619) & (abs(Phasefirstproporcional)< 1621))  |
((abs(Phasefirstproporcional)> 1979) & (abs(Phasefirstproporcional)< 1981))  |
((abs(Phasefirstproporcional)> 2339) & (abs(Phasefirstproporcional)< 2341))  )

                %Montagem dos vetores armazenando os dados importantes:
                  k1=k1+1;
                  samplingperiodfirstproporcional(k1)=T;
                  frequencyfirstproporcional(k1)=w;

                 modulofirstproporcional(k1)=10^(MagDBfirstproporcional/20);
                  moduloDBfirstproporcional(k1)=MagDBfirstproporcional;
                  fasefirstproporcional(k1)=Phasefirstproporcional;
                  %Grava arquivo formato .MAT:
                  save sttheoryfirstproporcional
samplingperiodfirstproporcional frequencyfirstproporcional modulofirstproporcional
moduloDBfirstproporcional fasefirstproporcional
                end
        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
        % Forward
```

```
%************************************************************************

                    if (abs(MagDBfirstderivativoforward)<0.01) &   (
((abs(Phasefirstderivativoforward)> 179.9) & (abs(Phasefirstderivativoforward)<
180.1)) | ((abs(Phasefirstderivativoforward)> 539) &
(abs(Phasefirstderivativoforward)< 541)) | ((abs(Phasefirstderivativoforward)> 899)
& (abs(Phasefirstderivativoforward)< 901)) | ((abs(Phasefirstderivativoforward)>
1259) & (abs(Phasefirstderivativoforward)< 1261)) |
((abs(Phasefirstderivativoforward)> 1619) & (abs(Phasefirstderivativoforward)<
1621))  | ((abs(Phasefirstderivativoforward)> 1979) &
(abs(Phasefirstderivativoforward)< 1981))  | ((abs(Phasefirstderivativoforward)>
2339) & (abs(Phasefirstderivativoforward)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k2=k2+1;
                      samplingperiodfirstderivativoforward(k2)=T;
                      frequencyfirstderivativoforward(k2)=w;

modulofirstderivativoforward(k2)=10^(MagDBfirstderivativoforward/20);

moduloDBfirstderivativoforward(k2)=MagDBfirstderivativoforward;
                      fasefirstderivativoforward(k2)=Phasefirstderivativoforward;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstderivativoforward
samplingperiodfirstderivativoforward frequencyfirstderivativoforward
modulofirstderivativoforward moduloDBfirstderivativoforward
fasefirstderivativoforward
                    end
          %************************************************************************
          % H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
          % Backward
```

```
%**************************************************************************

                    if (abs(MagDBfirstderivativobackward)<0.01) &   (
((abs(Phasefirstderivativobackward)> 179.9) &
(abs(Phasefirstderivativobackward)< 180.1)) |
((abs(Phasefirstderivativobackward)> 539) & (abs(Phasefirstderivativobackward)<
541)) | ((abs(Phasefirstderivativobackward)> 899) &
(abs(Phasefirstderivativobackward)< 901)) | ((abs(Phasefirstderivativobackward)>
1259) & (abs(Phasefirstderivativobackward)< 1261)) |
((abs(Phasefirstderivativobackward)> 1619) & (abs(Phasefirstderivativobackward)<
1621))  | ((abs(Phasefirstderivativobackward)> 1979) &
(abs(Phasefirstderivativobackward)< 1981))  |
((abs(Phasefirstderivativobackward)> 2339) & (abs(Phasefirstderivativobackward)<
2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k3=k3+1;
                      samplingperiodfirstderivativobackward(k3)=T;
                      frequencyfirstderivativobackward(k3)=w;

modulofirstderivativobackward(k3)=10^(MagDBfirstderivativobackward/20);
moduloDBfirstderivativobackward(k3)=MagDBfirstderivativobackward;
fasefirstderivativobackward(k3)=Phasefirstderivativobackward;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstderivativobackward
samplingperiodfirstderivativobackward frequencyfirstderivativobackward
modulofirstderivativobackward moduloDBfirstderivativobackward
fasefirstderivativobackward
                    end
```

```
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% Tustin
%**************************************************************************

                    if (abs(MagDBfirstderivativotustin)<0.01) &   (
((abs(Phasefirstderivativotustin)> 179.9) & (abs(Phasefirstderivativotustin)< 180.1))
| ((abs(Phasefirstderivativotustin)> 539) & (abs(Phasefirstderivativotustin)< 541)) |
((abs(Phasefirstderivativotustin)> 899) & (abs(Phasefirstderivativotustin)< 901)) |
((abs(Phasefirstderivativotustin)> 1259) & (abs(Phasefirstderivativotustin)< 1261)) |
((abs(Phasefirstderivativotustin)> 1619) & (abs(Phasefirstderivativotustin)< 1621))
| ((abs(Phasefirstderivativotustin)> 1979) & (abs(Phasefirstderivativotustin)< 1981))
| ((abs(Phasefirstderivativotustin)> 2339) & (abs(Phasefirstderivativotustin)< 2341))
)

                     %Montagem dos vetores armazenando os dados importantes:
                       k4=k4+1;
                       samplingperiodfirstderivativotustin(k4)=T;
                       frequencyfirstderivativotustin(k4)=w;

modulofirstderivativotustin(k4)=10^(MagDBfirstderivativotustin/20);
moduloDBfirstderivativotustin(k4)=MagDBfirstderivativotustin;
                       fasefirstderivativotustin(k4)=Phasefirstderivativotustin;
                       %Grava arquivo formato .MAT:
                       save sttheoryfirstderivativotustin
samplingperiodfirstderivativotustin frequencyfirstderivativotustin
modulofirstderivativotustin moduloDBfirstderivativotustin fasefirstderivativotustin
                    end
```

```
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% ST1
%************************************************************************

                    if (abs(MagDBfirstderivativost1)<0.01) &   (
((abs(Phasefirstderivativost1)> 179.9) & (abs(Phasefirstderivativost1)< 180.1)) |
((abs(Phasefirstderivativost1)> 539) & (abs(Phasefirstderivativost1)< 541)) |
((abs(Phasefirstderivativost1)> 899) & (abs(Phasefirstderivativost1)< 901)) |
((abs(Phasefirstderivativost1)> 1259) & (abs(Phasefirstderivativost1)< 1261)) |
((abs(Phasefirstderivativost1)> 1619) & (abs(Phasefirstderivativost1)< 1621))  |
((abs(Phasefirstderivativost1)> 1979) & (abs(Phasefirstderivativost1)< 1981))  |
((abs(Phasefirstderivativost1)> 2339) & (abs(Phasefirstderivativost1)< 2341))  )

                     %Montagem dos vetores armazenando os dados importantes:
                       k5=k5+1;
                       samplingperiodfirstderivativost1(k5)=T;
                       frequencyfirstderivativost1(k5)=w;

                      modulofirstderivativost1(k5)=10^(MagDBfirstderivativost1/20);
                       moduloDBfirstderivativost1(k5)=MagDBfirstderivativost1;
                       fasefirstderivativost1(k5)=Phasefirstderivativost1;
                       %Grava arquivo formato .MAT:
                       save sttheoryfirstderivativost1
samplingperiodfirstderivativost1 frequencyfirstderivativost1 modulofirstderivativost1
moduloDBfirstderivativost1 fasefirstderivativost1
                    end
```

```
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% ST2
%************************************************************************

                if (abs(MagDBfirstderivativost2)<0.01) &   (
((abs(Phasefirstderivativost2)> 179.9) & (abs(Phasefirstderivativost2)< 180.1)) |
((abs(Phasefirstderivativost2)> 539) & (abs(Phasefirstderivativost2)< 541)) |
((abs(Phasefirstderivativost2)> 899) & (abs(Phasefirstderivativost2)< 901)) |
((abs(Phasefirstderivativost2)> 1259) & (abs(Phasefirstderivativost2)< 1261)) |
((abs(Phasefirstderivativost2)> 1619) & (abs(Phasefirstderivativost2)< 1621))  |
((abs(Phasefirstderivativost2)> 1979) & (abs(Phasefirstderivativost2)< 1981))  |
((abs(Phasefirstderivativost2)> 2339) & (abs(Phasefirstderivativost2)< 2341))  )

                  %Montagem dos vetores armazenando os dados importantes:
                    k6=k6+1;
                    samplingperiodfirstderivativost2(k6)=T;
                    frequencyfirstderivativost2(k6)=w;

                   modulofirstderivativost2(k6)=10^(MagDBfirstderivativost2/20);
                    moduloDBfirstderivativost2(k6)=MagDBfirstderivativost2;
                    fasefirstderivativost2(k6)=Phasefirstderivativost2;
                    %Grava arquivo formato .MAT:
                    save sttheoryfirstderivativost2
samplingperiodfirstderivativost2 frequencyfirstderivativost2 modulofirstderivativost2
moduloDBfirstderivativost2 fasefirstderivativost2
                end
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Derivativo
% Schneider
```

```
%**************************************************************************

                    if (abs(MagDBfirstderivativoschneider)<0.01) &   (
((abs(Phasefirstderivativoschneider)> 179.9) &
(abs(Phasefirstderivativoschneider)< 180.1)) |
((abs(Phasefirstderivativoschneider)> 539) & (abs(Phasefirstderivativoschneider)<
541)) | ((abs(Phasefirstderivativoschneider)> 899) &
(abs(Phasefirstderivativoschneider)< 901)) | ((abs(Phasefirstderivativoschneider)>
1259) & (abs(Phasefirstderivativoschneider)< 1261)) |
((abs(Phasefirstderivativoschneider)> 1619) &
(abs(Phasefirstderivativoschneider)< 1621))  |
((abs(Phasefirstderivativoschneider)> 1979) &
(abs(Phasefirstderivativoschneider)< 1981))  |
((abs(Phasefirstderivativoschneider)> 2339) &
(abs(Phasefirstderivativoschneider)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k7=k7+1;
                      samplingperiodfirstderivativoschneider(k7)=T;
                      frequencyfirstderivativoschneider(k7)=w;

modulofirstderivativoschneider(k7)=10^(MagDBfirstderivativoschneider/20);
moduloDBfirstderivativoschneider(k7)=MagDBfirstderivativoschneider;
fasefirstderivativoschneider(k7)=Phasefirstderivativoschneider;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstderivativoschneider
samplingperiodfirstderivativoschneider frequencyfirstderivativoschneider
modulofirstderivativoschneider moduloDBfirstderivativoschneider
fasefirstderivativoschneider
                    end
```

```
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% Forward
%**************************************************************************

                    if (abs(MagDBfirstintegralforward)<0.01) &   (
((abs(Phasefirstintegralforward)> 179.9) & (abs(Phasefirstintegralforward)< 180.1))
| ((abs(Phasefirstintegralforward)> 539) & (abs(Phasefirstintegralforward)< 541)) |
((abs(Phasefirstintegralforward)> 899) & (abs(Phasefirstintegralforward)< 901)) |
((abs(Phasefirstintegralforward)> 1259) & (abs(Phasefirstintegralforward)< 1261)) |
((abs(Phasefirstintegralforward)> 1619) & (abs(Phasefirstintegralforward)< 1621))
| ((abs(Phasefirstintegralforward)> 1979) & (abs(Phasefirstintegralforward)< 1981))
| ((abs(Phasefirstintegralforward)> 2339) & (abs(Phasefirstintegralforward)< 2341))
)

                     %Montagem dos vetores armazenando os dados importantes:
                       k8=k8+1;
                       samplingperiodfirstintegralforward(k8)=T;
                       frequencyfirstintegralforward(k8)=w;

modulofirstintegralforward(k8)=10^(MagDBfirstintegralforward/20);

moduloDBfirstintegralforward(k8)=MagDBfirstintegralforward;
                       fasefirstintegralforward(k8)=Phasefirstintegralforward;
                       %Grava arquivo formato .MAT:
                       save sttheoryfirstintegralforward
samplingperiodfirstintegralforward frequencyfirstintegralforward
modulofirstintegralforward moduloDBfirstintegralforward fasefirstintegralforward
                    end
```

```
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% Backward
%**************************************************************************

                    if (abs(MagDBfirstintegralbackward)<0.01) &   (
((abs(Phasefirstintegralbackward)> 179.9) & (abs(Phasefirstintegralbackward)<
180.1)) | ((abs(Phasefirstintegralbackward)> 539) &
(abs(Phasefirstintegralbackward)< 541)) | ((abs(Phasefirstintegralbackward)> 899)
& (abs(Phasefirstintegralbackward)< 901)) | ((abs(Phasefirstintegralbackward)>
1259) & (abs(Phasefirstintegralbackward)< 1261)) |
((abs(Phasefirstintegralbackward)> 1619) & (abs(Phasefirstintegralbackward)<
1621))  | ((abs(Phasefirstintegralbackward)> 1979) &
(abs(Phasefirstintegralbackward)< 1981))  | ((abs(Phasefirstintegralbackward)>
2339) & (abs(Phasefirstintegralbackward)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k9=k9+1;
                      samplingperiodfirstintegralbackward(k9)=T;
                      frequencyfirstintegralbackward(k9)=w;

modulofirstintegralbackward(k9)=10^(MagDBfirstintegralbackward/20);
moduloDBfirstintegralbackward(k9)=MagDBfirstintegralbackward;
                      fasefirstintegralbackward(k9)=Phasefirstintegralbackward;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstintegralbackward
samplingperiodfirstintegralbackward frequencyfirstintegralbackward
modulofirstintegralbackward moduloDBfirstintegralbackward
fasefirstintegralbackward
                    end
```

```
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
% Tustin
%**************************************************************************

                if (abs(MagDBfirstintegraltustin)<0.01) &   (
((abs(Phasefirstintegraltustin)> 179.9) & (abs(Phasefirstintegraltustin)< 180.1)) |
((abs(Phasefirstintegraltustin)> 539) & (abs(Phasefirstintegraltustin)< 541)) |
((abs(Phasefirstintegraltustin)> 899) & (abs(Phasefirstintegraltustin)< 901)) |
((abs(Phasefirstintegraltustin)> 1259) & (abs(Phasefirstintegraltustin)< 1261)) |
((abs(Phasefirstintegraltustin)> 1619) & (abs(Phasefirstintegraltustin)< 1621))  |
((abs(Phasefirstintegraltustin)> 1979) & (abs(Phasefirstintegraltustin)< 1981))  |
((abs(Phasefirstintegraltustin)> 2339) & (abs(Phasefirstintegraltustin)< 2341))  )

                 %Montagem dos vetores armazenando os dados importantes:
                   k10=k10+1;
                   samplingperiodfirstintegraltustin(k10)=T;
                   frequencyfirstintegraltustin(k10)=w;

                modulofirstintegraltustin(k10)=10^(MagDBfirstintegraltustin/20);
                   moduloDBfirstintegraltustin(k10)=MagDBfirstintegraltustin;
                   fasefirstintegraltustin(k10)=Phasefirstintegraltustin;
                   %Grava arquivo formato .MAT:
                   save sttheoryfirstintegraltustin
samplingperiodfirstintegraltustin frequencyfirstintegraltustin
modulofirstintegraltustin moduloDBfirstintegraltustin fasefirstintegraltustin
                end
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle Integral ST1
%**************************************************************************
```

```
                    if (abs(MagDBfirstintegralst1)<0.01) &   (
((abs(Phasefirstintegralst1)> 179.9) & (abs(Phasefirstintegralst1)< 180.1)) |
((abs(Phasefirstintegralst1)> 539) & (abs(Phasefirstintegralst1)< 541)) |
((abs(Phasefirstintegralst1)> 899) & (abs(Phasefirstintegralst1)< 901)) |
((abs(Phasefirstintegralst1)> 1259) & (abs(Phasefirstintegralst1)< 1261)) |
((abs(Phasefirstintegralst1)> 1619) & (abs(Phasefirstintegralst1)< 1621))  |
((abs(Phasefirstintegralst1)> 1979) & (abs(Phasefirstintegralst1)< 1981))  |
((abs(Phasefirstintegralst1)> 2339) & (abs(Phasefirstintegralst1)< 2341))  )

                     %Montagem dos vetores armazenando os dados importantes:
                       k11=k11+1;
                       samplingperiodfirstintegralst1(k11)=T;
                       frequencyfirstintegralst1(k11)=w;
                       modulofirstintegralst1(k11)=10^(MagDBfirstintegralst1/20);
                       moduloDBfirstintegralst1(k11)=MagDBfirstintegralst1;
                       fasefirstintegralst1(k11)=Phasefirstintegralst1;
                       %Grava arquivo formato .MAT:
                       save sttheoryfirstintegralst1 samplingperiodfirstintegralst1
frequencyfirstintegralst1 modulofirstintegralst1 moduloDBfirstintegralst1
fasefirstintegralst1
                    end
               %************************************************************************
               % H*(s) para Planta de 1a. Ordem (com hold) e controle Integral ST2
               %************************************************************************

                    if (abs(MagDBfirstintegralst2)<0.01) &   (
((abs(Phasefirstintegralst2)> 179.9) & (abs(Phasefirstintegralst2)< 180.1)) |
((abs(Phasefirstintegralst2)> 539) & (abs(Phasefirstintegralst2)< 541)) |
((abs(Phasefirstintegralst2)> 899) & (abs(Phasefirstintegralst2)< 901)) |
((abs(Phasefirstintegralst2)> 1259) & (abs(Phasefirstintegralst2)< 1261)) |
```

```
((abs(Phasefirstintegralst2)> 1619) & (abs(Phasefirstintegralst2)< 1621))  |
((abs(Phasefirstintegralst2)> 1979) & (abs(Phasefirstintegralst2)< 1981))  |
((abs(Phasefirstintegralst2)> 2339) & (abs(Phasefirstintegralst2)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k12=k12+1;
                      samplingperiodfirstintegralst2(k12)=T;
                      frequencyfirstintegralst2(k12)=w;
                      modulofirstintegralst2(k12)=10^(MagDBfirstintegralst2/20);
                      moduloDBfirstintegralst2(k12)=MagDBfirstintegralst2;
                      fasefirstintegralst2(k12)=Phasefirstintegralst2;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstintegralst2 samplingperiodfirstintegralst2
frequencyfirstintegralst2 modulofirstintegralst2 moduloDBfirstintegralst2
fasefirstintegralst2
                    end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle Integral
            % Schneider
            %************************************************************************

                    if (abs(MagDBfirstintegralschneider)<0.01) &   (
((abs(Phasefirstintegralschneider)> 179.9) & (abs(Phasefirstintegralschneider)<
180.1)) | ((abs(Phasefirstintegralschneider)> 539) &
(abs(Phasefirstintegralschneider)< 541)) | ((abs(Phasefirstintegralschneider)> 899)
& (abs(Phasefirstintegralschneider)< 901)) | ((abs(Phasefirstintegralschneider)>
1259) & (abs(Phasefirstintegralschneider)< 1261)) |
((abs(Phasefirstintegralschneider)> 1619) & (abs(Phasefirstintegralschneider)<
1621))  | ((abs(Phasefirstintegralschneider)> 1979) &
(abs(Phasefirstintegralschneider)< 1981))  | ((abs(Phasefirstintegralschneider)>
2339) & (abs(Phasefirstintegralschneider)< 2341))  )
```

```
                    %Montagem dos vetores armazenando os dados importantes:
                      k13=k13+1;
                      samplingperiodfirstintegralschneider(k13)=T;
                      frequencyfirstintegralschneider(k13)=w;

modulofirstintegralschneider(k13)=10^(MagDBfirstintegralschneider/20);

moduloDBfirstintegralschneider(k13)=MagDBfirstintegralschneider;
                      fasefirstintegralchneider(k13)=Phasefirstintegralschneider;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstintegralschneider
samplingperiodfirstintegralschneider frequencyfirstintegralschneider
modulofirstintegralschneider moduloDBfirstintegralschneider
fasefirstintegralschneider
                    end

          %************************************************************************
          % H*(s) para Planta de 1a. Ordem (com hold) e controle PD Forward
          %************************************************************************

                    if (abs(MagDBfirstpdforward)<0.01) &   (
((abs(Phasefirstpdforward)> 179.9) & (abs(Phasefirstpdforward)< 180.1)) |
((abs(Phasefirstpdforward)> 539) & (abs(Phasefirstpdforward)< 541)) |
((abs(Phasefirstpdforward)> 899) & (abs(Phasefirstpdforward)< 901)) |
((abs(Phasefirstpdforward)> 1259) & (abs(Phasefirstpdforward)< 1261)) |
((abs(Phasefirstpdforward)> 1619) & (abs(Phasefirstpdforward)< 1621))  |
((abs(Phasefirstpdforward)> 1979) & (abs(Phasefirstpdforward)< 1981))  |
((abs(Phasefirstpdforward)> 2339) & (abs(Phasefirstpdforward)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
```

```
                    k14=k14+1;
                    samplingperiodfirstpdforward(k14)=T;
                    frequencyfirstpdforward(k14)=w;
                    modulofirstpdforward(k14)=10^(MagDBfirstpdforward/20);
                    moduloDBfirstpdforward(k14)=MagDBfirstpdforward;
                    fasefirstpdforward(k14)=Phasefirstpdforward;
                    %Grava arquivo formato .MAT:
                    save sttheoryfirstpdforward samplingperiodfirstpdforward
frequencyfirstpdforward modulofirstpdforward moduloDBfirstpdforward
fasefirstpdforward
                end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle PD Backward
            %************************************************************************

                if (abs(MagDBfirstpdbackward)<0.01) &   (
((abs(Phasefirstpdbackward)> 179.9) & (abs(Phasefirstpdbackward)< 180.1)) |
((abs(Phasefirstpdbackward)> 539) & (abs(Phasefirstpdbackward)< 541)) |
((abs(Phasefirstpdbackward)> 899) & (abs(Phasefirstpdbackward)< 901)) |
((abs(Phasefirstpdbackward)> 1259) & (abs(Phasefirstpdbackward)< 1261)) |
((abs(Phasefirstpdbackward)> 1619) & (abs(Phasefirstpdbackward)< 1621))  |
((abs(Phasefirstpdbackward)> 1979) & (abs(Phasefirstpdbackward)< 1981))  |
((abs(Phasefirstpdbackward)> 2339) & (abs(Phasefirstpdbackward)< 2341))  )

                %Montagem dos vetores armazenando os dados importantes:
                    k15=k15+1;
                    samplingperiodfirstpdbackward(k15)=T;
                    frequencyfirstpdbackward(k15)=w;

                modulofirstpdbackward(k15)=10^(MagDBfirstpdbackward/20);
                    moduloDBfirstpdbackward(k15)=MagDBfirstpdbackward;
```

```
                    fasefirstpdbackward(k15)=Phasefirstpdbackward;
                    %Grava arquivo formato .MAT:
                    save sttheoryfirstpdbackward
samplingperiodfirstpdbackward frequencyfirstpdbackward modulofirstpdbackward
moduloDBfirstpdbackward fasefirstpdbackward
                end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle PD Tustin
            %************************************************************************

                if (abs(MagDBfirstpdtustin)<0.01) &   (
((abs(Phasefirstpdtustin)> 179.9) & (abs(Phasefirstpdtustin)< 180.1)) |
((abs(Phasefirstpdtustin)> 539) & (abs(Phasefirstpdtustin)< 541)) |
((abs(Phasefirstpdtustin)> 899) & (abs(Phasefirstpdtustin)< 901)) |
((abs(Phasefirstpdtustin)> 1259) & (abs(Phasefirstpdtustin)< 1261)) |
((abs(Phasefirstpdtustin)> 1619) & (abs(Phasefirstpdtustin)< 1621))  |
((abs(Phasefirstpdtustin)> 1979) & (abs(Phasefirstpdtustin)< 1981))  |
((abs(Phasefirstpdtustin)> 2339) & (abs(Phasefirstpdtustin)< 2341))  )

                 %Montagem dos vetores armazenando os dados importantes:
                    k16=k16+1;
                    samplingperiodfirstpdtustin(k16)=T;
                    frequencyfirstpdtustin(k16)=w;
                    modulofirstpdtustin(k16)=10^(MagDBfirstpdtustin/20);
                    moduloDBfirstpdtustin(k16)=MagDBfirstpdtustin;
                    fasefirstpdtustin(k16)=Phasefirstpdtustin;
                    %Grava arquivo formato .MAT:
                    save sttheoryfirstpdtustin samplingperiodfirstpdtustin
frequencyfirstpdtustin modulofirstpdtustin moduloDBfirstpdtustin fasefirstpdtustin
                end
```

```
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD ST1
%**************************************************************************

                if (abs(MagDBfirstpdst1)<0.01) &   ( ((abs(Phasefirstpdst1)>
179.9) & (abs(Phasefirstpdst1)< 180.1)) | ((abs(Phasefirstpdst1)> 539) &
(abs(Phasefirstpdst1)< 541)) | ((abs(Phasefirstpdst1)> 899) &
(abs(Phasefirstpdst1)< 901)) | ((abs(Phasefirstpdst1)> 1259) &
(abs(Phasefirstpdst1)< 1261)) | ((abs(Phasefirstpdst1)> 1619) &
(abs(Phasefirstpdst1)< 1621))  | ((abs(Phasefirstpdst1)> 1979) &
(abs(Phasefirstpdst1)< 1981))  | ((abs(Phasefirstpdst1)> 2339) &
(abs(Phasefirstpdst1)< 2341))  )

                 %Montagem dos vetores armazenando os dados importantes:
                   k17=k17+1;
                   samplingperiodfirstpdst1(k17)=T;
                   frequencyfirstpdst1(k17)=w;
                   modulofirstpdst1(k17)=10^(MagDBfirstpdst1/20);
                   moduloDBfirstpdst1(k17)=MagDBfirstpdst1;
                   fasefirstpdst1(k17)=Phasefirstpdst1;
                   %Grava arquivo formato .MAT:
                   save sttheoryfirstpdst1 samplingperiodfirstpdst1
frequencyfirstpdst1 modulofirstpdst1 moduloDBfirstpdst1 fasefirstpdst1
                end
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle PD ST2
%**************************************************************************

                if (abs(MagDBfirstpdst2)<0.01) &   ( ((abs(Phasefirstpdst2)>
179.9) & (abs(Phasefirstpdst2)< 180.1)) | ((abs(Phasefirstpdst2)> 539) &
```

```
(abs(Phasefirstpdst2)< 541)) | ((abs(Phasefirstpdst2)> 899) &
(abs(Phasefirstpdst2)< 901)) | ((abs(Phasefirstpdst2)> 1259) &
(abs(Phasefirstpdst2)< 1261)) | ((abs(Phasefirstpdst2)> 1619) &
(abs(Phasefirstpdst2)< 1621))  | ((abs(Phasefirstpdst2)> 1979) &
(abs(Phasefirstpdst2)< 1981))  | ((abs(Phasefirstpdst2)> 2339) &
(abs(Phasefirstpdst2)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k18=k18+1;
                      samplingperiodfirstpdst2(k18)=T;
                      frequencyfirstpdst2(k18)=w;
                      modulofirstpdst2(k18)=10^(MagDBfirstpdst2/20);
                      moduloDBfirstpdst2(k18)=MagDBfirstpdst2;
                      fasefirstpdst2(k18)=Phasefirstpdst2;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstpdst2 samplingperiodfirstpdst2
frequencyfirstpdst2 modulofirstpdst2 moduloDBfirstpdst2 fasefirstpdst2
                    end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle PD
            % Schneider
            %************************************************************************

                    if (abs(MagDBfirstpdschneider)<0.01) &   (
((abs(Phasefirstpdschneider)> 179.9) & (abs(Phasefirstpdschneider)< 180.1)) |
((abs(Phasefirstpdschneider)> 539) & (abs(Phasefirstpdschneider)< 541)) |
((abs(Phasefirstpdschneider)> 899) & (abs(Phasefirstpdschneider)< 901)) |
((abs(Phasefirstpdschneider)> 1259) & (abs(Phasefirstpdschneider)< 1261)) |
((abs(Phasefirstpdschneider)> 1619) & (abs(Phasefirstpdschneider)< 1621))  |
((abs(Phasefirstpdschneider)> 1979) & (abs(Phasefirstpdschneider)< 1981))  |
((abs(Phasefirstpdschneider)> 2339) & (abs(Phasefirstpdschneider)< 2341))  )
```

```
                    %Montagem dos vetores armazenando os dados importantes:
                      k19=k19+1;
                      samplingperiodfirstpdschneider(k19)=T;
                      frequencyfirstpdschneider(k19)=w;

                     modulofirstpdschneider(k19)=10^(MagDBfirstpdschneider/20);
                      moduloDBfirstpdschneider(k19)=MagDBfirstpdschneider;
                      fasefirstpdchneider(k19)=Phasefirstpdschneider;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstpdschneider
samplingperiodfirstpdschneider frequencyfirstpdschneider modulofirstpdschneider
moduloDBfirstpdschneider fasefirstpdschneider
                    end

          %************************************************************************
          % H*(s) para Planta de 1a. Ordem (com hold) e controle PID Forward
          %************************************************************************

                    if (abs(MagDBfirstpidforward)<0.01) &   (
((abs(Phasefirstpidforward)> 179.9) & (abs(Phasefirstpidforward)< 180.1)) |
((abs(Phasefirstpidforward)> 539) & (abs(Phasefirstpidforward)< 541)) |
((abs(Phasefirstpidforward)> 899) & (abs(Phasefirstpidforward)< 901)) |
((abs(Phasefirstpidforward)> 1259) & (abs(Phasefirstpidforward)< 1261)) |
((abs(Phasefirstpidforward)> 1619) & (abs(Phasefirstpidforward)< 1621))  |
((abs(Phasefirstpidforward)> 1979) & (abs(Phasefirstpidforward)< 1981))  |
((abs(Phasefirstpidforward)> 2339) & (abs(Phasefirstpidforward)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k20=k20+1;
                      samplingperiodfirstpidforward(k20)=T;
```

```
                frequencyfirstpidforward(k20)=w;
                modulofirstpidforward(k20)=10^(MagDBfirstpidforward/20);
                moduloDBfirstpidforward(k20)=MagDBfirstpidforward;
                fasefirstpidforward(k20)=Phasefirstpidforward;
                %Grava arquivo formato .MAT:
                save sttheoryfirstpidforward samplingperiodfirstpidforward
frequencyfirstpidforward modulofirstpidforward moduloDBfirstpidforward
fasefirstpidforward
            end
        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle PID
        % Backward
        %************************************************************************

            if (abs(MagDBfirstpidbackward)<0.01) &   (
((abs(Phasefirstpidbackward)> 179.9) & (abs(Phasefirstpidbackward)< 180.1)) |
((abs(Phasefirstpidbackward)> 539) & (abs(Phasefirstpidbackward)< 541)) |
((abs(Phasefirstpidbackward)> 899) & (abs(Phasefirstpidbackward)< 901)) |
((abs(Phasefirstpidbackward)> 1259) & (abs(Phasefirstpidbackward)< 1261)) |
((abs(Phasefirstpidbackward)> 1619) & (abs(Phasefirstpidbackward)< 1621))  |
((abs(Phasefirstpidbackward)> 1979) & (abs(Phasefirstpidbackward)< 1981))  |
((abs(Phasefirstpidbackward)> 2339) & (abs(Phasefirstpidbackward)< 2341))  )

             %Montagem dos vetores armazenando os dados importantes:
                k21=k21+1;
                samplingperiodfirstpidbackward(k21)=T;
                frequencyfirstpidbackward(k21)=w;

            modulofirstpidbackward(k21)=10^(MagDBfirstpidbackward/20);
                moduloDBfirstpidbackward(k21)=MagDBfirstpidbackward;
                fasefirstpidbackward(k21)=Phasefirstpidbackward;
```

```
                %Grava arquivo formato .MAT:
                save sttheoryfirstpidbackward
samplingperiodfirstpidbackward frequencyfirstpidbackward modulofirstpidbackward
moduloDBfirstpidbackward fasefirstpidbackward
                end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle PID Tustin
            %************************************************************************

                if (abs(MagDBfirstpidtustin)<0.01) &   (
((abs(Phasefirstpidtustin)> 179.9) & (abs(Phasefirstpidtustin)< 180.1)) |
((abs(Phasefirstpidtustin)> 539) & (abs(Phasefirstpidtustin)< 541)) |
((abs(Phasefirstpidtustin)> 899) & (abs(Phasefirstpidtustin)< 901)) |
((abs(Phasefirstpidtustin)> 1259) & (abs(Phasefirstpidtustin)< 1261)) |
((abs(Phasefirstpidtustin)> 1619) & (abs(Phasefirstpidtustin)< 1621))  |
((abs(Phasefirstpidtustin)> 1979) & (abs(Phasefirstpidtustin)< 1981))  |
((abs(Phasefirstpidtustin)> 2339) & (abs(Phasefirstpidtustin)< 2341))  )

                %Montagem dos vetores armazenando os dados importantes:
                k22=k22+1;
                samplingperiodfirstpidtustin(k22)=T;
                frequencyfirstpidtustin(k22)=w;
                modulofirstpidtustin(k22)=10^(MagDBfirstpidtustin/20);
                moduloDBfirstpidtustin(k22)=MagDBfirstpidtustin;
                fasefirstpidtustin(k22)=Phasefirstpidtustin;
                %Grava arquivo formato .MAT:
                save sttheoryfirstpidtustin samplingperiodfirstpidtustin
frequencyfirstpidtustin modulofirstpidtustin moduloDBfirstpidtustin fasefirstpidtustin
                end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle PID ST1
```

```
%**************************************************************************

                if (abs(MagDBfirstpidst1)<0.01) &   ( ((abs(Phasefirstpidst1)>
179.9) & (abs(Phasefirstpidst1)< 180.1)) | ((abs(Phasefirstpidst1)> 539) &
(abs(Phasefirstpidst1)< 541)) | ((abs(Phasefirstpidst1)> 899) &
(abs(Phasefirstpidst1)< 901)) | ((abs(Phasefirstpidst1)> 1259) &
(abs(Phasefirstpidst1)< 1261)) | ((abs(Phasefirstpidst1)> 1619) &
(abs(Phasefirstpidst1)< 1621))  | ((abs(Phasefirstpidst1)> 1979) &
(abs(Phasefirstpidst1)< 1981))  | ((abs(Phasefirstpidst1)> 2339) &
(abs(Phasefirstpidst1)< 2341))  )

                 %Montagem dos vetores armazenando os dados importantes:
                   k23=k23+1;
                   samplingperiodfirstpidst1(k23)=T;
                   frequencyfirstpidst1(k23)=w;
                   modulofirstpidst1(k23)=10^(MagDBfirstpidst1/20);
                   moduloDBfirstpidst1(k23)=MagDBfirstpidst1;
                   fasefirstpidst1(k23)=Phasefirstpidst1;
                   %Grava arquivo formato .MAT:
                   save sttheoryfirstpidst1 samplingperiodfirstpidst1
frequencyfirstpidst1 modulofirstpidst1 moduloDBfirstpidst1 fasefirstpidst1
                end
            %**************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle PID ST2
            %**************************************************************************

                if (abs(MagDBfirstpidst2)<0.01) &   ( ((abs(Phasefirstpidst2)>
179.9) & (abs(Phasefirstpidst2)< 180.1)) | ((abs(Phasefirstpidst2)> 539) &
(abs(Phasefirstpidst2)< 541)) | ((abs(Phasefirstpidst2)> 899) &
(abs(Phasefirstpidst2)< 901)) | ((abs(Phasefirstpidst2)> 1259) &
(abs(Phasefirstpidst2)< 1261)) | ((abs(Phasefirstpidst2)> 1619) &
```

```
(abs(Phasefirstpidst2)< 1621))  | ((abs(Phasefirstpidst2)> 1979) &
(abs(Phasefirstpidst2)< 1981))  | ((abs(Phasefirstpidst2)> 2339) &
(abs(Phasefirstpidst2)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k24=k24+1;
                      samplingperiodfirstpidst2(k24)=T;
                      frequencyfirstpidst2(k24)=w;
                      modulofirstpidst2(k24)=10^(MagDBfirstpidst2/20);
                      moduloDBfirstpidst2(k24)=MagDBfirstpidst2;
                      fasefirstpidst2(k24)=Phasefirstpidst2;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstpidst2 samplingperiodfirstpidst2
frequencyfirstpidst2 modulofirstpidst2 moduloDBfirstpidst2 fasefirstpidst2
                  end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle PID
            % Schneider
            %************************************************************************

                  if (abs(MagDBfirstpidschneider)<0.01) &   (
((abs(Phasefirstpidschneider)> 179.9) & (abs(Phasefirstpidschneider)< 180.1)) |
((abs(Phasefirstpidschneider)> 539) & (abs(Phasefirstpidschneider)< 541)) |
((abs(Phasefirstpidschneider)> 899) & (abs(Phasefirstpidschneider)< 901)) |
((abs(Phasefirstpidschneider)> 1259) & (abs(Phasefirstpidschneider)< 1261)) |
((abs(Phasefirstpidschneider)> 1619) & (abs(Phasefirstpidschneider)< 1621))  |
((abs(Phasefirstpidschneider)> 1979) & (abs(Phasefirstpidschneider)< 1981))  |
((abs(Phasefirstpidschneider)> 2339) & (abs(Phasefirstpidschneider)< 2341))  )

                    %Montagem dos vetores armazenando os dados importantes:
                      k25=k25+1;
```

```
                samplingperiodfirstpidschneider(k25)=T;
                frequencyfirstpidschneider(k25)=w;

            modulofirstpidschneider(k25)=10^(MagDBfirstpidschneider/20);
                moduloDBfirstpidschneider(k25)=MagDBfirstpidschneider;
                fasefirstpidschneider(k25)=Phasefirstpidschneider;
                %Grava arquivo formato .MAT:
                save sttheoryfirstpidschneider
samplingperiodfirstpidschneider frequencyfirstpidschneider
modulofirstpidschneider moduloDBfirstpidschneider fasefirstpidschneider
            end

        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD
        % Forward
        %************************************************************************

            if (abs(MagDBfirstleadforward)<0.01) &   (
((abs(Phasefirstleadforward)> 179.9) & (abs(Phasefirstleadforward)< 180.1)) |
((abs(Phasefirstleadforward)> 539) & (abs(Phasefirstleadforward)< 541)) |
((abs(Phasefirstleadforward)> 899) & (abs(Phasefirstleadforward)< 901)) |
((abs(Phasefirstleadforward)> 1259) & (abs(Phasefirstleadforward)< 1261)) |
((abs(Phasefirstleadforward)> 1619) & (abs(Phasefirstleadforward)< 1621))  |
((abs(Phasefirstleadforward)> 1979) & (abs(Phasefirstleadforward)< 1981))  |
((abs(Phasefirstleadforward)> 2339) & (abs(Phasefirstleadforward)< 2341))  )

             %Montagem dos vetores armazenando os dados importantes:
                k26=k26+1;
                samplingperiodfirstleadforward(k26)=T;
                frequencyfirstleadforward(k26)=w;
```

```
                    modulofirstleadforward(k26)=10^(MagDBfirstleadforward/20);
                     moduloDBfirstleadforward(k26)=MagDBfirstleadforward;
                     fasefirstleadforward(k26)=Phasefirstleadforward;
                     %Grava arquivo formato .MAT:
                     save sttheoryfirstleadforward samplingperiodfirstleadforward
frequencyfirstleadforward modulofirstleadforward moduloDBfirstleadforward
fasefirstleadforward
                end
          %************************************************************************
          % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD
          % Backward
          %************************************************************************

                if (abs(MagDBfirstleadbackward)<0.01) &   (
((abs(Phasefirstleadbackward)> 179.9) & (abs(Phasefirstleadbackward)< 180.1)) |
((abs(Phasefirstleadbackward)> 539) & (abs(Phasefirstleadbackward)< 541)) |
((abs(Phasefirstleadbackward)> 899) & (abs(Phasefirstleadbackward)< 901)) |
((abs(Phasefirstleadbackward)> 1259) & (abs(Phasefirstleadbackward)< 1261)) |
((abs(Phasefirstleadbackward)> 1619) & (abs(Phasefirstleadbackward)< 1621))  |
((abs(Phasefirstleadbackward)> 1979) & (abs(Phasefirstleadbackward)< 1981))  |
((abs(Phasefirstleadbackward)> 2339) & (abs(Phasefirstleadbackward)< 2341))  )

                 %Montagem dos vetores armazenando os dados importantes:
                   k27=k27+1;
                   samplingperiodfirstleadbackward(k27)=T;
                   frequencyfirstleadbackward(k27)=w;

modulofirstleadbackward(k27)=10^(MagDBfirstleadbackward/20);
                   moduloDBfirstleadbackward(k27)=MagDBfirstleadbackward;
                   fasefirstleadbackward(k27)=Phasefirstleadbackward;
```

```
                %Grava arquivo formato .MAT:
                save sttheoryfirstleadbackward
samplingperiodfirstleadbackward frequencyfirstleadbackward
modulofirstleadbackward moduloDBfirstleadbackward fasefirstleadbackward
                end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD Tustin
            %************************************************************************

                if (abs(MagDBfirstleadtustin)<0.01) &   (
((abs(Phasefirstleadtustin)> 179.9) & (abs(Phasefirstleadtustin)< 180.1)) |
((abs(Phasefirstleadtustin)> 539) & (abs(Phasefirstleadtustin)< 541)) |
((abs(Phasefirstleadtustin)> 899) & (abs(Phasefirstleadtustin)< 901)) |
((abs(Phasefirstleadtustin)> 1259) & (abs(Phasefirstleadtustin)< 1261)) |
((abs(Phasefirstleadtustin)> 1619) & (abs(Phasefirstleadtustin)< 1621))  |
((abs(Phasefirstleadtustin)> 1979) & (abs(Phasefirstleadtustin)< 1981))  |
((abs(Phasefirstleadtustin)> 2339) & (abs(Phasefirstleadtustin)< 2341))  )

                %Montagem dos vetores armazenando os dados importantes:
                k28=k28+1;
                samplingperiodfirstleadtustin(k28)=T;
                frequencyfirstleadtustin(k28)=w;
                modulofirstleadtustin(k28)=10^(MagDBfirstleadtustin/20);
                moduloDBfirstleadtustin(k28)=MagDBfirstleadtustin;
                fasefirstleadtustin(k28)=Phasefirstleadtustin;
                %Grava arquivo formato .MAT:
                save sttheoryfirstleadtustin samplingperiodfirstleadtustin
frequencyfirstleadtustin modulofirstleadtustin moduloDBfirstleadtustin
fasefirstleadtustin
                end
```

```
%*************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD ST1
%*************************************************************************

                if (abs(MagDBfirstleadst1)<0.01) &   (
((abs(Phasefirstleadst1)> 179.9) & (abs(Phasefirstleadst1)< 180.1)) |
((abs(Phasefirstleadst1)> 539) & (abs(Phasefirstleadst1)< 541)) |
((abs(Phasefirstleadst1)> 899) & (abs(Phasefirstleadst1)< 901)) |
((abs(Phasefirstleadst1)> 1259) & (abs(Phasefirstleadst1)< 1261)) |
((abs(Phasefirstleadst1)> 1619) & (abs(Phasefirstleadst1)< 1621))  |
((abs(Phasefirstleadst1)> 1979) & (abs(Phasefirstleadst1)< 1981))  |
((abs(Phasefirstleadst1)> 2339) & (abs(Phasefirstleadst1)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k29=k29+1;
                      samplingperiodfirstleadst1(k29)=T;
                      frequencyfirstleadst1(k29)=w;
                      modulofirstleadst1(k29)=10^(MagDBfirstleadst1/20);
                      moduloDBfirstleadst1(k29)=MagDBfirstleadst1;
                      fasefirstleadst1(k29)=Phasefirstleadst1;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstleadst1 samplingperiodfirstleadst1
frequencyfirstleadst1 modulofirstleadst1 moduloDBfirstleadst1 fasefirstleadst1
                end
%*************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD ST2
%*************************************************************************

                if (abs(MagDBfirstleadst2)<0.01) &   (
((abs(Phasefirstleadst2)> 179.9) & (abs(Phasefirstleadst2)< 180.1)) |
((abs(Phasefirstleadst2)> 539) & (abs(Phasefirstleadst2)< 541)) |
```

```
((abs(Phasefirstleadst2)> 899) & (abs(Phasefirstleadst2)< 901)) |
((abs(Phasefirstleadst2)> 1259) & (abs(Phasefirstleadst2)< 1261)) |
((abs(Phasefirstleadst2)> 1619) & (abs(Phasefirstleadst2)< 1621))  |
((abs(Phasefirstleadst2)> 1979) & (abs(Phasefirstleadst2)< 1981))  |
((abs(Phasefirstleadst2)> 2339) & (abs(Phasefirstleadst2)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k30=k30+1;
                      samplingperiodfirstleadst2(k30)=T;
                      frequencyfirstleadst2(k30)=w;
                      modulofirstleadst2(k30)=10^(MagDBfirstleadst2/20);
                      moduloDBfirstleadst2(k30)=MagDBfirstleadst2;
                      fasefirstleadst2(k30)=Phasefirstleadst2;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstleadst2 samplingperiodfirstleadst2
frequencyfirstleadst2 modulofirstleadst2 moduloDBfirstleadst2 fasefirstleadst2
                  end
            %*************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD
            % Schneider
            %*************************************************************************

                  if (abs(MagDBfirstleadschneider)<0.01) &   (
((abs(Phasefirstleadschneider)> 179.9) & (abs(Phasefirstleadschneider)< 180.1)) |
((abs(Phasefirstleadschneider)> 539) & (abs(Phasefirstleadschneider)< 541)) |
((abs(Phasefirstleadschneider)> 899) & (abs(Phasefirstleadschneider)< 901)) |
((abs(Phasefirstleadschneider)> 1259) & (abs(Phasefirstleadschneider)< 1261)) |
((abs(Phasefirstleadschneider)> 1619) & (abs(Phasefirstleadschneider)< 1621))  |
((abs(Phasefirstleadschneider)> 1979) & (abs(Phasefirstleadschneider)< 1981))  |
((abs(Phasefirstleadschneider)> 2339) & (abs(Phasefirstleadschneider)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k31=k31+1;
```

```
                samplingperiodfirstleadschneider(k31)=T;
                frequencyfirstleadschneider(k31)=w;

            modulofirstleadschneider(k31)=10^(MagDBfirstleadschneider/20);
                moduloDBfirstleadschneider(k31)=MagDBfirstleadschneider;
                fasefirstleadschneider(k31)=Phasefirstleadschneider;
                %Grava arquivo formato .MAT:
                save sttheoryfirstleadschneider
samplingperiodfirstleadschneider frequencyfirstleadschneider
modulofirstleadschneider moduloDBfirstleadschneider fasefirstleadschneider
                end

        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle LAG Forward
        %************************************************************************

                if (abs(MagDBfirstlagforward)<0.01) &   (
((abs(Phasefirstlagforward)> 179.9) & (abs(Phasefirstlagforward)< 180.1)) |
((abs(Phasefirstlagforward)> 539) & (abs(Phasefirstlagforward)< 541)) |
((abs(Phasefirstlagforward)> 899) & (abs(Phasefirstlagforward)< 901)) |
((abs(Phasefirstlagforward)> 1259) & (abs(Phasefirstlagforward)< 1261)) |
((abs(Phasefirstlagforward)> 1619) & (abs(Phasefirstlagforward)< 1621))  |
((abs(Phasefirstlagforward)> 1979) & (abs(Phasefirstlagforward)< 1981))  |
((abs(Phasefirstlagforward)> 2339) & (abs(Phasefirstlagforward)< 2341))  )
                %Montagem dos vetores armazenando os dados importantes:
                k32=k32+1;
                samplingperiodfirstlagforward(k32)=T;
                frequencyfirstlagforward(k32)=w;
                modulofirstlagforward(k32)=10^(MagDBfirstlagforward/20);
                moduloDBfirstlagforward(k32)=MagDBfirstlagforward;
                fasefirstlagforward(k32)=Phasefirstlagforward;
```

```
                %Grava arquivo formato .MAT:
                save sttheoryfirstlagforward samplingperiodfirstlagforward
frequencyfirstlagforward modulofirstlagforward moduloDBfirstlagforward
fasefirstlagforward
            end
        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle LAG
        % Backward
        %************************************************************************

            if (abs(MagDBfirstlagbackward)<0.01) &   (
((abs(Phasefirstlagbackward)> 179.9) & (abs(Phasefirstlagbackward)< 180.1)) |
((abs(Phasefirstlagbackward)> 539) & (abs(Phasefirstlagbackward)< 541)) |
((abs(Phasefirstlagbackward)> 899) & (abs(Phasefirstlagbackward)< 901)) |
((abs(Phasefirstlagbackward)> 1259) & (abs(Phasefirstlagbackward)< 1261)) |
((abs(Phasefirstlagbackward)> 1619) & (abs(Phasefirstlagbackward)< 1621))  |
((abs(Phasefirstlagbackward)> 1979) & (abs(Phasefirstlagbackward)< 1981))  |
((abs(Phasefirstlagbackward)> 2339) & (abs(Phasefirstlagbackward)< 2341))  )
             %Montagem dos vetores armazenando os dados importantes:
                k33=k33+1;
                samplingperiodfirstlagbackward(k33)=T;
                frequencyfirstlagbackward(k33)=w;

            modulofirstlagbackward(k33)=10^(MagDBfirstlagbackward/20);
                moduloDBfirstlagbackward(k33)=MagDBfirstlagbackward;
                fasefirstlagbackward(k33)=Phasefirstlagbackward;
                %Grava arquivo formato .MAT:
                save sttheoryfirstlagbackward
samplingperiodfirstlagbackward frequencyfirstlagbackward modulofirstlagbackward
moduloDBfirstlagbackward fasefirstlagbackward
            end
```

```
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle LAG Tustin
%**************************************************************************

                if (abs(MagDBfirstlagtustin)<0.01) &   (
((abs(Phasefirstlagtustin)> 179.9) & (abs(Phasefirstlagtustin)< 180.1)) |
((abs(Phasefirstlagtustin)> 539) & (abs(Phasefirstlagtustin)< 541)) |
((abs(Phasefirstlagtustin)> 899) & (abs(Phasefirstlagtustin)< 901)) |
((abs(Phasefirstlagtustin)> 1259) & (abs(Phasefirstlagtustin)< 1261)) |
((abs(Phasefirstlagtustin)> 1619) & (abs(Phasefirstlagtustin)< 1621))  |
((abs(Phasefirstlagtustin)> 1979) & (abs(Phasefirstlagtustin)< 1981))  |
((abs(Phasefirstlagtustin)> 2339) & (abs(Phasefirstlagtustin)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k34=k34+1;
                      samplingperiodfirstlagtustin(k34)=T;
                      frequencyfirstlagtustin(k34)=w;
                      modulofirstlagtustin(k34)=10^(MagDBfirstlagtustin/20);
                      moduloDBfirstlagtustin(k34)=MagDBfirstlagtustin;
                      fasefirstlagtustin(k34)=Phasefirstlagtustin;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstlagtustin samplingperiodfirstlagtustin
frequencyfirstlagtustin modulofirstlagtustin moduloDBfirstlagtustin fasefirstlagtustin
                end
%**************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle LAG ST1
%**************************************************************************

                if (abs(MagDBfirstlagst1)<0.01) &   ( ((abs(Phasefirstlagst1)>
179.9) & (abs(Phasefirstlagst1)< 180.1)) | ((abs(Phasefirstlagst1)> 539) &
(abs(Phasefirstlagst1)< 541)) | ((abs(Phasefirstlagst1)> 899) &
```

```
(abs(Phasefirstlagst1)< 901)) | ((abs(Phasefirstlagst1)> 1259) &
(abs(Phasefirstlagst1)< 1261)) | ((abs(Phasefirstlagst1)> 1619) &
(abs(Phasefirstlagst1)< 1621))  | ((abs(Phasefirstlagst1)> 1979) &
(abs(Phasefirstlagst1)< 1981))  | ((abs(Phasefirstlagst1)> 2339) &
(abs(Phasefirstlagst1)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k35=k35+1;
                      samplingperiodfirstlagst1(k35)=T;
                      frequencyfirstlagst1(k35)=w;
                      modulofirstlagst1(k35)=10^(MagDBfirstlagst1/20);
                      moduloDBfirstlagst1(k35)=MagDBfirstlagst1;
                      fasefirstlagst1(k35)=Phasefirstlagst1;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstlagst1 samplingperiodfirstlagst1
frequencyfirstlagst1 modulofirstlagst1 moduloDBfirstlagst1 fasefirstlagst1
                    end
            %************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle LAG ST2
            %************************************************************************

                    if (abs(MagDBfirstlagst2)<0.01) &   ( ((abs(Phasefirstlagst2)>
179.9) & (abs(Phasefirstlagst2)< 180.1)) | ((abs(Phasefirstlagst2)> 539) &
(abs(Phasefirstlagst2)< 541)) | ((abs(Phasefirstlagst2)> 899) &
(abs(Phasefirstlagst2)< 901)) | ((abs(Phasefirstlagst2)> 1259) &
(abs(Phasefirstlagst2)< 1261)) | ((abs(Phasefirstlagst2)> 1619) &
(abs(Phasefirstlagst2)< 1621))  | ((abs(Phasefirstlagst2)> 1979) &
(abs(Phasefirstlagst2)< 1981))  | ((abs(Phasefirstlagst2)> 2339) &
(abs(Phasefirstlagst2)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k36=k36+1;
                      samplingperiodfirstlagst2(k36)=T;
```

```
                frequencyfirstlagst2(k36)=w;
                modulofirstlagst2(k36)=10^(MagDBfirstlagst2/20);
                moduloDBfirstlagst2(k36)=MagDBfirstlagst2;
                fasefirstlagst2(k36)=Phasefirstlagst2;
                %Grava arquivo formato .MAT:
                save sttheoryfirstlagst2 samplingperiodfirstlagst2
frequencyfirstlagst2 modulofirstlagst2 moduloDBfirstlagst2 fasefirstlagst2
            end
        %************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle LAG
        % Schneider
        %************************************************************************

            if (abs(MagDBfirstlagschneider)<0.01) &   (
((abs(Phasefirstlagschneider)> 179.9) & (abs(Phasefirstlagschneider)< 180.1)) |
((abs(Phasefirstlagschneider)> 539) & (abs(Phasefirstlagschneider)< 541)) |
((abs(Phasefirstlagschneider)> 899) & (abs(Phasefirstlagschneider)< 901)) |
((abs(Phasefirstlagschneider)> 1259) & (abs(Phasefirstlagschneider)< 1261)) |
((abs(Phasefirstlagschneider)> 1619) & (abs(Phasefirstlagschneider)< 1621))  |
((abs(Phasefirstlagschneider)> 1979) & (abs(Phasefirstlagschneider)< 1981))  |
((abs(Phasefirstlagschneider)> 2339) & (abs(Phasefirstlagschneider)< 2341))  )
             %Montagem dos vetores armazenando os dados importantes:
                k37=k37+1;
                samplingperiodfirstlagschneider(k37)=T;
                frequencyfirstlagschneider(k37)=w;

               modulofirstlagschneider(k37)=10^(MagDBfirstlagschneider/20);
                moduloDBfirstlagschneider(k37)=MagDBfirstlagschneider;
                fasefirstlagschneider(k37)=Phasefirstlagschneider;
                %Grava arquivo formato .MAT:
```

```
                save sttheoryfirstlagschneider
samplingperiodfirstlagschneider frequencyfirstlagschneider
modulofirstlagschneider moduloDBfirstlagschneider fasefirstlagschneider
                end

        %*************************************************************************
        % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD.LAG
        % Forward
        %*************************************************************************

                if (abs(MagDBfirstleadlagforward)<0.01) &   (
((abs(Phasefirstleadlagforward)> 179.9) & (abs(Phasefirstleadlagforward)< 180.1))
| ((abs(Phasefirstleadlagforward)> 539) & (abs(Phasefirstleadlagforward)< 541)) |
((abs(Phasefirstleadlagforward)> 899) & (abs(Phasefirstleadlagforward)< 901)) |
((abs(Phasefirstleadlagforward)> 1259) & (abs(Phasefirstleadlagforward)< 1261)) |
((abs(Phasefirstleadlagforward)> 1619) & (abs(Phasefirstleadlagforward)< 1621))  |
((abs(Phasefirstleadlagforward)> 1979) & (abs(Phasefirstleadlagforward)< 1981))  |
((abs(Phasefirstleadlagforward)> 2339) & (abs(Phasefirstleadlagforward)< 2341))
)
                %Montagem dos vetores armazenando os dados importantes:
                  k38=k38+1;
                  samplingperiodfirstleadlagforward(k38)=T;
                  frequencyfirstleadlagforward(k38)=w;

              modulofirstleadlagforward(k38)=10^(MagDBfirstleadlagforward/20);
                   moduloDBfirstleadlagforward(k38)=MagDBfirstleadlagforward;
                  fasefirstleadlagforward(k38)=Phasefirstleadlagforward;
                  %Grava arquivo formato .MAT:
                  save sttheoryfirstleadlagforward
samplingperiodfirstleadlagforward frequencyfirstleadlagforward
modulofirstleadlagforward moduloDBfirstleadlagforward fasefirstleadlagforward
```

```
                end
            %**************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD.LAG
            % Backward
            %**************************************************************************

                    if (abs(MagDBfirstleadlagbackward)<0.01) &   (
((abs(Phasefirstleadlagbackward)> 179.9) & (abs(Phasefirstleadlagbackward)<
180.1)) | ((abs(Phasefirstleadlagbackward)> 539) &
(abs(Phasefirstleadlagbackward)< 541)) | ((abs(Phasefirstleadlagbackward)> 899)
& (abs(Phasefirstleadlagbackward)< 901)) | ((abs(Phasefirstleadlagbackward)>
1259) & (abs(Phasefirstleadlagbackward)< 1261)) |
((abs(Phasefirstleadlagbackward)> 1619) & (abs(Phasefirstleadlagbackward)<
1621))  | ((abs(Phasefirstleadlagbackward)> 1979) &
(abs(Phasefirstleadlagbackward)< 1981))  | ((abs(Phasefirstleadlagbackward)>
2339) & (abs(Phasefirstleadlagbackward)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k39=k39+1;
                      samplingperiodfirstlagbackward(k39)=T;
                      frequencyfirstleadlagbackward(k39)=w;

            modulofirstleadlagbackward(k39)=10^(MagDBfirstleadlagbackward/20);
                  moduloDBfirstleadlagbackward(k39)=MagDBfirstleadlagbackward;
                      fasefirstleadlagbackward(k38)=Phasefirstleadlagbackward;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstleadlagbackward
samplingperiodfirstleadlagbackward frequencyfirstleadlagbackward
modulofirstleadlagbackward moduloDBfirstleadlagbackward
fasefirstleadlagbackward
                    end
```

```
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD.LAG
% Tustin
%************************************************************************

                if (abs(MagDBfirstleadlagtustin)<0.01) &   (
((abs(Phasefirstleadlagtustin)> 179.9) & (abs(Phasefirstleadlagtustin)< 180.1)) |
((abs(Phasefirstleadlagtustin)> 539) & (abs(Phasefirstleadlagtustin)< 541)) |
((abs(Phasefirstleadlagtustin)> 899) & (abs(Phasefirstleadlagtustin)< 901)) |
((abs(Phasefirstleadlagtustin)> 1259) & (abs(Phasefirstleadlagtustin)< 1261)) |
((abs(Phasefirstleadlagtustin)> 1619) & (abs(Phasefirstleadlagtustin)< 1621))  |
((abs(Phasefirstleadlagtustin)> 1979) & (abs(Phasefirstleadlagtustin)< 1981))  |
((abs(Phasefirstleadlagtustin)> 2339) & (abs(Phasefirstleadlagtustin)< 2341))  )
                 %Montagem dos vetores armazenando os dados importantes:
                   k40=k40+1;
                   samplingperiodfirstleadlagtustin(k40)=T;
                   frequencyfirstleadlagtustin(k40)=w;

                 modulofirstleadlagtustin(k40)=10^(MagDBfirstleadlagtustin/20);
                   moduloDBfirstleadlagtustin(k40)=MagDBfirstleadlagtustin;
                   fasefirstleadlagtustin(k40)=Phasefirstleadlagtustin;
                   %Grava arquivo formato .MAT:
                   save sttheoryfirstleadlagtustin
samplingperiodfirstleadlagtustin frequencyfirstleadlagtustin modulofirstleadlagtustin
moduloDBfirstleadlagtustin fasefirstleadlagtustin
                end
%************************************************************************
% H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD.LAG
% ST1
%************************************************************************
```

```
                    if (abs(MagDBfirstleadlagst1)<0.01) &   (
((abs(Phasefirstleadlagst1)> 179.9) & (abs(Phasefirstleadlagst1)< 180.1)) |
((abs(Phasefirstleadlagst1)> 539) & (abs(Phasefirstleadlagst1)< 541)) |
((abs(Phasefirstleadlagst1)> 899) & (abs(Phasefirstleadlagst1)< 901)) |
((abs(Phasefirstleadlagst1)> 1259) & (abs(Phasefirstleadlagst1)< 1261)) |
((abs(Phasefirstleadlagst1)> 1619) & (abs(Phasefirstleadlagst1)< 1621))  |
((abs(Phasefirstleadlagst1)> 1979) & (abs(Phasefirstleadlagst1)< 1981))  |
((abs(Phasefirstleadlagst1)> 2339) & (abs(Phasefirstleadlagst1)< 2341))  )
                     %Montagem dos vetores armazenando os dados importantes:
                       k41=k41+1;
                       samplingperiodfirstleadlagst1(k41)=T;
                       frequencyfirstleadlagst1(k41)=w;
                       modulofirstleadlagst1(k41)=10^(MagDBfirstleadlagst1/20);
                       moduloDBfirstleadlagst1(k41)=MagDBfirstleadlagst1;
                       fasefirstleadlagst1(k41)=Phasefirstleadlagst1;
                       %Grava arquivo formato .MAT:
                       save sttheoryfirstleadlagst1 samplingperiodfirstleadlagst1
frequencyfirstleadlagst1 modulofirstleadlagst1 moduloDBfirstleadlagst1
fasefirstleadlagst1
                    end
               %************************************************************************
               % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD.LAG
               % ST2
               %************************************************************************

                    if (abs(MagDBfirstleadlagst2)<0.01) &   (
((abs(Phasefirstleadlagst2)> 179.9) & (abs(Phasefirstleadlagst2)< 180.1)) |
((abs(Phasefirstleadlagst2)> 539) & (abs(Phasefirstleadlagst2)< 541)) |
((abs(Phasefirstleadlagst2)> 899) & (abs(Phasefirstleadlagst2)< 901)) |
((abs(Phasefirstleadlagst2)> 1259) & (abs(Phasefirstleadlagst2)< 1261)) |
((abs(Phasefirstleadlagst2)> 1619) & (abs(Phasefirstleadlagst2)< 1621))  |
```

```
((abs(Phasefirstleadlagst2)> 1979) & (abs(Phasefirstleadlagst2)< 1981))  |
((abs(Phasefirstleadlagst2)> 2339) & (abs(Phasefirstleadlagst2)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k42=k42+1;
                      samplingperiodfirstleadlagst2(k42)=T;
                      frequencyfirstleadlagst2(k42)=w;
                      modulofirstleadlagst2(k42)=10^(MagDBfirstleadlagst2/20);
                      moduloDBfirstleadlagst2(k42)=MagDBfirstleadlagst2;
                      fasefirstleadlagst2(k42)=Phasefirstleadlagst2;
                      %Grava arquivo formato .MAT:
                      save sttheoryfirstleadlagst2 samplingperiodfirstleadlagst2
frequencyfirstleadlagst2 modulofirstleadlagst2 moduloDBfirstleadlagst2
fasefirstleadlagst2
                  end
            %**************************************************************************
            % H*(s) para Planta de 1a. Ordem (com hold) e controle LEAD.LAG
            % Schneider
            %**************************************************************************

                  if (abs(MagDBfirstleadlagschneider)<0.01) &   (
((abs(Phasefirstleadlagschneider)> 179.9) & (abs(Phasefirstleadlagschneider)<
180.1)) | ((abs(Phasefirstleadlagschneider)> 539) &
(abs(Phasefirstleadlagschneider)< 541)) | ((abs(Phasefirstleadlagschneider)> 899)
& (abs(Phasefirstleadlagschneider)< 901)) | ((abs(Phasefirstleadlagschneider)>
1259) & (abs(Phasefirstleadlagschneider)< 1261)) |
((abs(Phasefirstleadlagschneider)> 1619) & (abs(Phasefirstleadlagschneider)<
1621))  | ((abs(Phasefirstleadlagschneider)> 1979) &
(abs(Phasefirstleadlagschneider)< 1981))  | ((abs(Phasefirstleadlagschneider)>
2339) & (abs(Phasefirstleadlagschneider)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k43=k43+1;
```

```
                    samplingperiodfirstleadlagschneider(k43)=T;
                    frequencyfirstleadlagschneider(k43)=w;

          modulofirstleadlagschneider(k43)=10^(MagDBfirstleadlagschneider/20);
               moduloDBfirstleadlagschneider(k43)=MagDBfirstleadlagschneider;
                    fasefirstleadlagschneider(k43)=Phasefirstleadlagschneider;
                    %Grava arquivo formato .MAT:
                    save sttheoryfirstleadlagschneider
samplingperiodfirstleadlagschneider frequencyfirstleadlagschneider
modulofirstleadlagschneider moduloDBfirstleadlagschneider
fasefirstleadlagschneider
                    end

%*********************************************
%*********************************************
%
%       PLANTA DE SEGUNDA ORDEM
%
%*********************************************
%*********************************************

          %************************************************************************
          % H*(s) para Planta de 2a. Ordem (com hold) e controle Proporcional
          %************************************************************************

                    if (abs(MagDBsecondproporcional)<0.01) &   (
((abs(Phasesecondproporcional)> 179.9) & (abs(Phasesecondproporcional)<
180.1)) | ((abs(Phasesecondproporcional)> 539) &
(abs(Phasesecondproporcional)< 541)) | ((abs(Phasesecondproporcional)> 899) &
(abs(Phasesecondproporcional)< 901)) | ((abs(Phasesecondproporcional)> 1259)
& (abs(Phasesecondproporcional)< 1261)) | ((abs(Phasesecondproporcional)>
```

```
1619) & (abs(Phasesecondproporcional)< 1621))  |
((abs(Phasesecondproporcional)> 1979) & (abs(Phasesecondproporcional)<
1981))  | ((abs(Phasesecondproporcional)> 2339) &
(abs(Phasesecondproporcional)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k44=k44+1;
                          samplingperiodsecondproporcional(k44)=T;
                          frequencysecondproporcional(k44)=w;

                  modulosecondproporcional(k44)=10^(MagDBsecondproporcional/20);
                       moduloDBsecondproporcional(k44)=MagDBsecondproporcional;
                          fasesecondproporcional(k44)=Phasesecondproporcional;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondproporcional
samplingperiodsecondproporcional frequencysecondproporcional
modulosecondproporcional moduloDBsecondproporcional fasesecondproporcional
                             end
                  %**************************************************************************
                  % H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
                  % Forward
                  %**************************************************************************

                          if (abs(MagDBsecondderivativoforward)<0.01) &   (
((abs(Phasesecondderivativoforward)> 179.9) &
(abs(Phasesecondderivativoforward)< 180.1)) |
((abs(Phasesecondderivativoforward)> 539) &
(abs(Phasesecondderivativoforward)< 541)) |
((abs(Phasesecondderivativoforward)> 899) &
(abs(Phasesecondderivativoforward)< 901)) |
((abs(Phasesecondderivativoforward)> 1259) &
(abs(Phasesecondderivativoforward)< 1261)) |
```

```
((abs(Phasesecondderivativoforward)> 1619) &
(abs(Phasesecondderivativoforward)< 1621))  |
((abs(Phasesecondderivativoforward)> 1979) &
(abs(Phasesecondderivativoforward)< 1981))  |
((abs(Phasesecondderivativoforward)> 2339) &
(abs(Phasesecondderivativoforward)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k45=k45+1;
                          samplingperiodsecondderivativoforward(k45)=T;
                          frequencysecondderivativoforward(k45)=w;

modulosecondderivativoforward(k45)=10^(MagDBsecondderivativoforward/20);
moduloDBsecondderivativoforward(k45)=MagDBsecondderivativoforward;
fasesecondderivativoforward(k45)=Phasesecondderivativoforward;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondderivativoforward
samplingperiodsecondderivativoforward frequencysecondderivativoforward
modulosecondderivativoforward moduloDBsecondderivativoforward
fasesecondderivativoforward
                        end
                %************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
                % Backward
                %************************************************************************

                        if (abs(MagDBsecondderivativobackward)<0.01) &   (
((abs(Phasesecondderivativobackward)> 179.9) &
(abs(Phasesecondderivativobackward)< 180.1)) |
((abs(Phasesecondderivativobackward)> 539) &
(abs(Phasesecondderivativobackward)< 541)) |
((abs(Phasesecondderivativobackward)> 899) &
```

```
(abs(Phasesecondderivativobackward)< 901)) |
((abs(Phasesecondderivativobackward)> 1259) &
(abs(Phasesecondderivativobackward)< 1261)) |
((abs(Phasesecondderivativobackward)> 1619) &
(abs(Phasesecondderivativobackward)< 1621))  |
((abs(Phasesecondderivativobackward)> 1979) &
(abs(Phasesecondderivativobackward)< 1981))  |
((abs(Phasesecondderivativobackward)> 2339) &
(abs(Phasesecondderivativobackward)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k46=k46+1;
                          samplingperiodsecondderivativobackward(k46)=T;
                          frequencysecondderivativobackward(k46)=w;

modulosecondderivativobackward(k46)=10^(MagDBsecondderivativobackward/20);
moduloDBsecondderivativobackward(k46)=MagDBsecondderivativobackward;

fasesecondderivativobackward(k46)=Phasesecondderivativobackward;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondderivativobackward
samplingperiodsecondderivativobackward frequencysecondderivativobackward
modulosecondderivativobackward moduloDBsecondderivativobackward
fasesecondderivativobackward
                        end
                %**************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
                % Tustin
                %**************************************************************************

                        if (abs(MagDBsecondderivativotustin)<0.01) &   (
((abs(Phasesecondderivativotustin)> 179.9) & (abs(Phasesecondderivativotustin)<
```

```
180.1)) | ((abs(Phasesecondderivativotustin)> 539) &
(abs(Phasesecondderivativotustin)< 541)) | ((abs(Phasesecondderivativotustin)>
899) & (abs(Phasesecondderivativotustin)< 901)) |
((abs(Phasesecondderivativotustin)> 1259) & (abs(Phasesecondderivativotustin)<
1261)) | ((abs(Phasesecondderivativotustin)> 1619) &
(abs(Phasesecondderivativotustin)< 1621))  | ((abs(Phasesecondderivativotustin)>
1979) & (abs(Phasesecondderivativotustin)< 1981))  |
((abs(Phasesecondderivativotustin)> 2339) & (abs(Phasesecondderivativotustin)<
2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k47=k47+1;
                          samplingperiodsecondderivativotustin(k47)=T;
                          frequencysecondderivativotustin(k47)=w;

modulosecondderivativotustin(k47)=10^(MagDBsecondderivativotustin/20);
moduloDBsecondderivativotustin(k47)=MagDBsecondderivativotustin;
fasesecondderivativotustin(k47)=Phasesecondderivativotustin;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondderivativotustin
samplingperiodsecondderivativotustin frequencysecondderivativotustin
modulosecondderivativotustin moduloDBsecondderivativotustin
fasesecondderivativotustin
                        end
                %**************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
                % ST1
                %**************************************************************************

                        if (abs(MagDBsecondderivativost1)<0.01) &   (
((abs(Phasesecondderivativost1)> 179.9) & (abs(Phasesecondderivativost1)<
180.1)) | ((abs(Phasesecondderivativost1)> 539) &
```

```
(abs(Phasesecondderivativost1)< 541)) | ((abs(Phasesecondderivativost1)> 899) &
(abs(Phasesecondderivativost1)< 901)) | ((abs(Phasesecondderivativost1)> 1259)
& (abs(Phasesecondderivativost1)< 1261)) | ((abs(Phasesecondderivativost1)>
1619) & (abs(Phasesecondderivativost1)< 1621))  |
((abs(Phasesecondderivativost1)> 1979) & (abs(Phasesecondderivativost1)<
1981))  | ((abs(Phasesecondderivativost1)> 2339) &
(abs(Phasesecondderivativost1)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k48=k48+1;
                          samplingperiodsecondderivativost1(k48)=T;
                          frequencysecondderivativost1(k48)=w;

modulosecondderivativost1(k48)=10^(MagDBsecondderivativost1/20);
moduloDBsecondderivativost1(k48)=MagDBsecondderivativost1;
                          fasesecondderivativost1(k48)=Phasesecondderivativost1;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondderivativost1
samplingperiodsecondderivativost1 frequencysecondderivativost1
modulosecondderivativost1 moduloDBsecondderivativost1
fasesecondderivativost1
                        end
                %**************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
                % ST2
                %**************************************************************************

                        if (abs(MagDBsecondderivativost2)<0.01) &   (
((abs(Phasesecondderivativost2)> 179.9) & (abs(Phasesecondderivativost2)<
180.1)) | ((abs(Phasesecondderivativost2)> 539) &
(abs(Phasesecondderivativost2)< 541)) | ((abs(Phasesecondderivativost2)> 899) &
(abs(Phasesecondderivativost2)< 901)) | ((abs(Phasesecondderivativost2)> 1259)
```

```
& (abs(Phasesecondderivativost2)< 1261)) | ((abs(Phasesecondderivativost2)>
1619) & (abs(Phasesecondderivativost2)< 1621))  |
((abs(Phasesecondderivativost2)> 1979) & (abs(Phasesecondderivativost2)<
1981))  | ((abs(Phasesecondderivativost2)> 2339) &
(abs(Phasesecondderivativost2)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k49=k49+1;
                          samplingperiodsecondderivativost2(k49)=T;
                          frequencysecondderivativost2(k49)=w;

modulosecondderivativost2(k49)=10^(MagDBsecondderivativost2/20);
moduloDBsecondderivativost2(k49)=MagDBsecondderivativost2;
                          fasesecondderivativost2(k49)=Phasesecondderivativost2;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondderivativost2
samplingperiodsecondderivativost2 frequencysecondderivativost2
modulosecondderivativost2 moduloDBsecondderivativost2
fasesecondderivativost2
                        end
                %************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle Derivativo
                % Schneider
                %************************************************************************

                        if (abs(MagDBsecondderivativoschneider)<0.01) &   (
((abs(Phasesecondderivativoschneider)> 179.9) &
(abs(Phasesecondderivativoschneider)< 180.1)) |
((abs(Phasesecondderivativoschneider)> 539) &
(abs(Phasesecondderivativoschneider)< 541)) |
((abs(Phasesecondderivativoschneider)> 899) &
(abs(Phasesecondderivativoschneider)< 901)) |
```

```
((abs(Phasesecondderivativoschneider)> 1259) &
(abs(Phasesecondderivativoschneider)< 1261)) |
((abs(Phasesecondderivativoschneider)> 1619) &
(abs(Phasesecondderivativoschneider)< 1621))  |
((abs(Phasesecondderivativoschneider)> 1979) &
(abs(Phasesecondderivativoschneider)< 1981))  |
((abs(Phasesecondderivativoschneider)> 2339) &
(abs(Phasesecondderivativoschneider)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k50=k50+1;
                          samplingperiodsecondderivativoschneider(k50)=T;
                          frequencysecondderivativoschneider(k50)=w;

modulosecondderivativoschneider(k50)=10^(MagDBsecondderivativoschneider/20);
moduloDBsecondderivativoschneider(k50)=MagDBsecondderivativoschneider;
fasesecondderivativoschneider(k50)=Phasesecondderivativoschneider;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondderivativoschneider
samplingperiodsecondderivativoschneider frequencysecondderivativoschneider
modulosecondderivativoschneider moduloDBsecondderivativoschneider
fasesecondderivativoschneider
                        end

                %**************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
                % Forward
                %**************************************************************************

                        if (abs(MagDBsecondintegralforward)<0.01) &   (
((abs(Phasesecondintegralforward)> 179.9) & (abs(Phasesecondintegralforward)<
180.1)) | ((abs(Phasesecondintegralforward)> 539) &
```

```
(abs(Phasesecondintegralforward)< 541)) | ((abs(Phasesecondintegralforward)>
899) & (abs(Phasesecondintegralforward)< 901)) |
((abs(Phasesecondintegralforward)> 1259) & (abs(Phasesecondintegralforward)<
1261)) | ((abs(Phasesecondintegralforward)> 1619) &
(abs(Phasesecondintegralforward)< 1621))  | ((abs(Phasesecondintegralforward)>
1979) & (abs(Phasesecondintegralforward)< 1981))  |
((abs(Phasesecondintegralforward)> 2339) & (abs(Phasesecondintegralforward)<
2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k51=k51+1;
                          samplingperiodsecondintegralforward(k51)=T;
                          frequencysecondintegralforward(k51)=w;

modulosecondintegralforward(k51)=10^(MagDBsecondintegralforward/20);
moduloDBsecondintegralforward(k51)=MagDBsecondintegralforward;
fasesecondintegralforward(k51)=Phasesecondintegralforward;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondintegralforward
samplingperiodsecondintegralforward frequencysecondintegralforward
modulosecondintegralforward moduloDBsecondintegralforward
fasesecondintegralforward
                    end
               %**************************************************************************
               % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
               % Backward
               %**************************************************************************

                    if (abs(MagDBsecondintegralbackward)<0.01) &   (
((abs(Phasesecondintegralbackward)> 179.9) &
(abs(Phasesecondintegralbackward)< 180.1)) |
((abs(Phasesecondintegralbackward)> 539) &
```

```
(abs(Phasesecondintegralbackward)< 541)) |
((abs(Phasesecondintegralbackward)> 899) &
(abs(Phasesecondintegralbackward)< 901)) |
((abs(Phasesecondintegralbackward)> 1259) &
(abs(Phasesecondintegralbackward)< 1261)) |
((abs(Phasesecondintegralbackward)> 1619) &
(abs(Phasesecondintegralbackward)< 1621))  |
((abs(Phasesecondintegralbackward)> 1979) &
(abs(Phasesecondintegralbackward)< 1981))  |
((abs(Phasesecondintegralbackward)> 2339) &
(abs(Phasesecondintegralbackward)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k52=k52+1;
                      samplingperiodsecondintegralbackward(k52)=T;
                      frequencysecondintegralbackward(k52)=w;

modulosecondintegralbackward(k52)=10^(MagDBsecondintegralbackward/20);
moduloDBsecondintegralbackward(52)=MagDBsecondintegralbackward;
fasesecondintegralbackward(52)=Phasesecondintegralbackward;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondintegralbackward
samplingperiodsecondintegralbackward frequencysecondintegralbackward
modulosecondintegralbackward moduloDBsecondintegralbackward
fasesecondintegralbackward
                        end
              %**************************************************************************
              % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
              % Tustin
              %**************************************************************************
```

```
                    if (abs(MagDBsecondintegraltustin)<0.01) &   (
((abs(Phasesecondintegraltustin)> 179.9) & (abs(Phasesecondintegraltustin)<
180.1)) | ((abs(Phasesecondintegraltustin)> 539) &
(abs(Phasesecondintegraltustin)< 541)) | ((abs(Phasesecondintegraltustin)> 899)
& (abs(Phasesecondintegraltustin)< 901)) | ((abs(Phasesecondintegraltustin)>
1259) & (abs(Phasesecondintegraltustin)< 1261)) |
((abs(Phasesecondintegraltustin)> 1619) & (abs(Phasesecondintegraltustin)<
1621))  | ((abs(Phasesecondintegraltustin)> 1979) &
(abs(Phasesecondintegraltustin)< 1981))  | ((abs(Phasesecondintegraltustin)>
2339) & (abs(Phasesecondintegraltustin)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k53=k53+1;
                      samplingperiodsecondintegraltustin(k53)=T;
                      frequencysecondintegraltustin(k53)=w;

modulosecondintegraltustin(k53)=10^(MagDBsecondintegraltustin/20);
moduloDBsecondintegraltustin(k53)=MagDBsecondintegraltustin;
                      fasesecondintegraltustin(k53)=Phasesecondintegraltustin;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondintegraltustin
samplingperiodsecondintegraltustin frequencysecondintegraltustin
modulosecondintegraltustin moduloDBsecondintegraltustin
fasesecondintegraltustin
                    end
          %************************************************************************
          % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral ST1
          %************************************************************************

                    if (abs(MagDBsecondintegralst1)<0.01) &   (
((abs(Phasesecondintegralst1)> 179.9) & (abs(Phasesecondintegralst1)< 180.1)) |
((abs(Phasesecondintegralst1)> 539) & (abs(Phasesecondintegralst1)< 541)) |
```

```
((abs(Phasesecondintegralst1)> 899) & (abs(Phasesecondintegralst1)< 901)) |
((abs(Phasesecondintegralst1)> 1259) & (abs(Phasesecondintegralst1)< 1261)) |
((abs(Phasesecondintegralst1)> 1619) & (abs(Phasesecondintegralst1)< 1621))  |
((abs(Phasesecondintegralst1)> 1979) & (abs(Phasesecondintegralst1)< 1981))  |
((abs(Phasesecondintegralst1)> 2339) & (abs(Phasesecondintegralst1)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k54=k54+1;
                      samplingperiodsecondintegralst1(k54)=T;
                      frequencysecondintegralst1(k54)=w;

                  modulosecondintegralst1(k54)=10^(MagDBsecondintegralst1/20);
                      moduloDBsecondintegralst1(k54)=MagDBsecondintegralst1;
                      fasesecondintegralst1(k54)=Phasesecondintegralst1;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondintegralst1
samplingperiodsecondintegralst1 frequencysecondintegralst1
modulosecondintegralst1 moduloDBsecondintegralst1 fasesecondintegralst1
                    end
            %************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral ST2
            %************************************************************************

                    if (abs(MagDBsecondintegralst2)<0.01) &   (
((abs(Phasesecondintegralst2)> 179.9) & (abs(Phasesecondintegralst2)< 180.1)) |
((abs(Phasesecondintegralst2)> 539) & (abs(Phasesecondintegralst2)< 541)) |
((abs(Phasesecondintegralst2)> 899) & (abs(Phasesecondintegralst2)< 901)) |
((abs(Phasesecondintegralst2)> 1259) & (abs(Phasesecondintegralst2)< 1261)) |
((abs(Phasesecondintegralst2)> 1619) & (abs(Phasesecondintegralst2)< 1621))  |
((abs(Phasesecondintegralst2)> 1979) & (abs(Phasesecondintegralst2)< 1981))  |
((abs(Phasesecondintegralst2)> 2339) & (abs(Phasesecondintegralst2)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
```

```
                k55=k55+1;
                samplingperiodsecondintegralst2(k55)=T;
                frequencysecondintegralst2(k55)=w;

            modulosecondintegralst2(k55)=10^(MagDBsecondintegralst2/20);
                moduloDBsecondintegralst2(k55)=MagDBsecondintegralst2;
                fasesecondintegralst2(k55)=Phasesecondintegralst2;
                %Grava arquivo formato .MAT:
                save sttheorysecondintegralst2
samplingperiodsecondintegralst2 frequencysecondintegralst2
modulosecondintegralst2 moduloDBsecondintegralst2 fasesecondintegralst2
                end
        %*************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle Integral
        % Schneider
        %*************************************************************************

                if (abs(MagDBsecondintegralschneider)<0.01) &   (
((abs(Phasesecondintegralschneider)> 179.9) &
(abs(Phasesecondintegralschneider)< 180.1)) |
((abs(Phasesecondintegralschneider)> 539) &
(abs(Phasesecondintegralschneider)< 541)) |
((abs(Phasesecondintegralschneider)> 899) &
(abs(Phasesecondintegralschneider)< 901)) |
((abs(Phasesecondintegralschneider)> 1259) &
(abs(Phasesecondintegralschneider)< 1261)) |
((abs(Phasesecondintegralschneider)> 1619) &
(abs(Phasesecondintegralschneider)< 1621))  |
((abs(Phasesecondintegralschneider)> 1979) &
(abs(Phasesecondintegralschneider)< 1981))  |
```

```
((abs(Phasesecondintegralschneider)> 2339) &
(abs(Phasesecondintegralschneider)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k56=k56+1;
                      samplingperiodsecondintegralschneider(k56)=T;
                      frequencysecondintegralschneider(k56)=w;

modulosecondintegralschneider(k56)=10^(MagDBsecondintegralschneider/20);
moduloDBsecondintegralschneider(k56)=MagDBsecondintegralschneider;
fasesecondintegralschneider(k56)=Phasesecondintegralschneider;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondintegralschneider
samplingperiodsecondintegralschneider frequencysecondintegralschneider
modulosecondintegralschneider moduloDBsecondintegralschneider
fasesecondintegralschneider
                    end

          %************************************************************************
          % H*(s) para Planta de 2a. Ordem (com hold) e controle PD Forward
          %************************************************************************

                    if (abs(MagDBsecondpdforward)<0.01) &   (
((abs(Phasesecondpdforward)> 179.9) & (abs(Phasesecondpdforward)< 180.1)) |
((abs(Phasesecondpdforward)> 539) & (abs(Phasesecondpdforward)< 541)) |
((abs(Phasesecondpdforward)> 899) & (abs(Phasesecondpdforward)< 901)) |
((abs(Phasesecondpdforward)> 1259) & (abs(Phasesecondpdforward)< 1261)) |
((abs(Phasesecondpdforward)> 1619) & (abs(Phasesecondpdforward)< 1621))  |
((abs(Phasesecondpdforward)> 1979) & (abs(Phasesecondpdforward)< 1981))  |
((abs(Phasesecondpdforward)> 2339) & (abs(Phasesecondpdforward)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k57=k57+1;
```

```
                samplingperiodsecondpdforward(k57)=T;
                frequencysecondpdforward(k57)=w;

            modulosecondpdforward(k57)=10^(MagDBsecondpdforward/20);
                moduloDBsecondpdforward(k57)=MagDBsecondpdforward;
                fasesecondpdforward(k57)=Phasesecondpdforward;
                %Grava arquivo formato .MAT:
                save sttheorysecondpdforward
samplingperiodsecondpdforward frequencysecondpdforward
modulosecondpdforward moduloDBsecondpdforward fasesecondpdforward
                end
        %************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle PD Backward
        %************************************************************************

                if (abs(MagDBsecondpdbackward)<0.01) &   (
((abs(Phasesecondpdbackward)> 179.9) & (abs(Phasesecondpdbackward)<
180.1)) | ((abs(Phasesecondpdbackward)> 539) &
(abs(Phasesecondpdbackward)< 541)) | ((abs(Phasesecondpdbackward)> 899) &
(abs(Phasesecondpdbackward)< 901)) | ((abs(Phasesecondpdbackward)> 1259)
& (abs(Phasesecondpdbackward)< 1261)) | ((abs(Phasesecondpdbackward)>
1619) & (abs(Phasesecondpdbackward)< 1621))  |
((abs(Phasesecondpdbackward)> 1979) & (abs(Phasesecondpdbackward)<
1981))  | ((abs(Phasesecondpdbackward)> 2339) &
(abs(Phasesecondpdbackward)< 2341))  )
                %Montagem dos vetores armazenando os dados importantes:
                k58=k58+1;
                samplingperiodsecondpdbackward(k58)=T;
                frequencysecondpdbackward(k58)=w;

            modulosecondpdbackward(k58)=10^(MagDBsecondpdbackward/20);
```

```
moduloDBsecondpdbackward(k58)=MagDBsecondpdbackward;
                    fasesecondpdbackward(k58)=Phasesecondpdbackward;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondpdbackward
samplingperiodsecondpdbackward frequencysecondpdbackward
modulosecondpdbackward moduloDBsecondpdbackward fasesecondpdbackward
                end
          %************************************************************************
          % H*(s) para Planta de 2a. Ordem (com hold) e controle PD Tustin
          %************************************************************************

                if (abs(MagDBsecondpdtustin)<0.01) &   (
((abs(Phasesecondpdtustin)> 179.9) & (abs(Phasesecondpdtustin)< 180.1)) |
((abs(Phasesecondpdtustin)> 539) & (abs(Phasesecondpdtustin)< 541)) |
((abs(Phasesecondpdtustin)> 899) & (abs(Phasesecondpdtustin)< 901)) |
((abs(Phasesecondpdtustin)> 1259) & (abs(Phasesecondpdtustin)< 1261)) |
((abs(Phasesecondpdtustin)> 1619) & (abs(Phasesecondpdtustin)< 1621))  |
((abs(Phasesecondpdtustin)> 1979) & (abs(Phasesecondpdtustin)< 1981))  |
((abs(Phasesecondpdtustin)> 2339) & (abs(Phasesecondpdtustin)< 2341))  )
                %Montagem dos vetores armazenando os dados importantes:
                  k59=k59+1;
                  samplingperiodsecondpdtustin(k59)=T;
                  frequencysecondpdtustin(k59)=w;

                  modulosecondpdtustin(k59)=10^(MagDBsecondpdtustin/20);
                  moduloDBsecondpdtustin(k59)=MagDBsecondpdtustin;
                  fasesecondpdtustin(k59)=Phasesecondpdtustin;
                  %Grava arquivo formato .MAT:
```

```
                save sttheorysecondpdtustin samplingperiodsecondpdtustin
frequencysecondpdtustin modulosecondpdtustin moduloDBsecondpdtustin
fasesecondpdtustin
                end
            %************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle PD ST1
            %************************************************************************

                if (abs(MagDBsecondpdst1)<0.01) &   (
((abs(Phasesecondpdst1)> 179.9) & (abs(Phasesecondpdst1)< 180.1)) |
((abs(Phasesecondpdst1)> 539) & (abs(Phasesecondpdst1)< 541)) |
((abs(Phasesecondpdst1)> 899) & (abs(Phasesecondpdst1)< 901)) |
((abs(Phasesecondpdst1)> 1259) & (abs(Phasesecondpdst1)< 1261)) |
((abs(Phasesecondpdst1)> 1619) & (abs(Phasesecondpdst1)< 1621))  |
((abs(Phasesecondpdst1)> 1979) & (abs(Phasesecondpdst1)< 1981))  |
((abs(Phasesecondpdst1)> 2339) & (abs(Phasesecondpdst1)< 2341))  )
                %Montagem dos vetores armazenando os dados importantes:
                  k60=k60+1;
                  samplingperiodsecondpdst1(k60)=T;
                  frequencysecondpdst1(k60)=w;
                  modulosecondpdst1(k60)=10^(MagDBsecondpdst1/20);
                  moduloDBsecondpdst1(k60)=MagDBsecondpdst1;
                  fasesecondpdst1(k60)=Phasesecondpdst1;
                  %Grava arquivo formato .MAT:
                  save sttheorysecondpdst1 samplingperiodsecondpdst1
frequencysecondpdst1 modulosecondpdst1 moduloDBsecondpdst1
fasesecondpdst1
                end
            %************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle PD ST2
            %************************************************************************
```

```
                    if (abs(MagDBsecondpdst2)<0.01) &   (
((abs(Phasesecondpdst2)> 179.9) & (abs(Phasesecondpdst2)< 180.1)) |
((abs(Phasesecondpdst2)> 539) & (abs(Phasesecondpdst2)< 541)) |
((abs(Phasesecondpdst2)> 899) & (abs(Phasesecondpdst2)< 901)) |
((abs(Phasesecondpdst2)> 1259) & (abs(Phasesecondpdst2)< 1261)) |
((abs(Phasesecondpdst2)> 1619) & (abs(Phasesecondpdst2)< 1621))  |
((abs(Phasesecondpdst2)> 1979) & (abs(Phasesecondpdst2)< 1981))  |
((abs(Phasesecondpdst2)> 2339) & (abs(Phasesecondpdst2)< 2341))  )
                     %Montagem dos vetores armazenando os dados importantes:
                       k61=k61+1;
                       samplingperiodsecondpdst2(k61)=T;
                       frequencysecondpdst2(k61)=w;
                       modulosecondpdst2(k61)=10^(MagDBsecondpdst2/20);
                       moduloDBsecondpdst2(k61)=MagDBsecondpdst2;
                       fasesecondpdst2(k61)=Phasesecondpdst2;
                       %Grava arquivo formato .MAT:
                       save sttheorysecondpdst2 samplingperiodsecondpdst2
frequencysecondpdst2 modulosecondpdst2 moduloDBsecondpdst2
fasesecondpdst2
                    end
               %************************************************************************
               % H*(s) para Planta de 2a. Ordem (com hold) e controle PD
               % Schneider
               %************************************************************************

                    if (abs(MagDBsecondpdschneider)<0.01) &   (
((abs(Phasesecondpdschneider)> 179.9) & (abs(Phasesecondpdschneider)<
180.1)) | ((abs(Phasesecondpdschneider)> 539) &
(abs(Phasesecondpdschneider)< 541)) | ((abs(Phasesecondpdschneider)> 899) &
(abs(Phasesecondpdschneider)< 901)) | ((abs(Phasesecondpdschneider)> 1259)
```

```
& (abs(Phasesecondpdschneider)< 1261)) | ((abs(Phasesecondpdschneider)>
1619) & (abs(Phasesecondpdschneider)< 1621))  |
((abs(Phasesecondpdschneider)> 1979) & (abs(Phasesecondpdschneider)<
1981))  | ((abs(Phasesecondpdschneider)> 2339) &
(abs(Phasesecondpdschneider)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k62=k62+1;
                          samplingperiodsecondpdschneider(k62)=T;
                          frequencysecondpdschneider(k62)=w;

modulosecondpdschneider(k62)=10^(MagDBsecondpdschneider/20);
moduloDBsecondpdschneider(k62)=MagDBsecondpdschneider;
                          fasesecondpdschneider(k62)=Phasesecondpdschneider;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondpdschneider
samplingperiodsecondpdschneider frequencysecondpdschneider
modulosecondpdschneider moduloDBsecondpdschneider fasesecondpdschneider
                        end

                %************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle PID Forward
                %************************************************************************

                        if (abs(MagDBsecondpidforward)<0.01) &   (
((abs(Phasesecondpidforward)> 179.9) & (abs(Phasesecondpidforward)< 180.1)) |
((abs(Phasesecondpidforward)> 539) & (abs(Phasesecondpidforward)< 541)) |
((abs(Phasesecondpidforward)> 899) & (abs(Phasesecondpidforward)< 901)) |
((abs(Phasesecondpidforward)> 1259) & (abs(Phasesecondpidforward)< 1261)) |
((abs(Phasesecondpidforward)> 1619) & (abs(Phasesecondpidforward)< 1621))  |
((abs(Phasesecondpidforward)> 1979) & (abs(Phasesecondpidforward)< 1981))  |
((abs(Phasesecondpidforward)> 2339) & (abs(Phasesecondpidforward)< 2341))  )
```

```
                    %Montagem dos vetores armazenando os dados importantes:
                      k63=k63+1;
                      samplingperiodsecondpidforward(k63)=T;
                      frequencysecondpidforward(k63)=w;

modulosecondpidforward(k63)=10^(MagDBsecondpidforward/20);
moduloDBsecondpidforward(k63)=MagDBsecondpidforward;
                      fasesecondpidforward(k63)=Phasesecondpidforward;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondpidforward
samplingperiodsecondpidforward frequencysecondpidforward
modulosecondpidforward moduloDBsecondpidforward fasesecondpidforward
                  end
            %*************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle PID
            % Backward
            %*************************************************************************

                  if (abs(MagDBsecondpidbackward)<0.01) &   (
((abs(Phasesecondpidbackward)> 179.9) & (abs(Phasesecondpidbackward)<
180.1)) | ((abs(Phasesecondpidbackward)> 539) &
(abs(Phasesecondpidbackward)< 541)) | ((abs(Phasesecondpidbackward)> 899) &
(abs(Phasesecondpidbackward)< 901)) | ((abs(Phasesecondpidbackward)> 1259)
& (abs(Phasesecondpidbackward)< 1261)) | ((abs(Phasesecondpidbackward)>
1619) & (abs(Phasesecondpidbackward)< 1621))  |
((abs(Phasesecondpidbackward)> 1979) & (abs(Phasesecondpidbackward)<
1981))  | ((abs(Phasesecondpidbackward)> 2339) &
(abs(Phasesecondpidbackward)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k64=k64+1;
                      samplingperiodsecondpidbackward(k64)=T;
```

```
                    frequencysecondpidbackward(k64)=w;

modulosecondpidbackward(k64)=10^(MagDBsecondpidbackward/20);
moduloDBsecondpidbackward(k64)=MagDBsecondpidbackward;
                    fasesecondpidbackward(k64)=Phasesecondpidbackward;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondpidbackward
samplingperiodsecondpidbackward frequencysecondpidbackward
modulosecondpidbackward moduloDBsecondpidbackward
fasesecondpidbackward
                  end
          %************************************************************************
          % H*(s) para Planta de 2a. Ordem (com hold) e controle PID Tustin
          %************************************************************************

                  if (abs(MagDBsecondpidtustin)<0.01) &   (
((abs(Phasesecondpidtustin)> 179.9) & (abs(Phasesecondpidtustin)< 180.1)) |
((abs(Phasesecondpidtustin)> 539) & (abs(Phasesecondpidtustin)< 541)) |
((abs(Phasesecondpidtustin)> 899) & (abs(Phasesecondpidtustin)< 901)) |
((abs(Phasesecondpidtustin)> 1259) & (abs(Phasesecondpidtustin)< 1261)) |
((abs(Phasesecondpidtustin)> 1619) & (abs(Phasesecondpidtustin)< 1621))  |
((abs(Phasesecondpidtustin)> 1979) & (abs(Phasesecondpidtustin)< 1981))  |
((abs(Phasesecondpidtustin)> 2339) & (abs(Phasesecondpidtustin)< 2341))  )
                  %Montagem dos vetores armazenando os dados importantes:
                    k65=k65+1;
                    samplingperiodsecondpidtustin(k65)=T;
                    frequencysecondpidtustin(k65)=w;

                   modulosecondpidtustin(k65)=10^(MagDBsecondpidtustin/20);
                    moduloDBsecondpidtustin(k65)=MagDBsecondpidtustin;
                    fasesecondpidtustin(k65)=Phasesecondpidtustin;
```

```
                %Grava arquivo formato .MAT:
                save sttheorysecondpidtustin
samplingperiodsecondpidtustin frequencysecondpidtustin modulosecondpidtustin
moduloDBsecondpidtustin fasesecondpidtustin
            end
        %************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle PID ST1
        %************************************************************************

            if (abs(MagDBsecondpidst1)<0.01) &   (
((abs(Phasesecondpidst1)> 179.9) & (abs(Phasesecondpidst1)< 180.1)) |
((abs(Phasesecondpidst1)> 539) & (abs(Phasesecondpidst1)< 541)) |
((abs(Phasesecondpidst1)> 899) & (abs(Phasesecondpidst1)< 901)) |
((abs(Phasesecondpidst1)> 1259) & (abs(Phasesecondpidst1)< 1261)) |
((abs(Phasesecondpidst1)> 1619) & (abs(Phasesecondpidst1)< 1621))  |
((abs(Phasesecondpidst1)> 1979) & (abs(Phasesecondpidst1)< 1981))  |
((abs(Phasesecondpidst1)> 2339) & (abs(Phasesecondpidst1)< 2341))  )
              %Montagem dos vetores armazenando os dados importantes:
                k66=k66+1;
                samplingperiodsecondpidst1(k66)=T;
                frequencysecondpidst1(k66)=w;
                modulosecondpidst1(k66)=10^(MagDBsecondpidst1/20);
                moduloDBsecondpidst1(k66)=MagDBsecondpidst1;
                fasesecondpidst1(k66)=Phasesecondpidst1;
                %Grava arquivo formato .MAT:
                save sttheorysecondpidst1 samplingperiodsecondpidst1
frequencysecondpidst1 modulosecondpidst1 moduloDBsecondpidst1
fasesecondpidst1
            end
```

```
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID ST2
%************************************************************************

                if (abs(MagDBsecondpidst2)<0.01) &   (
((abs(Phasesecondpidst2)> 179.9) & (abs(Phasesecondpidst2)< 180.1)) |
((abs(Phasesecondpidst2)> 539) & (abs(Phasesecondpidst2)< 541)) |
((abs(Phasesecondpidst2)> 899) & (abs(Phasesecondpidst2)< 901)) |
((abs(Phasesecondpidst2)> 1259) & (abs(Phasesecondpidst2)< 1261)) |
((abs(Phasesecondpidst2)> 1619) & (abs(Phasesecondpidst2)< 1621))  |
((abs(Phasesecondpidst2)> 1979) & (abs(Phasesecondpidst2)< 1981))  |
((abs(Phasesecondpidst2)> 2339) & (abs(Phasesecondpidst2)< 2341))  )
                  %Montagem dos vetores armazenando os dados importantes:
                    k67=k67+1;
                    samplingperiodsecondpidst2(k67)=T;
                    frequencysecondpidst2(k67)=w;
                    modulosecondpidst2(k67)=10^(MagDBsecondpidst2/20);
                    moduloDBsecondpidst2(k67)=MagDBsecondpidst2;
                    fasesecondpidst2(k67)=Phasesecondpidst2;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondpidst2 samplingperiodsecondpidst2
frequencysecondpidst2 modulosecondpidst2 moduloDBsecondpidst2
fasesecondpidst2
                end
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle PID
% Schneider
%************************************************************************

                if (abs(MagDBsecondpidschneider)<0.01) &   (
((abs(Phasesecondpidschneider)> 179.9) & (abs(Phasesecondpidschneider)<
```

```
180.1)) | ((abs(Phasesecondpidschneider)> 539) &
(abs(Phasesecondpidschneider)< 541)) | ((abs(Phasesecondpidschneider)> 899)
& (abs(Phasesecondpidschneider)< 901)) | ((abs(Phasesecondpidschneider)>
1259) & (abs(Phasesecondpidschneider)< 1261)) |
((abs(Phasesecondpidschneider)> 1619) & (abs(Phasesecondpidschneider)<
1621))  | ((abs(Phasesecondpidschneider)> 1979) &
(abs(Phasesecondpidschneider)< 1981))  | ((abs(Phasesecondpidschneider)>
2339) & (abs(Phasesecondpidschneider)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k68=k68+1;
                          samplingperiodsecondpidschneider(k68)=T;
                          frequencysecondpidschneider(k68)=w;

modulosecondpidschneider(k68)=10^(MagDBsecondpidschneider/20);
moduloDBsecondpidschneider(k68)=MagDBsecondpidschneider;
                          fasesecondpidschneider(k68)=Phasesecondpidschneider;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondpidschneider
samplingperiodsecondpidschneider frequencysecondpidschneider
modulosecondpidschneider moduloDBsecondpidschneider
fasesecondpidschneider
                        end

                %************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD
                % Forward
                %************************************************************************

                        if (abs(MagDBsecondleadforward)<0.01) &   (
((abs(Phasesecondleadforward)> 179.9) & (abs(Phasesecondleadforward)<
180.1)) | ((abs(Phasesecondleadforward)> 539) & (abs(Phasesecondleadforward)<
```

```
541)) | ((abs(Phasesecondleadforward)> 899) & (abs(Phasesecondleadforward)<
901)) | ((abs(Phasesecondleadforward)> 1259) & (abs(Phasesecondleadforward)<
1261)) | ((abs(Phasesecondleadforward)> 1619) &
(abs(Phasesecondleadforward)< 1621))  | ((abs(Phasesecondleadforward)> 1979)
& (abs(Phasesecondleadforward)< 1981))  | ((abs(Phasesecondleadforward)>
2339) & (abs(Phasesecondleadforward)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k69=k69+1;
                          samplingperiodsecondleadforward(k69)=T;
                          frequencysecondleadforward(k69)=w;

modulosecondleadforward(k69)=10^(MagDBsecondleadforward/20);
moduloDBsecondleadforward(k69)=MagDBsecondleadforward;
                          fasesecondleadforward(k69)=Phasesecondleadforward;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondleadforward
samplingperiodsecondleadforward frequencysecondleadforward
modulosecondleadforward moduloDBsecondleadforward fasesecondleadforward
                    end
               %**************************************************************************
               % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD
               % Backward
               %**************************************************************************

                    if (abs(MagDBsecondleadbackward)<0.01) &   (
((abs(Phasesecondleadbackward)> 179.9) & (abs(Phasesecondleadbackward)<
180.1)) | ((abs(Phasesecondleadbackward)> 539) &
(abs(Phasesecondleadbackward)< 541)) | ((abs(Phasesecondleadbackward)>
899) & (abs(Phasesecondleadbackward)< 901)) |
((abs(Phasesecondleadbackward)> 1259) & (abs(Phasesecondleadbackward)<
1261)) | ((abs(Phasesecondleadbackward)> 1619) &
```

```
(abs(Phasesecondleadbackward)< 1621))  | ((abs(Phasesecondleadbackward)>
1979) & (abs(Phasesecondleadbackward)< 1981))  |
((abs(Phasesecondleadbackward)> 2339) & (abs(Phasesecondleadbackward)<
2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k70=k70+1;
                      samplingperiodsecondleadbackward(k70)=T;
                      frequencysecondleadbackward(k70)=w;

modulosecondleadbackward(k70)=10^(MagDBsecondleadbackward/20);
moduloDBsecondleadbackward(k70)=MagDBsecondleadbackward;
                      fasesecondleadbackward(k70)=Phasesecondleadbackward;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondleadbackward
samplingperiodsecondleadbackward frequencysecondleadbackward
modulosecondleadbackward moduloDBsecondleadbackward
fasesecondleadbackward
                  end
            %************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD Tustin
            %************************************************************************

                  if (abs(MagDBsecondleadtustin)<0.01) &   (
((abs(Phasesecondleadtustin)> 179.9) & (abs(Phasesecondleadtustin)< 180.1)) |
((abs(Phasesecondleadtustin)> 539) & (abs(Phasesecondleadtustin)< 541)) |
((abs(Phasesecondleadtustin)> 899) & (abs(Phasesecondleadtustin)< 901)) |
((abs(Phasesecondleadtustin)> 1259) & (abs(Phasesecondleadtustin)< 1261)) |
((abs(Phasesecondleadtustin)> 1619) & (abs(Phasesecondleadtustin)< 1621))  |
((abs(Phasesecondleadtustin)> 1979) & (abs(Phasesecondleadtustin)< 1981))  |
((abs(Phasesecondleadtustin)> 2339) & (abs(Phasesecondleadtustin)< 2341))  )
                  %Montagem dos vetores armazenando os dados importantes:
```

```
                    k71=k71+1;
                    samplingperiodsecondleadtustin(k71)=T;
                    frequencysecondleadtustin(k71)=w;

                modulosecondleadtustin(k71)=10^(MagDBsecondleadtustin/20);
                    moduloDBsecondleadtustin(k71)=MagDBsecondleadtustin;
                    fasesecondleadtustin(k71)=Phasesecondleadtustin;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondleadtustin
samplingperiodsecondleadtustin frequencysecondleadtustin
modulosecondleadtustin moduloDBsecondleadtustin fasesecondleadtustin
                    end
            %************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD ST1
            %************************************************************************

                    if (abs(MagDBsecondleadst1)<0.01) &   (
((abs(Phasesecondleadst1)> 179.9) & (abs(Phasesecondleadst1)< 180.1)) |
((abs(Phasesecondleadst1)> 539) & (abs(Phasesecondleadst1)< 541)) |
((abs(Phasesecondleadst1)> 899) & (abs(Phasesecondleadst1)< 901)) |
((abs(Phasesecondleadst1)> 1259) & (abs(Phasesecondleadst1)< 1261)) |
((abs(Phasesecondleadst1)> 1619) & (abs(Phasesecondleadst1)< 1621))  |
((abs(Phasesecondleadst1)> 1979) & (abs(Phasesecondleadst1)< 1981))  |
((abs(Phasesecondleadst1)> 2339) & (abs(Phasesecondleadst1)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                    k72=k72+1;
                    samplingperiodsecondleadst1(k72)=T;
                    frequencysecondleadst1(k72)=w;
                    modulosecondleadst1(k72)=10^(MagDBsecondleadst1/20);
                    moduloDBsecondleadst1(k72)=MagDBsecondleadst1;
                    fasesecondleadst1(k72)=Phasesecondleadst1;
```

```
                %Grava arquivo formato .MAT:
                save sttheorysecondleadst1 samplingperiodsecondleadst1
frequencysecondleadst1 modulosecondleadst1 moduloDBsecondleadst1
fasesecondleadst1
            end
        %*************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD ST2
        %*************************************************************************

            if (abs(MagDBsecondleadst2)<0.01) &   (
((abs(Phasesecondleadst2)> 179.9) & (abs(Phasesecondleadst2)< 180.1)) |
((abs(Phasesecondleadst2)> 539) & (abs(Phasesecondleadst2)< 541)) |
((abs(Phasesecondleadst2)> 899) & (abs(Phasesecondleadst2)< 901)) |
((abs(Phasesecondleadst2)> 1259) & (abs(Phasesecondleadst2)< 1261)) |
((abs(Phasesecondleadst2)> 1619) & (abs(Phasesecondleadst2)< 1621))  |
((abs(Phasesecondleadst2)> 1979) & (abs(Phasesecondleadst2)< 1981))  |
((abs(Phasesecondleadst2)> 2339) & (abs(Phasesecondleadst2)< 2341))  )
              %Montagem dos vetores armazenando os dados importantes:
                k73=k73+1;
                samplingperiodsecondleadst2(k73)=T;
                frequencysecondleadst2(k73)=w;
                modulosecondleadst2(k73)=10^(MagDBsecondleadst2/20);
                moduloDBsecondleadst2(k73)=MagDBsecondleadst2;
                fasesecondleadst2(k73)=Phasesecondleadst2;
                %Grava arquivo formato .MAT:
                save sttheorysecondleadst2 samplingperiodsecondleadst2
frequencysecondleadst2 modulosecondleadst2 moduloDBsecondleadst2
fasesecondleadst2
            end
```

```
%************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD
% Schneider
%************************************************************************

                    if (abs(MagDBsecondleadschneider)<0.01) &   (
((abs(Phasesecondleadschneider)> 179.9) & (abs(Phasesecondleadschneider)<
180.1)) | ((abs(Phasesecondleadschneider)> 539) &
(abs(Phasesecondleadschneider)< 541)) | ((abs(Phasesecondleadschneider)>
899) & (abs(Phasesecondleadschneider)< 901)) |
((abs(Phasesecondleadschneider)> 1259) & (abs(Phasesecondleadschneider)<
1261)) | ((abs(Phasesecondleadschneider)> 1619) &
(abs(Phasesecondleadschneider)< 1621))  | ((abs(Phasesecondleadschneider)>
1979) & (abs(Phasesecondleadschneider)< 1981))  |
((abs(Phasesecondleadschneider)> 2339) & (abs(Phasesecondleadschneider)<
2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k74=k74+1;
                      samplingperiodsecondleadschneider(k74)=T;
                      frequencysecondleadschneider(k74)=w;

modulosecondleadschneider(k74)=10^(MagDBsecondleadschneider/20);
moduloDBsecondleadschneider(k74)=MagDBsecondleadschneider;
                      fasesecondleadschneider(k74)=Phasesecondleadschneider;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondleadschneider
samplingperiodsecondleadschneider frequencysecondleadschneider
modulosecondleadschneider moduloDBsecondleadschneider
fasesecondleadschneider
                    end
```

```
%**************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle LAG Forward
%**************************************************************************

                if (abs(MagDBsecondlagforward)<0.01) &   (
((abs(Phasesecondlagforward)> 179.9) & (abs(Phasesecondlagforward)< 180.1)) |
((abs(Phasesecondlagforward)> 539) & (abs(Phasesecondlagforward)< 541)) |
((abs(Phasesecondlagforward)> 899) & (abs(Phasesecondlagforward)< 901)) |
((abs(Phasesecondlagforward)> 1259) & (abs(Phasesecondlagforward)< 1261)) |
((abs(Phasesecondlagforward)> 1619) & (abs(Phasesecondlagforward)< 1621))  |
((abs(Phasesecondlagforward)> 1979) & (abs(Phasesecondlagforward)< 1981))  |
((abs(Phasesecondlagforward)> 2339) & (abs(Phasesecondlagforward)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k75=k75+1;
                      samplingperiodsecondlagforward(k75)=T;
                      frequencysecondlagforward(k75)=w;

modulosecondlagforward(k75)=10^(MagDBsecondlagforward/20);
moduloDBsecondlagforward(k75)=MagDBsecondlagforward;
                      fasesecondlagforward(k75)=Phasesecondlagforward;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondlagforward
samplingperiodsecondlagforward frequencysecondlagforward
modulosecondlagforward moduloDBsecondlagforward fasesecondlagforward
                  end
%**************************************************************************
% H*(s) para Planta de 2a. Ordem (com hold) e controle LAG
% Backward
%**************************************************************************
```

```
                    if (abs(MagDBsecondlagbackward)<0.01) &   (
((abs(Phasesecondlagbackward)> 179.9) & (abs(Phasesecondlagbackward)<
180.1)) | ((abs(Phasesecondlagbackward)> 539) &
(abs(Phasesecondlagbackward)< 541)) | ((abs(Phasesecondlagbackward)> 899) &
(abs(Phasesecondlagbackward)< 901)) | ((abs(Phasesecondlagbackward)> 1259)
& (abs(Phasesecondlagbackward)< 1261)) | ((abs(Phasesecondlagbackward)>
1619) & (abs(Phasesecondlagbackward)< 1621))  |
((abs(Phasesecondlagbackward)> 1979) & (abs(Phasesecondlagbackward)<
1981))  | ((abs(Phasesecondlagbackward)> 2339) &
(abs(Phasesecondlagbackward)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k76=k76+1;
                      samplingperiodsecondlagbackward(k76)=T;
                      frequencysecondlagbackward(k76)=w;

modulosecondlagbackward(k76)=10^(MagDBsecondlagbackward/20);
moduloDBsecondlagbackward(k76)=MagDBsecondlagbackward;
                      fasesecondlagbackward(k76)=Phasesecondlagbackward;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondlagbackward
samplingperiodsecondlagbackward frequencysecondlagbackward
modulosecondlagbackward moduloDBsecondlagbackward
fasesecondlagbackward
                    end
          %************************************************************************
          % H*(s) para Planta de 2a. Ordem (com hold) e controle LAG Tustin
          %************************************************************************

                    if (abs(MagDBsecondlagtustin)<0.01) &   (
((abs(Phasesecondlagtustin)> 179.9) & (abs(Phasesecondlagtustin)< 180.1)) |
```

```
((abs(Phasesecondlagtustin)> 539) & (abs(Phasesecondlagtustin)< 541)) |
((abs(Phasesecondlagtustin)> 899) & (abs(Phasesecondlagtustin)< 901)) |
((abs(Phasesecondlagtustin)> 1259) & (abs(Phasesecondlagtustin)< 1261)) |
((abs(Phasesecondlagtustin)> 1619) & (abs(Phasesecondlagtustin)< 1621))  |
((abs(Phasesecondlagtustin)> 1979) & (abs(Phasesecondlagtustin)< 1981))  |
((abs(Phasesecondlagtustin)> 2339) & (abs(Phasesecondlagtustin)< 2341))  )
                        %Montagem dos vetores armazenando os dados importantes:
                          k77=k77+1;
                          samplingperiodsecondlagtustin(k77)=T;
                          frequencysecondlagtustin(k77)=w;

                         modulosecondlagtustin(k77)=10^(MagDBsecondlagtustin/20);
                          moduloDBsecondlagtustin(k77)=MagDBsecondlagtustin;
                          fasesecondlagtustin(k77)=Phasesecondlagtustin;
                          %Grava arquivo formato .MAT:
                          save sttheorysecondlagtustin
samplingperiodsecondlagtustin frequencysecondlagtustin modulosecondlagtustin
moduloDBsecondlagtustin fasesecondlagtustin
                        end
                %************************************************************************
                % H*(s) para Planta de 2a. Ordem (com hold) e controle LAG ST1
                %************************************************************************

                        if (abs(MagDBsecondlagst1)<0.01) &   (
((abs(Phasesecondlagst1)> 179.9) & (abs(Phasesecondlagst1)< 180.1)) |
((abs(Phasesecondlagst1)> 539) & (abs(Phasesecondlagst1)< 541)) |
((abs(Phasesecondlagst1)> 899) & (abs(Phasesecondlagst1)< 901)) |
((abs(Phasesecondlagst1)> 1259) & (abs(Phasesecondlagst1)< 1261)) |
((abs(Phasesecondlagst1)> 1619) & (abs(Phasesecondlagst1)< 1621))  |
((abs(Phasesecondlagst1)> 1979) & (abs(Phasesecondlagst1)< 1981))  |
((abs(Phasesecondlagst1)> 2339) & (abs(Phasesecondlagst1)< 2341))  )
```

```
                    %Montagem dos vetores armazenando os dados importantes:
                      k78=k78+1;
                      samplingperiodsecondlagst1(k78)=T;
                      frequencysecondlagst1(k78)=w;
                      modulosecondlagst1(k78)=10^(MagDBsecondlagst1/20);
                      moduloDBsecondlagst1(k78)=MagDBsecondlagst1;
                      fasesecondlagst1(k78)=Phasesecondlagst1;
                      %Grava arquivo formato .MAT:
                      save sttheorysecondlagst1 samplingperiodsecondlagst1
frequencysecondlagst1 modulosecondlagst1 moduloDBsecondlagst1
fasesecondlagst1
                    end
              %*************************************************************************
              % H*(s) para Planta de 2a. Ordem (com hold) e controle LAG ST2
              %*************************************************************************

                    if (abs(MagDBsecondlagst2)<0.01) &   (
((abs(Phasesecondlagst2)> 179.9) & (abs(Phasesecondlagst2)< 180.1)) |
((abs(Phasesecondlagst2)> 539) & (abs(Phasesecondlagst2)< 541)) |
((abs(Phasesecondlagst2)> 899) & (abs(Phasesecondlagst2)< 901)) |
((abs(Phasesecondlagst2)> 1259) & (abs(Phasesecondlagst2)< 1261)) |
((abs(Phasesecondlagst2)> 1619) & (abs(Phasesecondlagst2)< 1621))  |
((abs(Phasesecondlagst2)> 1979) & (abs(Phasesecondlagst2)< 1981))  |
((abs(Phasesecondlagst2)> 2339) & (abs(Phasesecondlagst2)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k79=k79+1;
                      samplingperiodsecondlagst2(k79)=T;
                      frequencysecondlagst2(k79)=w;
                      modulosecondlagst2(k79)=10^(MagDBsecondlagst2/20);
                      moduloDBsecondlagst2(k79)=MagDBsecondlagst2;
                      fasesecondlagst2(k79)=Phasesecondlagst2;
```

```
                    %Grava arquivo formato .MAT:
                    save sttheorysecondlagst2 samplingperiodsecondlagst2
frequencysecondlagst2 modulosecondlagst2 moduloDBsecondlagst2
fasesecondlagst2
                end
            %*************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle LAG
            % Schneider
            %*************************************************************************

                if (abs(MagDBsecondlagschneider)<0.01) &   (
((abs(Phasesecondlagschneider)> 179.9) & (abs(Phasesecondlagschneider)<
180.1)) | ((abs(Phasesecondlagschneider)> 539) &
(abs(Phasesecondlagschneider)< 541)) | ((abs(Phasesecondlagschneider)> 899)
& (abs(Phasesecondlagschneider)< 901)) | ((abs(Phasesecondlagschneider)>
1259) & (abs(Phasesecondlagschneider)< 1261)) |
((abs(Phasesecondlagschneider)> 1619) & (abs(Phasesecondlagschneider)<
1621))  | ((abs(Phasesecondlagschneider)> 1979) &
(abs(Phasesecondlagschneider)< 1981))  | ((abs(Phasesecondlagschneider)>
2339) & (abs(Phasesecondlagschneider)< 2341))  )
                %Montagem dos vetores armazenando os dados importantes:
                    k80=k80+1;
                    samplingperiodsecondlagschneider(k80)=T;
                    frequencysecondlagschneider(k80)=w;

modulosecondlagschneider(k80)=10^(MagDBsecondlagschneider/20);
moduloDBsecondlagschneider(k80)=MagDBsecondlagschneider;
                    fasesecondlagschneider(k80)=Phasesecondlagschneider;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondlagschneider
samplingperiodsecondlagschneider frequencysecondlagschneider
```

```
modulosecondlagschneider moduloDBsecondlagschneider
fasesecondlagschneider
                    end

            %**************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD.LAG
            % Forward
            %**************************************************************************

                    if (abs(MagDBsecondleadlagforward)<0.01) &   (
((abs(Phasesecondleadlagforward)> 179.9) & (abs(Phasesecondleadlagforward)<
180.1)) | ((abs(Phasesecondleadlagforward)> 539) &
(abs(Phasesecondleadlagforward)< 541)) | ((abs(Phasesecondleadlagforward)>
899) & (abs(Phasesecondleadlagforward)< 901)) |
((abs(Phasesecondleadlagforward)> 1259) & (abs(Phasesecondleadlagforward)<
1261)) | ((abs(Phasesecondleadlagforward)> 1619) &
(abs(Phasesecondleadlagforward)< 1621))  | ((abs(Phasesecondleadlagforward)>
1979) & (abs(Phasesecondleadlagforward)< 1981))  |
((abs(Phasesecondleadlagforward)> 2339) & (abs(Phasesecondleadlagforward)<
2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k81=k81+1;
                      samplingperiodsecondleadlagforward(k81)=T;
                      frequencysecondleadlagforward(k81)=w;

modulosecondleadlagforward(k81)=10^(MagDBsecondleadlagforward/20);
moduloDBsecondleadlagforward(k81)=MagDBsecondleadlagforward;
fasesecondleadlagforward(k81)=Phasesecondleadlagforward;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondleadlagforward
samplingperiodsecondleadlagforward frequencysecondleadlagforward
```

```
modulosecondleadlagforward moduloDBsecondleadlagforward
fasesecondleadlagforward
                    end
            %**************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD.LAG
            % Backward
            %**************************************************************************

                    if (abs(MagDBsecondleadlagbackward)<0.01) &   (
((abs(Phasesecondleadlagbackward)> 179.9) &
(abs(Phasesecondleadlagbackward)< 180.1)) |
((abs(Phasesecondleadlagbackward)> 539) &
(abs(Phasesecondleadlagbackward)< 541)) |
((abs(Phasesecondleadlagbackward)> 899) &
(abs(Phasesecondleadlagbackward)< 901)) |
((abs(Phasesecondleadlagbackward)> 1259) &
(abs(Phasesecondleadlagbackward)< 1261)) |
((abs(Phasesecondleadlagbackward)> 1619) &
(abs(Phasesecondleadlagbackward)< 1621))  |
((abs(Phasesecondleadlagbackward)> 1979) &
(abs(Phasesecondleadlagbackward)< 1981))  |
((abs(Phasesecondleadlagbackward)> 2339) &
(abs(Phasesecondleadlagbackward)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k82=k82+1;
                      samplingperiodsecondleadlagbackward(k82)=T;
                      frequencysecondleadlagbackward(k82)=w;

modulosecondleadlagbackward(k82)=10^(MagDBsecondleadlagbackward/20);
moduloDBsecondleadlagbackward(k82)=MagDBsecondleadlagbackward;
```

```
fasesecondleadlagbackward(k82)=Phasesecondleadlagbackward;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondleadlagbackward
samplingperiodsecondleadlagbackward frequencysecondleadlagbackward
modulosecondleadlagbackward moduloDBsecondleadlagbackward
fasesecondleadlagbackward
                    end
          %************************************************************************
          % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD.LAG
          % Tustin
          %************************************************************************

                    if (abs(MagDBsecondleadlagtustin)<0.01) &   (
((abs(Phasesecondleadlagtustin)> 179.9) & (abs(Phasesecondleadlagtustin)<
180.1)) | ((abs(Phasesecondleadlagtustin)> 539) &
(abs(Phasesecondleadlagtustin)< 541)) | ((abs(Phasesecondleadlagtustin)> 899) &
(abs(Phasesecondleadlagtustin)< 901)) | ((abs(Phasesecondleadlagtustin)> 1259)
& (abs(Phasesecondleadlagtustin)< 1261)) | ((abs(Phasesecondleadlagtustin)>
1619) & (abs(Phasesecondleadlagtustin)< 1621))  |
((abs(Phasesecondleadlagtustin)> 1979) & (abs(Phasesecondleadlagtustin)<
1981))  | ((abs(Phasesecondleadlagtustin)> 2339) &
(abs(Phasesecondleadlagtustin)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k83=k83+1;
                      samplingperiodsecondleadlagtustin(k83)=T;
                      frequencysecondleadlagtustin(k83)=w;

modulosecondleadlagtustin(k83)=10^(MagDBsecondleadlagtustin/20);
moduloDBsecondleadlagtustin(k83)=MagDBsecondleadlagtustin;
                      fasesecondleadlagtustin(k83)=Phasesecondleadlagtustin;
```

```
                    %Grava arquivo formato .MAT:
                    save sttheorysecondleadlagtustin
samplingperiodsecondleadlagtustin frequencysecondleadlagtustin
modulosecondleadlagtustin moduloDBsecondleadlagtustin
fasesecondleadlagtustin
                end
            %************************************************************************
            % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD.LAG
            % ST1
            %************************************************************************

                if (abs(MagDBsecondleadlagst1)<0.01) &   (
((abs(Phasesecondleadlagst1)> 179.9) & (abs(Phasesecondleadlagst1)< 180.1)) |
((abs(Phasesecondleadlagst1)> 539) & (abs(Phasesecondleadlagst1)< 541)) |
((abs(Phasesecondleadlagst1)> 899) & (abs(Phasesecondleadlagst1)< 901)) |
((abs(Phasesecondleadlagst1)> 1259) & (abs(Phasesecondleadlagst1)< 1261)) |
((abs(Phasesecondleadlagst1)> 1619) & (abs(Phasesecondleadlagst1)< 1621))  |
((abs(Phasesecondleadlagst1)> 1979) & (abs(Phasesecondleadlagst1)< 1981))  |
((abs(Phasesecondleadlagst1)> 2339) & (abs(Phasesecondleadlagst1)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                    k84=k84+1;
                    samplingperiodsecondleadlagst1(k84)=T;
                    frequencysecondleadlagst1(k84)=w;

                modulosecondleadlagst1(k84)=10^(MagDBsecondleadlagst1/20);
                    moduloDBsecondleadlagst1(k84)=MagDBsecondleadlagst1;
                    fasesecondleadlagst1(k84)=Phasesecondleadlagst1;
                    %Grava arquivo formato .MAT:
                    save sttheorysecondleadlagst1
samplingperiodsecondleadlagst1 frequencysecondleadlagst1
modulosecondleadlagst1 moduloDBsecondleadlagst1 fasesecondleadlagst1
```

```
            end
        %**************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD.LAG
        % ST2
        %**************************************************************************

            if (abs(MagDBsecondleadlagst2)<0.01) &   (
((abs(Phasesecondleadlagst2)> 179.9) & (abs(Phasesecondleadlagst2)< 180.1)) |
((abs(Phasesecondleadlagst2)> 539) & (abs(Phasesecondleadlagst2)< 541)) |
((abs(Phasesecondleadlagst2)> 899) & (abs(Phasesecondleadlagst2)< 901)) |
((abs(Phasesecondleadlagst2)> 1259) & (abs(Phasesecondleadlagst2)< 1261)) |
((abs(Phasesecondleadlagst2)> 1619) & (abs(Phasesecondleadlagst2)< 1621))  |
((abs(Phasesecondleadlagst2)> 1979) & (abs(Phasesecondleadlagst2)< 1981))  |
((abs(Phasesecondleadlagst2)> 2339) & (abs(Phasesecondleadlagst2)< 2341))  )
              %Montagem dos vetores armazenando os dados importantes:
                k85=k85+1;
                samplingperiodsecondleadlagst2(k85)=T;
                frequencysecondleadlagst2(k85)=w;

            modulosecondleadlagst2(k85)=10^(MagDBsecondleadlagst2/20);
                moduloDBsecondleadlagst2(k85)=MagDBsecondleadlagst2;
                fasesecondleadlagst2(k85)=Phasesecondleadlagst2;
                %Grava arquivo formato .MAT:
                save sttheorysecondleadlagst2
samplingperiodsecondleadlagst2 frequencysecondleadlagst2
modulosecondleadlagst2 moduloDBsecondleadlagst2 fasesecondleadlagst2
            end
        %**************************************************************************
        % H*(s) para Planta de 2a. Ordem (com hold) e controle LEAD.LAG
        % Schneider
        %**************************************************************************
```

```
                    if (abs(MagDBsecondleadlagschneider)<0.01) &   (
((abs(Phasesecondleadlagschneider)> 179.9) &
(abs(Phasesecondleadlagschneider)< 180.1)) |
((abs(Phasesecondleadlagschneider)> 539) &
(abs(Phasesecondleadlagschneider)< 541)) |
((abs(Phasesecondleadlagschneider)> 899) &
(abs(Phasesecondleadlagschneider)< 901)) |
((abs(Phasesecondleadlagschneider)> 1259) &
(abs(Phasesecondleadlagschneider)< 1261)) |
((abs(Phasesecondleadlagschneider)> 1619) &
(abs(Phasesecondleadlagschneider)< 1621))  |
((abs(Phasesecondleadlagschneider)> 1979) &
(abs(Phasesecondleadlagschneider)< 1981))  |
((abs(Phasesecondleadlagschneider)> 2339) &
(abs(Phasesecondleadlagschneider)< 2341))  )
                    %Montagem dos vetores armazenando os dados importantes:
                      k86=k86+1;
                      samplingperiodsecondleadlagschneider(k86)=T;
                      frequencysecondleadlagschneider(k86)=w;

modulosecondleadlagschneider(k86)=10^(MagDBsecondleadlagschneider/20);
moduloDBsecondleadlagschneider(k86)=MagDBsecondleadlagschneider;
fasesecondleadlagschneider(k86)=Phasesecondleadlagschneider;
                      %Grava arquivo formato .MAT:
                      save sttherysecondleadlagschneider
samplingperiodsecondleadlagschneider frequencysecondleadlagschneider
modulosecondleadlagschneider moduloDBsecondleadlagschneider
fasesecondleadlagschneider
                    end
```

```
        %pause
    end         %End do loop da frequencia w
end         %End do loop do periodo de amostragem T

disp(' ')
disp('FIM')
disp(' ')
```

G.2. EXEMPLO DE SCRIPT MATLAB PARA GERAR DIAGRAMAS FASORIAIS DAS FUNÇÕES DE e^{sT}. ESSE SCRIPT REQUER O USO DAS FUNÇÕES MATLAB "CIRCLE" E "ARROW" (DISPONÍVEIS NO WEBSITE DA EMPRESA THE MATHWORKS)

```
%*********************************************************************
% TESE DE DOUTORADO NO INPE/DMC (CURSO ETE) - M. TREDINNICK
% ORIENTADOR: DR. MARCELO LOPES DE OLIVEIRA E SOUZA
%*********************************************************************
% DIAGRAMAS DE BODE
%*********************************************************************
% CASO 001:
%         PLANTA DE PRIMEIRA ORDEM
%         CONTROLE INTEGRAL
%         MAPEAMENTO FORWARD
%*********************************************************************

clc; clear all

%*************************************************************************
% INICIALIZAÇAO DE CONSTANTES ITERATIVAS PARA A GRAVAÇAO DE
% VALORES DE H*(s)
%*************************************************************************
k=0;       %constante iterativa
a=1;       %constante existente na planta de segunda ordem G(s)=a/[s(s+a)]
T=1;     %periodo de amostragem identificado como provavel periodo instabilizador
          % no script ESCANEADOR
ganho = 1;  %ganho de controle
for w=.2:.01:(2*pi)*10
```

```
%************************************************************************
% PLANTA DE PRIMEIRA ORDEM JA COM ZERO ORDER HOLD:
% (Franklin (1981), equaçao 4.39)
% G(s) = a/(s+a)
% (1-exp(-sT)).(G(s)/s)* = 0.5/(exp(sT)-0.5)
% convençao usada no livro Franklin (1981): exp(-aT)=0.5
%************************************************************************
        expG =  0.5 ./ [exp(i*w*T)-0.5];

%************************************************************************
%       CONTROLE
%************************************************************************
        %************************************************************************
        % CONTROLE INTEGRAL (mapeamento Forward)
        %************************************************************************
                D = ganho * (T) ./ (exp(i*w*T) - 1);

%************************************************************************
%************************************************************************
% FUNÇAO DE TRANSFERENCIA DE MALHA-ABERTA H*(S) = (ZERO
% ORDER HOLD).D*.(G/s)*
% H*(S) = (1-exp(-sT)) . D* . (G/s)*
% Ja calculei o HOLD junto com (G/s)* acima, nesse script.
%************************************************************************
%************************************************************************
          H=1.000* D .* expG;  %tem que colocar o Hold para o calculo da
                               %planta de segunda ordem
```

```
    o = [0 0]; %origem
    Hfasor = [real(H) imag(H)]
    Dfasor = [real(D) imag(D)]
    Gfasor = [real(expG) imag(expG)]
    faseH=angle(H);faseHgraus=faseH*180/pi;
    faseD=angle(D);faseDgraus=faseD*180/pi;
    faseexpG=angle(expG);faseexpGgraus=faseexpG*180/pi;
    figure(1)
    hhh=arrow(o,Hfasor,40);set(hhh,'FaceColor',[1 0 0]) %vermelho
    ddd=arrow(o,Dfasor,40);set(ddd,'FaceColor',[0 0 1]) %azul
    ggg=arrow(o,Gfasor,40);set(ggg,'FaceColor',[0 0 0]) %preto
    title(['  w(rd/s)= ' num2str(w) '      T=' num2str(T) '      w*T=' num2str(w*T) '
|D*|=' num2str(norm(Dfasor,2)) ' ,Fase D*=' num2str(faseDgraus) '      |(1-exp(-
sT)[G/s]*|=' num2str(norm(Gfasor,2)) ' ,Fase=' num2str(faseexpGgraus) '      |H*|='
num2str(norm(Hfasor,2)) ' ,Fase H*=' num2str(faseHgraus)]);
    xlabel('Imag');ylabel('Real');
    legend('|H*|','|D*|','(1-exp(sT)[G/s]*')
    grid on;axis([-1.2 1.2 -1.2 1.2]);
    hold on
    circle(o,1,1000);
    pause
    clf
end
```

G.3. SCRIPTS MATLAB PARA GERAR DIAGRAMAS DE BODE DAS FUNÇÕES DE e^{sT}

CASO 1: PLANTA DE PRIMEIRA ORDEM E CONTROLE INTEGRAL MAPEADO PELO MÉTODO FORWARD

```
%*********************************************************************
% TESE DE DOUTORADO NO INPE/DMC (CURSO ETE) - M. TREDINNICK
% ORIENTADOR: DR. MARCELO LOPES DE OLIVEIRA E SOUZA
%*********************************************************************
% DIAGRAMAS DE BODE
%*********************************************************************
% CASO 001:
%       PLANTA DE PRIMEIRA ORDEM
%       CONTROLE INTEGRAL
%       MAPEAMENTO FORWARD
%*********************************************************************

clc; clear all

%*************************************************************************
% INICIALIZAÇAO DE CONSTANTES ITERATIVAS PARA A GRAVAÇAO DE
% VALORES DE H*(s)
%*************************************************************************
k=0;      %constante iterativa
a=1;      %constante existente na planta de segunda ordem G(s)=a/[s(s+a)]
T=1;      %periodo de amostragem identificado como provavel periodo
          % instabilizador no script ESCANEADOR
ganho = 1;  % ganho do controle

for w=.5:.005:1.5;
   w   % apresenta o valor de w
```

```
%*************************************************************************
% PLANTA DE PRIMEIRA ORDEM JA COM ZERO ORDER HOLD:
% (Franklin (1981), equaçao 4.39)
% G(s) = a/(s+a)
% (1-exp(-sT)).(G(s)/s)* = 0.5/(exp(sT)-0.5)
% convençao usada no livro Franklin (1981): exp(-aT)=0.5
%*************************************************************************
        expG =  0.5 ./ [exp(i*w*T)-0.5];
        expGlog=20*log10(abs(expG));
        expGFase=angle(expG);
        expGFaseGraus=expGFase*180/pi;

        if expGFaseGraus > 0
            expGFaseGraus=-360+expGFaseGraus;
         end
        if (expGFaseGraus > -359.9788 & w > 10^0.7982)
            expGFaseGraus=-360+expGFaseGraus;
         end
        if (expGFaseGraus > -720 & w > 10^1.0992)
            expGFaseGraus=-360+expGFaseGraus;
         end
        if (expGFaseGraus > -1080 & w > 10^1.27529)
            expGFaseGraus=-360+expGFaseGraus;
         end
%*************************************************************************
%       CONTROLE
%*************************************************************************
         %*************************************************************************
         % CONTROLE INTEGRAL (mapeamento Forward)
         %*************************************************************************
```

```
D = ganho * (T) ./ (exp(i*w*T) - 1);
Dlog=20*log10(abs(D));
Dfase=angle(D);
DfaseGraus=Dfase*180/pi;

    if DfaseGraus > 0
        DfaseGraus=-360+DfaseGraus;
     end
    if (DfaseGraus >= -269.9948 & w > 10^0.7982)
        DfaseGraus=-180+DfaseGraus;
     end
    if (DfaseGraus >= -430.44 & w > 10^0.74819)
        DfaseGraus=+00+DfaseGraus;
     end
    if (DfaseGraus >= -626.5 & w > 10^1.099)
        DfaseGraus=-180+DfaseGraus;
     end
    if (DfaseGraus >= -806.4 & w > 10^1.275)
        DfaseGraus=-180+DfaseGraus;
     end
%**************************************************************************
%**************************************************************************
% FUNÇAO DE TRANSFERENCIA DE MALHA-ABERTA H*(S) = (ZERO
% ORDER HOLD).D*.(G/s)*
% H*(S) = (1-exp(-sT)) . D* . (G/s)*
% Ja calculei o HOLD junto com (G/s)* acima, nesse script.
%**************************************************************************
%**************************************************************************
    H= D .* expG;  %tem que colocar o Hold para o calculo da planta
                   % de segunda ordem
    modH=20*log10(abs(H));
```

```
 faseH=angle(H);
 faseHgraus=faseH*180/pi;

 if faseHgraus > 0
    faseHgraus=-360+faseHgraus;
 end
 %Ifs especificos apenas para dar continuidade aos diagramas
 if (faseHgraus >= -179.8 & w > 10^-0.1423)
    faseHgraus=-360+faseHgraus;
 end
 if (faseHgraus >= -539.5 & w > 10^0.7447)
    faseHgraus=-360+faseHgraus;
 end
 if (faseHgraus >= -899.45 & w > 10^0.7982)
   faseHgraus=+180+faseHgraus;
end
 if (faseHgraus >= -719.405 & w > 10^0.845098)
   faseHgraus=-360+faseHgraus;
end
 if (faseHgraus >= -1080 & w > 10^1.07326)
   faseHgraus=-360+faseHgraus;
end
 if (faseHgraus >= -1450 & w > 10^1.0992)
  faseHgraus=+180+faseHgraus;
end
 if (faseHgraus >= -1260 & w > 10^1.12333)
  faseHgraus=-360+faseHgraus;
end
 if (faseHgraus >= -1619 & w > 10^1.25816)
  faseHgraus=-360+faseHgraus;
end
```

```
                if (faseHgraus >= -1980 & w > 10^1.27531)
                  faseHgraus=+180+faseHgraus;
                end
                if (faseHgraus >= -1799 & w > 10^1.29141)
                  faseHgraus=-360+faseHgraus;
                end
%montagem dos vetores de dados:
   k=k+1;
   freq(k)=w;
   vDlog(k)=Dlog;
   vDfaseGraus(k)=DfaseGraus;
   vexpG(k)=expGlog;
   vexpGFaseGraus(k)=expGFaseGraus;
   vmodH(k)=modH;
   vfaseHgraus(k)=faseHgraus;

end
%Plotagem de graficos Modulos e Fases:
   figure(1)
%    clf;
%                plotyy(freq,vDlog,freq,vDfaseGraus);grid on; ylabel('|D*| and
Phase(right)');title(['T = ' num2str(T) ' seconds']);
   %subplot(3,1,1), plotyy(freq,vDlog,freq,vDfaseGraus);grid on; ylabel('|D*| e
fase(a direita)');title(['T = ' num2str(T) ' seconds']);
   %subplot(3,1,2)
%    figure(2)
%    clf;
%    plotyy(freq,vexpG,freq,-180);
%    hold on;
%       plotyy(freq,vexpG,freq,vexpGFaseGraus);grid on; ylabel('20log|expG*| and
Phase(right)');
```

```
%    title(['Sampling Period T = ' num2str(T) ' seconds']);
   %subplot(3,1,3)
%    figure(3)
%    clf;
%    plotyy(freq,vmodH,freq,-180);
%    hold on;
%            plotyy(freq,vmodH,freq,vfaseHgraus);grid on; ylabel('20log|H*| and
Phase(right)');
%    title(['Sampling Period T = ' num2str(T) ' seconds']);
%   figure(4)
   clf
   subplot(2,1,1),
      jump=1:50:k;semilogx(freq(jump),vmodH(jump),'*r');hold
on;semilogx(freq(jump),vexpG(jump),'^b');hold
on;semilogx(freq(jump),vDlog(jump),'vk')
      legend('|H*|','|(1-exp(-sT)(G/s)*|','|D*|');
      semilogx(freq,vmodH,'r');hold on;semilogx(freq,vexpG,'b');hold
on;semilogx(freq,vDlog,'k');grid on; ylabel('20.log | . |');
      title(['Sampling Period T = ' num2str(T) ' seconds']);
   subplot(2,1,2),
   %clf
      jump=1:50:k;semilogx(freq(jump),vfaseHgraus(jump),'*r');hold
on;semilogx(freq(jump),vexpGFaseGraus(jump),'^b');hold
on;semilogx(freq(jump),vDfaseGraus(jump),'vk')
      legend('Phase H*','Phase (1-exp(-sT)(G/s)*','Phase D*');
      semilogx(freq,-180,'k');
      %hold on;semilogx(freq,-540,'g');
      %hold on;semilogx(freq,-900,'g');
      %hold on;semilogx(freq,-1260,'g');
      %hold on;semilogx(freq,-1620,'g');
      %hold on;semilogx(freq,-1980,'g');
```

```
%hold on;semilogx(freq,-2340,'g');
hold on; semilogx(freq,vfaseHgraus,'r');hold on;semilogx(freq,vexpGFaseGraus,'b');hold on;semilogx(freq,vDfaseGraus,'k'); grid on;ylabel('Phase (degrees)');
xlabel('w (rd/s)');

disp('FIM')
```

CASO 2: PLANTA DE PRIMEIRA ORDEM E CONTROLE INTEGRAL MAPEADO PELO MÉTODO TUSTIN

```
%*****************************************************************
% TESE DE DOUTORADO NO INPE/DMC (CURSO ETE) - M. TREDINNICK
% ORIENTADOR: DR. MARCELO LOPES DE OLIVEIRA E SOUZA
%*****************************************************************
% DIAGRAMAS DE BODE
%*****************************************************************
% CASO 002:
%         PLANTA DE PRIMEIRA ORDEM
%         CONTROLE INTEGRAL
%         MAPEAMENTO TUSTIN
%*****************************************************************

clc; clear all

%*************************************************************************
% INICIALIZAÇAO DE CONSTANTES ITERATIVAS PARA A GRAVAÇAO DE
% VALORES DE H*(s)
%*************************************************************************
k=0;       %constante iterativa
a=1;       %constante existente na planta de segunda ordem G(s)=a/[s(s+a)]
T=2;       %periodo de amostragem identificado como provavel periodo
           % instabilizador no script ESCANEADOR
ganho = 1;  %ganho do controle

for w=10^-0.5:.005:10^-.1;
   w
```

```
%*************************************************************************
% PLANTA DE PRIMEIRA ORDEM JA COM ZERO ORDER HOLD:
% (Franklin (1981), equaçao 4.39)
% G(s) = a/(s+a)
% (1-exp(-sT)).(G(s)/s)* = 0.5/(exp(sT)-0.5)
% convençao usada no livro Franklin (1981): exp(-aT)=0.5
%*************************************************************************
        expG =  0.5 ./ [exp(i*w*T)-0.5];
        expGlog=20*log10(abs(expG));
        expGFase=angle(expG);
        expGFaseGraus=expGFase*180/pi;

        if expGFaseGraus > 0
            expGFaseGraus=-360+expGFaseGraus;
         end
        if (expGFaseGraus > -359.6351 & w > 10^0.49693)
            expGFaseGraus=-360+expGFaseGraus;
         end
        if (expGFaseGraus > -719.3 & w > 10^0.7981)
            expGFaseGraus=-360+expGFaseGraus;
         end
        if (expGFaseGraus > -1078.91 & w > 10^0.974051)
            expGFaseGraus=-360+expGFaseGraus;
         end
%*************************************************************************
%       CONTROLE
%*************************************************************************
         %*************************************************************************
         % CONTROLE INTEGRAL (mapeamento Tustin)
         %*************************************************************************
```

```
D = ganho * (T/2) * (exp(i*w*T) + 1) ./ (exp(i*w*T) - 1);
Dlog=20*log10(abs(D));
Dfase=angle(D);
DfaseGraus=Dfase*180/pi;

      if DfaseGraus > 0
         DfaseGraus=-360+DfaseGraus;
       end
      if (DfaseGraus >= -270.1 & w > 10^0.1959)
         DfaseGraus=+180+DfaseGraus;
       end
      if (DfaseGraus >= -90.1 & w > 10^0.49695)
         DfaseGraus=-180+DfaseGraus;
       end
      if (DfaseGraus >= -270.1 & w > 10^0.6732)
         DfaseGraus=+180+DfaseGraus;
       end
      if (DfaseGraus >= -90.1 & w > 10^0.79796)
         DfaseGraus=-180+DfaseGraus;
       end
      if (DfaseGraus >= -270.1 & w > 10^0.894871)
         DfaseGraus=+180+DfaseGraus;
       end
      if (DfaseGraus >= -90.1 & w > 10^0.974051)
         DfaseGraus=-180+DfaseGraus;
       end
      if (DfaseGraus >= -270.1 & w > 10^1.0412)
         DfaseGraus=+180+DfaseGraus;
       end
```

```
%************************************************************************
%************************************************************************
% FUNÇAO DE TRANSFERENCIA DE MALHA-ABERTA H*(S) = (ZERO
% ORDER HOLD).D*.(G/s)*
% H*(S) = (1-exp(-sT)) . D* . (G/s)*
% Ja calculei o HOLD junto com (G/s)* acima, nesse script.
%************************************************************************
%************************************************************************
            H= D .* expG;  %tem que colocar o Hold para o calculo da planta
                           % de segunda ordem
            modH=20*log10(abs(H));
            faseH=angle(H);
            faseHgraus=faseH*180/pi;

            if faseHgraus > 0
               faseHgraus=-360+faseHgraus;
            end
            %Ifs especificos apenas para dar continuidade aos diagramas
            if (faseHgraus >= -179.8 & w > 10^-0.1423)
               faseHgraus=+180+faseHgraus;
            end
            if (faseHgraus >= -550 & w > 10^0.196)
               faseHgraus=-360+faseHgraus;
            end
            %para o tracejado: T=0.5
                        % if (faseHgraus >= -516.1015 & (w > 10^-0.142065 &
w<10^0.1))
                         %  faseHgraus=+360+faseHgraus;
                         %end
            if (faseHgraus >= -630 & w > 10^0.417344) %<----------
```

```
    faseHgraus=+180+faseHgraus;
 end
  if (faseHgraus >= -450 & w > 10^0.497)
    faseHgraus=-360+faseHgraus;
 end
  if (faseHgraus >= -810 & w > 10^0.56407)
    faseHgraus=+180+faseHgraus;
 end
  if (faseHgraus >= -629.82 & w > 10^0.6733)
   faseHgraus=-360+faseHgraus;
end
  if (faseHgraus >= -990 & w > 10^0.7603)
   faseHgraus=+180+faseHgraus;
end
  if (faseHgraus >= -809.28 & w > 10^0.79796)
   faseHgraus=-360+faseHgraus;
end
  if (faseHgraus >= -1170 & w > 10^0.833147)
   faseHgraus=+180+faseHgraus;
end
  if (faseHgraus >= -988.95 & w > 10^0.8945)
   faseHgraus=-360+faseHgraus;
end
  if (faseHgraus >= -1350 & w > 10^0.94939)
   faseHgraus=+180+faseHgraus;
end
  if (faseHgraus >= -1169 & w > 10^0.974051)
   faseHgraus=-360+faseHgraus;
end
  if (faseHgraus >= -1529 & w > 10^0.9978)
   faseHgraus=+180+faseHgraus;
```

```
                end
                 if (faseHgraus >= -1348.811 & w > 10^1.0408)
                   faseHgraus=-360+faseHgraus;
                end
%montagem dos vetores de dados:
   k=k+1;
   freq(k)=w;
   vDlog(k)=Dlog;
   vDfaseGraus(k)=DfaseGraus;
   vexpG(k)=expGlog;
   vexpGFaseGraus(k)=expGFaseGraus;
   vmodH(k)=modH;
   vfaseHgraus(k)=faseHgraus;

end
%Plotagem de graficos Modulos e Fases:
   figure(1)
%    clf;
%                plotyy(freq,vDlog,freq,vDfaseGraus);grid on; ylabel('|D*| and
Phase(right)');title(['T = ' num2str(T) ' seconds']);
   %subplot(3,1,1), plotyy(freq,vDlog,freq,vDfaseGraus);grid on; ylabel('|D*| e
fase(a direita)');title(['T = ' num2str(T) ' seconds']);
   %subplot(3,1,2)
%    figure(2)
%    clf;
%    plotyy(freq,vexpG,freq,-180);
%    hold on;
%       plotyy(freq,vexpG,freq,vexpGFaseGraus);grid on; ylabel('20log|expG*| and
Phase(right)');
%    title(['Sampling Period T = ' num2str(T) ' seconds']);
   %subplot(3,1,3)
```

```
%    figure(3)
%    clf;
%    plotyy(freq,vmodH,freq,-180);
%    hold on;
%              plotyy(freq,vmodH,freq,vfaseHgraus);grid   on;   ylabel('20log|H*|   and
Phase(right)');
%    title(['Sampling Period T = ' num2str(T) ' seconds']);
%   figure(4)
%clf
   subplot(2,1,1),
      jump=1:30:k;semilogx(freq(jump),vmodH(jump),'*r');hold
on;semilogx(freq(jump),vexpG(jump),'^b');hold
on;semilogx(freq(jump),vDlog(jump),'vk')
      legend('|H*|','|(1-exp(-sT)(G/s)*|','|D*|');
      semilogx(freq,vmodH,'-r');hold on;semilogx(freq,vexpG,'-b');hold
on;semilogx(freq,vDlog,'-k');grid on; ylabel('20.log | . |');
      title(['Sampling Period T = ' num2str(T) ' seconds']);
   subplot(2,1,2),
   %clf
      jump=1:30:k;semilogx(freq(jump),vfaseHgraus(jump),'*r');hold
on;semilogx(freq(jump),vexpGFaseGraus(jump),'^b');hold
on;semilogx(freq(jump),vDfaseGraus(jump),'vk')
      legend('Phase H*','Phase (1-exp(-sT)(G/s)*','Phase D*');
      semilogx(freq,-180,'k');

      %hold on;semilogx(freq,-540,'g');
      %hold on;semilogx(freq,-900,'g');
      %hold on;semilogx(freq,-1260,'g');
      %hold on;semilogx(freq,-1620,'g');
      %hold on;semilogx(freq,-1980,'g');
      %hold on;semilogx(freq,-2340,'g');
```

```
        hold on; semilogx(freq,vfaseHgraus,'-r');hold
on;semilogx(freq,vexpGFaseGraus,'-b');hold on;semilogx(freq,vDfaseGraus,'-k');
grid on;ylabel('Phase (degrees)');
        xlabel('w (rd/s)');

disp('FIM')
```

G.4. PLOTAGEM GRÁFICA DA EXCURSÃO DE AUTOVALORES COMO FUNÇÃO DO PERÍODO DE AMOSTRAGEM

```
%*****************************************************************
% TESE DE DOUTORADO NO INPE/DMC (CURSO ETE) - M. TREDINNICK
% ORIENTADOR: DR. MARCELO LOPES DE OLIVEIRA E SOUZA
%*****************************************************************

% Foi usada a funçao do Matlab "pade", onde os coeficientes sao dispostos
% em ordem polinomial crescente. Aqui foi feito um truque para obter os
% coefientes Pade, usou-se o delay unitario no primeiro parametro da funçao
% e inverteu-se a ordem num para dem e dem para num, pois o primeiro termo
% desssa funçao e utilizado internamente como um valor negativo na
% exponencial, logo para obter os coefientes na forma correta da exp(.)
% basta inverter num com den. Por ex: [DEN,NUM] = pade(1,12)
clf;clear all
k=0;kk=0;pula=0;
for T=0.01:0.5:140; %sampling period
        A=[0 1; -3 -5]; %dynamics matrix
        Q=[50 100; 40 100];
        P=lyap(A,Q); %Liapunov equation AP+PA'=-Q with P and Q positive
                        % definites matrices.
        LAMBminQ=min(eig(Q)); %minimum eigenvalue of Q
        %Approxitation Padé[3,3] of the exponential exp(A*T):
        ExpAppr=inv(120*eye(2)-60*T*A+12*T^2*A^2-
T^3*A^3)*(120*eye(2)+60*T*A+12*T^2*A^2+T^3*A^3);
        %Approxitation Padé[6,6] of the exponential exp(A*T):
   ExpApprP6=inv(1*T^6*A^6 -42*T^5*A^5+ 840*T^4*A^4 -10080*T^3*A^3 +
75600*T^2*A^2 -332640*T^1*A^1+ 665280*eye(2) ) * (1*T^6*A^6 +42*T^5*A^5+
840*T^4*A^4 +10080*T^3*A^3 + 75600*T^2*A^2 +332640*T^1*A^1+
665280*eye(2) );
        %Approxitation Padé[8,8] of the exponential exp(A*T):
        ExpApprP8=inv(518918400*eye(2)-259459200*T*A+60540480*T^2*A^2-
8648640*T^3*A^3+831600*T^4*A^4-55440*T^5*A^5+2520*T^6*A^6-
72*T^7*A^7+T^8*A^8)*(518918400*eye(2)+259459200*T*A+60540480*T^2*A^2+8
```

```
648640*T^3*A^3+831600*T^4*A^4+55440*T^5*A^5+2520*T^6*A^6+72*T^7*A^7+
T^8*A^8);

   %Approxitation Padé[10,10] of the exponential exp(A*T):

   ExpApprP10=inv(1*T^10*A^10 -110*T^9*A^9 +  5940*T^8*A^8  -
205920*T^7*A^7 + 5045040*T^6*A^6 -90810720*T^5*A^5 + 1210809600*T^4*A^4
-11762150400*T^3*A^3 +  79394515200*T^2*A^2  -335221286400*T^1*A^1 +
670442572800*eye(2) ) * (1*T^10*A^10 +110*T^9*A^9 +  5940*T^8*A^8
+205920*T^7*A^7 + 5045040*T^6*A^6 +90810720*T^5*A^5 +
1210809600*T^4*A^4  +11762150400*T^3*A^3 +  79394515200*T^2*A^2
+335221286400*T^1*A^1 + 670442572800*eye(2) );

   %Approxitation Padé[12,12] of the exponential exp(A*T):

   ExpApprP12=inv(1*T^12*A^12 -156*T^11*A^11 + 12012*T^10*A^10  -
600600*T^9*A^9 +  21621600*T^8*A^8  -588107520*T^7*A^7 +
12350257920*T^6*A^6  -201132771840*T^5*A^5 + 2514159648000*T^4*A^4  -
23465490048000*T^3*A^3  +  154872234316800*T^2*A^2  -
647647525324800*T^1*A^1 + 1.2952950506496e+015*eye(2) )*(1*T^12*A^12
+156*T^11*A^11 + 12012*T^10*A^10  + 600600*T^9*A^9 +  21621600*T^8*A^8
+588107520*T^7*A^7 +  12350257920*T^6*A^6  +201132771840*T^5*A^5 +
2514159648000*T^4*A^4  +23465490048000*T^3*A^3  +
154872234316800*T^2*A^2  +647647525324800*T^1*A^1 +
1.2952950506496e+015*eye(2) );

   %Approxitation Padé[14,14] of the exponential exp(A*T):

   ExpApprP14=inv(1*T^14*A^14  -210*T^13*A^13 + 21840*T^12*A^12  -
1485120*T^11*A^11  +  73513440*T^10*A^10  -2793510720*T^9*A^9 +
83805321600*T^8*A^8  -2011327718400*T^7*A^7 +  38718058579200*T^6*A^6
-593676898214400*T^5*A^5  +  7.1241227785728e+015*T^4*A^4  -
6.476475253248e+016*T^3*A^3  + 4.2097089146112e+017*T^2*A^2  -
1.74864831837696e+018*T^1*A^1  +  3.49729663675392e+018*eye(2) )
*(1*T^14*A^14  +210*T^13*A^13 + 21840*T^12*A^12  +1485120*T^11*A^11  +
73513440*T^10*A^10  +2793510720*T^9*A^9 +  83805321600*T^8*A^8
+2011327718400*T^7*A^7 +  38718058579200*T^6*A^6
+593676898214400*T^5*A^5  +  7.1241227785728e+015*T^4*A^4
+6.476475253248e+016*T^3*A^3  + 4.2097089146112e+017*T^2*A^2
+1.74864831837696e+018*T^1*A^1  +  3.49729663675392e+018*eye(2) ) ;

   %Approxitation Padé[16,16] of the exponential exp(A*T):

   ExpApprP16=inv(1*T^16*A^16 -272*T^15*A^15 + 36720*T^14*A^14  -
3255840*T^13*A^13  + 211629600*T^12*A^12  -10666131840*T^11*A^11 +
430200650880*T^10*A^10  -14135164243200*T^9*A^9 +
381649434566400*T^8*A^8  -8.48109854592e+015*T^7*A^7 +
1.54355993535744e+017*T^6*A^6  -2.27324281389005e+018*T^5*A^5 +
2.65211661620506e+019*T^4*A^4  -2.36650405753682e+020*T^3*A^3 +
1.52132403698796e+021*T^2*A^2  -6.28813935288355e+021*T^1*A^1 +
1.25762787057671e+022*eye(2) ) * (1*T^16*A^16 +272*T^15*A^15 +
```

36720*T^14*A^14 +3255840*T^13*A^13 + 211629600*T^12*A^12
+10666131840*T^11*A^11 + 430200650880*T^10*A^10
+14135164243200*T^9*A^9 + 381649434566400*T^8*A^8
+8.48109854592e+015*T^7*A^7 + 1.54355993535744e+017*T^6*A^6
+2.27324281389005e+018*T^5*A^5 + 2.65211661620506e+019*T^4*A^4
+2.36650405753682e+020*T^3*A^3 + 1.52132403698796e+021*T^2*A^2
+6.28813935288355e+021*T^1*A^1 + 1.25762787057671e+022*eye(2));

*%Approxitation Padé[20,20] of the exponential exp(A*T):*

% ExpApprP20=inv(1*T^20*A^20 -420*T^19*A^19 + 87780*T^18*A^18 -
12113640*T^17*A^17 + 1235591280*T^16*A^16 -98847302400*T^15*A^15 +
6425074656000*T^14*A^14 -346954031424000*T^13*A^13 +
1.5786408429792e+016*T^12*A^12 -6.10407792618624e+017*T^11*A^11 +
2.01434571564146e+019*T^10*A^10 -5.6767924713532e+020*T^9*A^9 +
1.36243019312477e+022*T^8*A^8 -2.76678131526876e+023*T^7*A^7 +
4.7035282359569e+024*T^6*A^6 -6.58493953033965e+025*T^5*A^5 +
7.40805697163211e+026*T^4*A^4 -6.44936724589149e+027*T^3*A^3 +
4.08459925573127e+028*T^2*A^2 -1.67683548393179e+029*T^1*A^1 +
3.35367096786357e+029*eye(2)) *(1*T^20*A^20 +420*T^19*A^19 +
87780*T^18*A^18 +12113640*T^17*A^17 + 1235591280*T^16*A^16
+98847302400*T^15*A^15 + 6425074656000*T^14*A^14
+346954031424000*T^13*A^13 + 1.5786408429792e+016*T^12*A^12
+6.10407792618624e+017*T^11*A^11 + 2.01434571564146e+019*T^10*A^10
+5.6767924713532e+020*T^9*A^9 + 1.36243019312477e+022*T^8*A^8
+2.76678131526876e+023*T^7*A^7 + 4.7035282359569e+024*T^6*A^6
+6.58493953033965e+025*T^5*A^5 + 7.40805697163211e+026*T^4*A^4
+6.44936724589149e+027*T^3*A^3 + 4.08459925573127e+028*T^2*A^2
+1.67683548393179e+029*T^1*A^1 + 3.35367096786357e+029*eye(2));

*%Approx. Matlab of the exponential exp(A*T):*

*%ExpAppr1=exp(A*T);*

*%LAMBmaxMAT=max(eig(ExpAppr1'*P*ExpAppr1)); %maximum eigenvalue by MatLab app.*

LAMBmaxPAD3=max(eig(ExpAppr'*P*ExpAppr)); %maximum eigenvalue by PADE 3 app.

LAMBmaxPAD6=max(eig(ExpApprP6'*P*ExpApprP6)); %maximum eigenvalue by PADE 6 app.

LAMBmaxPAD8=max(eig(ExpApprP8'*P*ExpApprP8)); %maximum eigenvalue by PADE 8 app.

LAMBmaxPAD10=max(eig(ExpApprP10'*P*ExpApprP10)); %maximum eigenvalue by PADE 10 app.

```
        LAMBmaxPAD12=max(eig(ExpApprP12'*P*ExpApprP12)); %maximum
eigenvalue by PADE 12 app.
        LAMBmaxPAD14=max(eig(ExpApprP14'*P*ExpApprP14)); %maximum
eigenvalue by PADE 14 app.
    LAMBmaxPAD16=max(eig(ExpApprP16'*P*ExpApprP16)); %maximum
eigenvalue by PADE 16 app.
%       LAMBmaxPAD20=max(eig(ExpApprP20'*P*ExpApprP20)); %maximum
eigenvalue by PADE 20 app.
    k=k+1;
        time(k)=T;
        gr1(k)=LAMBmaxPAD3;
        gr2(k)=LAMBmaxPAD6;
        gr3(k)=LAMBmaxPAD8;
        gr4(k)=LAMBmaxPAD10;
        gr5(k)=LAMBmaxPAD12;
        gr6(k)=LAMBmaxPAD14;
        gr7(k)=LAMBmaxPAD16;
    grp(k)=LAMBminQ;
    %loop para plotar marcadores nos graficos bem espaçados:
    kk=kk+1;
    if kk==40
       kk=0;pula=pula+1;
        timex(pula)=T;
              gr1x(pula)=LAMBmaxPAD3;
              gr2x(pula)=LAMBmaxPAD6;
              gr3x(pula)=LAMBmaxPAD8;
              gr4x(pula)=LAMBmaxPAD10;
              gr5x(pula)=LAMBmaxPAD12;
              gr6x(pula)=LAMBmaxPAD14;
              gr7x(pula)=LAMBmaxPAD16;
         grpx(pula)=LAMBminQ;
    end
end
figure(1)
```

```
hold on
%marcadores graficos:
plot(timex,gr1x,'+b','linewidth',2);
plot(timex,gr2x,'xm','linewidth',2);
plot(timex,gr3x,'*g','linewidth',2);
plot(timex,gr4x,'sc','linewidth',2);
plot(timex,gr5x,'vr','linewidth',2);
plot(timex,gr6x,'^r','linewidth',2);
plot(timex,gr7x,'pk','linewidth',2);

plot(time,grp,'--k','linewidth',2);
%plot(T,LAMBmaxMAT,'-r','linewidth',3);
plot(time,gr1,'-b','linewidth',2);
plot(time,gr2,'-m','linewidth',2);
plot(time,gr3,'-g','linewidth',2);
plot(time,gr4,'-c','linewidth',2);
plot(time,gr5,'-r','linewidth',2);
plot(time,gr6,'-r','linewidth',2);
plot(time,gr7,'-k','linewidth',2);
legend('Lamb Max Padé [3,3]','Lamb Max Padé [6,6]','Lamb Max Padé [8,8]','Lamb
Max Padé [10,10]','Lamb Max Padé [12,12]','Lamb Max Padé [14,14]','Lamb Max
Padé [16,16]','Lamb Min (P)')
grid on
xlabel('T (sec.)'); ylabel('Eigenvalues') ;title('Eigenvalues evolutions vs. Sampling
Period');
%-------------------------------------------------------
%>> [DEN,NUM] = PADE(1,6)

%DEN =      1        -42         840      -10080       75600     -332640      665280
%NUM =      1         42         840       10080       75600      332640      665280
%-------------------------------------------------------
%>> [DEN,NUM] = PADE(1,10)
%DEN =  1    -110     5940    -205920    5045040    -90810720    1210809600    -
%11762150400    79394515200   -335221286400    670442572800
```

```
%NUM =  1     110     5940     205920     5045040     90810720     1210809600
%11762150400     79394515200     335221286400     670442572800

%-------------------------------------------------------

%>> [DEN,NUM] = PADE(1,12)

%DEN = 1          -156         12012         -600600         21621600         -588107520
%12350257920         -201132771840         2514159648000         -
%23465490048000         154872234316800         -647647525324800
%1.2952950506496e+015

%NUM =  1         156         12012         600600         21621600         588107520
%12350257920         201132771840         2514159648000         23465490048000
%154872234316800         647647525324800     1.2952950506496e+015

%-------------------------------------------------------

%>> [DEN,NUM] = PADE(1,14)

%DEN =    1         -210         21840         -1485120         73513440         -2793510720
%83805321600         -2011327718400     38718058579200         -
%593676898214400     7.1241227785728e+015     -6.476475253248e+016
%4.2097089146112e+017     -1.74864831837696e+018
%3.49729663675392e+018

%NUM =    1         210         21840         1485120         73513440         2793510720
%83805321600         2011327718400     38718058579200
%593676898214400     7.1241227785728e+015     6.476475253248e+016
%4.2097089146112e+017     1.74864831837696e+018
%3.49729663675392e+018

%-------------------------------------------------------

%>> [DEN,NUM] = PADE(1,16)

%DEN =  1     -272     36720     -3255840     211629600     -10666131840
%430200650880     -14135164243200     381649434566400     -
%8.48109854592e+015     1.54355993535744e+017     -2.27324281389005e+018
%2.65211661620506e+019     -2.36650405753682e+020
%1.52132403698796e+021     -6.28813935288355e+021
%1.25762787057671e+022

%NUM =  1     272     36720     3255840     211629600     10666131840
%430200650880     14135164243200     381649434566400
%8.48109854592e+015     1.54355993535744e+017     2.27324281389005e+018
%2.65211661620506e+019     2.36650405753682e+020
%1.52132403698796e+021     6.28813935288355e+021
%1.25762787057671e+022

%-------------------------------------------------------

%>> [DEN,NUM] = PADE(1,20)
```

%DEN = 1 -420 87780 -12113640 1235591280 -98847302400
%6425074656000 -346954031424000 1.5786408429792e+016 -
%6.10407792618624e+017 2.01434571564146e+019 -
%5.6767924713532e+020 1.36243019312477e+022 -
%2.76678131526876e+023 4.7035282359569e+024 -
%6.58493953033965e+025 7.40805697163211e+026 -
%6.44936724589149e+027 4.08459925573127e+028 -
%1.67683548393179e+029 3.35367096786357e+029

%NUM = 1 420 87780 12113640 1235591280 98847302400
%6425074656000 346954031424000 1.5786408429792e+016
%6.10407792618624e+017 2.01434571564146e+019
%5.6767924713532e+020 1.36243019312477e+022
%2.76678131526876e+023 4.7035282359569e+024
%6.58493953033965e+025 7.40805697163211e+026
%6.44936724589149e+027 4.08459925573127e+028
%1.67683548393179e+029 3.35367096786357e+029

G.5. SCRIPT PARA GERAR AS SUPERFÍCIES DO TEOREMA 7.2 DO CAPÍTULO 7

```
%Geratriz de superficie T x Csi para o Teorema 2 desta Tese
A=[0 1; -3 -5]; %dynamics matrix
Q=[50 100; 40 100];
P=lyap(A,Q); %Liapunov equation AP+PA'=-Q with P and Q positive definites
                % matrices.
LAMBminP=min(eig(P)); %minimum eigenvalue of Q
[T,csi] = meshgrid(0:.5:140, 0:.00356:1);
   %Montagem da matriz de raios espectrais
   contT=0;
      for sp=0:.5:140   %sp é o sampling period para nao misturar o o T do
                        % meshgrid
         sp=sp+2.7;
         contT=contT+1
         contCSI=0;
         for cs=0:.00356:1   %cs é o CSI para nao misturar o o csi do meshgrid
            cs=cs+.0;
               contCSI=contCSI+1
               teorema2(contT,contCSI)=max(eig(
(2*eye(2)+cs*sp*A)'*inv(2*eye(2)*sp*A)'*P*inv(2*eye(2)*sp*A)*(2*eye(2)+cs*sp*A)
));   %maximum eigenvalue;
         end
      end
```

```
figure(1);
surfl(T,csi,teorema2);
shading interp
colormap(copper)
hold on;
[e,f]=size(teorema2);Z1=ones(e)*LAMBminP;
mesh(T,csi,Z1);
xlabel('T'); ylabel('Csi'); zlabel('Rho');
%shading faceted;
shading interp
colormap(copper)
```

G.6. SCRIPT PARA GERAR AS SUPERFÍCIES DO TEOREMA 7.3 DO CAPÍTULO 7

```
clc;clear all
%Geratriz de superficie T x Csi para o teorema 3 desta Tese
A=[0 1; -3 -5]; %dynamics matrix
Q=[50 100; 40 100];
P=lyap(A,Q); %Liapunov equation AP+PA'=-Q with P and Q positive definites
                % matrices.
LAMBminP=min(eig(P)); %minimum eigenvalue of Q
T=3;    % Atribui-se um valor para T e verifica-se o raio espectral funçao de T, csi1
        %e csi2 so que visualizado o grafico somente de rho x csi1 x csi2
 [csi1,csi2] = meshgrid(0:.01:1, 0:.01:1);
   %Montagem da matriz de raios espectrais
   contCSI1=0;
      for cs1=0:.01:1   %cs1 e o csi1 para nao confundir com a variavel do
                        % meshgrid
         %csi1=csi1+.1;
         contCSI1=contCSI1+1
         contCSI2=0;
         for cs2=0:.01:1   %cs2 e o csi2 para nao confundir com a variavel do
                           % meshgrid
            %csi2=csi2+.01;
              contCSI2=contCSI2+1
```

```
                teorema3(contCSI1,contCSI2) = max(eig(
(2*cs1*eye(2)+cs2*T*A)'*inv(2*eye(2)*T*A)'*P*inv(2*eye(2)*T*A)*(2*cs1*eye(2)+cs
2*T*A)   ));   %maximum eigenvalue;
            end
        end
figure(2);
surfl(csi1,csi2,teorema3);
shading interp
colormap(copper)
hold on;
[e,f]=size(teorema3);Z1=ones(e)*LAMBminP;
mesh(csi1,csi2,Z1);
xlabel('Csi1'); ylabel('Csi2'); zlabel('Rho');
title(['T = ' num2str(T) ' segundos']);
%shading faceted;
shading interp
colormap(copper)
```

G.7. SCRIPT PARA PLOTAR O GRÁFICO DE CONVERGÊNCIA PARA O SISTEMA EM MALHA FECHADA AMOSTRADOR E PLANTA DO CAPÍTULO 6

```
clc;clear all
k=0;soma=0;T=2;
for w=0:0.01:5;
    f=norm((1-exp(-w*T))/w,inf)
    %f=(1-exp(-w*T))/w;
    k=k+1;
    vector(k)=f;
    soma=soma+f
    x(k)=w;
end
figure (1); clf
plot(x,vector,'-k','linewidth',2);grid on
%axis([0 70 0 0.6]);
ylabel(['norm(1-exp(-wT))/w) ; T = ' num2str(T) ' segundos']);xlabel('w');
```

G.8. SCRIPT PARA PLOTAR O GRÁFICO DE CONVERGÊNCIA PARA A PLANTA G(s) DE SEGUNDA ORDEM USADA DO CAPÍTULO 6

```
clc;clear all
T=2;
a=1;b=.1;
w=-pi/T:.01:pi/T;
kk=0;soma=0;cont=0;
contW=0;
dimW=-pi/T:.1:pi/T;[qwer,aaa]=size(dimW)
[w,k] = meshgrid(dimW, 1:1:aaa);
   %Montagem da matriz
   contCSI1=0;
      for cw=-pi/T:.1:pi/T   %cs1 e o csi1 para nao confundir com a variavel do meshgrid
         %csi1=csi1+.1;
         contW=contW+1
         contK=0;
         for ck=1:1:aaa   %cw e o ck para nao confundir com a variavel do meshgrid
            %csi2=csi2+.01;
               contK=contK+1
               vector(contW,contK) = norm(1/( (cw-2*(ck+1)*pi/T)^2+(cw-
2*(ck+1)*pi/T)*a+b ),inf)  /  norm(1/( (cw-2*ck*pi/T)^2+(cw-2*ck*pi/T)*a+b ),inf);
         end
      end
surfl(w,k,vector);
shading interp
```

```
colormap(copper)
hold on;
[e,f]=size(vector);Z1=ones(e);
mesh(w,k,Z1);
xlabel('w'); ylabel('k'); zlabel('L');
title(['T = ' num2str(T) ' segundos']);
%shading faceted;
shading interp
colormap(copper)
```

G.9. SCRIPTS MATLAB “CIRCLE” E “ARROW” (DISPONÍVEIS NO WEBSITE DA EMPRESA THE MATHWORKS)

1.Script “arrow.m”:

```
function [h,yy,zz] = arrow(varargin)
% ARROW  Draw a line with an arrowhead.
%
%  ARROW(Start,Stop) draws a line with an arrow from Start to Stop (points
%       should be vectors of length 2 or 3, or matrices with 2 or 3
%       columns), and returns the graphics handle of the arrow(s).
%
%  ARROW uses the mouse (click-drag) to create an arrow.
%
%  ARROW DEMO & ARROW DEMO2 show 3-D & 2-D demos of the capabilities
of ARROW.
%
%  ARROW may be called with a normal argument list or a property-based list.
%       ARROW(Start,Stop,Length,BaseAngle,TipAngle,Width,Page,CrossDir) is
%       the full normal argument list, where all but the Start and Stop
%       points are optional.  If you need to specify a later argument (e.g.,
%       Page) but want default values of earlier ones (e.g., TipAngle),
%       pass an empty matrix for the earlier ones (e.g., TipAngle=[]).
%
%  ARROW('Property1',PropVal1,'Property2',PropVal2,...) creates arrows with the
%       given properties, using default values for any unspecified or given as
%       'default' or NaN.  Some properties used for line and patch objects are
%       used in a modified fashion, others are passed directly to LINE, PATCH,
%       or SET.  For a detailed properties explanation, call ARROW PROPERTIES.
%
%       Start         The starting points.                B
```

```
%       Stop          The end points.                  /|\          ^
%       Length        Length of the arrowhead in pixels.     /|||\          |
%       BaseAngle     Base angle in degrees (ADE).          //|||\\        L|
%       TipAngle      Tip angle in degrees (ABC).          ///|||\\\       e|
%       Width         Width of the base in pixels.        ////|||\\\\      n|
%       Page          Use hardcopy proportions.           /////|D|\\\\\     g|
%       CrossDir      Vector || to arrowhead plane.      ////  |||  \\\\    t|
%       NormalDir     Vector out of arrowhead plane.    ///    |||    \\\   h|
%       Ends          Which end has an arrowhead.      //<----->||      \\   |
%       ObjectHandles Vector of handles to update.   /   base |||       \  V
%                                              E    angle||<-------->C
%  ARROW(H,'Prop1',PropVal1,...), where H is a                |||tipangle
%       vector of handles to previously-created arrows        |||
%       and/or line objects, will update the previously-      |||
%       created arrows according to the current view       -->|A|<-- width
%       and any specified properties, and will convert
%       two-point line objects to corresponding arrows.  ARROW(H) will update
%       the arrows if the current view has changed.  Root, figure, or axes
%       handles included in H are replaced by all descendant Arrow objects.
%
%  A property list can follow any specified normal argument list, e.g.,
%  ARROW([1 2 3],[0 0 0],36,'BaseAngle',60) creates an arrow from (1,2,3) to
%  the origin, with an arrowhead of length 36 pixels and 60-degree base angle.
%
%  The basic arguments or properties can generally be vectorized to create
%  multiple arrows with the same call.  This is done by passing a property
%  with one row per arrow, or, if all arrows are to have the same property
%  value, just one row may be specified.
%
%   You may want to execute AXIS(AXIS) before calling ARROW so it doesn't
change
```

```
%  the axes on you; ARROW determines the sizes of arrow components BEFORE
the
%  arrow is plotted, so if ARROW changes axis limits, arrows may be malformed.
%
%  This version of ARROW uses features of MATLAB 5 and is incompatible with
%  earlier MATLAB versions (ARROW for MATLAB 4.2c is available separately);
%  some problems with perspective plots still exist.

% Copyright (c)1995-2002, Dr. Erik A. Johnson <JohnsonE@usc.edu>, 11/15/02

% Revision history:
%   11/15/02  EAJ  Accomodate how MATLAB 6.5 handles NaN and logicals
%    7/28/02  EAJ  Tried (but failed) work-around for MATLAB 6.x / OpenGL bug
%                  if zero 'Width' or not double-ended
%   11/10/99  EAJ  Add logical() to eliminate zero index problem in MATLAB 5.3.
%   11/10/99  EAJ  Corrected warning if axis limits changed on multiple axes.
%   11/10/99  EAJ  Update e-mail address.
%    2/10/99  EAJ  Some documentation updating.
%    2/24/98  EAJ  Fixed bug if Start~=Stop but both colinear with viewpoint.
%     8/14/97   EAJ   Added workaround for MATLAB 5.1 scalar logical transpose
bug.
%    7/21/97  EAJ  Fixed a few misc bugs.
%    7/14/97  EAJ  Make arrow([],'Prop',...) do nothing (no old handles)
%    6/23/97  EAJ  MATLAB 5 compatible version, release.
%    5/27/97  EAJ  Added Line Arrows back in.  Corrected a few bugs.
%    5/26/97  EAJ  Changed missing Start/Stop to mouse-selected arrows.
%    5/19/97  EAJ  MATLAB 5 compatible version, beta.
%    4/13/97  EAJ  MATLAB 5 compatible version, alpha.
%    1/31/97  EAJ  Fixed bug with multiple arrows and unspecified Z coords.
%   12/05/96  EAJ  Fixed one more bug with log plots and NormalDir specified
%   10/24/96  EAJ  Fixed bug with log plots and NormalDir specified
```

```
%   11/13/95  EAJ  Corrected handling for 'reverse' axis directions
%   10/06/95  EAJ  Corrected occasional conflict with SUBPLOT
%    4/24/95  EAJ  A major rewrite.
%    Fall 94  EAJ  Original code.

% Things to be done:
%  - segment parsing, computing, and plotting into separate subfunctions
%  - change computing from Xform to Camera paradigms
%     + this will help especially with 3-D perspective plots
%     + if the WarpToFill section works right, remove warning code
%     + when perpsective works properly, remove perspective warning code
%  - add cell property values and struct property name/values (like get/set)
%  - get rid of NaN as the "default" data label
%     + perhaps change userdata to a struct and don't include (or leave
%       empty) the values specified as default; or use a cell containing
%       an empty matrix for a default value
%  - add functionality of GET to retrieve current values of ARROW properties

% Many thanks to Keith Rogers <kerog@ai.mit.com> for his many excellent
% suggestions and beta testing.  Check out his shareware package MATDRAW
% (at ftp://ftp.mathworks.com/pub/contrib/v5/graphics/matdraw/) -- he has
% permission to distribute ARROW with MATDRAW.

% Permission is granted to distribute ARROW with the toolboxes for the book
% "Solving Solid Mechanics Problems with MATLAB 5", by F. Golnaraghi et al.
% (Prentice Hall, 1999).

% global variable initialization
global ARROW_PERSP_WARN ARROW_STRETCH_WARN ARROW_AXLIMITS
if isempty(ARROW_PERSP_WARN  ), ARROW_PERSP_WARN  =1; end;
if isempty(ARROW_STRETCH_WARN), ARROW_STRETCH_WARN=1; end;
```

```
% Handle callbacks
if (nargin>0 & isstr(varargin{1}) & strcmp(lower(varargin{1}),'callback')),
  arrow_callback(varargin{2:end}); return;
end;

% Are we doing the demo?
c = sprintf('\n');
if (nargin==1 & isstr(varargin{1})),
  arg1 = lower(varargin{1});
  if strncmp(arg1,'prop',4), arrow_props;
  elseif strncmp(arg1,'demo',4)
    clf reset
    demo_info = arrow_demo;
    if ~strncmp(arg1,'demo2',5),
        hh=arrow_demo3(demo_info);
    else,
        hh=arrow_demo2(demo_info);
    end;
    if (nargout>=1), h=hh; end;
  elseif strncmp(arg1,'fixlimits',3),
    arrow_fixlimits(ARROW_AXLIMITS);
    ARROW_AXLIMITS=[];
  elseif strncmp(arg1,'help',4),
    disp(help(mfilename));
  else,
    error([upper(mfilename) ' got an unknown single-argument string '"
deblank(arg1) '".']);
  end;
  return;
end;
```

```
% Check # of arguments
if (nargout>3), error([upper(mfilename) ' produces at most 3 output arguments.']);
end;

% find first property number
firstprop = nargin+1;
for k=1:length(varargin), if ~isnumeric(varargin{k}), firstprop=k; break; end; end;
lastnumeric = firstprop-1;

% check property list
if (firstprop<=nargin),
  for k=firstprop:2:nargin,
    curarg = varargin{k};
    if ~isstr(curarg) | sum(size(curarg)>1)>1,
        error([upper(mfilename) ' requires that a property name be a single string.']);
    end;
  end;
  if (rem(nargin-firstprop,2)~=1),
    error([upper(mfilename) ' requires that the property ''' ...
          varargin{nargin} ''' be paired with a property value.']);
  end;
end;

% default output
if (nargout>0), h=[]; end;
if (nargout>1), yy=[]; end;
if (nargout>2), zz=[]; end;

% set values to empty matrices
start      = [];
```

```
stop       = [];
len        = [];
baseangle  = [];
tipangle   = [];
wid        = [];
page       = [];
crossdir   = [];
ends       = [];
ax         = [];
oldh       = [];
ispatch    = [];
defstart      = [NaN NaN NaN];
defstop       = [NaN NaN NaN];
deflen        = 16;
defbaseangle  = 90;
deftipangle   = 16;
defwid        = 0;
defpage       = 0;
defcrossdir   = [NaN NaN NaN];
defends       = 1;
defoldh       = [];
defispatch    = 1;

% The 'Tag' we'll put on our arrows
ArrowTag = 'Arrow';

% check for oldstyle arguments
if (firstprop==2),
  % assume arg1 is a set of handles
  oldh = varargin{1}(:);
  if isempty(oldh), return; end;
```

```
elseif (firstprop>9),
  error([upper(mfilename) ' takes at most 8 non-property arguments.']);
elseif (firstprop>2),
  s = str2mat('start','stop','len','baseangle','tipangle','wid','page','crossdir');
  for k=1:firstprop-1, eval([deblank(s(k,:)) '=varargin{k};']); end;
end;

% parse property pairs
extraprops={};
for k=firstprop:2:nargin,
  prop = varargin{k};
  val  = varargin{k+1};
  prop = [lower(prop(:)') '      '];
  if     strncmp(prop,'start' ,5),   start      = val;
  elseif strncmp(prop,'stop'  ,4),   stop       = val;
  elseif strncmp(prop,'len'   ,3),   len        = val(:);
  elseif strncmp(prop,'base'  ,4),   baseangle  = val(:);
  elseif strncmp(prop,'tip'   ,3),   tipangle   = val(:);
  elseif strncmp(prop,'wid'   ,3),   wid        = val(:);
  elseif strncmp(prop,'page'  ,4),   page       = val;
  elseif strncmp(prop,'cross' ,5),   crossdir   = val;
  elseif    strncmp(prop,'norm'        ,4),           if    (isstr(val)),    crossdir=val;    else,
crossdir=val*sqrt(-1); end;
  elseif strncmp(prop,'end'   ,3),   ends       = val;
  elseif strncmp(prop,'object',6),   oldh       = val(:);
  elseif strncmp(prop,'handle',6),   oldh       = val(:);
  elseif strncmp(prop,'type'  ,4),   ispatch    = val;
  elseif strncmp(prop,'userd' ,5),   %ignore it
  else,
    % make sure it is a valid patch or line property
    eval('get(0,[''DefaultPatch'' varargin{k}]);err=0;','err=1;'); errstr=lasterr;
```

```
    if (err), eval('get(0,[''DefaultLine'' varargin{k}]);err=0;','err=1;'); end;
    if (err),
        errstr(1:max(find(errstr==setstr(13)|errstr==setstr(10)))) = '';
        error([upper(mfilename) ' got ' errstr]);
    end;
    extraprops={extraprops{:},varargin{k},val};
  end;
end;

% Check if we got 'default' values
start     = arrow_defcheck(start    ,defstart    ,'Start'        );
stop      = arrow_defcheck(stop     ,defstop     ,'Stop'         );
len       = arrow_defcheck(len      ,deflen      ,'Length'       );
baseangle = arrow_defcheck(baseangle,defbaseangle,'BaseAngle'    );
tipangle  = arrow_defcheck(tipangle ,deftipangle ,'TipAngle'     );
wid       = arrow_defcheck(wid      ,defwid      ,'Width'        );
crossdir  = arrow_defcheck(crossdir ,defcrossdir ,'CrossDir'     );
page      = arrow_defcheck(page     ,defpage     ,'Page'         );
ends      = arrow_defcheck(ends     ,defends     ,''             );
oldh      = arrow_defcheck(oldh     ,[]          ,'ObjectHandles');
ispatch   = arrow_defcheck(ispatch  ,defispatch  ,''             );

% check transpose on arguments
[m,n]=size(start   );   if any(m==[2 3])&(n==1|n>3),   start    = start';      end;
[m,n]=size(stop    );   if any(m==[2 3])&(n==1|n>3),   stop     = stop';       end;
[m,n]=size(crossdir);   if any(m==[2 3])&(n==1|n>3),   crossdir = crossdir';   end;

% convert strings to numbers
if ~isempty(ends) & isstr(ends),
  endsorig = ends;
  [m,n] = size(ends);
```

```
  col = lower([ends(:,1:min(3,n)) ones(m,max(0,3-n))*' ']);
  ends = NaN*ones(m,1);
  oo = ones(1,m);
  ii=find(all(col'==['non']'*oo)'); if ~isempty(ii), ends(ii)=ones(length(ii),1)*0; end;
  ii=find(all(col'==['sto']'*oo)'); if ~isempty(ii), ends(ii)=ones(length(ii),1)*1; end;
  ii=find(all(col'==['sta']'*oo)'); if ~isempty(ii), ends(ii)=ones(length(ii),1)*2; end;
  ii=find(all(col'==['bot']'*oo)'); if ~isempty(ii), ends(ii)=ones(length(ii),1)*3; end;
  if any(isnan(ends)),
    ii = min(find(isnan(ends)));
    error([upper(mfilename) ' does not recognize '" deblank(endsorig(ii,:)) '" as a
valid "Ends" value.']);
  end;
else,
  ends = ends(:);
end;
if ~isempty(ispatch) & isstr(ispatch),
  col = lower(ispatch(:,1));
  patchchar='p'; linechar='l'; defchar=' ';
  mask = col~=patchchar & col~=linechar & col~=defchar;
  if any(mask),
    error([upper(mfilename) ' does not recognize '"
deblank(ispatch(min(find(mask)),:)) '" as a valid "Type" value.']);
  end;
  ispatch = (col==patchchar)*1 + (col==linechar)*0 + (col==defchar)*defispatch;
else,
  ispatch = ispatch(:);
end;
oldh = oldh(:);

% check object handles
if ~all(ishandle(oldh)), error([upper(mfilename) ' got invalid object handles.']); end;
```

```
% expand root, figure, and axes handles
if ~isempty(oldh),
  ohtype = get(oldh,'Type');
  mask = strcmp(ohtype,'root') | strcmp(ohtype,'figure') | strcmp(ohtype,'axes');
  if any(mask),
    oldh = num2cell(oldh);
    for ii=find(mask)',
        oldh(ii) = {findobj(oldh{ii},'Tag',ArrowTag)};
    end;
    oldh = cat(1,oldh{:});
    if isempty(oldh), return; end; % no arrows to modify, so just leave
  end;
end;

% largest argument length
[mstart,junk]=size(start); [mstop,junk]=size(stop); [mcrossdir,junk]=size(crossdir);
argsizes = [length(oldh) mstart mstop                              ...
           length(len) length(baseangle) length(tipangle)         ...
          length(wid) length(page) mcrossdir length(ends) ];
args=['length(ObjectHandle)  '; ...
     '#rows(Start)          '; ...
     '#rows(Stop)           '; ...
     'length(Length)        '; ...
     'length(BaseAngle)     '; ...
     'length(TipAngle)      '; ...
     'length(Width)         '; ...
     'length(Page)          '; ...
     '#rows(CrossDir)       '; ...
    '#rows(Ends)           '];
if (any(imag(crossdir(:))~=0)),
```

```
  args(9,:) = '#rows(NormalDir)      ';
end;
if isempty(oldh),
  narrows = max(argsizes);
else,
  narrows = length(oldh);
end;
if (narrows<=0), narrows=1; end;

% Check size of arguments
ii = find((argsizes~=0)&(argsizes~=1)&(argsizes~=narrows));
if ~isempty(ii),
  s = args(ii',:);
  while ((size(s,2)>1)&((abs(s(:,size(s,2)))==0)|(abs(s(:,size(s,2)))==abs(' ')))),
    s = s(:,1:size(s,2)-1);
  end;
  s = [ones(length(ii),1)*[upper(mfilename) ' requires that  '] s ...
       ones(length(ii),1)*['  equal the # of arrows (' num2str(narrows) ').' c]];
  s = s';
  s = s(:)';
  s = s(1:length(s)-1);
  error(setstr(s));
end;

% check element length in Start, Stop, and CrossDir
if ~isempty(start),
  [m,n] = size(start);
  if (n==2),
    start = [start NaN*ones(m,1)];
  elseif (n~=3),
    error([upper(mfilename) ' requires 2- or 3-element Start points.']);
```

```
  end;
end;
if ~isempty(stop),
  [m,n] = size(stop);
  if (n==2),
    stop = [stop NaN*ones(m,1)];
  elseif (n~=3),
    error([upper(mfilename) ' requires 2- or 3-element Stop points.']);
  end;
end;
if ~isempty(crossdir),
  [m,n] = size(crossdir);
  if (n<3),
    crossdir = [crossdir NaN*ones(m,3-n)];
  elseif (n~=3),
    if (all(imag(crossdir(:))==0)),
        error([upper(mfilename) ' requires 2- or 3-element CrossDir vectors.']);
    else,
        error([upper(mfilename) ' requires 2- or 3-element NormalDir vectors.']);
    end;
  end;
end;

% fill empty arguments
if isempty(start     ),   start      = [Inf Inf Inf];      end;
if isempty(stop      ),   stop       = [Inf Inf Inf];      end;
if isempty(len       ),   len        = Inf;                end;
if isempty(baseangle ),   baseangle  = Inf;                end;
if isempty(tipangle  ),   tipangle   = Inf;                end;
if isempty(wid       ),   wid        = Inf;                end;
if isempty(page      ),   page       = Inf;                end;
```

```
if isempty(crossdir  ),   crossdir   = [Inf Inf Inf];      end;
if isempty(ends      ),   ends       = Inf;                end;
if isempty(ispatch   ),   ispatch    = Inf;                end;

% expand single-column arguments
o = ones(narrows,1);
if (size(start     ,1)==1),   start      = o * start     ;   end;
if (size(stop      ,1)==1),   stop       = o * stop      ;   end;
if (length(len       )==1),   len        = o * len       ;   end;
if (length(baseangle )==1),   baseangle  = o * baseangle ;   end;
if (length(tipangle  )==1),   tipangle   = o * tipangle  ;   end;
if (length(wid       )==1),   wid        = o * wid       ;   end;
if (length(page      )==1),   page       = o * page      ;   end;
if (size(crossdir  ,1)==1),   crossdir   = o * crossdir  ;   end;
if (length(ends      )==1),   ends       = o * ends      ;   end;
if (length(ispatch   )==1),   ispatch    = o * ispatch   ;   end;
ax = o * gca;

% if we've got handles, get the defaults from the handles
if ~isempty(oldh),
  for k=1:narrows,
    oh = oldh(k);
    ud = get(oh,'UserData');
    ax(k) = get(oh,'Parent');
    ohtype = get(oh,'Type');
    if strcmp(get(oh,'Tag'),ArrowTag), % if it's an arrow already
        if isinf(ispatch(k)), ispatch(k)=strcmp(ohtype,'patch'); end;
        % arrow UserData format: [start' stop' len base tip wid page crossdir' ends]
        start0 = ud(1:3);
        stop0  = ud(4:6);
        if (isinf(len(k))),          len(k)       = ud( 7);   end;
```

```
        if (isinf(baseangle(k))),     baseangle(k)  = ud( 8);   end;
        if (isinf(tipangle(k))),      tipangle(k)   = ud( 9);   end;
        if (isinf(wid(k))),           wid(k)        = ud(10);   end;
        if (isinf(page(k))),          page(k)       = ud(11);   end;
        if (isinf(crossdir(k,1))),    crossdir(k,1) = ud(12);   end;
        if (isinf(crossdir(k,2))),    crossdir(k,2) = ud(13);   end;
        if (isinf(crossdir(k,3))),    crossdir(k,3) = ud(14);   end;
        if (isinf(ends(k))),          ends(k)       = ud(15);   end;
    elseif strcmp(ohtype,'line')|strcmp(ohtype,'patch'), % it's a non-arrow line or patch
        convLineToPatch = 1; %set to make arrow patches when converting from lines.
        if isinf(ispatch(k)), ispatch(k)=convLineToPatch|strcmp(ohtype,'patch'); end;
        x=get(oh,'XData');  x=x(~isnan(x(:)));  if isempty(x), x=NaN; end;
        y=get(oh,'YData');  y=y(~isnan(y(:)));  if isempty(y), y=NaN; end;
        z=get(oh,'ZData');  z=z(~isnan(z(:)));  if isempty(z), z=NaN; end;
        start0 = [x(1)   y(1)   z(1)  ];
        stop0  = [x(end) y(end) z(end)];
    else,
        error([upper(mfilename) ' cannot convert ' ohtype ' objects.']);
    end;
    ii=find(isinf(start(k,:)));  if ~isempty(ii),  start(k,ii)=start0(ii);  end;
    ii=find(isinf(stop( k,:)));  if ~isempty(ii),  stop( k,ii)=stop0( ii);  end;
  end;
end;

% convert Inf's to NaN's
start(     isinf(start    )) = NaN;
stop(      isinf(stop     )) = NaN;
len(       isinf(len      )) = NaN;
baseangle( isinf(baseangle)) = NaN;
```

```
tipangle(  isinf(tipangle )) = NaN;
wid(       isinf(wid      )) = NaN;
page(      isinf(page     )) = NaN;
crossdir(  isinf(crossdir )) = NaN;
ends(      isinf(ends     )) = NaN;
ispatch(   isinf(ispatch  )) = NaN;

% set up the UserData data (here so not corrupted by log10's and such)
ud = [start stop len baseangle tipangle wid page crossdir ends];

% Set Page defaults
page = ~isnan(page) & trueornan(page);

% Get axes limits, range, min; correct for aspect ratio and log scale
axm       = zeros(3,narrows);
axr       = zeros(3,narrows);
axrev     = zeros(3,narrows);
ap        = zeros(2,narrows);
xyzlog    = zeros(3,narrows);
limmin    = zeros(2,narrows);
limrange  = zeros(2,narrows);
oldaxlims = zeros(narrows,7);
oneax = all(ax==ax(1));
if (oneax),
  T    = zeros(4,4);
  invT = zeros(4,4);
else,
  T    = zeros(16,narrows);
  invT = zeros(16,narrows);
end;
axnotdone = logical(ones(size(ax)));
```

```
while (any(axnotdone)),
  ii = min(find(axnotdone));
  curax = ax(ii);
  curpage = page(ii);
  % get axes limits and aspect ratio
  axl = [get(curax,'XLim'); get(curax,'YLim'); get(curax,'ZLim')];
  oldaxlims(min(find(oldaxlims(:,1)==0)),:) = [curax reshape(axl',1,6)];
  % get axes size in pixels (points)
  u = get(curax,'Units');
  axposoldunits = get(curax,'Position');
  really_curpage = curpage & strcmp(u,'normalized');
  if (really_curpage),
    curfig = get(curax,'Parent');
    pu = get(curfig,'PaperUnits');
    set(curfig,'PaperUnits','points');
    pp = get(curfig,'PaperPosition');
    set(curfig,'PaperUnits',pu);
    set(curax,'Units','pixels');
    curapscreen = get(curax,'Position');
    set(curax,'Units','normalized');
    curap = pp.*get(curax,'Position');
  else,
    set(curax,'Units','pixels');
    curapscreen = get(curax,'Position');
    curap = curapscreen;
  end;
  set(curax,'Units',u);
  set(curax,'Position',axposoldunits);
  % handle non-stretched axes position
  str_stretch = { 'DataAspectRatioMode'    ; ...
                  'PlotBoxAspectRatioMode' ; ...
```

```
            'CameraViewAngleMode'      };
str_camera  = { 'CameraPositionMode'  ; ...
            'CameraTargetMode'    ; ...
            'CameraViewAngleMode' ; ...
            'CameraUpVectorMode'    };
notstretched = strcmp(get(curax,str_stretch),'manual');
manualcamera = strcmp(get(curax,str_camera),'manual');
if ~arrow_WarpToFill(notstretched,manualcamera,curax),
  % give a warning that this has not been thoroughly tested
  if 0 & ARROW_STRETCH_WARN,
      ARROW_STRETCH_WARN = 0;
      strs = {str_stretch{1:2},str_camera{:}};
      strs = [char(ones(length(strs),1)*sprintf('\n    ')) char(strs)]';
      warning([upper(mfilename) ' may not yet work quite right ' ...
            'if any of the following are ''manual'':' strs(:).']);
  end;
  % find the true pixel size of the actual axes
  texttmp = text(axl(1,[1 2 2 1 1 2 2 1]), ...
            axl(2,[1 1 2 2 1 1 2 2]), ...
            axl(3,[1 1 1 1 2 2 2 2]),'');
  set(texttmp,'Units','points');
  textpos = get(texttmp,'Position');
  delete(texttmp);
  textpos = cat(1,textpos{:});
  textpos = max(textpos(:,1:2)) - min(textpos(:,1:2));
  % adjust the axes position
  if (really_curpage),
      % adjust to printed size
      textpos = textpos * min(curap(3:4)./textpos);
      curap = [curap(1:2)+(curap(3:4)-textpos)/2 textpos];
  else,
```

```
        % adjust for pixel roundoff
        textpos = textpos * min(curapscreen(3:4)./textpos);
        curap = [curap(1:2)+(curap(3:4)-textpos)/2 textpos];
    end;
  end;
  if ARROW_PERSP_WARN & ~strcmp(get(curax,'Projection'),'orthographic'),
    ARROW_PERSP_WARN = 0;
    warning([upper(mfilename) ' does not yet work right for 3-D perspective projection.']);
  end;
  % adjust limits for log scale on axes
  curxyzlog = [strcmp(get(curax,'XScale'),'log'); ...
               strcmp(get(curax,'YScale'),'log'); ...
               strcmp(get(curax,'ZScale'),'log')];
  if (any(curxyzlog)),
    ii = find([curxyzlog;curxyzlog]);
    if (any(axl(ii)<=0)),
        error([upper(mfilename) ' does not support non-positive limits on log-scaled axes.']);
    else,
        axl(ii) = log10(axl(ii));
    end;
  end;
  % correct for 'reverse' direction on axes;
  curreverse = [strcmp(get(curax,'XDir'),'reverse'); ...
                strcmp(get(curax,'YDir'),'reverse'); ...
                strcmp(get(curax,'ZDir'),'reverse')];
  ii = find(curreverse);
  if ~isempty(ii),
    axl(ii,[1 2])=-axl(ii,[2 1]);
  end;
```

```
% compute the range of 2-D values
curT = get(curax,'Xform');
lim = curT*[0 1 0 1 0 1 0 1;0 0 1 1 0 0 1 1;0 0 0 0 1 1 1 1;1 1 1 1 1 1 1 1];
lim = lim(1:2,:)./([1;1]*lim(4,:));
curlimmin = min(lim')';
curlimrange = max(lim')' - curlimmin;
curinvT = inv(curT);
if (~oneax),
  curT = curT.';
  curinvT = curinvT.';
  curT = curT(:);
  curinvT = curinvT(:);
end;
% check which arrows to which cur corresponds
ii = find((ax==curax)&(page==curpage));
oo = ones(1,length(ii));
axr(:,ii)      = diff(axl')' * oo;
axm(:,ii)      = axl(:,1)    * oo;
axrev(:,ii)    = curreverse  * oo;
ap(:,ii)       = curap(3:4)' * oo;
xyzlog(:,ii)   = curxyzlog   * oo;
limmin(:,ii)   = curlimmin   * oo;
limrange(:,ii) = curlimrange * oo;
if (oneax),
  T    = curT;
  invT = curinvT;
else,
  T(:,ii)    = curT    * oo;
  invT(:,ii) = curinvT * oo;
end;
axnotdone(ii) = zeros(1,length(ii));
```

```
end;
oldaxlims(oldaxlims(:,1)==0,:)=[];

% correct for log scales
curxyzlog = xyzlog.';
ii = find(curxyzlog(:));
if ~isempty(ii),
  start(   ii) = real(log10(start(   ii)));
  stop(    ii) = real(log10(stop(    ii)));
  if (all(imag(crossdir)==0)), % pulled (ii) subscript on crossdir, 12/5/96 eaj
    crossdir(ii) = real(log10(crossdir(ii)));
  end;
end;

% correct for reverse directions
ii = find(axrev.');
if ~isempty(ii),
  start(   ii) = -start(   ii);
  stop(    ii) = -stop(    ii);
  crossdir(ii) = -crossdir(ii);
end;

% transpose start/stop values
start     = start.';
stop      = stop.';

% take care of defaults, page was done above
ii=find(isnan(start(:)          ));   if  ~isempty(ii),   start(ii)          =  axm(ii)+axr(ii)/2;
end;
ii=find(isnan(stop(:)           ));   if  ~isempty(ii),   stop(ii)           =  axm(ii)+axr(ii)/2;
end;
```

```
ii=find(isnan(crossdir(:)      ));  if ~isempty(ii),  crossdir(ii)     = zeros(length(ii),1);
end;
ii=find(isnan(len              ));  if ~isempty(ii),  len(ii)          = ones(length(ii),1)*deflen;
end;
ii=find(isnan(baseangle        ));  if ~isempty(ii),  baseangle(ii)    =
ones(length(ii),1)*defbaseangle;  end;
ii=find(isnan(tipangle         ));  if ~isempty(ii),  tipangle(ii)     =
ones(length(ii),1)*deftipangle;   end;
ii=find(isnan(wid              ));  if ~isempty(ii),  wid(ii)          = ones(length(ii),1)*defwid;
end;
ii=find(isnan(ends             ));  if ~isempty(ii),  ends(ii)         =
ones(length(ii),1)*defends;       end;

% transpose rest of values
len       = len.';
baseangle = baseangle.';
tipangle  = tipangle.';
wid       = wid.';
page      = page.';
crossdir  = crossdir.';
ends      = ends.';
ax        = ax.';

% given x, a 3xN matrix of points in 3-space;
% want to convert to X, the corresponding 4xN 2-space matrix
%
%   tmp1=[(x-axm)./axr; ones(1,size(x,1))];
%   if (oneax), X=T*tmp1;
%   else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=T.*tmp1;
%         tmp2=zeros(4,4*N); tmp2(:)=tmp1(:);
%         X=zeros(4,N); X(:)=sum(tmp2)'; end;
```

```
%   X = X ./ (ones(4,1)*X(4,:));

% for all points with start==stop, start=stop-(verysmallvalue)*(up-direction);
ii = find(all(start==stop));
if ~isempty(ii),
  % find an arrowdir vertical on screen and perpendicular to viewer
  %     transform to 2-D
    tmp1 = [(stop(:,ii)-axm(:,ii))./axr(:,ii);ones(1,length(ii))];
    if (oneax), twoD=T*tmp1;
    else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=T(:,ii).*tmp1;
         tmp2=zeros(4,4*length(ii)); tmp2(:)=tmp1(:);
         twoD=zeros(4,length(ii)); twoD(:)=sum(tmp2)'; end;
    twoD=twoD./(ones(4,1)*twoD(4,:));
  %     move the start point down just slightly
    tmp1 = twoD + [0;-1/1000;0;0]*(limrange(2,ii)./ap(2,ii));
  %     transform back to 3-D
    if (oneax), threeD=invT*tmp1;
    else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=invT(:,ii).*tmp1;
         tmp2=zeros(4,4*length(ii)); tmp2(:)=tmp1(:);
         threeD=zeros(4,length(ii)); threeD(:)=sum(tmp2)'; end;
    start(:,ii) = (threeD(1:3,:)./(ones(3,1)*threeD(4,:))).*axr(:,ii)+axm(:,ii);
end;

% compute along-arrow points
%  transform Start points
  tmp1=[(start-axm)./axr;ones(1,narrows)];
  if (oneax), X0=T*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=T.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       X0=zeros(4,narrows); X0(:)=sum(tmp2)'; end;
  X0=X0./(ones(4,1)*X0(4,:));
```

```
%  transform Stop points
  tmp1=[(stop-axm)./axr;ones(1,narrows)];
  if (oneax), Xf=T*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=T.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       Xf=zeros(4,narrows); Xf(:)=sum(tmp2)'; end;
  Xf=Xf./(ones(4,1)*Xf(4,:));
%  compute pixel distance between points
  D = sqrt(sum(((Xf(1:2,:)-X0(1:2,:)).*(ap./limrange)).^2));
  D = D + (D==0);  %eaj new 2/24/98
%  compute and modify along-arrow distances
  len1 = len;
  len2 = len - (len.*tan(tipangle/180*pi)-wid/2).*tan((90-baseangle)/180*pi);
  slen0 = zeros(1,narrows);
  slen1 = len1 .* ((ends==2)|(ends==3));
  slen2 = len2 .* ((ends==2)|(ends==3));
  len0 = zeros(1,narrows);
  len1  = len1 .* ((ends==1)|(ends==3));
  len2  = len2 .* ((ends==1)|(ends==3));
  %    for no start arrowhead
    ii=find((ends==1)&(D<len2));
    if ~isempty(ii),
       slen0(ii) = D(ii)-len2(ii);
    end;
  %    for no end arrowhead
    ii=find((ends==2)&(D<slen2));
    if ~isempty(ii),
       len0(ii) = D(ii)-slen2(ii);
    end;
  len1  = len1  + len0;
  len2  = len2  + len0;
```

```
  slen1 = slen1 + slen0;
  slen2 = slen2 + slen0;
  % note:  the division by D below will probably not be accurate if both
  %       of the following are true:
  %          1. the ratio of the line length to the arrowhead
  %             length is large
  %          2. the view is highly perspective.
%  compute stoppoints
  tmp1=X0.*(ones(4,1)*(len0./D))+Xf.*(ones(4,1)*(1-len0./D));
  if (oneax), tmp3=invT*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=invT.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       tmp3=zeros(4,narrows); tmp3(:)=sum(tmp2)'; end;
  stoppoint = tmp3(1:3,:)./(ones(3,1)*tmp3(4,:)).*axr+axm;
%  compute tippoints
  tmp1=X0.*(ones(4,1)*(len1./D))+Xf.*(ones(4,1)*(1-len1./D));
  if (oneax), tmp3=invT*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=invT.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       tmp3=zeros(4,narrows); tmp3(:)=sum(tmp2)'; end;
  tippoint = tmp3(1:3,:)./(ones(3,1)*tmp3(4,:)).*axr+axm;
%  compute basepoints
  tmp1=X0.*(ones(4,1)*(len2./D))+Xf.*(ones(4,1)*(1-len2./D));
  if (oneax), tmp3=invT*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=invT.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       tmp3=zeros(4,narrows); tmp3(:)=sum(tmp2)'; end;
  basepoint = tmp3(1:3,:)./(ones(3,1)*tmp3(4,:)).*axr+axm;
%  compute startpoints
  tmp1=X0.*(ones(4,1)*(1-slen0./D))+Xf.*(ones(4,1)*(slen0./D));
  if (oneax), tmp3=invT*tmp1;
```

```
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=invT.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       tmp3=zeros(4,narrows); tmp3(:)=sum(tmp2)'; end;
  startpoint = tmp3(1:3,:)./(ones(3,1)*tmp3(4,:)).*axr+axm;
%  compute stippoints
  tmp1=X0.*(ones(4,1)*(1-slen1./D))+Xf.*(ones(4,1)*(slen1./D));
  if (oneax), tmp3=invT*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=invT.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       tmp3=zeros(4,narrows); tmp3(:)=sum(tmp2)'; end;
  stippoint = tmp3(1:3,:)./(ones(3,1)*tmp3(4,:)).*axr+axm;
%  compute sbasepoints
  tmp1=X0.*(ones(4,1)*(1-slen2./D))+Xf.*(ones(4,1)*(slen2./D));
  if (oneax), tmp3=invT*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=invT.*tmp1;
       tmp2=zeros(4,4*narrows); tmp2(:)=tmp1(:);
       tmp3=zeros(4,narrows); tmp3(:)=sum(tmp2)'; end;
  sbasepoint = tmp3(1:3,:)./(ones(3,1)*tmp3(4,:)).*axr+axm;

% compute cross-arrow directions for arrows with NormalDir specified
if (any(imag(crossdir(:))~=0)),
  ii = find(any(imag(crossdir)~=0));
  crossdir(:,ii) = cross((stop(:,ii)-start(:,ii))./axr(:,ii), ...
                    imag(crossdir(:,ii))).*axr(:,ii);
end;

% compute cross-arrow directions
basecross  = crossdir + basepoint;
tipcross   = crossdir + tippoint;
sbasecross = crossdir + sbasepoint;
stipcross  = crossdir + stippoint;
```

```
ii = find(all(crossdir==0)|any(isnan(crossdir)));
if ~isempty(ii),
  numii = length(ii);
  %    transform start points
    tmp1 = [basepoint(:,ii) tippoint(:,ii) sbasepoint(:,ii) stippoint(:,ii)];
    tmp1 = (tmp1-axm(:,[ii ii ii ii])) ./ axr(:,[ii ii ii ii]);
    tmp1 = [tmp1; ones(1,4*numii)];
    if (oneax), X0=T*tmp1;
    else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=T(:,[ii ii ii ii]).*tmp1;
         tmp2=zeros(4,16*numii); tmp2(:)=tmp1(:);
         X0=zeros(4,4*numii); X0(:)=sum(tmp2)'; end;
    X0=X0./(ones(4,1)*X0(4,:));
  %    transform stop points
    tmp1 = [(2*stop(:,ii)-start(:,ii)-axm(:,ii))./axr(:,ii);ones(1,numii)];
    tmp1 = [tmp1 tmp1 tmp1 tmp1];
    if (oneax), Xf=T*tmp1;
    else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=T(:,[ii ii ii ii]).*tmp1;
         tmp2=zeros(4,16*numii); tmp2(:)=tmp1(:);
         Xf=zeros(4,4*numii); Xf(:)=sum(tmp2)'; end;
    Xf=Xf./(ones(4,1)*Xf(4,:));
  %    compute perpendicular directions
    pixfact = ((limrange(1,ii)./limrange(2,ii)).*(ap(2,ii)./ap(1,ii))).^2;
    pixfact = [pixfact pixfact pixfact pixfact];
    pixfact = [pixfact;1./pixfact];
    [dummyval,jj] = max(abs(Xf(1:2,:)-X0(1:2,:)));
    jj1 = ((1:4)'*ones(1,length(jj))==ones(4,1)*jj);
    jj2 = ((1:4)'*ones(1,length(jj))==ones(4,1)*(3-jj));
    jj3 = jj1(1:2,:);
    Xf(jj1)=Xf(jj1)+(Xf(jj1)-X0(jj1)==0); %eaj new 2/24/98
    Xp = X0;
    Xp(jj2) = X0(jj2) + ones(sum(jj2(:)),1);
```

```
    Xp(jj1) = X0(jj1) - (Xf(jj2)-X0(jj2))./(Xf(jj1)-X0(jj1)) .* pixfact(jj3);
  %    inverse transform the cross points
    if (oneax), Xp=invT*Xp;
    else, tmp1=[Xp;Xp;Xp;Xp]; tmp1=invT(:,[ii ii ii ii]).*tmp1;
         tmp2=zeros(4,16*numii); tmp2(:)=tmp1(:);
         Xp=zeros(4,4*numii); Xp(:)=sum(tmp2)'; end;
    Xp=(Xp(1:3,:)./(ones(3,1)*Xp(4,:))).*axr(:,[ii ii ii ii])+axm(:,[ii ii ii ii]);
    basecross(:,ii)  = Xp(:,0*numii+(1:numii));
    tipcross(:,ii)   = Xp(:,1*numii+(1:numii));
    sbasecross(:,ii) = Xp(:,2*numii+(1:numii));
    stipcross(:,ii)  = Xp(:,3*numii+(1:numii));
end;

% compute all points
%  compute start points
  axm11 = [axm axm axm axm axm axm axm axm axm axm axm];
  axr11 = [axr axr axr axr axr axr axr axr axr axr axr];
  st = [stoppoint tippoint basepoint sbasepoint stippoint startpoint stippoint
sbasepoint basepoint tippoint stoppoint];
  tmp1 = (st - axm11) ./ axr11;
  tmp1 = [tmp1; ones(1,size(tmp1,2))];
  if (oneax), X0=T*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=[T T T T T T T T T T T].*tmp1;
       tmp2=zeros(4,44*narrows); tmp2(:)=tmp1(:);
       X0=zeros(4,11*narrows); X0(:)=sum(tmp2)'; end;
  X0=X0./(ones(4,1)*X0(4,:));
%  compute stop points
  tmp1 = ([start tipcross basecross sbasecross stipcross stop stipcross sbasecross
basecross tipcross start] ...
      - axm11) ./ axr11;
  tmp1 = [tmp1; ones(1,size(tmp1,2))];
```

```
  if (oneax), Xf=T*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=[T T T T T T T T T T].*tmp1;
       tmp2=zeros(4,44*narrows); tmp2(:)=tmp1(:);
       Xf=zeros(4,11*narrows); Xf(:)=sum(tmp2)'; end;
  Xf=Xf./(ones(4,1)*Xf(4,:));
%  compute lengths
  len0  = len.*((ends==1)|(ends==3)).*tan(tipangle/180*pi);
  slen0 = len.*((ends==2)|(ends==3)).*tan(tipangle/180*pi);
  le = [zeros(1,narrows) len0 wid/2 wid/2 slen0 zeros(1,narrows) -slen0 -wid/2 -
wid/2 -len0 zeros(1,narrows)];
  aprange = ap./limrange;
  aprange = [aprange aprange aprange aprange aprange aprange aprange
aprange aprange aprange aprange];
  D = sqrt(sum(((Xf(1:2,:)-X0(1:2,:)).*aprange).^2));
  Dii=find(D==0); if ~isempty(Dii), D=D+(D==0); le(Dii)=zeros(1,length(Dii)); end;
%should fix DivideByZero warnings
  tmp1 = X0.*(ones(4,1)*(1-le./D)) + Xf.*(ones(4,1)*(le./D));
%  inverse transform
  if (oneax), tmp3=invT*tmp1;
  else, tmp1=[tmp1;tmp1;tmp1;tmp1]; tmp1=[invT invT invT invT invT invT invT
invT invT invT invT].*tmp1;
       tmp2=zeros(4,44*narrows); tmp2(:)=tmp1(:);
       tmp3=zeros(4,11*narrows); tmp3(:)=sum(tmp2)'; end;
  pts = tmp3(1:3,:)./(ones(3,1)*tmp3(4,:)) .* axr11 + axm11;

% correct for ones where the crossdir was specified
ii = find(~(all(crossdir==0)|any(isnan(crossdir))));
if ~isempty(ii),
  D1 = [pts(:,1*narrows+ii)-pts(:,9*narrows+ii) ...
       pts(:,2*narrows+ii)-pts(:,8*narrows+ii) ...
       pts(:,3*narrows+ii)-pts(:,7*narrows+ii) ...
```

```
        pts(:,4*narrows+ii)-pts(:,6*narrows+ii) ...
        pts(:,6*narrows+ii)-pts(:,4*narrows+ii) ...
        pts(:,7*narrows+ii)-pts(:,3*narrows+ii) ...
        pts(:,8*narrows+ii)-pts(:,2*narrows+ii) ...
        pts(:,9*narrows+ii)-pts(:,1*narrows+ii)]/2;
  ii = ii'*ones(1,8) + ones(length(ii),1)*[1:4 6:9]*narrows;
  ii = ii(:)';
  pts(:,ii) = st(:,ii) + D1;
end;


% readjust for reverse directions
iicols=(1:narrows)'; iicols=iicols(:,ones(1,11)); iicols=iicols(:).';
tmp1=axrev(:,iicols);
ii = find(tmp1(:)); if ~isempty(ii), pts(ii)=-pts(ii); end;

% readjust for log scale on axes
tmp1=xyzlog(:,iicols);
ii = find(tmp1(:)); if ~isempty(ii), pts(ii)=10.^pts(ii); end;

% compute the x,y,z coordinates of the patches;
ii = narrows*(0:10)'*ones(1,narrows) + ones(11,1)*(1:narrows);
ii = ii(:)';
x = zeros(11,narrows);
y = zeros(11,narrows);
z = zeros(11,narrows);
x(:) = pts(1,ii)';
y(:) = pts(2,ii)';
z(:) = pts(3,ii)';

% do the output
```

```
if (nargout<=1),
%  % create or modify the patches
  newpatch                =                trueornan(ispatch)                &
(isempty(oldh)|~strcmp(get(oldh,'Type'),'patch'));
  newline = ~trueornan(ispatch) & (isempty(oldh)|~strcmp(get(oldh,'Type'),'line'));
  if isempty(oldh), H=zeros(narrows,1); else, H=oldh; end;
%  % make or modify the arrows
  for k=1:narrows,
    if all(isnan(ud(k,[3 6])))&arrow_is2DXY(ax(k)), zz=[]; else, zz=z(:,k); end;
    % work around a MATLAB 6.x OpenGL bug -- 7/28/02
      xx=x(:,k); yy=y(:,k);
      mask=any([ones(1,2+size(zz,2));diff([xx yy zz],[],1)],2);
      xx=xx(mask); yy=yy(mask); if ~isempty(zz), zz=zz(mask); end;
    % plot the patch or line
    xyz = {'XData',xx,'YData',yy,'ZData',zz,'Tag',ArrowTag};
    if newpatch(k)|newline(k),
        if newpatch(k),
                H(k) = patch(xyz{:});
        else,
                H(k) = line(xyz{:});
        end;
        if ~isempty(oldh), arrow_copyprops(oldh(k),H(k)); end;
    else,
        if ispatch(k), xyz={xyz{:},'CData',[]}; end;
        set(H(k),xyz{:});
    end;
  end;
  if ~isempty(oldh), delete(oldh(oldh~=H)); end;
%  % additional properties
  set(H,'Clipping','off');
  set(H,{'UserData'},num2cell(ud,2));
```

```
  if (length(extraprops)>0), set(H,extraprops{:}); end;
  % handle choosing arrow Start and/or Stop locations if unspecified
  [H,oldaxlims,errstr] = arrow_clicks(H,ud,x,y,z,ax,oldaxlims);
  if ~isempty(errstr), error([upper(mfilename) ' got ' errstr]); end;
  % set the output
  if (nargout>0), h=H; end;
  % make sure the axis limits did not change
  if isempty(oldaxlims),
    ARROW_AXLIMITS = [];
  else,
    lims = get(oldaxlims(:,1),{'XLim','YLim','ZLim'})';
    lims = reshape(cat(2,lims{:}),6,size(lims,2));
    mask = arrow_is2DXY(oldaxlims(:,1));
    oldaxlims(mask,6:7) = lims(5:6,mask)';
    ARROW_AXLIMITS = oldaxlims(find(any(oldaxlims(:,2:7)'~=lims)),:);
    if ~isempty(ARROW_AXLIMITS),
        warning(arrow_warnlimits(ARROW_AXLIMITS,narrows));
    end;
  end;
else,
  % don't create the patch, just return the data
  h=x;
  yy=y;
  zz=z;
end;



function out = arrow_defcheck(in,def,prop)
% check if we got 'default' values
  out = in;
```

```
  if ~isstr(in), return; end;
  if size(in,1)==1 & strncmp(lower(in),'def',3),
    out = def;
  elseif ~isempty(prop),
    error([upper(mfilename) ' does not recognize ''' in(:)' ''' as a valid ''' prop '''
string.']);
  end;



function [H,oldaxlims,errstr] = arrow_clicks(H,ud,x,y,z,ax,oldaxlims)
% handle choosing arrow Start and/or Stop locations if necessary
  errstr = '';
  if isempty(H)|isempty(ud)|isempty(x), return; end;
  % determine which (if any) need Start and/or Stop
  needStart = all(isnan(ud(:,1:3))')';
  needStop  = all(isnan(ud(:,4:6))')';
  mask = any(needStart|needStop);
  if ~any(mask), return; end;
  ud(~mask,:)=[]; ax(:,~mask)=[];
  x(:,~mask)=[]; y(:,~mask)=[]; z(:,~mask)=[];
  % make them invisible for the time being
  set(H,'Visible','off');
  % save the current axes and limits modes; set to manual for the time being
  oldAx  = gca;
  limModes=get(ax(:),{'XLimMode','YLimMode','ZLimMode'});
  set(ax(:),{'XLimMode','YLimMode','ZLimMode'},{'manual','manual','manual'});
  % loop over each arrow that requires attention
  jj = find(mask);
  for ii=1:length(jj),
    h = H(jj(ii));
```

```
    axes(ax(ii));
    % figure out correct call
    if needStart(ii), prop='Start'; else, prop='Stop'; end;
    [wasInterrupted,errstr] = arrow_click(needStart(ii)&needStop(ii),h,prop,ax(ii));
    % handle errors and control-C
    if wasInterrupted,
        delete(H(jj(ii:end)));
        H(jj(ii:end))=[];
        oldaxlims(jj(ii:end),:)=[];
        break;
    end;
  end;
  % restore the axes and limit modes
  axes(oldAx);
  set(ax(:),{'XLimMode','YLimMode','ZLimMode'},limModes);

function [wasInterrupted,errstr] = arrow_click(lockStart,H,prop,ax)
% handle the clicks for one arrow
  fig = get(ax,'Parent');
  % save some things
  oldFigProps = {'Pointer','WindowButtonMotionFcn','WindowButtonUpFcn'};
  oldFigValue = get(fig,oldFigProps);
  oldArrowProps = {'EraseMode'};
  oldArrowValue = get(H,oldArrowProps);
  set(H,'EraseMode','background'); %because 'xor' makes shaft invisible unless
Width>1
  global ARROW_CLICK_H ARROW_CLICK_PROP ARROW_CLICK_AX
ARROW_CLICK_USE_Z
  ARROW_CLICK_H=H; ARROW_CLICK_PROP=prop; ARROW_CLICK_AX=ax;
  ARROW_CLICK_USE_Z=~arrow_is2DXY(ax)|~arrow_planarkids(ax);
  set(fig,'Pointer','crosshair');
```

```
  % set up the WindowButtonMotion so we can see the arrow while moving around
  set(fig,'WindowButtonUpFcn','set(gcf,''WindowButtonUpFcn'','''')', ...
        'WindowButtonMotionFcn','');
  if ~lockStart,
    set(H,'Visible','on');
    set(fig,'WindowButtonMotionFcn',[mfilename '(''callback'',''motion'');']);
  end;
  % wait for the button to be pressed
  [wasKeyPress,wasInterrupted,errstr] = arrow_wfbdown(fig);
  % if we wanted to click-drag, set the Start point
  if lockStart & ~wasInterrupted,
    pt = arrow_point(ARROW_CLICK_AX,ARROW_CLICK_USE_Z);
    feval(mfilename,H,'Start',pt,'Stop',pt);
    set(H,'Visible','on');
    ARROW_CLICK_PROP='Stop';
    set(fig,'WindowButtonMotionFcn',[mfilename '(''callback'',''motion'');']);
    % wait for the mouse button to be released
    eval('waitfor(fig,''WindowButtonUpFcn'','''');','wasInterrupted=1;');
    if wasInterrupted, errstr=lasterr; end;
  end;
  if ~wasInterrupted, feval(mfilename,'callback','motion'); end;
  % restore some things
  set(gcf,oldFigProps,oldFigValue);
  set(H,oldArrowProps,oldArrowValue);

function arrow_callback(varargin)
% handle redrawing callbacks
  if nargin==0, return; end;
  str = varargin{1};
  if ~isstr(str), error([upper(mfilename) ' got an invalid Callback command.']); end;
  s = lower(str);
```

```
  if strcmp(s,'motion'),
    % motion callback
    global ARROW_CLICK_H ARROW_CLICK_PROP ARROW_CLICK_AX
ARROW_CLICK_USE_Z

  feval(mfilename,ARROW_CLICK_H,ARROW_CLICK_PROP,arrow_point(ARRO
W_CLICK_AX,ARROW_CLICK_USE_Z));
    drawnow;
  else,
    error([upper(mfilename) ' does not recognize '' str(:).' '' as a valid Callback
option.']);
  end;

function out = arrow_point(ax,use_z)
% return the point on the given axes
  if nargin==0, ax=gca; end;
  if nargin<2, use_z=~arrow_is2DXY(ax)|~arrow_planarkids(ax); end;
  out = get(ax,'CurrentPoint');
  out = out(1,:);
  if ~use_z, out=out(1:2); end;

function [wasKeyPress,wasInterrupted,errstr] = arrow_wfbdown(fig)
% wait for button down ignoring object ButtonDownFcn's
  if nargin==0, fig=gcf; end;
  errstr = '';
  % save ButtonDownFcn values
  objs = findobj(fig);
  buttonDownFcns = get(objs,'ButtonDownFcn');
  mask=~strcmp(buttonDownFcns,'');                    objs=objs(mask);
buttonDownFcns=buttonDownFcns(mask);
  set(objs,'ButtonDownFcn','');
```

```
  % save other figure values
  figProps = {'KeyPressFcn','WindowButtonDownFcn'};
  figValue = get(fig,figProps);
  % do the real work
  set(fig,'KeyPressFcn','set(gcf,''KeyPressFcn'',"","WindowButtonDownFcn'',"");', ...
        'WindowButtonDownFcn','set(gcf,"WindowButtonDownFcn'',"")');
  lasterr('');
  wasInterrupted=0;
eval('waitfor(fig,"WindowButtonDownFcn'',"");','wasInterrupted=1;');
  wasKeyPress = ~wasInterrupted & strcmp(get(fig,'KeyPressFcn'),'');
  if wasInterrupted, errstr=lasterr; end;
  % restore ButtonDownFcn and other figure values
  set(objs,'ButtonDownFcn',buttonDownFcns);
  set(fig,figProps,figValue);



function [out,is2D] = arrow_is2DXY(ax)
% check if axes are 2-D X-Y plots
  % may not work for modified camera angles, etc.
  out = logical(zeros(size(ax))); % 2-D X-Y plots
  is2D = out;                   % any 2-D plots
  views = get(ax(:),{'View'});
  views = cat(1,views{:});
  out(:) = abs(views(:,2))==90;
  is2D(:) = out(:) | all(rem(views',90)==0)';

function out = arrow_planarkids(ax)
% check if axes descendents all have empty ZData (lines,patches,surfaces)
  out = logical(ones(size(ax)));
  allkids = get(ax(:),{'Children'});
```

```
  for k=1:length(allkids),
    kids = get([findobj(allkids{k},'flat','Type','line')
                findobj(allkids{k},'flat','Type','patch')
                findobj(allkids{k},'flat','Type','surface')],{'ZData'});
    for j=1:length(kids),
        if ~isempty(kids{j}), out(k)=logical(0); break; end;
    end;
  end;



function arrow_fixlimits(axlimits)
% reset the axis limits as necessary
  if isempty(axlimits), disp([upper(mfilename) ' does not remember any axis limits to reset.']); end;
  for k=1:size(axlimits,1),
    if                                      any(get(axlimits(k,1),'XLim')~=axlimits(k,2:3)), set(axlimits(k,1),'XLim',axlimits(k,2:3)); end;
    if                                      any(get(axlimits(k,1),'YLim')~=axlimits(k,4:5)), set(axlimits(k,1),'YLim',axlimits(k,4:5)); end;
    if                                      any(get(axlimits(k,1),'ZLim')~=axlimits(k,6:7)), set(axlimits(k,1),'ZLim',axlimits(k,6:7)); end;
  end;



function out = arrow_WarpToFill(notstretched,manualcamera,curax)
% check if we are in "WarpToFill" mode.
  out = strcmp(get(curax,'WarpToFill'),'on');
  % 'WarpToFill' is undocumented, so may need to replace this by
  % out = ~( any(notstretched) & any(manualcamera) );
```

```
function out = arrow_warnlimits(axlimits,narrows)
% create a warning message if we've changed the axis limits
  msg = '';
  switch (size(axlimits,1))
    case 1, msg='';
    case 2, msg='on two axes ';
    otherwise, msg='on several axes ';
  end;
  msg = [upper(mfilename) ' changed the axis limits ' msg ...
        'when adding the arrow'];
  if (narrows>1), msg=[msg 's']; end;
  out = [msg '.' sprintf('\n') '         Call ' upper(mfilename) ...
        ' FIXLIMITS to reset them now.'];



function arrow_copyprops(fm,to)
% copy line properties to patches
  props  = {'EraseMode','LineStyle','LineWidth','Marker','MarkerSize',...
            'MarkerEdgeColor','MarkerFaceColor','ButtonDownFcn',      ...
            'Clipping','DeleteFcn','BusyAction','HandleVisibility',   ...
            'Selected','SelectionHighlight','Visible'};
  lineprops  = {'Color',    props{:}};
  patchprops = {'EdgeColor',props{:}};
  patch2props = {'FaceColor',patchprops{:}};
  fmpatch = strcmp(get(fm,'Type'),'patch');
  topatch = strcmp(get(to,'Type'),'patch');
```

```
  set(to( fmpatch& topatch),patch2props,get(fm( fmpatch& topatch),patch2props));
%p->p
  set(to(~fmpatch&~topatch),lineprops,  get(fm(~fmpatch&~topatch),lineprops  ));
%l->l
  set(to( fmpatch&~topatch),lineprops,  get(fm( fmpatch&~topatch),patchprops ));
%p->l
  set(to(~fmpatch&  topatch),patchprops,  get(fm(~fmpatch&  topatch),lineprops)
,'FaceColor','none'); %l->p



function arrow_props
% display further help info about ARROW properties
  c = sprintf('\n');
  disp([c ...
  'ARROW Properties:  Default values are given in [square brackets], and other' c
...
  '                 acceptable equivalent property names are in (parenthesis).' c c ...
  '  Start          The starting points. For N arrows,           B' c ...
  '                 this should be a Nx2 or Nx3 matrix.         /|\          ^' c ...
  '  Stop           The end points. For N arrows, this          /|||\         |' c ...
  '                 should be a Nx2 or Nx3 matrix.             //|||\\       L|' c ...
  '  Length         Length of the arrowhead (in pixels on     ///|||\\\      e|' c ...
  '                 screen, points on a page). [16] (Len)    ////|||\\\\     n|' c ...
  '  BaseAngle      Angle (degrees) of the base angle       /////|D|\\\\\    g|' c ...
  '                 ADE.  For a simple stick arrow, use     ////  |||  \\\\   t|' c ...
  '                 BaseAngle=TipAngle. [90] (Base)        ///    |||    \\\  h|' c ...
  '  TipAngle       Angle (degrees) of tip angle ABC.     //<----->||      \\  |' c ...
  '                 [16] (Tip)                          /   base |||       \  V' c ...
  '  Width          Width of the base in pixels.  Not  E   angle ||<-------->C' c ...
  '                 the "LineWidth" prop. [0] (Wid)            |||tipangle' c ...
```

```
  ' Page          If provided, non-empty, and not NaN,         |||' c ...
  '               this causes ARROW to use hardcopy            |||' c ...
  '               rather than onscreen proportions.            A' c ...
  '               This is important if screen aspect        -->   <-- width' c ...
  '               ratio and hardcopy aspect ratio are    ----CrossDir---->' c ...
  '               vastly different. []' c...
  ' CrossDir      A vector giving the direction towards which the fletches' c ...
  '               on the arrow should go.  [computed such that it is perpen-' c ...
  '               dicular to both the arrow direction and the view direction' c ...
  '               (i.e., as if it was pasted on a normal 2-D graph)]  (Note' c ...
  '               that CrossDir is a vector.  Also note that if an axis is' c ...
  '               plotted on a log scale, then the corresponding component' c ...
  '               of CrossDir must also be set appropriately, i.e., to 1 for' c ...
  '               no change in that direction, >1 for a positive change, >0' c ...
  '               and <1 for negative change.)' c ...
  ' NormalDir     A vector normal to the fletch direction (CrossDir is then' c ...
  '               computed by the vector cross product [Line]x[NormalDir]). []' c ...
  '               (Note that NormalDir is a vector.  Unlike CrossDir,' c ...
  '               NormalDir is used as is regardless of log-scaled axes.)' c ...
  ' Ends          Set which end has an arrowhead.  Valid values are ''none'',' c ...
  '               ''stop'', ''start'', and ''both''. [''stop''] (End)' c...
  ' ObjectHandles   Vector of handles to previously-created arrows to be' c ...
  '               updated or line objects to be converted to arrows.' c ...
  '               [] (Object,Handle)' c ]);



function out = arrow_demo
  % demo
  % create the data
  [x,y,z] = peaks;
```

```
  [ddd,out.iii]=max(z(:));
  out.axlim = [min(x(:)) max(x(:)) min(y(:)) max(y(:)) min(z(:)) max(z(:))];

  % modify it by inserting some NaN's
  [m,n] = size(z);
  m = floor(m/2);
  n = floor(n/2);
  z(1:m,1:n) = NaN*ones(m,n);

  % graph it
  clf('reset');
  out.hs=surf(x,y,z);
  out.x=x; out.y=y; out.z=z;
  xlabel('x'); ylabel('y');

function h = arrow_demo3(in)
  % set the view
  axlim = in.axlim;
  axis(axlim);
  zlabel('z');
  %set(in.hs,'FaceColor','interp');
  view(viewmtx(-37.5,30,20));
  title(['Demo of the capabilities of the ARROW function in 3-D']);

  % Normal blue arrow
  h1 = feval(mfilename,[axlim(1) axlim(4) 4],[-.8 1.2 4], ...
          'EdgeColor','b','FaceColor','b');

  % Normal white arrow, clipped by the surface
  h2 = feval(mfilename,axlim([1 4 6]),[0 2 4]);
  t=text(-2.4,2.7,7.7,'arrow clipped by surf');
```

```
% Baseangle<90
h3 = feval(mfilename,[3 .125 3.5],[1.375 0.125 3.5],30,50);
t2=text(3.1,.125,3.5,'local maximum');

% Baseangle<90, fill and edge colors different
h4 = feval(mfilename,axlim(1:2:5)*.5,[0 0 0],36,60,25, ...
        'EdgeColor','b','FaceColor','c');
t3=text(axlim(1)*.5,axlim(3)*.5,axlim(5)*.5-.75,'origin');
set(t3,'HorizontalAlignment','center');

% Baseangle>90, black fill
h5 = feval(mfilename,[-2.9 2.9 3],[-1.3 .4 3.2],30,120,[],6, ...
        'EdgeColor','r','FaceColor','k','LineWidth',2);

% Baseangle>90, no fill
h6 = feval(mfilename,[-2.9 2.9 1.3],[-1.3 .4 1.5],30,120,[],6, ...
        'EdgeColor','r','FaceColor','none','LineWidth',2);

% Stick arrow
h7 = feval(mfilename,[-1.6 -1.65 -6.5],[0 -1.65 -6.5],[],16,16);
t4=text(-1.5,-1.65,-7.25,'global mininum');
set(t4,'HorizontalAlignment','center');

% Normal, black fill
h8 = feval(mfilename,[-1.4 0 -7.2],[-1.4 0 -3],'FaceColor','k');
t5=text(-1.5,0,-7.75,'local minimum');
set(t5,'HorizontalAlignment','center');

% Gray fill, crossdir specified, 'LineStyle' --
h9 = feval(mfilename,[-3 2.2 -6],[-3 2.2 -.05],36,[],27,6,[],[0 -1 0], ...
```

```
        'EdgeColor','k','FaceColor',.75*[1 1 1],'LineStyle','--');

  % a series of normal arrows, linearly spaced, crossdir specified
  h10y=(0:4)'/3;
  h10 = feval(mfilename,[-3*ones(size(h10y)) h10y -6.5*ones(size(h10y))], ...
        [-3*ones(size(h10y)) h10y -.05*ones(size(h10y))], ...
        12,[],[],[],[],[0 -1 0]);

  % a series of normal arrows, linearly spaced
  h11x=(1:.33:2.8)';
  h11 = feval(mfilename,[h11x -3*ones(size(h11x)) 6.5*ones(size(h11x))], ...
        [h11x -3*ones(size(h11x)) -.05*ones(size(h11x))]);

  % series of magenta arrows, radially oriented, crossdir specified
  h12x=2; h12y=-3; h12z=axlim(5)/2; h12xr=1; h12zr=h12z; ir=.15;or=.81;
  h12t=(0:11)'/6*pi;
  h12 = feval(mfilename,                                    ...
        [h12x+h12xr*cos(h12t)*ir h12y*ones(size(h12t))      ...
         h12z+h12zr*sin(h12t)*ir],[h12x+h12xr*cos(h12t)*or  ...
         h12y*ones(size(h12t)) h12z+h12zr*sin(h12t)*or],    ...
        10,[],[],[],[],                                     ...
        [-h12xr*sin(h12t) zeros(size(h12t)) h12zr*cos(h12t)],...
        'FaceColor','none','EdgeColor','m');

  % series of normal arrows, tangentially oriented, crossdir specified
  or13=.91; h13t=(0:.5:12)'/6*pi;
  locs = [h12x+h12xr*cos(h13t)*or13 h12y*ones(size(h13t))
h12z+h12zr*sin(h13t)*or13];
  h13 = feval(mfilename,locs(1:end-1,:),locs(2:end,:),6);

  % arrow with no line ==> oriented downwards
```

```
  h14 = feval(mfilename,[3 3 .100001],[3 3 .1],30);
  t6=text(3,3,3.6,'no line'); set(t6,'HorizontalAlignment','center');

  % arrow with arrowheads at both ends
  h15 = feval(mfilename,[-.5 -3 -3],[1 -3 -3],'Ends','both','FaceColor','g', ...
           'Length',20,'Width',3,'CrossDir',[0 0 1],'TipAngle',25);

  h=[h1;h2;h3;h4;h5;h6;h7;h8;h9;h10;h11;h12;h13;h14;h15];

function h = arrow_demo2(in)
  axlim = in.axlim;
  dolog = 1;
  if (dolog), set(in.hs,'YData',10.^get(in.hs,'YData')); end;
  shading('interp');
  view(2);
  title(['Demo of the capabilities of the ARROW function in 2-D']);
  hold on; [C,H]=contour(in.x,in.y,in.z,20,'-'); hold off;
  for k=H',
    set(k,'ZData',(axlim(6)+1)*ones(size(get(k,'XData'))),'Color','k');
    if (dolog), set(k,'YData',10.^get(k,'YData')); end;
  end;
  if (dolog), axis([axlim(1:2) 10.^axlim(3:4)]); set(gca,'YScale','log');
  else,       axis(axlim(1:4)); end;

  % Normal blue arrow
  start = [axlim(1) axlim(4) axlim(6)+2];
  stop  = [in.x(in.iii) in.y(in.iii) axlim(6)+2];
  if (dolog), start(:,2)=10.^start(:,2); stop(:,2)=10.^stop(:,2); end;
  h1 = feval(mfilename,start,stop,'EdgeColor','b','FaceColor','b');

  % three arrows with varying fill, width, and baseangle
```

```
  start = [-3   -3   10; -3   -1.5 10; -1.5 -3   10];
  stop  = [-.03 -.03 10; -.03 -1.5 10; -1.5 -.03 10];
  if (dolog), start(:,2)=10.^start(:,2); stop(:,2)=10.^stop(:,2); end;
  h2                                                                                    =
feval(mfilename,start,stop,24,[90;60;120],[],[0;0;4],'Ends',str2mat('both','stop','stop')
);
  set(h2(2),'EdgeColor',[0 .35 0],'FaceColor',[0 .85 .85]);
  set(h2(3),'EdgeColor','r','FaceColor',[1 .5 1]);
  h=[h1;h2];

function out = trueornan(x)
if isempty(x),
  out=x;
else,
  out = isnan(x);
  out(~out) = x(~out);
end;
```

2.Script “circle.m”:

```
function H=circle(center,radius,NOP,style)
%-------------------------------------------------------------------------------------------
% H=CIRCLE(CENTER,RADIUS,NOP,STYLE)
% This routine draws a circle with center defined as
% a vector CENTER, radius as a scaler RADIS. NOP is
% the number of points on the circle. As to STYLE,
% use it the same way as you use the rountine PLOT.
% Since the handle of the object is returned, you
% use routine SET to get the best result.
%
```

```
%   Usage Examples,
%
%   circle([1,3],3,1000,':');
%   circle([2,4],2,1000,'--');
%
%   Zhenhai Wang <zhenhai@ieee.org>
%   Version 1.00
%   December, 2002
%-------------------------------------------------------------------------------------------

if (nargin <3),
 error('Please see help for INPUT DATA.');
elseif (nargin==3)
    style='b-';
end;
THETA=linspace(0,2*pi,NOP);
RHO=ones(1,NOP)*radius;
[X,Y] = pol2cart(THETA,RHO);
X=X+center(1);
Y=Y+center(2);
H=plot(X,Y,style);
axis square;
```