

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

MURILO EDUARDO CASTEROBA BENTO

Um Estudo Comparativo entre Dois Algoritmos de
Resolução de Desigualdades Matriciais Lineares
Aplicados ao Projeto de Controladores por
Realimentação de Saída

SÃO CARLOS
NOVEMBRO/2013

MURILO EDUARDO CASTEROBA BENTO

**UM ESTUDO COMPARATIVO ENTRE DOIS
ALGORITMOS DE RESOLUÇÃO DE
DESIGUALDADES MATRICIAIS LINEARES
APLICADOS AO PROJETO DE
CONTROLADORES POR REALIMENTAÇÃO DE
SAÍDA**

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos,
da Universidade de São Paulo

Curso de Engenharia Elétrica com Ênfase em
Sistemas de Energia e Automação

ORIENTADOR: Dr. Rodrigo Andrade Ramos

SÃO CARLOS

2013

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

B478u Bento, Murilo Eduardo Casteroba
Um estudo comparativo entre dois algoritmos de
resolução de desigualdades matriciais lineares
aplicados ao projeto de controladores por realimentação
de saída / Murilo Eduardo Casteroba Bento; orientador
Rodrigo Andrade Ramos. São Carlos, 2013.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Sistemas de Energia e Automação) -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2013.

1. Sistemas Elétricos de Potência. 2. Controle
Robusto. 3. Estabilidade Angular . 4. Desigualdades
Matriciais Lineares. 5. Compensador de Fase. 6. Sistema
com Realimentação de Saída. I. Título.

FOLHA DE APROVAÇÃO

Nome: Murilo Eduardo Casteroba Bento

Título: "Um estudo comparativo entre dois algoritmos de resolução de desigualdades matriciais lineares aplicados ao projeto de controladores de realimentação de saída"

Trabalho de Conclusão de Curso defendido e aprovado
em 25/11/2013,

com NOTA 9,0 (NOVE, ZERO), pela Comissão Julgadora:

Prof. Associado Rodrigo Andrade Ramos - (Orientador - SEL/EESC/USP)

Mestre Daniel Souto Siqueira - (Doutorando - SEL/EESC/USP)

Mestre Daiane Cristina Bortolin - (Doutoranda - SEL/EESC/USP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Homero Schiabel

*Dedico este trabalho de conclusão de curso
à minha mãe Carmen, à minha avó Maria
e ao meu irmão Matheus.*

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter sempre me acompanhado a vencer meus desafios e realizar meus sonhos.

À minha mãe **Carmen**, à minha avó **Maria Carlota** e ao meu irmão **Matheus** pelo amor, carinho e incentivo nestes cinco anos de graduação.

Aos meus amigos **Eleandro**, **Gabriel Rogatto**, **Israel**, **Lara**, **Lucas Tibiano**, **Luis Fernando** e **Osmar** que, sem eles, com certeza minha graduação não teria sido tão especial.

Ao professor **Rodrigo Andrade Ramos** pela orientação neste trabalho e nos projetos de Iniciação Científica.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro concedido no período de Iniciação Científica.

À Universidade de São Paulo pelo ensino gratuito e de qualidade que me proporcionou crescimento pessoal e profissional.

RESUMO

BENTO, M. E. C. (2013). **Um Estudo Comparativo entre Dois Algoritmos de Resolução de Desigualdades Matriciais Lineares Aplicados ao Projeto de Controladores por Realimentação de Saída**. Trabalho de Conclusão de Curso - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2013.

O presente trabalho de conclusão de curso teve como objetivo a avaliação comparativa de desempenho de diferentes *solvers* de Desigualdades Matriciais Lineares (LMI's do inglês *Linear Matrix Inequalities*) aplicados ao projeto de controladores por realimentação de saída utilizando uma técnica em particular denominada iteração V-K. Bons resultados da literatura já foram obtidos utilizando-se o *solver* mincx, o qual requer licença de uso. Esta pesquisa pretendeu avaliar o desempenho de outros dois *solvers*: SeDuMi e SDPT3, ambos de distribuição gratuita e código aberto. O sistema teste para os três *solvers* tratou-se de um gerador síncrono conectado a um barramento infinito cujo amortecimento mínimo era inferior a 3% para 880 pontos de operação, ou seja, havia 880 matrizes distintas em espaço de estados que definiam a máquina síncrona. O êxito das implementações dos três *solvers* no algoritmo de iteração V-K proporcionou um único controlador por realimentação de saída para os 880 pontos de operação com amortecimento mínimo para o sistema superior a 5%, respeitando todas as desigualdades matriciais impostas. Ao final, o desempenho dos *solvers* SeDuMi e SDPT3 foi comparado com relação ao *solver* mincx em tempo de processamento, precisão numérica requerida e facilidade de uso.

Palavras-chaves: Sistemas Elétricos de Potência, Controle Robusto, Estabilidade Angular, Desigualdades Matriciais Lineares, Compensador de Fase, Sistema com Realimentação de Saída.

ABSTRACT

BENTO, M. E. C. (2013). **A Comparative Study between Two Algorithms for Solving Linear Matrix Inequalities Applied to the Design of Controllers for Output Feedback**. Trabalho de Conclusão de Curso - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2013.

This study course completion aimed benchmarking performance of different solvers of Linear Matrix Inequalities applied to the design of output feedback controllers using a particular technique called iteration V-K. Good results have been obtained in the literature using the solver mincx, which requires license. This study sought to evaluate the performance of two other solvers: SeDuMi and SDPT3 both free distribution and open source. The test system for the three solvers treated to a synchronous generator connected to an infinite bus whose minimum damping was less than 3% to 880 operating points, in other words, there were 880 different matrices in state space that defined the synchronous machine. The successful implementations of the three solvers iteration of the algorithm V-K provided a single output feedback controller for the 880 points of operating with minimum damping to the system exceeds 5%, respecting all the matrix inequalities imposed. At the end, the performance of solvers SeDuMi and SDPT3 was compared with respect to the solver mincx in processing time, numerical accuracy required and ease of use.

Keywords: Electrical Power Systems, Robust Control, Angle Stability, Linear Matrix Inequalities, Phase Compensator, System with Output Feedback.

Sumário

1	Introdução	9
2	Sistema	11
2.1	Malha Direta	11
2.2	PSS	13
2.3	Sistema em Malha Fechada	17
3	Algoritmo de Iteração V-K	19
3.1	Condições Iniciais	19
3.2	Algoritmo	19
3.3	Pseudo-códigos	20
3.3.1	Pseudo-código do mincx	22
3.3.2	Pseudo-código do SeDuMi e SDPT3	25
4	Código Implementado	29
4.1	Condições Iniciais	29
5	Resultados de Cada Código	31
5.1	Resultados	31
5.2	Cumprimento das Desigualdades Impostas	31
5.2.1	Resultados mincx	32
5.2.2	Resultados SeDuMi	32
5.2.3	Resultados SDPT3	32
5.3	Resposta ao Degrau	33
5.3.1	Ponto de Operação 150	34
5.3.2	Ponto de Operação 450	34
5.3.3	Ponto de Operação 750	35
6	Conclusões	37
7	Referências Bibliográficas	39
A	Álgebra Linear	41
A.1	Autovalores e Autovetores de uma Matriz	41
A.2	Polinômios e Matrizes	43
A.3	Existência de Autovetores e Autovalores	45
A.4	Polinômio Característico	47
A.5	Propriedades de Autovalores e Autovetores	48

B	Equações Diferenciais	51
B.1	Equações Diferenciais	51
B.2	Equação Diferencial de Primeira Ordem	53
B.3	Equação Diferencial Linear de Segunda Ordem	55
B.4	Transformada de Laplace	63
B.5	Transformada Inversa de Laplace	66
B.6	Sistema de Equações Diferenciais Lineares de Primeira Ordem	67
C	Sistemas de Controle	73
C.1	Sistemas de Controle	73
C.2	Modelagem Matemática de Sistemas de Controle	75
C.3	Diagrama de Blocos	77
C.4	Controladores Automáticos	77
C.4.1	Controladores de duas posições ou <i>on-off</i>	77
C.4.2	Controladores proporcionais	78
C.4.3	Controladores integrais	78
C.4.4	Controladores proporcional-integrais	79
C.4.5	Controladores proporcional-derivativos	79
C.4.6	Controladores proporcional-integral-derivativos	79
C.5	Modelagem em Espaço de Estados	80
D	Desigualdade Matricial Linear	89
D.1	Definição	89
D.2	Propriedades	91
D.3	Otimização	92
D.4	Resolução Numérica de LMI's	94
E	Solvers em Estudo	95
E.1	LMILAB	95
E.1.1	Setlmis	95
E.1.2	Lmivar	95
E.1.3	Lmiterm	96
E.1.4	Getlmis	97
E.1.5	Decnbr	97
E.1.6	Defcx	97
E.1.7	Dec2mat	98
E.2	YALMIP	98
E.2.1	SDPVAR	98

E.2.2	SET	98
E.2.3	SDPSETTINGS	99
E.2.4	SOLVESDP	99
E.3	Mincx	100
E.4	SeDuMi	101
E.5	SDPT3	102
F	Código Implementado com o mincx	107
G	Código Implementado com o SeDuMi	113
H	Código Implementado com o SDPT3	119

Lista de Figuras

1	Sistema-teste para o estudo	11
2	Diagrama Típico do PSS	13
3	PSS a ser determinado	13
4	Lugar de autovalores de \bar{A}_j	18
5	Amortecimento <i>versus</i> Ponto de Operação	32
6	Amortecimento Mínimo de Cada Ponto de Operação segundo o mincx	33
7	Amortecimento Mínimo de Cada Ponto de Operação segundo o SeDuMi	33
8	Amortecimento Mínimo de Cada Ponto de Operação segundo o SDPT3	34
9	Resposta ao Degrau para o Ponto de Operação 150	35
10	Resposta ao Degrau para o Ponto de Operação 450	35
11	Resposta ao Degrau para o Ponto de Operação 750	36
12	Modelo de um Sistema com Realimentação de Saída	74
13	Modelo de um Sistema com Realimentação de Estados	74
14	Modelo de um Sistema em Espaço de Estados com Realimentação de Saída	74
15	Modelo de um Sistema em Malha Aberta	75
16	Função convexa $f(\xi)$	94

Lista de Tabelas

1	Resultado das Implementações	32
---	--	----

1 Introdução

Estudos de estabilidade em sistemas elétricos de potência (SEP's) vêm recebendo a atenção dos pesquisadores e engenheiros desde a segunda metade do século passado, quando os sistemas elétricos que até então eram ilhados passaram a se interligar uns aos outros, trazendo diversos benefícios operacionais e econômicos (Kundur, 1994).

Um dos principais pontos estudados, na área de estabilidade a pequenas perturbações, se refere às oscilações eletromecânicas de baixa frequência, intrínsecas a qualquer sistema elétrico dotado de geradores síncronos. Estas oscilações são observáveis em diversas grandezas do sistema como, por exemplo, no ângulo do rotor dos geradores. Um baixo amortecimento de tais oscilações pode induzir o aparecimento de oscilações em diversas variáveis do sistema, por um tempo considerável, cujos impactos negativos podem ser vistos na forma de desgastes mecânicos das máquinas e problemas de qualidade da potência entregue aos consumidores (Edwards, 2000).

A principal solução em termos de custo/benefício capaz de fornecer amortecimento adicional às oscilações eletromecânicas pouco amortecidas é o uso de Estabilizadores de Sistemas de Potência (ESP's ou mais conhecidos por PSS's, sigla em inglês para *Power System Stabilizers*) conectados aos Reguladores Automáticos de Tensão (ou do inglês, AVR's, *Automatic Voltage Regulators*) dos geradores síncronos. Neste ponto, grandes avanços têm sido realizados ao longo das últimas décadas no sentido de explicar detalhadamente o comportamento dinâmico de um SEP sujeito a essas perturbações de baixa magnitude, incluindo o desenvolvimento de técnicas para projeto e análise dos diversos dispositivos de controle do sistema, no caso, dos AVR's e dos PSS's, e das relações existentes entre eles (Kundur, 1994).

Apenas a inserção de controladores do tipo PSS não é suficiente para garantir um bom amortecimento das oscilações do sistema, fazendo-se necessário uma boa sintonia dos parâmetros deste controlador. Inicialmente, os projetos de sintonia envolviam métodos de tentativa e erro, os quais tendem a ser demorados, dispendendo assim um grande tempo por parte do projetista. Com o avanço dos microcomputadores, métodos de sintonia automática começaram a ser propostos, utilizando métodos de otimização baseados em desigualdades matriciais lineares (LMI's do inglês *Linear Matrix Inequalities*)(Chilali, 1999). Os métodos baseados em LMI's tendem a ser mais rápidos, uma vez que promovem uma busca local por uma solução factível (espaço de busca menor). Porém, por se tratarem de métodos de busca locais, necessitam de uma boa condição inicial para encontrar uma solução factível. Este trabalho de pesquisa propõe uma comparação entre três *solvers* de LMI's aplicados a um método de busca local conhecido como iteração V-K para sintonia de PSS's, partindo de uma mesma condição inicial.

Pesquisas já foram realizadas utilizando-se o *solver* mincx através da técnica de iteração

V-K, para o projeto de controladores de amortecimento como, por exemplo, em (Oliveira, 2009) e em (Kuiava, 2009). O objetivo deste projeto foi testar, também, a eficiência dos *solvers* SeDuMi e SDPT3, de distribuição gratuita, no projeto de controladores por realimentação de saída. O sistema teste tratou-se de um gerador síncrono conectado ao barramento infinito com uma carga próxima ao gerador, cujo amortecimento mínimo para 880 pontos de operação era inferior a 3%.

Este trabalho está organizado da seguinte maneira: o capítulo 2 fornece informações sobre o sistema em estudo, inclusive sua modelagem. A metodologia empregada para o projeto de controladores está descrito no capítulo 3. O código implementado para cada um dos *solvers* encontra-se no capítulo 4. Os resultados das implementações estão no capítulo 5, onde discuti-se o cumprimento das desigualdades impostas. A conclusão geral encontra-se no capítulo 6, onde os três *solvers* em análise foram avaliados segundo seu tempo de processamento, precisão requerida e facilidade de utilização nas implementações de Desigualdades Matriciais Lineares.

2 Sistema

O sistema que será objeto de estudo dos *solvers* é um gerador síncrono distribuído conectado a um barramento infinito com uma carga próxima ao gerador onde as informações foram originalmente propostas em (Abreu, 2005) e cujo controlador por realimentação de saída trata-se de um Estabilizador de Sistema de Potência.

2.1 Malha Direta

O sistema em estudo, gerador síncrono distribuído conectado a um barramento infinito com uma carga próxima ao gerador, está descrito na Figura 1.

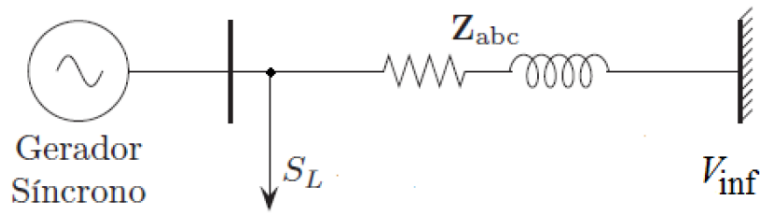


Figura 1: Sistema-teste para o estudo

onde S_L representa a carga, Z_{abc} a impedância trifásica que liga o gerador ao barramento infinito e V_{inf} a tensão do barramento infinito.

O sistema em malha direta a qual se aplicará o projeto de controlador por realimentação de saída é o gerador síncrono distribuído. A fim de o sistema apresentar baixo amortecimento que justifique a utilização de um controlador de amortecimento por realimentação de saída, o mesmo foi submetido a um conjunto de pequenas perturbações (Salim, 2011). O conjunto de perturbações compreendeu:

- Desequilíbrio de carga representado por um fator de desequilíbrio de carga l . Este fator l representa o grau de variação da potência aparente da carga de duas fases, mantendo-se a terceira fase com carga constante. A variação em uma das duas fases é incremental, ao passo que na outra fase ela é decremental, mantendo assim a potência aparente trifásica constante em todos os casos. O cálculo da nova potência aparente das cargas modificadas é realizado da seguinte forma:

$$S_{il} = (1 + l)S_{monofasica} \quad (1)$$

$$S_{dl} = (1 - l)S_{monofasica} \quad (2)$$

$$S_{ul} = S_{monofasica} \quad (3)$$

onde $S_{monofasica}$ é a potência aparente monofásica da carga no caso equilibrado, e S_{il} , S_{dl} e S_{ul} são respectivamente as potências aparentes da carga incrementada, decrementada e constante.

Foram 11 variações (0, 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100%);

- Fator de potência da carga: 0,90 a 0,98;

Foram 5 variações (0,90; 0,92; 0,94; 0,96; 0,98);

- Potência aparente da carga: 1 a 8 pu;

Foram 8 variações (1, 2, 3, 4, 5, 6, 7 e 8 pu);

- Potência do gerador: 0,3 a 0,6 pu.

Foram 2 variações (0,3 e 0,6 pu).

Este conjunto de perturbações permitiu obter $11 \times 5 \times 8 \times 2 = 880$ pontos de operação distintos para o sistema com gerador síncrono distribuído e com amortecimento mínimo inferior a 3%.

Com estes 880 pontos de operação, o gerador síncrono distribuído foi modelado em espaço de estados. As equações algébricas diferenciais que descrevem o sistema são:

$$\dot{\tilde{x}} = f(\tilde{x}, \tilde{u}, z, \lambda), \quad (4)$$

$$0 = g(\tilde{x}, \tilde{u}, z, \lambda), \quad (5)$$

$$\tilde{y} = h(\tilde{x}, \tilde{u}, z, \lambda). \quad (6)$$

onde $\tilde{x} \in \mathbb{R}^n$ é o vetor de estados, $\tilde{u} \in \mathbb{R}^p$ é o vetor de entrada de controle, $\tilde{y} \in \mathbb{R}^q$ é a saída medida, $z \in \mathbb{R}^m$ é o vetor de variáveis algébricas representando o acoplamento da rede de transmissão entre as variáveis de estado e $\lambda \in \mathbb{R}^l$ é o vetor de parâmetros representando os níveis de carga e outras quantidades definindo as condições do sistema de operação.

O sistema de equações pode ser linearizado em torno de um ponto de operação de interesse, resultando em:

$$\dot{x}_j = A_j x_j + B_j u_j, \quad (7)$$

$$y_j = C_j x_j + D_j u_j. \quad (8)$$

onde $x_j \in \mathbb{R}^n$ representa um desvio do ponto de equilíbrio \tilde{x}_{je} obtido de um valor particular do vetor de parâmetros λ . Em um modo similar, $u_j \in \mathbb{R}^p$ e $y_j \in \mathbb{R}^q$ representam o desvio de \tilde{u}_{je} e \tilde{y}_{je} , respectivamente. Detalhes do modelo linearizado, incluindo o equacionamento

da parte elétrica e da parte mecânica, tanto as equações diferenciais como as algébricas encontram-se em (Salim, 2011).

Assim, há 880 pontos de operação (A_j, B_j, C_j, D_j) para o projeto de um único controlador PSS por realimentação de saída que atenda todos estes pontos de operação.

2.2 PSS

O *Power System Stabilizer* é um controlador com a função de melhorar o amortecimento de um sistema que apresenta oscilações eletromecânicas através da inserção de um sinal estabilizante na malha do regulador de tensão dos geradores.

O PSS é composto basicamente por três blocos como pode ser visto na Figura 2: bloco *washout*, blocos de avanço-atraso de fase e bloco de ganho K .

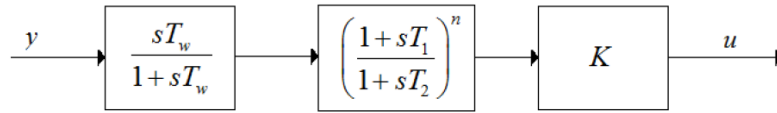


Figura 2: Diagrama Típico do PSS

- **Bloco *washout*:** o bloco *washout* é um filtro passa-altas com a constante de tempo T_w alta o suficiente para que somente os sinais associados às oscilações de velocidade passem sem nenhuma alteração. Normalmente, T_w assume valores entre 1 a 20s;
- **Bloco de avanço-atraso:** compensa o atraso de fase do sistema de forma a produzir um torque elétrico em fase com os desvios de velocidade do sistema. O índice n indica o número de estágios de avanço-atraso necessários para a compensação de fase;
- **K:** corresponde ao ganho puro inserido pelo controlador no sistema

Para o sistema em estudo foi adotado um bloco de avanço e atraso de três estágios, ou seja, $n = 3$ devido ao alto valor de compensação de fase requerido pelo sistema. Assim, o PSS a ser determinado está na Figura 3.

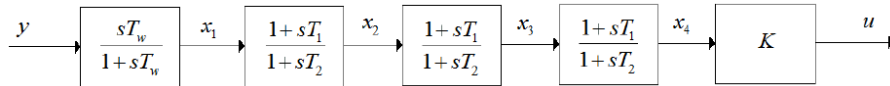


Figura 3: PSS a ser determinado

Devido o sistema em malha direta estar em espaço de estados, o PSS também será modelado em espaço de estados. Como pode ser visto pela Figura 3, cada bloco do PSS está associado a um estado $(x_1, x_2, x_3$ e $x_4)$. Assim, para o estado x_1 :

$$x_1(s) = \frac{sT_w}{1 + sT_w}y(s), \quad (9)$$

$$x_1(s) \left[\frac{1 + sT_w}{T_w} \right] = sy(s), \quad (10)$$

$$x_1(s) \left[s + \frac{1}{T_w} \right] = sy(s). \quad (11)$$

Aplicando a transformada de Laplace inversa, tem-se:

$$\dot{x}_1 + \frac{1}{T_w}x_1 = \dot{y}, \quad (12)$$

e assim o estado x_1 :

$$\dot{x}_1 = -\frac{1}{T_w}x_1 + \dot{y}. \quad (13)$$

Para o estado x_2 tem-se:

$$x_2(s) = \frac{1 + sT_1}{1 + sT_2}x_1(s), \quad (14)$$

$$x_2(s)[1 + sT_2] = x_1(s)[1 + sT_1]. \quad (15)$$

Aplicando a transformada inversa de Laplace:

$$T_2\dot{x}_2 + x_2 = T_1\dot{x}_1 + x_1, \quad (16)$$

$$\dot{x}_2 = \frac{T_1}{T_2}\dot{x}_1 + \frac{1}{T_2}x_1 - \frac{1}{T_2}x_2. \quad (17)$$

Substituindo o estado \dot{x}_1 já obtido anteriormente, tem-se:

$$\dot{x}_2 = \frac{T_1}{T_2} \left[-\frac{1}{T_w}x_1 + \dot{y} \right] + \frac{1}{T_2}x_1 - \frac{1}{T_2}x_2. \quad (18)$$

e o estado x_2 é:

$$\dot{x}_2 = \left[\frac{1}{T_2} - \frac{T_1}{T_w T_2} \right] x_1 - \frac{1}{T_2}x_2 + \frac{T_1}{T_2}\dot{y}. \quad (19)$$

Para o estado x_3 :

$$x_3(s) = \frac{1 + sT_1}{1 + sT_2}x_2(s), \quad (20)$$

$$x_3(s)[1 + sT_2] = x_2(s)[1 + sT_1]. \quad (21)$$

Aplicando a transformada inversa de Laplace:

$$T_2\dot{x}_3 + x_3 = T_1\dot{x}_2 + x_2, \quad (22)$$

$$\dot{x}_3 = \frac{T_1}{T_2}\dot{x}_2 + \frac{1}{T_2}x_2 - \frac{1}{T_2}x_3. \quad (23)$$

Substituindo o estado \dot{x}_2 já obtido anteriormente, tem-se:

$$\dot{x}_3 = \frac{T_1}{T_2} \left\{ \left[\frac{1}{T_2} - \frac{T_1}{T_w T_2} \right] x_1 - \frac{1}{T_2} x_2 + \frac{T_1}{T_2} \dot{y} \right\} + \frac{1}{T_2} x_2 - \frac{1}{T_2} x_3, \quad (24)$$

e o estado x_3 é

$$\dot{x}_3 = \left[\frac{T_1}{T_2^2} - \frac{T_1^2}{T_w T_2^2} \right] x_1 + \left[\frac{1}{T_2} - \frac{T_1}{T_2^2} \right] x_2 - \frac{1}{T_2} x_3 + \left(\frac{T_1}{T_2} \right)^2 \dot{y}. \quad (25)$$

Para o estado x_4 :

$$x_4(s) = \frac{1 + sT_1}{1 + sT_2} x_3(s), \quad (26)$$

$$x_4(s)[1 + sT_2] = x_3(s)[1 + sT_1]. \quad (27)$$

Aplicando a transformada inversa de Laplace:

$$T_2\dot{x}_4 + x_4 = T_1\dot{x}_3 + x_3, \quad (28)$$

$$\dot{x}_4 = \frac{T_1}{T_2}\dot{x}_3 + \frac{1}{T_2}x_3 - \frac{1}{T_2}x_4. \quad (29)$$

Substituindo o estado \dot{x}_3 já obtido anteriormente, tem-se:

$$\dot{x}_4 = \frac{T_1}{T_2} \left\{ \left[\frac{T_1}{T_2^2} - \frac{T_1^2}{T_w T_2^2} \right] x_1 + \left[\frac{1}{T_2} - \frac{T_1}{T_2^2} \right] x_2 - \frac{1}{T_2} x_3 + \frac{T_1^2}{T_2^2} \dot{y} \right\} + \frac{1}{T_2} x_3 - \frac{1}{T_2} x_4, \quad (30)$$

e o estado x_4 é:

$$\dot{x}_4 = \left[\frac{T_1^2}{T_2^3} - \frac{T_1^3}{T_w T_2^3} \right] x_1 + \left[\frac{T_1}{T_2^2} - \frac{T_1^2}{T_2^3} \right] x_2 + \left[\frac{1}{T_2} - \frac{T_1}{T_2^2} \right] x_3 - \frac{1}{T_2} x_4 + \left(\frac{T_1}{T_2} \right)^3 \dot{y}. \quad (31)$$

Fazendo as substituições de variáveis:

$$\alpha = \frac{1}{T_w}, \quad (32)$$

$$\beta = \frac{T_1}{T_2}, \quad (33)$$

$$\gamma = \frac{1}{T_2}. \quad (34)$$

As novas representações dos estados ficam:

$$\begin{cases} \dot{x}_1 = -\alpha x_1 + \dot{y} \\ \dot{x}_2 = (\gamma - \alpha\beta)x_1 - \gamma x_2 + \beta\dot{y} \\ \dot{x}_3 = (\beta\gamma - \alpha\beta^2)x_1 + (\gamma - \beta\gamma)x_2 - \gamma x_3 + \beta^2\dot{y} \\ \dot{x}_4 = (\beta^2\gamma - \alpha\beta^3)x_1 + (\beta\gamma - \beta^2\gamma)x_2 + (\gamma - \beta\gamma)x_3 - \gamma x_4 + \beta^3\dot{y} \end{cases} \quad (35)$$

Além disso, tem-se que a entrada u relaciona-se com um estado do sistema:

$$u = Kx_4. \quad (36)$$

Portanto, o sistema em espaço de estados dos PSS é:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -\alpha & 0 & 0 & 0 \\ \gamma - \alpha\beta & -\gamma & 0 & 0 \\ \beta\gamma - \alpha\beta^2 & \gamma - \beta\gamma & -\gamma & 0 \\ \beta^2\gamma - \alpha\beta^3 & \beta\gamma - \beta^2\gamma & \gamma - \beta\gamma & -\gamma \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \beta^3 \end{bmatrix} \dot{y}, \quad (37)$$

$$u = \begin{bmatrix} 0 & 0 & 0 & K \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad (38)$$

onde

$$A_c = \begin{bmatrix} -\alpha & 0 & 0 & 0 \\ \gamma - \alpha\beta & -\gamma & 0 & 0 \\ \beta\gamma - \alpha\beta^2 & \gamma - \beta\gamma & -\gamma & 0 \\ \beta^2\gamma - \alpha\beta^3 & \beta\gamma - \beta^2\gamma & \gamma - \beta\gamma & -\gamma \end{bmatrix}, \quad (39)$$

$$B_c = \begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \beta^3 \end{bmatrix}, \quad (40)$$

$$C_c = \begin{bmatrix} 0 & 0 & 0 & K \end{bmatrix}, \quad (41)$$

$$x_c = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \quad (42)$$

A representação simplificada fica:

$$\begin{cases} \dot{x}_c = A_c x_c + B_c \dot{y} \\ u = C_c x_c \end{cases} \quad (43)$$

Embora a saída derivada (\dot{y}) apareça como uma entrada para a formulação em espaço de estados do controlador, não há necessidade de uma medição direta desta derivada. Esta quantidade é exibida na formulação devido à presença do termo *washout* sT_w na função de transferência do PSS, que sintetiza uma aproximação para a derivada da saída medida. No entanto, a implementação atual do projeto dos controladores será baseado somente na saída medida, usando a relação:

$$\dot{y} = C_j A_j x_j. \quad (44)$$

Mais detalhes deste assunto podem ser vistos em (Ramos, 2005).

2.3 Sistema em Malha Fechada

A fim de analisar os sistemas em malha fechada como um todo (gerador síncrono distribuído mais controlador PSS) define-se:

$$\bar{x}_j = \begin{bmatrix} x_j \\ x_c \end{bmatrix}, \quad (45)$$

e

$$\bar{A}_j = \begin{bmatrix} A_j & B_j C_c \\ B_c C_j A_j & A_c \end{bmatrix}, \quad (46)$$

onde $j = 1, \dots, 880$, representando os 880 pontos de operação e $c = 1$, representando um único controlador PSS que proverá amortecimento adicional para todos os 880 pontos de operação.

O problema em estudo consistiu em encontrar as matrizes do controlador (A_c, B_c, C_c) considerando um critério de desempenho desejado. Um dos critérios mais normalmente usados é que todos os autovalores da matriz \bar{A}_j apresentem um amortecimento mínimo ζ_0 desejado. Esta especificação define o lugar de autovalores de \bar{A}_j com desempenho aceitável

como um cone no plano complexo. A Figura 4 ilustra isto, onde $\theta = \arccos(\zeta_0)$.

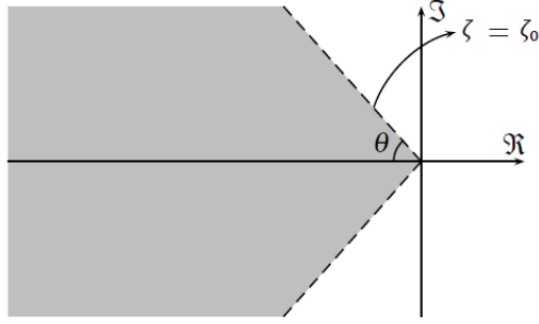


Figura 4: Lugar de autovalores de \bar{A}_j

Foi utilizado a teoria de estabilidade quadrática de Lyapunov. A ideia consiste em encontrar as matrizes A_c , B_c e C_c e uma matriz P que satisfaçam:

$$P^T = P \succ 0 \quad (47)$$

,

$$\bar{A}_j^T P + P \bar{A}_j \prec 0 \quad (48)$$

,

para $j = 1, \dots, 880$, onde as notações $M \succ 0$ e $N \prec 0$ indicam matrizes definidas positiva e negativa respectivamente. Se existir as matrizes que satisfaçam estas condições, cada \tilde{x}_{je} de equilíbrio é estável localmente. Entretanto, ser estável localmente não é um critério suficiente de desempenho. Além disso, o amortecimento mínimo de \bar{A}_j também é um critério a ser alcançado. Para tanto, deve-se encontrar as matrizes A_c , B_c e C_c e uma matriz P que satisfaçam:

$$\begin{bmatrix} \text{sen}(\theta)(\bar{A}_j^T P + P \bar{A}_j) & \text{cos}(\theta)(\bar{A}_j^T P - P \bar{A}_j) \\ \text{cos}(\theta)(\bar{A}_j^T P - P \bar{A}_j)^T & \text{sen}(\theta)(\bar{A}_j^T P + P \bar{A}_j) \end{bmatrix} \prec 0 \quad (49)$$

,

onde $\theta = \arccos(\zeta_0)$.

O cumprimento desta desigualdade garante estabilidade quadrática para o sistema no ponto de equilíbrio além de proporcionar o amortecimento desejado.

3 Algoritmo de Iteração V-K

A fim de encontrar as matrizes A_c , B_c e C_c do controlador e uma única matriz P que satisfaçam as desigualdades matriciais impostas na seção anterior proporcionando o amortecimento mínimo ζ_0 desejado e estabilidade quadrática, foi utilizado o algoritmo de iteração V-K (Oliveira, 2009 e Kuiuava, 2009).

3.1 Condições Iniciais

Os dados da planta (A_j , B_j e C_j) já são conhecidos. São necessários sete estados para definir a máquina síncrona, assim $A_j \in \mathbb{R}^7$, $B_j \in \mathbb{R}^{7 \times 1}$ e $C_j \in \mathbb{R}^{1 \times 7}$.

O algoritmo de iteração V-K é um método baseado em desigualdades matriciais lineares que promovem uma busca local por uma solução factível. Por conta disso, necessitam de uma boa condição inicial para encontrar uma solução factível. As condições iniciais para o controlador PSS nos três algoritmos foram:

$$\alpha = 0,10 \tag{50}$$

,

$$\beta = 7,2079 \tag{51}$$

,

$$\gamma = 26,3157 \tag{52}$$

,

$$K = 0,000104 \tag{53}$$

.

Estas condições iniciais proporcionaram um amortecimento mínimo (ζ) para o sistema de 4% e foi solicitado no algoritmo de iteração V-K um amortecimento mínimo de 5% ($\zeta_0 = 0,05$). As únicas variáveis do controlador PSS foram γ e K . O parâmetro α manteve-se fixo porque ele está associado à constante de tempo T_w do filtro passa-alta, que deseja ser fixo para que a seleção dos sinais associados às oscilações de velocidade também seja fixa. O parâmetro β também manteve-se fixo porque caso não fosse ele deixaria o sistema não linear já que a matriz B_c possui β^2 e β^3 em sua composição, e deseja-se trabalhar com sistemas lineares.

3.2 Algoritmo

Passo 1: Encontrar uma única matriz P que satisfaça (51) e (52) com as condições iniciais das matrizes A_c , B_c e C_c para inicializar o algoritmo;

$$P^T = P \succ 0 \quad (54)$$

$$\bar{A}_j^T P + P \bar{A}_j \prec 0 \quad (55)$$

onde $j = 1, \dots, 880$.

Passo 2: Fixe os valores da matriz P obtidos anteriormente e minimize σ sujeito às entradas γ de A_c e K de C_c :

$$\begin{bmatrix} \text{sen}\theta(\bar{A}_j^T P + P \bar{A}_j) & \text{cos}\theta(\bar{A}_j^T P - P \bar{A}_j) \\ \text{cos}\theta(\bar{A}_j^T P - P \bar{A}_j)^T & \text{sen}\theta(\bar{A}_j^T P + P \bar{A}_j) \end{bmatrix} - \sigma I \prec 0 \quad (56)$$

onde $\theta = \arccos(\zeta_0)$

Passo 3: Se $\sigma < 0$, pare; se não, fixe os valores de A_c e C_c obtidos anteriormente e minimize σ sujeito às entradas de P , $P^T = P \succ 0$ e (53).

Passo 4: Se $\sigma < 0$, pare; se não, retorne ao passo 2.

Para resolver as desigualdades matriciais lineares deste algoritmo foram utilizados três *solvers* para serem comparados: mincx, SeDuMi e SDPT3, um para cada algoritmo V-K implementado. A implementação do algoritmo com o *solver* mincx foi efetuada na plataforma LMILAB e os *solvers* SeDuMi e SDPT3 na plataforma YALMIP.

3.3 Pseudo-códigos

Em virtude dos *solvers* mincx, SeDuMi e SDPT3 terem sido implementados em plataformas diferentes, serão apresentados dois pseudo-códigos nesta seção. A implementação do código com o *solver* mincx requereu alterações nas matrizes do controlador a fim isolar a variável a se determinar (γ e K).

A matriz A_c foi definida como:

$$A_c = \begin{bmatrix} -\alpha & 0 & 0 & 0 \\ \gamma - \alpha\beta & -\gamma & 0 & 0 \\ \beta\gamma - \alpha\beta^2 & \gamma - \beta\gamma & -\gamma & 0 \\ \beta^2\gamma - \alpha\beta^3 & \beta\gamma - \beta^2\gamma & \gamma - \beta\gamma & -\gamma \end{bmatrix}. \quad (57)$$

O código no *solver* mincx exigiu:

$$A_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \beta & 1 & 0 \\ 0 & \beta^2 & \beta & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ \gamma & -\gamma & 0 & 0 \\ 0 & \gamma & -\gamma & 0 \\ 0 & 0 & \gamma & -\gamma \end{bmatrix} + \begin{bmatrix} -\alpha & 0 & 0 & 0 \\ -\alpha\beta & 0 & 0 & 0 \\ -\alpha\beta^2 & 0 & 0 & 0 \\ -\alpha\beta^3 & 0 & 0 & 0 \end{bmatrix} \quad (58)$$

Definindo as matrizes:

$$A_{c2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \beta & 1 & 0 \\ 0 & \beta^2 & \beta & 1 \end{bmatrix} \quad (59)$$

$$A_{c2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \gamma & -\gamma & 0 & 0 \\ 0 & \gamma & -\gamma & 0 \\ 0 & 0 & \gamma & -\gamma \end{bmatrix} \quad (60)$$

e

$$A_{c3} = \begin{bmatrix} -\alpha & 0 & 0 & 0 \\ -\alpha\beta & 0 & 0 & 0 \\ -\alpha\beta^2 & 0 & 0 & 0 \\ -\alpha\beta^3 & 0 & 0 & 0 \end{bmatrix} \quad (61)$$

Logo a matriz A_c é definida como:

$$A_c = A_{c2}A_{c1} + A_{c3} \quad (62)$$

onde a matriz A_{c1} possui a variável a se determinar (γ).

Então, a matriz \bar{A}_j é:

$$\bar{A}_j = \begin{bmatrix} A_j & B_j C_c \\ B_c C_j A_j & A_{c2} A_{c1} + A_{c3} \end{bmatrix} \quad (63)$$

Definiu-se também:

$$P = \begin{bmatrix} X & Z \\ Z' & Y \end{bmatrix} \quad (64)$$

Assim, as setenças matriciais ficam:

$$\bar{A}_j^T P + P \bar{A}_j = \begin{bmatrix} XA_j + A_jX + & ZA_{c2}A_{c1} + ZA_{c3} + A_j^T Z + \\ ZB_c C_j A_j + A_j^T C_j^T B_c^T Z^T & XB_j C_c + A_j^T C_j^T B_c^T Y \\ Z^T A_j + A_{c1}^T A_{c2}^T Z^T + A_{c3}^T Z^T + & YA_{c2}A_{c1} + YA_{c1} + A_{c1}^T A_{c2}^T Y + A_{c3}^T Y + \\ YB_c C_j A_j + C_c^T B_j^T X & Z^T B_j C_c + C_c^T B_j^T Z \end{bmatrix} \quad (65)$$

$$\bar{A}_j^T P - P \bar{A}_j = \begin{bmatrix} A_j X - X A_j + & A_j^T Z - Z A_{c2} A_{c1} - Z A_{c3} + \\ A_j^T C_j^T B_c^T Z^T - Z B_c C_j A_j & A_j^T C_j^T B_c^T Y - X B_j C_c \\ A_{c1}^T A_{c2}^T Z^T + A_{c3}^T Z^T - Z^T A_j + & A_{c1}^T A_{c2}^T Y - Y A_{c2} A_{c1} - Y A_{c1} - A_{c3}^T Y + \\ C_c^T B_j^T X - Y B_c C_j A_j & C_c^T B_j^T Z - Z^T B_j C_c \end{bmatrix} \quad (66)$$

Estas duas setenças matriciais compõem a desigualdade matricial 53.

3.3.1 Pseudo-código do mincx

Início

Var i: inteiro

Para i de 1 até 880

Aj(i) < - dadodeentrada(i)

Fim Para

Bj < - [0 0 0 0 866,6667 0 0]'

Cj < - [0 1 0 0 0 0 0]

alfa < - 0,1

beta < - 7,2079

gama < - 26,3157

K < - 0,000104

zeta < - 0,05

teta < - arccos(zeta)

Ac < - [-alfa, 0, 0, 0; gama - alfa x beta, -gama, 0, 0; beta x gama - alfa x beta x beta, gama - beta x gama, -gama, 0; beta x beta x gama - alfa x beta x beta x beta, beta x gama - beta x beta x gama, gama - beta x gama, -gama]

Bc < - [1; beta; beta x beta; beta x beta x beta]

CC < - [0, 0, 0, K]

Para i de 1 até 880

A(i) < - [Aj(i), Bj x Cc; Bc x Cj x Aj, Ac]

Fim Para

Var P[11][11]: real

ineq1 < - P >= 0

Para i de 1 até 880

ineq1 < - ineq1 + A(i)' x P + P x A <= 0

Fim Para

Solução < - *solver(ineq1)*

Var passo < - 0

Para i de 1 até 10

passo < - passo + 1

Var gama: real

Var K: real

Ac1 < - [0, 0, 0, 0; γ , $-\gamma$, 0, 0; 0, γ , $-\gamma$, 0; 0, 0, γ , $-\gamma$]

Ac2 < - [1, 0, 0, 0; 0, 1, 0, 0; 0, β , 1, 0; 0, β^2 , β , 1]

Ac3 < - [$-\alpha$, 0, 0, 0; $-\alpha\beta$, 0, 0, 0; $-\alpha\beta^2$, 0, 0, 0; $-\alpha\beta^3$, 0, 0, 0]

Bc < - [1; beta; beta x beta; beta x beta x beta]

CC < - [0, 0, 0, K]

X < - P(1:7,1:7)

Y < - P(8:11,8:11)

Z < - P(1:7,8:11)

Var sigma: real

I1 < - Matriz Identidade (7 x 7)

I2 < - Matriz Identidade (4 x 4)

Para i de 1 até 880

ineq2[1][1] < - sen(teta) x [Aj(i)' x X + X x Aj(i) + Z x Bc x Cj x Aj(i) + Aj(i)' x Cj' x Bc' x Z' - σ x I1]

$$\text{ineq2}[1][2] < - \text{sen}(\text{teta}) \times [\text{Aj}(i)' \times Z + Z \times \text{Ac2} \times \text{Ac1} + Z \times \text{Ac3} + \text{Aj}(i)' \times \text{Cj}' \times \text{Bc}' \times Y + X \times \text{Bj} \times \text{Cc}]$$

$$\text{ineq2}[2][2] < - \text{sen}(\text{teta}) \times [\text{Ac1}' \times \text{Ac2}' \times Y + Y \times \text{Ac2} \times \text{Ac1} + \text{Ac3}' \times Y + \text{Ac3} \times Y + \text{Cc}' \times \text{Bj}' \times Z + Z \times \text{Bj} \times \text{Cc} - \sigma \times \text{I2}]$$

$$\text{ineq2}[1][3] < - \text{cos}(\text{teta}) \times [\text{Aj}(i)' \times X - X \times \text{Aj}(i) + \text{Aj}(i)' \times \text{Cj}' \times \text{Bc}' \times Z' + Z \times \text{Bc} \times \text{Cj} \times \text{Aj}(i)]$$

$$\text{ineq2}[1][4] < - \text{cos}(\text{teta}) \times [\text{Aj}(i)' \times Z - Z \times \text{Ac2} \times \text{Ac1} - Z \times \text{Ac3} + \text{Aj}(i)' \times \text{Cj}' \times \text{Bc}' \times Y - X \times \text{Bj} \times \text{Cc}]$$

$$\text{ineq2}[2][3] < - \text{cos}(\text{teta}) \times [Z' \times \text{Aj}(i) + \text{Ac1}' \times \text{Ac2}' \times Z' + \text{Ac3}' \times Z - Y \times \text{Bc} \times \text{Cj} \times \text{Aj}(i) + \text{Cc}' \times \text{Bj}' \times X]$$

$$\text{ineq2}[2][4] < - \text{cos}(\text{teta}) \times [\text{Ac1}' \times \text{Ac2}' \times Y - Y \times \text{Ac2} \times \text{Ac1} + \text{Ac3}' \times Y + Y \times \text{Ac3} - Z' \times \text{Bj} \times \text{Cc} + \text{Cc}' \times \text{Bj}' \times Z]$$

$$\text{ineq2}[3][3] < - \text{sen}(\text{teta}) \times [X \times \text{Aj}(i) + \text{Aj}(i)' \times X + Z \times \text{Bc} \times \text{Cj} \times \text{Aj}(i) + \text{Aj}(i)' \times \text{Cj}' \times \text{Bc}' \times Z' - \sigma \times \text{I1}]$$

$$\text{ineq2}[3][4] < - \text{sen}(\text{teta}) \times [\text{Aj}(i)' \times Z + Z \times \text{Ac2} \times \text{Ac1} + Z \times \text{Ac3} + \text{Aj}(i)' \times \text{Cj}' \times \text{Bc}' \times Y + X \times \text{Bj} \times \text{Cc}]$$

$$\text{ineq2}[4][4] < - \text{sen}(\text{teta}) \times [\text{Ac1}' \times \text{Ac2}' \times Y + Y \times \text{Ac2} \times \text{Ac1} + \text{Ac3}' \times Y + Y \times \text{Ac3} + \text{Cc}' \times \text{Bj}' \times Z + Z' \times \text{Bj} \times \text{Cc} - \sigma \times \text{I2}]$$

Fim Para

Solução < - *solver*(ineq2)

Se sigma < 0

Parar o Algoritmo

Fim Se

Var P[11][11]: real

Var sigma: real

Para i de 1 até 880

$$M(i) < - [\text{sen}(\text{teta}) \times (\text{A}(i)' \times P + P \times \text{A}(i)), \text{cos}(\text{teta}) \times (\text{A}(i)' \times P - P \times \text{A}(i)); \text{cos}(\text{teta}) \times (\text{A}(i)' \times P - P \times \text{A}(i))', \text{sen}(\text{teta}) \times (\text{A}(i)' \times P + P \times \text{A}(i))]$$

Fim Para

$$\text{ineq3} < - P > 0$$

Para i de 1 até 880

$$\text{ineq3} < - \text{ineq3} + (M(i) + \text{sigma} \times \text{I} \leq 0)$$

Fim Para

Solução \leftarrow *solver*(ineq3)

Se $\sigma < 0$

Parar o Algoritmo

Fim Se

Fim Para

Fim Inicio

3.3.2 Pseudo-código do SeDuMi e SDPT3

Inicio

Var i: inteiro

Para i de 1 até 880

Aj(i) \leftarrow *dadodeentrada*(i)

Fim Para

Bj \leftarrow [0 0 0 0 866,6667 0 0]'

Cj \leftarrow [0 1 0 0 0 0 0]

alfa \leftarrow -0,1

beta \leftarrow -7,2079

gama \leftarrow -26,3157

K \leftarrow -0,000104

zeta \leftarrow -0,05

teta \leftarrow *arccos*(zeta)

Ac \leftarrow [-alfa, 0, 0, 0; gama - alfa x beta, -gama, 0, 0; beta x gama - alfa x beta x beta, gama - beta x gama, -gama, 0; beta x beta x gama - alfa x beta x beta x beta, beta x gama - beta x beta x gama, gama - beta x gama, -gama]

Bc \leftarrow [1; beta; beta x beta; beta x beta x beta]

CC \leftarrow [0, 0, 0, K]

Para i de 1 até 880

A(i) \leftarrow [Aj(i), Bj x Cc; Bc x Cj x Aj, Ac]

Fim Para

Var P[11][11]: real

ineq1 \leftarrow P \geq 0

Para i de 1 até 880

$ineq1 < - ineq1 + A(i)' \times P + P \times A \leq 0$

Fim Para

Solução < - *solver(ineq1)*

Var passo < - 0

Para i de 1 até 10

passo < - passo + 1

Var gama: real

Var K: real

Ac < - [-alfa, 0, 0, 0; gama - alfa x beta, -gama, 0, 0; beta x gama - alfa x beta x
beta, gama - beta x gama, -gama, 0; beta x beta x gama - alfa x beta x beta x beta, beta
x gama - beta x beta x gama, gama - beta x gama, -gama]

Bc < - [1; beta; beta x beta; beta x beta x beta]

CC < - [0, 0, 0, K]

Para i de 1 até 880

A(i) < - [Aj(i), Bj x Cc; Bc x Cj x Aj, Ac]

Fim Para

Para i de 1 até 880

M(i) < - [sen(teta) x (A(i)' x P + P x A(i)), cos(teta) x (A(i)' x P - P x A(i));
cos(teta) x (A(i)' x P - P x A(i))', sen(teta) x (A(i)' x P + P x A(i))]

Fim Para

Var sigma: real

I < - Matriz Identidade (22x22)

ineq2 < - 0

Para i de 1 até 880

ineq2 < - ineq2 + (M(i) + sigma x I <= 0)

Fim Para

Solução < - *solver(ineq2)*

Se sigma < 0

Parar o Algoritmo

Fim Se

Var P[11][11]: real

Var sigma: real

Para i de 1 até 880

$M(i) < - [\text{sen}(\text{teta}) \times (A(i)' \times P + P \times A(i)), \text{cos}(\text{teta}) \times (A(i)' \times P - P \times A(i));$
 $\text{cos}(\text{teta}) \times (A(i)' \times P - P \times A(i))', \text{sen}(\text{teta}) \times (A(i)' \times P + P \times A(i))]$

Fim Para

$\text{ineq3} < - P > 0$

Para i de 1 até 880

$\text{ineq3} < - \text{ineq3} + (M(i) + \text{sigma} \times I \leq 0)$

Fim Para

Solução < - *solver*(ineq3)

Se sigma < 0

Parar o Algoritmo

Fim Se

Fim Para

Fim Inicio

4 Código Implementado

Neste capítulo será apresentado cada uma das implementações dos três *solvers*: mincx na plataforma LMILAB e SeDuMi e SDPT3 na plataforma YALMIP, para a obtenção do controlador por realimentação de saída (PSS) do sistema em estudo (máquina síncrona) através do algoritmo de iteração V-K.

Os códigos foram executados no MatLab versão 7.9.0 (R2009b) num computador de 64 bits, com processador Intel Core i3 de 2,53 GHz, com 2,0 GB de RAM.

Os apêndices F, G e H apresentam as implementações para a obtenção de um único controlador para os 880 pontos de operação do sistema através dos *solvers* mincx, SeDuMi e SDPT3 respectivamente. Detalhes dos principais comando utilizados estão presentes no Apêndice E.

4.1 Condições Iniciais

A fim dos algoritmos convergirem, são necessárias boas condições iniciais para os parâmetros das matrizes A_c , B_c e C_c . Para os três algoritmos implementados as condições iniciais foram: $\alpha = 0,1$, $\beta = 7,2079$, $\gamma = 26,3157$, $K = 0.000104$ e $\zeta = 0,05$. Estas condições iniciais proporcionam um amortecimento mínimo do sistema próximo de 0,04 ou 4% para os 880 pontos de operação e deseja-se que o algoritmo encontre um amortecimento mínimo superior a 5% para os 880 pontos de operação, variando somente os parâmetros γ da matriz A_c e K da matriz C_c .

5 Resultados de Cada Código

As implementações do *solver* mincx na plataforma LMILAB e dos *solvers* SeDuMi e SDPT3 na plataforma YALMIP alcançaram seu objetivo, permitindo obter os resultados deste capítulo.

5.1 Resultados

As implementações dos três *solvers* obtiveram êxito. Os programas encontraram um único controlador PSS satisfazendo todas as desigualdades matriciais impostas, além de encontrar uma matriz P e a exigência de amortecimento mínimo de 5% para os 880 pontos de operação.

A Tabela 1 fornece o tempo para cada *solver* para encontrar um controlador para os 880 pontos de operação, a sua precisão requerida e o amortecimento mínimo alcançado.

Percebe-se pela Tabela 1 que o programa implementado pelo *solver* SeDuMi apresentou o menor tempo de processamento para convergir. Quanto à precisão, percebe-se que o *solver* SDPT3 apresentou a maior precisão entre os demais *solvers*. Além disso, os três *solvers* cumpriram a exigência de amortecimento mínimo superior a 5%, com maior valor do *solver* mincx.

Os três algoritmos implementados convergiram na primeira iteração e no Passo 3 da iteração V-K.

5.2 Cumprimento das Desigualdades Impostas

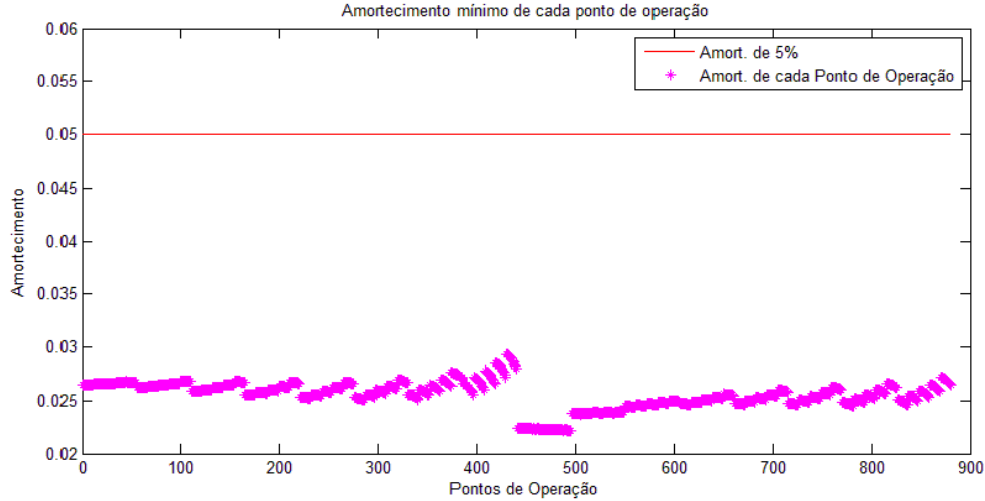
O cumprimento das desigualdades impostas no algoritmo de interação V-K fornece um amortecimento mínimo ζ_0 desejado, além de garantir estabilidade local do ponto de equilíbrio. No caso das implementações realizadas $\zeta_0 = 5\%$. Este é o benefício e propósito do algoritmo: aumentar o amortecimento mínimo da planta. No caso a planta é uma máquina síncrona conectada ao barramento infinito e aumentar seu amortecimento mínimo significa reduzir os danos impostos à máquina devido às oscilações de baixa frequência quando submetidos a pequenas perturbações. Assim sendo, a função do PSS é prover um amortecimento adicional à máquina síncrona.

Como pode ser visto na Figura 5, a malha direta do sistema em estudo, ou seja, somente a máquina síncrona apresenta um amortecimento mínimo para os 880 pontos de operação inferior a 5% (reta), no caso menor ainda que 2,93%.

As implementações dos três *solvers* cumpriram as desigualdades impostas no algoritmo de interação V-K fornecendo um amortecimento mínimo maior ou igual a 5% para cada ponto de operação, além de garantir estabilidade local do ponto de equilíbrio.

Tabela 1: Resultado das Implementações

<i>Solver</i>	Tempo (h:min:s)	Precisão	Amortecimento mínimo (%)
mincx	00:46:09	1×10^{-3}	17,3
SeDuMi	00:18:32	1×10^{-4}	12,8
SDPT3	00:27:00	1×10^{-5}	6,3

Figura 5: Amortecimento *versus* Ponto de Operação

5.2.1 Resultados mincx

A convergência do *solver* mincx forneceu $K = 1,042$ e $\gamma = 167,886$, as únicas variáveis do controlador. Assim sendo, o controlador PSS apresenta todos os seus parâmetros definidos. A Figura 6 fornece o amortecimento mínimo de cada ponto de operação com estes valores dos parâmetros convergidos do controlador PSS, onde se observa que o amortecimento mínimo de 5% foi alcançado.

5.2.2 Resultados SeDuMi

A convergência do *solver* SeDuMi forneceu $K = 0,0006$ e $\gamma = 23,7490$, as únicas variáveis do controlador. Assim sendo, o controlador PSS apresenta todos os seus parâmetros definidos. A Figura 7 fornece o amortecimento mínimo de cada ponto de operação com estes valores dos parâmetros convergidos do controlador PSS, onde se observa que o amortecimento mínimo de 5% foi alcançado.

5.2.3 Resultados SDPT3

A convergência do *solver* SDPT3 forneceu $K = 0,00026$ e $\gamma = 26,26750$, as únicas variáveis do controlador. Assim sendo, o controlador PSS apresenta todos os seus parâmetros definidos. A Figura 8 fornece o amortecimento mínimo de cada ponto de operação

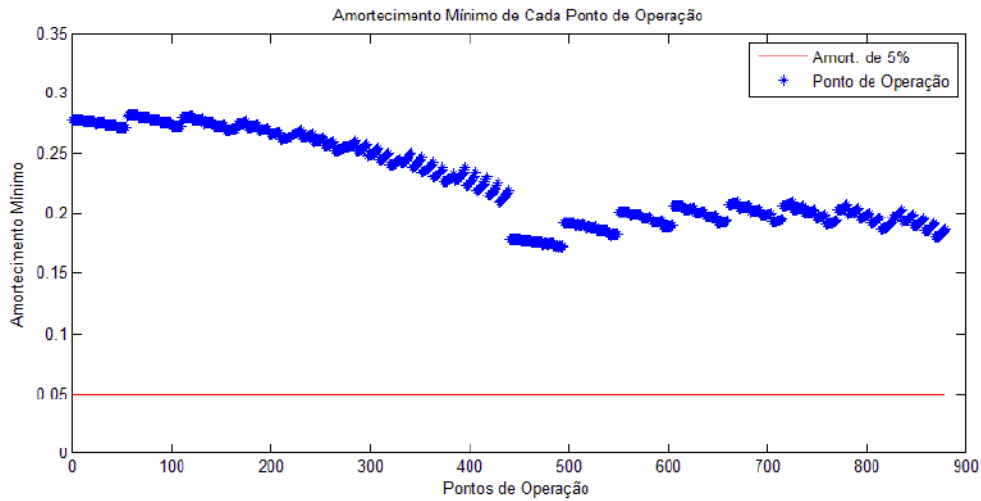


Figura 6: Amortecimento Mínimo de Cada Ponto de Operação segundo o mincx

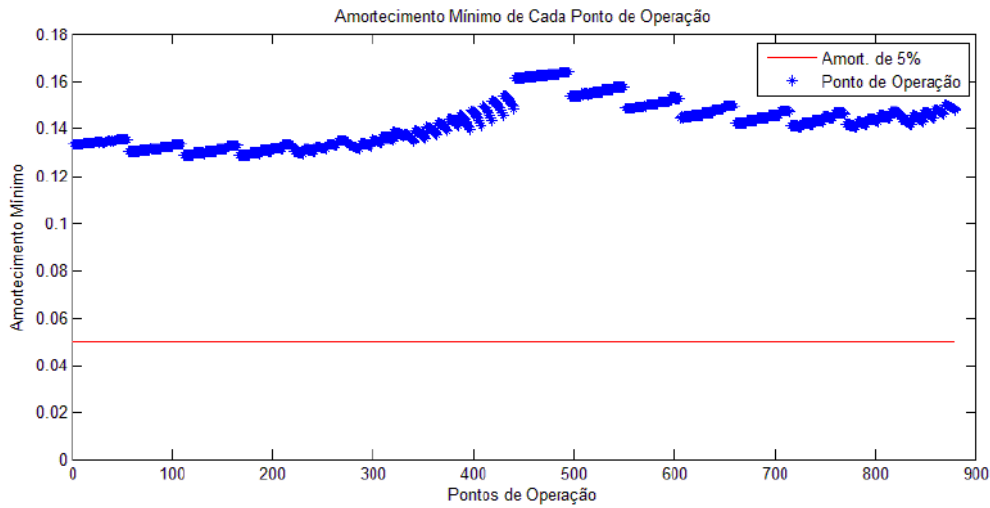


Figura 7: Amortecimento Mínimo de Cada Ponto de Operação segundo o SeDuMi

com estes valores dos parâmetros convergidos do controlador PSS, onde se observa que o amortecimento mínimo de 5% foi alcançado.

5.3 Resposta ao Degrau

A fim de analisar os efeitos do controlador PSS encontrado para cada *solver*, foram escolhidos três pontos de operação. Para cada um destes pontos de operação, a entrada recebeu um degrau unitário e foi analisada a saída do sistema.

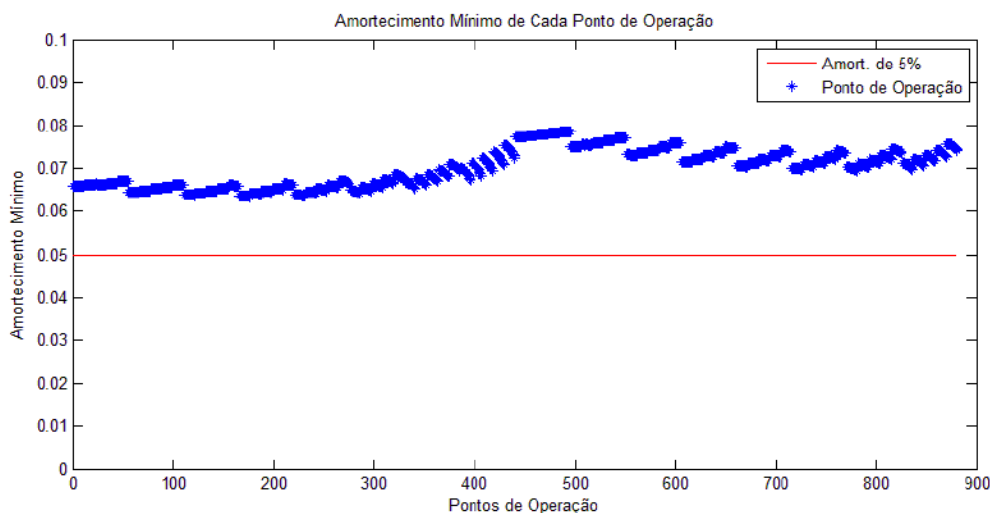


Figura 8: Amortecimento Mínimo de Cada Ponto de Operação segundo o SDPT3

5.3.1 Ponto de Operação 150

Para este primeiro exemplo foi escolhido o ponto de operação de número 150 no conjunto de 880 pontos. Este ponto de operação proporcionou um amortecimento mínimo de 27,3% pelo *solver* mincx, 13,1% pelo *solver* SeDuMi e 6,5% pelo *solver* SDPT3.

A Figura 9 ilustra a resposta ao degrau para o sistema para um tempo de 10 s. Observa-se que os *solvers* apresentaram um sinal de saída com tempo menor para chegar ao regime permanente que o sinal de saída da planta, que é característico de sistemas mais amortecidos. Percebe-se pela A Figura 9 que a ordem crescente de respostas que chegaram ao regime permanente (mincx, SeDuMi, SDPT3 e Planta) é justamente a ordem decrescente de amortecimento mínimo encontrado para o ponto de operação 150 (mincx, SeDuMi, SDPT3 e Planta).

5.3.2 Ponto de Operação 450

Para este exemplo foi escolhido o ponto de operação de número 450 no conjunto de 880 pontos. Este ponto de operação proporcionou um amortecimento mínimo de 17,9% pelo *solver* mincx, 16,2% pelo *solver* SeDuMi e 7,7% pelo *solver* SDPT3.

A Figura 10 ilustra a resposta ao degrau para o sistema para um tempo de 10 s. Observa-se que os *solvers* apresentaram um sinal de saída com tempo menor para chegar ao regime permanente que o sinal de saída da planta, que é característico de sistemas mais amortecidos. Percebe-se pela A Figura 10 que a ordem crescente de respostas que chegaram ao regime permanente (mincx, SeDuMi, SDPT3 e Planta) é justamente a ordem decrescente de amortecimento mínimo encontrado para o ponto de operação 450 (mincx, SeDuMi, SDPT3 e Planta).

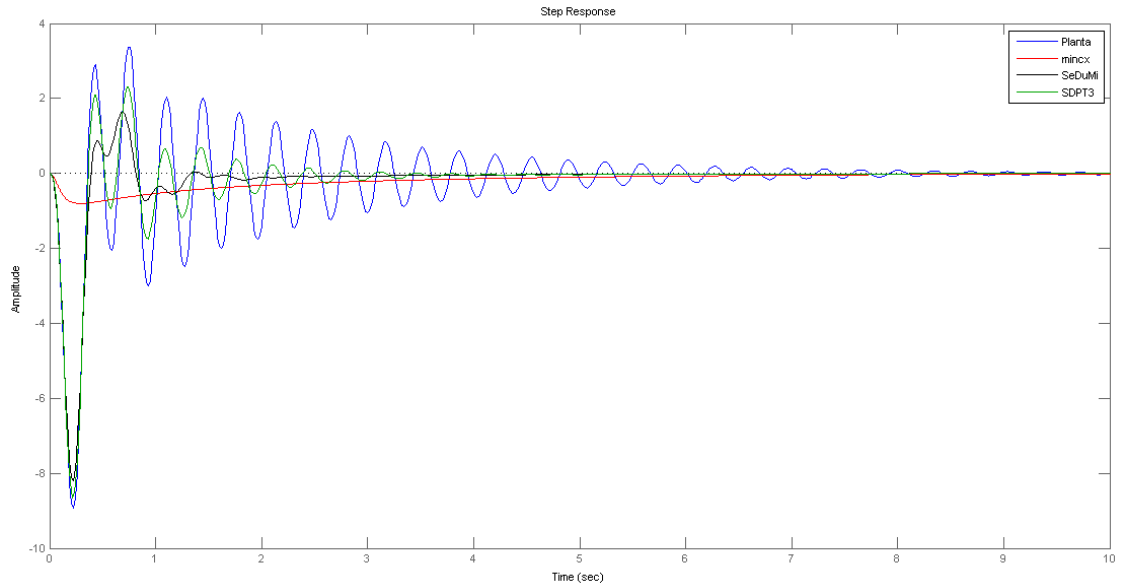


Figura 9: Resposta ao Degrau para o Ponto de Operação 150

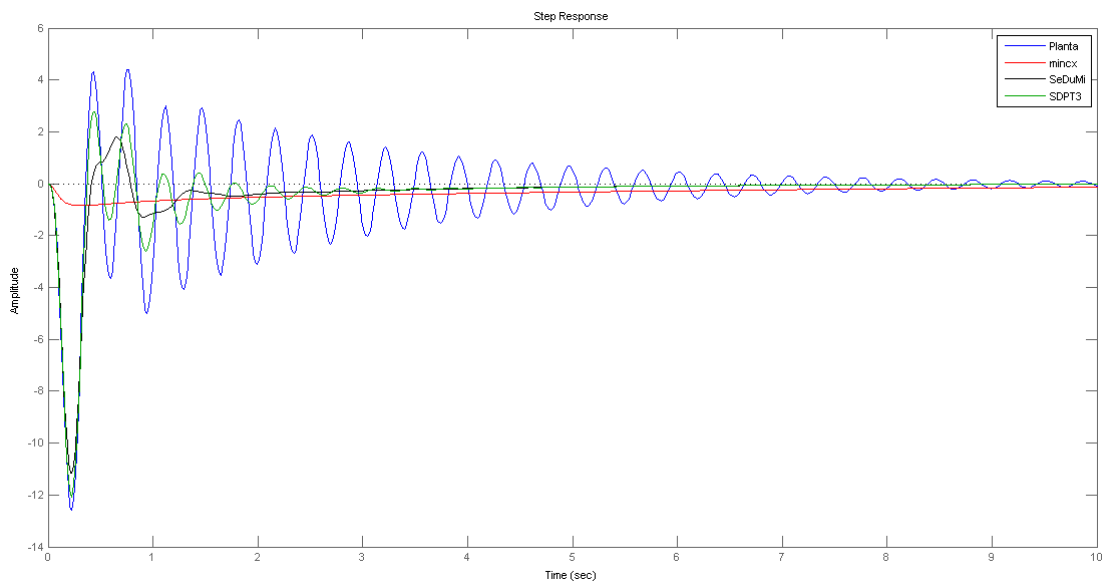


Figura 10: Resposta ao Degrau para o Ponto de Operação 450

5.3.3 Ponto de Operação 750

Para este exemplo foi escolhido o ponto de operação de número 750 no conjunto de 880 pontos. Este ponto de operação proporcionou um amortecimento mínimo de 19,6% pelo *solver* mincx, 14,5% pelo *solver* SeDuMi e 7,3% pelo *solver* SDPT3.

A Figura 11 ilustra a resposta ao degrau para o sistema para um tempos de 10 s. Observa-se que os *solvers* apresentaram um sinal de saída com tempo menor para chegar ao regime permanente que o sinal de saída da planta, que é característico de sistemas

mais amortecidos. Percebe-se pela A Figura 11 que a ordem crescente de respostas que chegaram ao regime permanente (mincx, SeDuMi, SDPT3 e Planta) é justamente a ordem decrescente de amortecimento mínimo encontrado para o ponto de operação 750 (mincx, SeDuMi, SDPT3 e Planta).

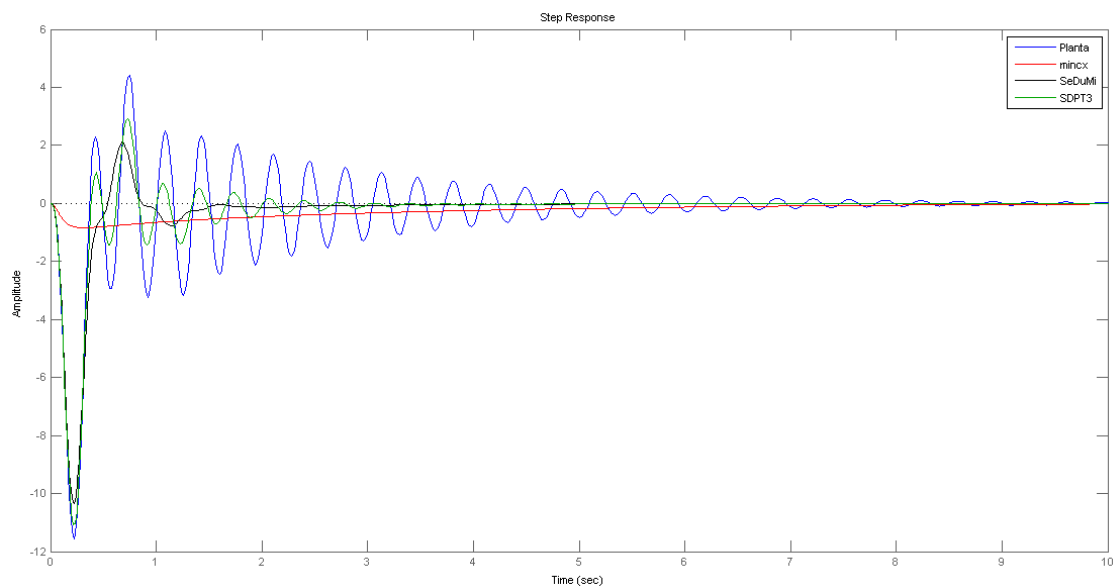


Figura 11: Resposta ao Degrau para o Ponto de Operação 750

6 Conclusões

Todas as etapas deste projeto de pesquisa foram cumpridas, permitindo tirar as conclusões a seguir.

Os três *solvers* em análise: mincx, SeDuMi e SDPT3 cumpriram com seu objetivo: fornecer um único controlador com realimentação de saída (PSS) para todos os 880 pontos de operação do gerador síncrono distribuído respeitando todas as desigualdades impostas da técnica de iteração V-K. Como resultado, o sistema como um todo apresentou um amortecimento mínimo superior a 5% (vide Figuras 6, 7 e 8) para todos os pontos de operação do gerador síncrono distribuído, além de garantir estabilidade quadrática para todos os pontos de equilíbrio do sistema.

Os três *solvers* conseguiram cumprir seu objetivo para diferentes níveis de precisão requeridos e tempos de processamento. Como pode ser observado na Tabela 1, o algoritmo *solver* SeDuMi convergiu em menor tempo. No entanto, o que apresentou maior precisão durante o processo de convergência foi o *solver* SDPT3. Assim, os *solvers* SeDuMi e SDPT3, de licença gratuita, apresentaram desempenho semelhante ou melhor em relação ao *solver* mincx, que requer licença paga de uso, no projeto de controladores por realimentação de saída.

Outra medida de comparação que pode ser analisada é a facilidade no uso dos *solvers* segundo sua plataforma. O *solver* mincx requer sua implementação na plataforma LMILAB enquanto os *solvers* SeDuMi e SDPT3, na plataforma YALMIP. A implementação dos *solvers* na plataforma YALMIP apresentou maior facilidade, pois as definições das desigualdades não exigiam que fossem em função das variáveis ou matrizes a se determinar como foi o caso na plataforma LMILAB. A implementação na plataforma LMILAB exigiu mais linhas de código que na implementação na plataforma YALMIP.

Assim sendo, o *solver* que apresentou melhor desempenho segundo o tempo de processamento necessário foi o SeDuMi. No entanto, os *solvers* SDPT3 e mincx também cumpriram com o seu objetivo de projeto: fornecer um controlador por realimentação de saída para todos os 880 pontos de operação.

7 Referências Bibliográficas

- [1] Kundur, P. (1994), *Power System Stability and Control*, New York, NY: McGraw-Hill, 1994.
- [2] Edwards, F. V.; Dudgeon, G. J. W.; McDonald, J. R.; Leithead, W. E. (2000), *Dynamics of distribution networks with distributed generation*, in Proc. IEEE Power Engineering Society Summer Meeting, vol. 2, pp. 1032-1037, 16-20 July 2000.
- [3] Chilali, M.; Gahinet, P.; Apkarian, P. (1999), *Robust Pole Placement in LMI Regions*, IEEE Transactions on Automatic Control, vol. 44, pp. 2257-2270, 1999.
- [4] Oliveira, R. V. de; Ramos, R. A.; Bretas, N. G. (2009), *An algorithm for computerized automatic tuning of power system stabilizers*, Control Engineering Practice, vol. 18, pp. 45-54, 2009.
- [5] Kuiava, R.; Oliveira, R. V. de; Ramos, R. A.; Bretas, N. G. (2009), *Automatic tuning method for the design of supplementary damping controllers for flexible alternating current transmission system devices*, IET Generation, Transmission & Distribution, vol. 3, no 10, pp. 919-929, 2009.
- [6] Abreu, L. V. L. de (2005), *Análise de Desempenho Dinâmico de Geradores Síncronos Distribuídos de Energia Elétrica.*, Dissertação (Mestrado em Engenharia Elétrica), UNICAMP, Campinas-SP, 2005.
- [7] Salim, Rodrigo Hartsein (2011), *Uma Nova Abordagem para a Análise da Estabilidade a Pequenas Perturbações em Sistemas de Distribuição de Energia Elétrica com Geradores Síncronos Distribuídos*, Tese (Doutorado em Engenharia Elétrica), USP, São Carlos-SP, 2011.
- [8] Ramos, R. A.; Martins, A. C. P.; Bretas, N. G. (2005), *An improved methodology for the design of power system damping controllers*, IEEE Transactions on Power Systems, vol. 20, no 4, pp. 1938-1945, 2005.
- [9] Guidorizzi, H. L.(2002), *Um curso de cálculo*, Editora LTC, vol. 4, Rio de Janeiro, 2002.
- [10] Nise, N. S. (2002), *Engenharia de Sistemas de Controle*, Editora LTC, 3 ed.,Rio de Janeiro, 2002.

[11] Ogata, K. (2010), *Engenharia de Controle Moderno*, Editora Pearson Prentice, 5 ed., São Paulo, 2010.

[12] Löfberg, J. (2004), *YALMIP: A Toolbox for Modeling and Optimization in MATLAB*, Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

A Álgebra Linear

Este capítulo fornece uma revisão dos conceitos fundamentais de álgebra linear que embasam o problema em estudo.

A.1 Autovalores e Autovetores de uma Matriz

Definição A.1.1: Seja A uma matriz quadrada de ordem n , $\mathbf{x} \neq \mathbf{0}$ é um vetor em C^n , e λ é um escalar em C . Então \mathbf{x} é um autovetor de A com autovalor λ se

$$A\mathbf{x} = \lambda\mathbf{x}. \quad (67)$$

Exemplo A.1.1: Considere a matriz:

$$\mathbf{A} = \begin{bmatrix} 204 & 98 & 26 & -10 \\ -280 & -134 & 36 & 14 \\ 716 & 348 & -90 & -36 \\ -472 & -232 & 60 & 28 \end{bmatrix} \quad (68)$$

e os vetores

$$\mathbf{x} = \begin{bmatrix} 1 \\ -1 \\ 2 \\ 5 \end{bmatrix} \quad (69)$$

$$\mathbf{y} = \begin{bmatrix} -3 \\ 4 \\ -10 \\ 4 \end{bmatrix} \quad (70)$$

$$\mathbf{z} = \begin{bmatrix} -3 \\ 7 \\ 0 \\ 8 \end{bmatrix} \quad (71)$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ -1 \\ 4 \\ 0 \end{bmatrix} \quad (72)$$

Então

$$\mathbf{Ax} = \begin{bmatrix} 204 & 98 & -26 & -10 \\ -280 & -134 & 36 & 14 \\ 716 & 348 & -90 & -36 \\ -472 & -232 & 60 & 28 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \\ 8 \\ 20 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ -1 \\ 2 \\ 5 \end{bmatrix} = 4\mathbf{x} \quad (73)$$

e dessa forma \mathbf{x} é um autovetor de \mathbf{A} com autovalor $\lambda = 4$. Tem-se também

$$\mathbf{Ay} = \begin{bmatrix} 204 & 98 & -26 & -10 \\ -280 & -134 & 36 & 14 \\ 716 & 348 & -90 & -36 \\ -472 & -232 & 60 & 28 \end{bmatrix} \begin{bmatrix} -3 \\ 4 \\ -10 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0 \begin{bmatrix} -3 \\ 4 \\ -10 \\ 4 \end{bmatrix} = 0\mathbf{y} \quad (74)$$

e dessa forma \mathbf{y} é um autovetor de \mathbf{A} com autovalor $\lambda = 0$. Tem-se também

$$\mathbf{Az} = \begin{bmatrix} 204 & 98 & -26 & -10 \\ -280 & -134 & 36 & 14 \\ 716 & 348 & -90 & -36 \\ -472 & -232 & 60 & 28 \end{bmatrix} \begin{bmatrix} -3 \\ 7 \\ 0 \\ 8 \end{bmatrix} = \begin{bmatrix} -6 \\ 14 \\ 0 \\ 16 \end{bmatrix} = 2 \begin{bmatrix} -3 \\ 7 \\ 0 \\ 8 \end{bmatrix} = 2\mathbf{z} \quad (75)$$

e dessa forma \mathbf{z} é um autovetor de \mathbf{A} com autovalor $\lambda = 2$. Tem-se também

$$\mathbf{Aw} = \begin{bmatrix} 204 & 98 & -26 & -10 \\ -280 & -134 & 36 & 14 \\ 716 & 348 & -90 & -36 \\ -472 & -232 & 60 & 28 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 8 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ -1 \\ 4 \\ 0 \end{bmatrix} = 2\mathbf{w} \quad (76)$$

e dessa forma \mathbf{w} é um autovetor de \mathbf{A} com autovalor $\lambda = 2$.

Neste exemplo, foram mostrados quatro autovalores de \mathbf{A} . Entretanto, qualquer escalar não-nulo múltiplo de um autovetor é também um autovetor. No exemplo acima, definindo $\mathbf{u} = 30\mathbf{x}$, tem-se

$$\mathbf{Au} = \mathbf{A}(30\mathbf{x}) = 30\mathbf{Ax} = 30(4\mathbf{x}) = 4(30\mathbf{x}) = 4\mathbf{u}, \quad (77)$$

e assim, \mathbf{u} também é um autovetor de \mathbf{A} para o mesmo autovalor, $\lambda = 4$.

Os vetores \mathbf{z} e \mathbf{w} são ambos autovetores de \mathbf{A} para o mesmo autovalor $\lambda = 2$, no entanto \mathbf{z} e \mathbf{w} não são múltiplos escalares um do outro. Quando são somados para formar $\mathbf{v} = \mathbf{z} + \mathbf{w}$, e multiplicados por \mathbf{A} , tem-se

$$\mathbf{A}\mathbf{v} = \mathbf{A}(\mathbf{z} + \mathbf{w}) = \mathbf{A}\mathbf{z} + \mathbf{A}\mathbf{w} = 2\mathbf{z} + 2\mathbf{w} = 2(\mathbf{z} + \mathbf{w}) = 2\mathbf{v}, \quad (78)$$

e \mathbf{v} também é um autovetor de \mathbf{A} para o autovalor $\lambda = 2$.

O vetor \mathbf{y} é um autovetor de \mathbf{A} para o autovalor $\lambda = 0$, assim $\mathbf{A}\mathbf{y} = 0\mathbf{y} = 0$. Mas isso também significa que \mathbf{y} pertence ao núcleo de \mathbf{A} ($\mathbf{y} \in N(\mathbf{A})$).

A.2 Polinômios e Matrizes

Se uma matriz é quadrada, todas as operações de um polinômio preservam a ordem da matriz. Então, é natural relacionar um polinômio com uma matriz, substituindo a variável do polinômio por uma matriz.

Exemplo A.2.1: Seja

$$p(x) = 14 - 19x - 3x^2 - 7x^3 + x^4, \quad (79)$$

$$\mathbf{D} = \begin{bmatrix} -1 & 3 & 2 \\ 1 & 0 & -2 \\ -3 & 1 & 1 \end{bmatrix} \quad (80)$$

Então será calculado $p(D)$. Primeiro, calcula-se as potências necessárias de D . Note que D^0 é definido como a matriz identidade I_3 .

$$\mathbf{D}^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (81)$$

$$\mathbf{D}^1 = \mathbf{D} = \begin{bmatrix} -1 & 3 & 2 \\ 1 & 0 & -2 \\ -3 & 1 & 1 \end{bmatrix} \quad (82)$$

$$\mathbf{D}^2 = \mathbf{D}\mathbf{D}^1 = \begin{bmatrix} -1 & 3 & 2 \\ 1 & 0 & -2 \\ -3 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & 2 \\ 1 & 0 & -2 \\ -3 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -1 & -6 \\ 5 & 1 & 0 \\ 1 & -8 & -7 \end{bmatrix} \quad (83)$$

$$\mathbf{D}^3 = \mathbf{D}\mathbf{D}^2 = \begin{bmatrix} -1 & 3 & 2 \\ 1 & 0 & -2 \\ -3 & 1 & 1 \end{bmatrix} \begin{bmatrix} -2 & -1 & -6 \\ 5 & 1 & 0 \\ 1 & -8 & -7 \end{bmatrix} = \begin{bmatrix} 19 & -12 & -8 \\ -4 & 15 & 8 \\ 12 & -4 & 11 \end{bmatrix} \quad (84)$$

$$\mathbf{D}^4 = \mathbf{D}\mathbf{D}^3 = \begin{bmatrix} -1 & 3 & 2 \\ 1 & 0 & -2 \\ -3 & 1 & 1 \end{bmatrix} \begin{bmatrix} 19 & -12 & -8 \\ -4 & 15 & 8 \\ 12 & -4 & 11 \end{bmatrix} = \begin{bmatrix} -7 & 49 & 54 \\ -5 & -4 & -30 \\ -49 & 47 & 43 \end{bmatrix} \quad (85)$$

Então,

$$p(\mathbf{D}) = 14 + 19\mathbf{D} - 3\mathbf{D}^2 - 7\mathbf{D}^3 + \mathbf{D}^4, \quad (86)$$

$$p(\mathbf{D}) = 14 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 19 \begin{bmatrix} -1 & 3 & 2 \\ 1 & 0 & -2 \\ -3 & 1 & 1 \end{bmatrix} - 3 \begin{bmatrix} -2 & -1 & -6 \\ 5 & 1 & 0 \\ 1 & -8 & -7 \end{bmatrix} - \quad (87)$$

$$7 \begin{bmatrix} 19 & -12 & -8 \\ -4 & 15 & 8 \\ 12 & -4 & 11 \end{bmatrix} + \begin{bmatrix} -7 & 49 & 54 \\ -5 & -4 & -30 \\ -49 & 47 & 43 \end{bmatrix} \quad (88)$$

$$p(\mathbf{D}) = \begin{bmatrix} -139 & 193 & 166 \\ 27 & -98 & -124 \\ -193 & 118 & 20 \end{bmatrix} \quad (89)$$

Note que $p(x)$ pode ser fatorado da seguinte forma

$$p(x) = 14 + 19x - 3x^2 - 7x^3 + x^4 = (x - 2)(x - 7)(x + 1)^2 \quad (90)$$

Como \mathbf{D} comuta-se consigo mesmo, pode-se utilizar a distributividade da multiplicação de matrizes e adição de matrizes, assim calcula-se $p(\mathbf{D})$ usando a forma de $p(x)$,

$$p(\mathbf{D}) = 14 + 19\mathbf{D} - 3\mathbf{D}^2 - 7\mathbf{D}^3 + \mathbf{D}^4 = (\mathbf{D} - 2I_3)(\mathbf{D} - 7I_3)(\mathbf{D} + I_3)^2 \quad (91)$$

$$= \begin{bmatrix} -3 & 3 & 2 \\ 1 & -2 & -2 \\ -3 & 1 & 1 \end{bmatrix} \begin{bmatrix} -8 & 3 & 2 \\ 1 & -7 & -2 \\ -3 & 1 & -6 \end{bmatrix} \begin{bmatrix} 0 & 3 & 2 \\ 1 & 1 & -2 \\ -3 & 1 & 2 \end{bmatrix}^2 = \begin{bmatrix} -139 & 193 & 166 \\ 27 & -98 & -124 \\ -193 & 118 & 20 \end{bmatrix} \quad (92)$$

Este exemplo mostrou que é comum calcular um polinômio com uma matriz e que a forma fatorada do polinômio é tão boa quanto a forma expandida.

A.3 Existência de Autovetores e Autovalores

Neste tópico será demonstrado que toda matriz tem pelo menos um autovalor (e um autovetor relacionado a este autovalor).

Teorema 2.3.1: Toda matriz tem um autovalor

Seja \mathbf{A} uma matriz quadrada. Então \mathbf{A} tem pelo menos um autovalor.

Prova: Suponha que \mathbf{A} tenha ordem n , tome \mathbf{x} como qualquer vetor não-nulo de C^n . Considere o conjunto

$$S = \{\mathbf{x}, \mathbf{A}\mathbf{x}, \mathbf{A}^2\mathbf{x}, \mathbf{A}^3\mathbf{x}, \dots, \mathbf{A}^n\mathbf{x}\} \quad (93)$$

Este é um conjunto de $n+1$ vetores de C^n , e dessa forma S é um conjunto linearmente dependente. Seja $a_0, a_1, a_2, \dots, a_n$ um conjunto de $n+1$ elementos de C , não todos nulos, que fornece uma relação de dependência linear em S . Assim sendo,

$$a_0\mathbf{x} + a_1\mathbf{A}\mathbf{x} + a_2\mathbf{A}^2\mathbf{x} + a_3\mathbf{A}^3\mathbf{x} + \dots + a_n\mathbf{A}^n\mathbf{x} = 0 \quad (94)$$

Um dentre os escalares a_i deve ser não-nulo. Assim, suponha que $a_0 \neq 0$, e $a_1 = a_2 = a_3 = \dots = a_n = 0$. Então $a_0\mathbf{x} = 0$, e assim $a_0 = 0$ ou $\mathbf{x} = 0$, o que é uma contradição. Desta forma, $a_i \neq 0$ para algum $i \geq 1$. Seja m o maior inteiro tal que $a_m \neq 0$. Como visto anteriormente, $m \geq 1$. Pode-se assumir que $a_m = 1$, caso não seja, basta substituir cada a_i por a_i/a_m para obter os escalares que fornecem uma relação de dependência linear em S .

Definindo o polinômio

$$p(\mathbf{x}) = a_0 + a_1\mathbf{x} + a_2\mathbf{x}^2 + a_3\mathbf{x}^3 + \dots + a_m\mathbf{x}^m, \quad (95)$$

sabe-se que $p(\mathbf{x})$ pode ser fatorado em fatores da forma $(\mathbf{x} - b_i)$, em que $b_i \in C$. Assim,

existem escalares $b_1, b_2, b_3, \dots, b_m$, tais que

$$p(\mathbf{x}) = (\mathbf{x} - b_m)(\mathbf{x} - b_{m-1}) \cdots (\mathbf{x} - b_3)(\mathbf{x} - b_2)(\mathbf{x} - b_1) \quad (96)$$

Assim sendo,

$$0 = a_0\mathbf{x} + a_1\mathbf{A}\mathbf{x} + a_2\mathbf{A}^2\mathbf{x} + a_3\mathbf{A}^3\mathbf{x} + \cdots + a_n\mathbf{A}^n\mathbf{x} \quad (97)$$

$$= a_0\mathbf{x} + a_1\mathbf{A}\mathbf{x} + a_2\mathbf{A}^2\mathbf{x} + a_3\mathbf{A}^3\mathbf{x} + \cdots + a_m\mathbf{A}^m\mathbf{x} \quad (98)$$

$$= (a_0I_n + a_1\mathbf{A} + a_2\mathbf{A}^2 + a_3\mathbf{A}^3 + \cdots + a_m\mathbf{A}^m)\mathbf{x} = p(\mathbf{A})\mathbf{x} \quad (99)$$

$$= (\mathbf{A} - b_mI_n)(\mathbf{A} - b_{m-1}I_n) \cdots (\mathbf{A} - b_3I_n)(\mathbf{A} - b_2I_n)(\mathbf{A} - b_1I_n)\mathbf{x} \quad (100)$$

Seja k o menor inteiro tal que

$$(\mathbf{A} - b_kI_n)(\mathbf{A} - b_{k-1}I_n) \cdots (\mathbf{A} - b_3I_n)(\mathbf{A} - b_2I_n)(\mathbf{A} - b_1I_n)\mathbf{x} = 0 \quad (101)$$

Da equação anterior, sabe-se que $k \leq m$. Definindo o vetor \mathbf{z} como

$$\mathbf{z} = (\mathbf{A} - b_{k-1}I_n) \cdots (\mathbf{A} - b_3I_n)(\mathbf{A} - b_2I_n)(\mathbf{A} - b_1I_n)\mathbf{x} \quad (102)$$

note que por definição de k , o vetor \mathbf{z} deve ser não-nulo. No caso em que $k = 1$, \mathbf{z} é definido por $\mathbf{z} = \mathbf{x}$, sendo \mathbf{z} ainda não-nulo. Agora tem-se

$$(\mathbf{A} - b_kI_n)\mathbf{z} = (\mathbf{A} - b_kI_n)(\mathbf{A} - b_{k-1}I_n) \cdots (\mathbf{A} - b_3I_n)(\mathbf{A} - b_2I_n)(\mathbf{A} - b_1I_n)\mathbf{x} = 0 \quad (103)$$

o que permite escrever

$$\mathbf{A}\mathbf{z} = (\mathbf{A} + 0)\mathbf{z} = (\mathbf{A} - b_kI_n + b_kI_n)\mathbf{z} = (\mathbf{A} - b_kI_n)\mathbf{z} + b_kI_n\mathbf{z} = 0 + b_kI_n\mathbf{z} = b_kI_n\mathbf{z} = b_k\mathbf{z} \quad (104)$$

Como $\mathbf{z} \neq 0$, esta equação diz que \mathbf{z} é um autovetor de \mathbf{A} para o autovalor $\lambda = b_k$. Dessa forma, demonstrou-se que toda matriz quadrada \mathbf{A} tem pelo menos um autovalor.

A.4 Polinômio Característico

Definição A.4.1: Seja \mathbf{A} uma matriz quadrada de ordem n . Então, o polinômio característico de \mathbf{A} é o polinômio $p_A(x)$ definido por

$$p_A(x) = \det(\mathbf{A} - \mathbf{x}I_n). \quad (105)$$

Exemplo A.4.1: Polinômio característico de uma matriz de ordem 3. Considere a matriz

$$\mathbf{F} = \begin{bmatrix} -13 & -8 & -4 \\ 12 & 7 & 4 \\ 24 & 16 & 7 \end{bmatrix} \quad (106)$$

Então

$$p_F(\mathbf{x}) = \det(\mathbf{F} - \mathbf{x}I_3) = \begin{vmatrix} -13 - \mathbf{x} & -8 & -4 \\ 12 & 7 - \mathbf{x} & 4 \\ 24 & 16 & 7 - \mathbf{x} \end{vmatrix} \quad (107)$$

$$= (-13 - \mathbf{x}) \begin{vmatrix} 7 - \mathbf{x} & 4 \\ 16 & 7 - \mathbf{x} \end{vmatrix} + (-8)(-1) \begin{vmatrix} 12 & 4 \\ 24 & 7 - \mathbf{x} \end{vmatrix} + (-4) \begin{vmatrix} 12 & 7 - \mathbf{x} \\ 24 & 16 \end{vmatrix} \quad (108)$$

$$= (-13 - \mathbf{x})((7 - \mathbf{x})(7 - \mathbf{x}) - 64) + 8(12(7 - \mathbf{x}) - 96) + (-4)(12(16) - 24(7 - \mathbf{x})) \quad (109)$$

$$= 3 + 5\mathbf{x} + \mathbf{x}^2 - \mathbf{x}^3 = -(\mathbf{x} - 3)(\mathbf{x} + 1)^2 \quad (110)$$

Teorema A.4.1: Autovalores de uma matriz são raízes do polinômio característico.

Seja \mathbf{A} uma matriz quadrada. Então λ é um autovalor de \mathbf{A} se, e somente se, $p(\mathbf{A} - \lambda I_n) = 0$.

Prova: Suponha que \mathbf{A} tenha ordem n .

λ é autovalor de $\mathbf{A} \iff$ existe $\mathbf{x} \neq 0$ tal que $\mathbf{Ax} = \lambda\mathbf{x} \iff$ existe $\mathbf{x} \neq 0$ tal que $\mathbf{Ax} - \lambda\mathbf{x} = 0 \iff$ existe $\mathbf{x} \neq 0$ tal que $\mathbf{Ax} - \lambda I_n \mathbf{x} = 0 \iff$ existe $\mathbf{x} \neq 0$ tal que $(\mathbf{A} - \lambda I_n)\mathbf{x} = 0 \iff (\mathbf{A} - \lambda I_n)$ é singular $\iff \det(\mathbf{A} - \lambda I_n) = 0 \iff p_A(\mathbf{x}) = 0$

Exemplo A.4.2: Autovalores de uma matriz de ordem 3.

Como visto no exemplo A.4.1, o polinômio característico da matriz

$$\mathbf{F} = \begin{bmatrix} -13 & -8 & -4 \\ 12 & 7 & 4 \\ 24 & 16 & 7 \end{bmatrix} \quad (111)$$

,
é dado por $p_F(\mathbf{x}) = -(\mathbf{x} - 3)(\mathbf{x} + 1)^2$. Do teorema 2.4.1, pode-se encontrar os autovalores da matriz \mathbf{F} facilmente, sendo eles as raízes do polinômio característico, $\mathbf{x} = 3$ e $\mathbf{x} = -1$

A.5 Propriedades de Autovalores e Autovetores

Teorema A.5.1: Autovetores de autovalores distintos são linearmente independentes.

Seja \mathbf{A} uma matriz quadrada de ordem n e $S = \{x_1, x_2, x_3, \dots, x_p\}$ um conjunto de autovetores associados aos autovalores $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_p$ tal que $\lambda_i \neq \lambda_j$ para $i \neq j$. Então S é um conjunto linearmente independente.

Teorema A.5.2: Matrizes singulares possuem autovalores nulos.

Seja \mathbf{A} uma matriz quadrada de ordem n . Então \mathbf{A} é singular se, e somente se, $\lambda = 0$ é um autovalor de \mathbf{A} .

Equivalência da matriz não singular

Seja \mathbf{A} uma matriz quadrada de ordem n . As seguintes afirmações são equivalentes:

1. \mathbf{A} é não singular
2. O espaço nulo de \mathbf{A} contém apenas o vetor nulo, $N(\mathbf{A}) = \mathbf{0}$;
3. O sistema linear $\mathbf{Ax} = \mathbf{b}$ tem solução única para qualquer escolha possível do vetor \mathbf{b} ;
4. As colunas de \mathbf{A} formam um conjunto linear independente;
5. \mathbf{A} é invertível;
6. As colunas de \mathbf{A} formam uma base de C^n ;
7. O posto de \mathbf{A} é n , $rank(\mathbf{A}) = n$;
8. O determinante de \mathbf{A} é não-nulo, $det(\mathbf{A}) \neq 0$;
9. $\lambda = 0$ não é um autovalor de \mathbf{A} .

Teorema A.5.3: Seja \mathbf{A} uma matriz quadrada e λ um autovalor de \mathbf{A} . Então $\alpha\lambda$ é um autovalor de $\alpha\mathbf{A}$.

Prova: Seja $\mathbf{x} \neq 0$ um autovalor de \mathbf{A} para λ . Então

$$(\alpha\mathbf{A})\mathbf{x} = \alpha(\mathbf{A}\mathbf{x}) = \alpha(\lambda\mathbf{x}) = (\alpha\lambda)\mathbf{x} \quad (112)$$

E dessa forma, $\mathbf{x} \neq 0$ é um autovetor de $\alpha\mathbf{A}$ para o autovalor $\alpha\lambda$.

Teorema A.5.4: Autovalores de Potências de Matrizes.

Seja \mathbf{A} uma matriz quadrada, λ um autovalor de \mathbf{A} e $s \geq 0$ um inteiro. Então λ^s é um autovalor de \mathbf{A}^s .

Teorema A.5.5: Autovalores de uma Matriz Inversa.

Seja \mathbf{A} uma matriz quadrada não singular e λ um autovalor de \mathbf{A} . Então $1/\lambda$ é um autovalor da matriz \mathbf{A}^{-1} .

Teorema A.5.6: Autovalores de uma Matriz Transposta.

Seja \mathbf{A} uma matriz quadrada e λ um autovalor de \mathbf{A} . Então λ é um autovalor da matriz \mathbf{A}^t .

Teorema A.5.7: Autovalores de matrizes reais vêm em pares conjugados.

Seja \mathbf{A} uma matriz quadrada em valores reais e \mathbf{x} um autovetor de \mathbf{A} para o autovalor λ . Então $\bar{\mathbf{x}}$ é um autovetor de \mathbf{A} para o autovalor $\bar{\lambda}$.

Prova:

$$\mathbf{A}\bar{\mathbf{x}} = \bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{A}}\mathbf{x} = \bar{\lambda}\mathbf{x} = \bar{\lambda}\bar{\mathbf{x}} \quad (113)$$

Assim, $\bar{\mathbf{x}}$ é um autovetor de \mathbf{A} para o autovalor $\bar{\lambda}$.

Teorema A.5.8: Número máximo de autovalores de uma matriz.

Seja \mathbf{A} uma matriz quadrada de ordem n . Então \mathbf{A} não pode ter mais do que n autovalores distintos.

B Equações Diferenciais

Este capítulo fornecerá conceitos fundamentais de equações diferenciais que serão úteis para o problema em estudo.

B.1 Equações Diferenciais

Uma equação diferencial é uma relação entre uma 'função incógnita' e suas derivadas ou diferenciais.

Exemplo B.1.1:

1. $\dot{y}(t) = f(t)$, em que \dot{y} denota $\frac{dy}{dt}$;
2. $\ddot{y}(t) - 2y(t) = 0$;
3. $y^{(3)}(t) + (\cos(t))\ddot{y}(t) - ty(t) = 0$;
4. $\frac{\partial v(t,x)^2}{\partial t^2} + \frac{\partial v(t,x)^2}{\partial x^2} = 0$;
5. $M(x, y)dx + N(x, y)dy = 0$

Definição B.1.1: Uma equação diferencial ordinária (E. D. O.) é uma equação diferencial na qual a função incógnita depende apenas de uma variável.

Os itens 1, 2, 3 e 5 do Exemplo B.1.1 são exemplos de E.D.O.s.

Definição B.1.2: Uma equação diferencial parcial (E. D. P.) é uma equação diferencial na qual a função incógnita depende de mais de uma variável.

O item 4 do Exemplo B.1.1 é um exemplo de E.D.P.

Definição B.1.3: A ordem de uma equação diferencial é a ordem da mais alta derivada da função incógnita.

Assim, a equação do item 1 do Exemplo B.1.1 é de primeira ordem, o item 2 é de segunda ordem e o item 3 é de terceira ordem.

Definição B.1.4: Seja $f : [a, b] \rightarrow R$ uma função contínua. O Teorema Fundamental do Cálculo implica que a função

$$F(t) = \int_a^t f(s)ds, a \leq t \leq b \quad (114)$$

, e ainda $F(a) = 0$. Neste caso, $F(t)$ é uma solução do problema de valor inicial (P.V.I.)

$$\begin{cases} \dot{y}(t) = f(t) \\ y(a) = 0 \end{cases} \quad (115)$$

Exemplo B.1.2:

1. $\begin{cases} \dot{y} = y \\ y(0) = 1 \end{cases}$ é um P.V.I. cuja solução é $y(t) = e^t$.

Prova:

$$\dot{y} = y \quad (116)$$

$$\frac{dy}{dt} = y \quad (117)$$

$$\frac{d(e^t)}{dt} = e^t \quad (118)$$

$$e^t = e^t \quad (119)$$

$$y(0) = e^0 = 1 \quad (120)$$

2. $\begin{cases} \dot{y} = 2t(y + 1) \\ y(0) = 0 \end{cases}$ é um P.V.I. cuja solução é $y(t) = e^{t^2} - 1$.

Prova:

$$\dot{y} = 2t(y + 1) \quad (121)$$

$$\frac{dy}{dt} = 2t(y + 1) \quad (122)$$

$$\frac{d(e^{t^2} - 1)}{dt} = 2t[(e^{t^2} - 1) + 1] \quad (123)$$

$$2te^{t^2} = 2te^{t^2} \quad (124)$$

$$y(0) = e^{0^2} - 1 = 1 - 1 = 0 \quad (125)$$

3. $\begin{cases} \ddot{y} = -y \\ y(0) = 0 \end{cases}$ é um P.V.I. cuja solução é $y(t) = \text{sen}(t)$.

Prova:

$$\ddot{y} = -y \quad (126)$$

$$\frac{d^2y}{dt^2} = -y \quad (127)$$

$$\frac{d^2(\text{sen}(t))}{dt^2} = -\text{sen}(t) \quad (128)$$

$$\frac{d(\text{cos}(t))}{dt} = -\text{sen}(t) \quad (129)$$

$$-\text{sen}(t) = -\text{sen}(t) \quad (130)$$

$$y(0) = \text{sen}(0) = 0 \quad (131)$$

B.2 Equação Diferencial de Primeira Ordem

Definição B.2.1: Uma equação diferencial linear de primeira ordem é uma equação da forma:

$$\dot{y} + a(t)y = b(t) \quad (132)$$

em que $a(t)$ e $b(t)$ são funções contínuas num intervalo I .

Exemplo B.2.1:

1. $\dot{y} = 3y$ é uma equação diferencial de primeira ordem pois $g(t, y) = 3y$ é linear em y ;
2. $\dot{y} = t^2y + \cos(t)$ é uma E.D.O. linear de primeira ordem, pois $g(t, y) = t^2y$ é linear em y ;
3. $\dot{y} = ty^2 + \cos(t)$ não é E.D.O. linear de primeira ordem, pois $g(t, y) = ty^2$ não é linear em y ;
4. $\dot{y} = t\cos(y) + t$ não é E.D.O. linear de primeira ordem, pois $g(t, y) = t\cos(y)$ não é linear em y .

Definição B.2.2: Uma equação diferencial linear é homogênea quando $b(t) = 0$. Assim, uma equação diferencial linear homogênea é definida por:

$$\dot{y} + a(t)y = 0 \quad (133)$$

em que $a(t)$ é uma função contínua num intervalo I .

A equação pode ser resolvida facilmente. Dividindo ambos os membros da equação por y , obtém-se

$$\frac{\dot{y}}{y} = -a(t) \quad (134)$$

lembrando-se que $\frac{\dot{y}}{y} = \frac{d(\ln|y(t)|)}{dt}$ tem-se que a equação pode ser reescrita como:

$$\frac{d}{dt}(\ln |y(t)|) = -a(t) \quad (135)$$

,
integrando-se ambos os membros, obtêm-se:

$$\ln |y(t)| = - \int a(t)dt + k_1 \quad (136)$$

,
em que k_1 é uma constante de integração. Resolvendo encontra-se

$$|y(t)| = \exp(- \int a(t)dt + k_1) \quad (137)$$

.
Logo,

$$y(t) = k \exp(- \int a(t)dt). \quad (138)$$

Exemplo B.2.2: Determinar a solução geral da equação:

$$\dot{y} + 2ty = 0 \quad (139)$$

Solução: Neste caso $a(t) = 2t$. Assim,

$$y(t) = k e^{- \int a(t)dt} = k e^{- \int 2t dt} = k e^{-t^2} \quad (140)$$

.
Portanto,

$$y(t) = k e^{-t^2}, \quad (141)$$

é a solução geral.

Definição B.2.3: Uma equação diferencial linear é não homogênea quando $b(t) \neq 0$. Assim, uma equação diferencial linear não homogênea é definida por:

$$\dot{y} + a(t)y = b(t) \quad (142)$$

,
em que $a(t)$ e $b(t)$ são funções contínuas num intervalo I.

B.3 Equação Diferencial Linear de Segunda Ordem

Definição B.3.1: Uma equação diferencial linear de segunda ordem é uma equação da forma

$$\ddot{y} + a(t)\dot{y} + b(t)y = g(t) \quad (143)$$

em que $a(t)$, $b(t)$ e $g(t)$ são funções contínuas num intervalo I .

Quando $g(t) \neq 0$ a equação é chamada linear não homogênea.

Exemplo B.3.1: São exemplos de equações diferenciais lineares de segunda ordem:

1. $\ddot{y} - 2\dot{y} + y = 0$;
2. $2t^2\ddot{y} + 3t\dot{y} - y = 0$;
3. $\ddot{y} + \dot{y} - 6y = \text{sen}(t) + te^{2t}$.

Teorema B.3.1: Se as funções $a(t)$, $b(t)$ e $g(t)$ forem contínuas num intervalo I , então dados $t_0 \in I$ e $y_0, z_0 \in R$, o P.V.I.

$$\begin{cases} \ddot{y} + a(t)\dot{y} + b(t)y = g(t) \\ y(t_0) = y_0 \\ \dot{y}(t_0) = z_0 \end{cases} \quad (144)$$

possui uma única solução $y = y(t)$, a qual está definida para todo $t \in I$.

Exemplo B.3.2: A função $y(t) = 1 - e^{-t}$ é a solução única do P.V.I.

$$\begin{cases} \ddot{y} + \dot{y} = 0, \\ y(0) = 0, \\ \dot{y}(0) = 1. \end{cases} \quad (145)$$

Prova:

$$\ddot{y} + \dot{y} = 0 \quad (146)$$

$$\frac{d^2y}{dt^2} + \frac{dy}{dt} = 0 \quad (147)$$

$$\frac{d^2}{dt^2}(1 - e^{-t}) + \frac{d}{dt}(1 - e^{-t}) \quad (148)$$

$$\frac{d}{dt}(e^{-t}) + (e^{-t}) = 0 \quad (149)$$

$$-e^{-t} + e^{-t} = 0 \quad (150)$$

$$y(0) = 1 - e^{-0} = 1 - 1 = 0 \quad (151)$$

$$\dot{y}(0) = e^{-0} = 1 \quad (152)$$

Teorema B.3.2: Suponha-se que $a(t)$ e $b(t)$ sejam funções contínuas num intervalo I . Então existem duas soluções $y_1(t)$ e $y_2(t)$ da equação

$$\ddot{y} + a(t)\dot{y} + b(t)y = 0 \quad (153)$$

para todo $t \in I$. Além disso, a solução geral desta equação é dada por $k_1y_1(t) + k_2y_2(t)$, em que k_1 e k_2 são constantes arbitrárias.

Equações diferenciais lineares de segunda ordem apresentam diversos métodos de resolução. Será apresentado um dos métodos, sendo que os demais podem ser encontrados em, por exemplo, (Guidorizzi, 2002). Em seguida é apresentado o Teorema de Laplace, muito útil na resolução de equações diferenciais de qualquer ordem.

Equações Homogêneas com Coeficientes Constantes

Equações homogêneas com coeficientes constantes são da forma:

$$a\ddot{y} + b\dot{y} + cy = 0 \quad (154)$$

em que a , b e c são constantes reais com $a \neq 0$.

Pelo Teorema B.3.2 há duas soluções $y_1(t)$ e $y_2(t)$ linearmente independentes e todas as demais serão combinações lineares destas.

Se $y = \varphi(t)$ é uma solução da equação acima, então a soma dos termos $a\ddot{\varphi}(t)$, $b\dot{\varphi}(t)$ e $c\varphi(t)$ deverá ser nula para todo t . Assim sendo, a função $y(t) = e^{\lambda t}$, onde λ é constante, apresenta a propriedade de que tanto $\dot{y}(t)$ como $\ddot{y}(t)$ são múltiplos de $y(t)$. Portanto, substituindo $y(t) = e^{\lambda t}$ na equação acima obtém-se

$$a(e^{\lambda t})'' + b(e^{\lambda t})' + c(e^{\lambda t}) = 0 \quad (155)$$

$$e^{\lambda t}(a\lambda^2 + b\lambda + c) = 0 \quad (156)$$

Implicando em

$$a\lambda^2 + b\lambda + c = 0 \quad (157)$$

Portanto, $y(t) = e^{\lambda t}$ é uma solução se, e somente se, λ é a raiz da equação acima. A equação acima é chamada Equação Característica possuindo as raízes

$$\lambda_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (158)$$

$$\lambda_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (159)$$

Agora serão analisadas as três possibilidades do discriminante $b^2 - 4ac$.

1. $b^2 - 4ac > 0$: Raízes reais distintas

Assim, $e^{\lambda_1 t}$ e $e^{\lambda_2 t}$ são soluções para todo $t \in R$ e qualquer solução é da forma

$$y(t) = k_1 e^{\lambda_1 t} + k_2 e^{\lambda_2 t} \quad (160)$$

2. $b^2 - 4ac = 0$: Raízes reais iguais

Nesta situação, $\lambda_1 = \lambda_2 = -\frac{b}{2a}$ e assim uma solução é $y_1 = e^{-\frac{b}{2a}t}$. Agora deve-se encontrar uma outra solução não múltipla desta última. Para tanto será procurado uma $v(t)$ não constante tal que $y_2(t) = v(t)e^{-\frac{b}{2a}t}$ seja solução. Substituindo y_2 na equação original tem-se

$$a\ddot{y} + b\dot{y} + cy = 0 \quad (161)$$

$$a \frac{d^2}{dt^2}(v(t)e^{-\frac{b}{2a}t}) + b \frac{d}{dt}(v(t)e^{-\frac{b}{2a}t}) + c(v(t)e^{-\frac{b}{2a}t}) = 0 \quad (162)$$

$$a(\ddot{v} - \frac{\dot{v}b}{2a} - \frac{\dot{v}b}{2a} + \frac{vb^2}{4a})e^{-\frac{b}{2a}t} + b(\dot{v} - \frac{vb}{2a})e^{-\frac{b}{2a}t} + cv e^{-\frac{b}{2a}t} = 0 \quad (163)$$

$$e^{-\frac{b}{2a}t}[a\ddot{v} + (\frac{b^2}{4a} - \frac{b^2}{2a} + c)v] = 0 \quad (164)$$

como $e^{-\frac{b}{2a}t} \neq 0$ para todo t e $b^2 - 4ac = 0$, tem-se que

$$\ddot{v} = 0 \Rightarrow v(t) = \alpha t + \beta, \alpha, \beta \in R \quad (165)$$

Pode-se tomar $\alpha = 1$ e $\beta = 0$, pois deseja-se apenas uma solução. Assim, $v(t) = t$. Portanto outra solução é

$$y_2 = te^{-\frac{b}{2a}t} \quad (166)$$

Exemplo B.3.3: Encontrar a resposta do P.V.I.

$$\begin{cases} \ddot{y} + 4\dot{y} + 4y = 0, \\ y(0) = 1, \\ \dot{y}(0) = 2. \end{cases} \quad (167)$$

Solução:

$$y = e^{\lambda t} \Rightarrow \lambda^2 + 4\lambda + 4 = 0 \Rightarrow \lambda_1 = \lambda_2 = -2. \quad (168)$$

Portanto, a solução geral é

$$y(t) = (k_1 + k_2t)e^{-2t}. \quad (169)$$

Como $y(0) = 1$, tem-se que $k_1 = 1$ e $\dot{y}(t) = (k_2 - 2k_2t - 2k_1)e^{-2t}$ e $\dot{y}(0) = 2$. Assim, $k_2 = 4$. Portanto a solução geral do P.V.I. é

$$y(t) = e^{-2t} + 4te^{-2t} \quad (170)$$

3. $b^2 - 4ac < 0$: Raízes Complexas

Neste caso,

$$\lambda_1 = \frac{-b}{2a} + \frac{i\sqrt{4ac - b^2}}{2a} \quad (171)$$

$$\lambda_2 = \frac{-b}{2a} - \frac{i\sqrt{4ac - b^2}}{2a} \quad (172)$$

Para se encontrar as soluções $e^{\lambda_1 t}$ e $e^{\lambda_2 t}$ deve-se definir $e^{\lambda t}$ para λ complexo e encontrar soluções reais.

Definição 3.3.2: Se $F(t) = u(t) + iv(t)$, define-se $\dot{F}(t) = \dot{u}(t) + i\dot{v}(t)$.

Proposição 3.3.1: Se $y(t) = u(t) + iv(t)$ é uma solução a valores complexos, então $u(t)$ e $v(t)$ são soluções reais.

Demonstração: Note que

$$a\ddot{y}(t) + b\dot{y}(t) + cy(t) = 0 \quad (173)$$

$$[a\ddot{u}(t) + b\dot{u}(t) + cu(t)] + i[a\ddot{v}(t) + b\dot{v}(t) + cv(t)] = 0 \quad (174)$$

Assim, u e v são soluções.

A partir disso, é possível definir $e^{\lambda t}$ para λ complexo. Se $\lambda = \alpha + i\beta$, tem-se

$$e^{\lambda t} = e^{\alpha t + i\beta t} = e^{\alpha t} \cdot e^{i\beta t}. \quad (175)$$

Então basta apenas definir $e^{i\beta t}$. De (Guidorizzi, 2002), tem-se para todo x real:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad (176)$$

A equação acima tem sentido mesmo para x complexo. Assim, pode-se empregar:

$$\begin{aligned} e^{i\theta} &= 1 + i\theta + \frac{(i\theta)^2}{2!} + \frac{(i\theta)^3}{3!} + \dots \quad (177) \\ &= 1 + i\theta - \frac{\theta^2}{2!} - \frac{i\theta^3}{3!} + \frac{\theta^4}{4!} + \frac{i\theta^5}{5!} - \dots \\ &= \left[1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \dots\right] + i\left[\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots\right]. \end{aligned}$$

Como

$$\cos(\theta) = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \dots \quad (178)$$

$$\text{sen}(\theta) = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots \quad (179)$$

Então, é possível definir

$$e^{i\theta} = \cos\theta + i\text{sen}\theta \quad (180)$$

e portanto,

$$e^{\lambda t} = e^{(\alpha + i\beta)t} = e^{\alpha t}(\cos(\beta t) + i\text{sen}(\beta t)) \quad (181)$$

Exemplo B.3.4: Resolver o P.V.I.

$$\begin{cases} \ddot{y} + 2\dot{y} + 5y = 0, \\ y(0) = 1, \\ \dot{y}(0) = 3. \end{cases} \quad (182)$$

Solução:

$$\lambda^2 + 2\lambda + 5 = 0 \quad (183)$$

$$\lambda_1 = -1 + 2i \quad (184)$$

$$\lambda_2 = -1 - 2i \quad (185)$$

Assim,

$$e^{\lambda_1 t} = e^{(-1+2i)t} = e^{-t} \cos(2t) + ie^{-t} \sen(2t) \quad (186)$$

é uma solução com valores complexos de $\ddot{y} + 2\dot{y} + 5y = 0$. Pela Proposição 3.3.1, tem-se

$$y_1(t) = \Re(e^{\lambda_1 t}) = e^{-t} \cos(2t) \quad (187)$$

$$y_2(t) = \Im(e^{\lambda_1 t}) = e^{-t} \sen(2t) \quad (188)$$

são soluções reais da equação. Logo, a solução geral é:

$$y(t) = e^{-t}(k_1 \cos(2t) + k_2 \sen(2t)) \quad (189)$$

Onde k_1 e k_2 são constantes reais. Como $y(0) = 1$, tem-se que $k_1 = 1$. Assim,

$$y(t) = e^{-t}(\cos(2t) + k_2 \sen(2t)) \quad (190)$$

$$\dot{y}(t) = -e^{-t}(\cos(2t) + k_2 \sen(2t)) + e^{-t}(-2\sen(2t) + 2k_2 \cos(2t)) \quad (191)$$

$$\dot{y}(0) = 3 \quad (192)$$

$$k_2 = 2 \tag{193}$$

Portanto a solução geral do P.V.I. é

$$y(t) = e^{-t}(\cos(2t) + 2\text{sen}(2t)) \tag{194}$$

Equações Não Homogêneas

Equação não homogênea é da forma

$$\ddot{y} + a(t)\dot{y} + b(t)y = g(t) \tag{195}$$

onde $a(t)$, $b(t)$, $c(t)$ e $g(t)$ são funções contínuas num intervalo I e $g(t) \neq 0$.

A resolução deste tipo de equação envolve dois processos: a solução geral desta equação para $g(t) = 0$ (equação homogênea) e uma solução particular da equação não homogênea. Uma vez que a solução geral da equação homogênea já foi discutida, será tratada agora a solução da equação não homogênea.

O método utilizado para a resolução denomina-se Método dos Coeficientes a Determinar. As possíveis formas de $g(t)$ são:

1. $P_n(t) = a_n t^n + a_{n-1} t^{n-1} + \dots + a_1 t + a_0$;
2. $e^{\alpha t} P_n(t)$;
3. $e^{\alpha t} P_n(t) \text{sen}(\beta t)$ ou $e^{\alpha t} P_n(t) \text{cos}(\beta t)$;
4. Combinações lineares das anteriores.

Caso 1: Se $g(t) = P_n(t)$, e $a_n \neq 0$, então a equação torna-se:

$$a\ddot{y} + b\dot{y} + cy = a_n t^n + a_{n-1} t^{n-1} + \dots + a_1 t + a_0 \tag{196}$$

Deve-se procurar $y_p(t)$ de maneira que a combinação $a\dot{y}_p + b\dot{y}_p + cy_p$ seja um polinômio de grau n . A escolha natural é:

$$y_p(t) = A_n t^n + A_{n-1} t^{n-1} + \dots + A_1 t + A_0 \tag{197}$$

com os coeficientes A_0, A_1, \dots, A_n a serem determinados.

Exemplo B.3.5: Determinar a solução geral da equação $\ddot{y} + y = t$

Solução: Determina-se primeiro a solução da equação homogênea: $\ddot{y} + y = 0$. A equação característica $\lambda^2 + 1 = 0$ possui raízes $\lambda_1 = i$ e $\lambda_2 = -i$. Assim sendo, $\varphi(t) = e^{it} = \cos(t) + i\sin(t)$ é uma solução a valores complexos. Então $y_1(t) = \cos(t)$ e $y_2(t) = \sin(t)$ são soluções linearmente independentes de $\ddot{y} + y = 0$. Agora para se encontrar uma solução particular basta fazer $y_p(t) = A_1t + A_0$ e substituir na equação original.

$$\ddot{y}_p + y = t \quad (198)$$

$$\frac{d^2}{dt^2}(A_1t + A_0) + (A_1t + A_0) = t \quad (199)$$

$$A_1t + A_0 = t \quad (200)$$

Pela última equação conclui-se que $A_1 = 1$ e $A_0 = 0$. Portanto, a solução geral da equação é:

$$y(t) = k_1\cos(t) + k_2\sin(t) + t \quad (201)$$

Caso 2: Para $g(t) = e^{\alpha t}P_n(t)$ a equação diferencial torna-se:

$$a\ddot{y} + b\dot{y} + cy = e^{\alpha t}P_n(t) \quad (202)$$

Tomando-se $y_p(t) = e^{\alpha t}v$ e substituindo-se na equação acima obtém-se:

$$a\ddot{y}_p + b\dot{y}_p + cy_p = e^{\alpha t}P_n(t) \quad (203)$$

$$a\frac{d^2}{dt^2}(e^{\alpha t}v) + b\frac{d}{dt}(e^{\alpha t}v) + c(e^{\alpha t}v) = e^{\alpha t}P_n(t) \quad (204)$$

$$e^{\alpha t}[a\ddot{v} + (2a\alpha + b)\dot{v} + (a\alpha^2 + b\alpha + c)v] = e^{\alpha t}P_n(t) \quad (205)$$

$$a\ddot{v} + (2a\alpha + b)\dot{v} + (a\alpha^2 + b\alpha + c)v = P_n(t) \quad (206)$$

Portanto, a última equação pode ser resolvida pelo Caso 1 e determina-se assim a solução geral da equação do Caso 2.

Caso 3: Para $g(t) = e^{\alpha t}P_n(t)\sin(\beta t)$ a equação diferencial torna-se:

$$a\ddot{y} + b\dot{y} + cy = e^{\alpha t}P_n(t)\sin(\beta t) \quad (207)$$

Esta equação pode ser resolvida notando-se que

- $e^{i\beta t} = \cos(\beta t) + i\sin(\beta t)$
- se $y(t) = u(t) + iv(t)$ é uma solução com valores complexos da equação:

$$a\ddot{y} + b\dot{y} + cy = g_1(t) + ig_2(t) \quad (208)$$

em que a, b e c são constantes reais, então

$$\begin{cases} a\ddot{u} + b\dot{u} + cu = g_1(t) \\ a\ddot{v} + b\dot{v} + cv = g_2(t) \end{cases} \quad (209)$$

Caso 4: Para $g(t)$ uma combinação linear de funções dos Casos 1, 2 e 3, o problema pode ser resolvido aplicando-se o Princípio da Superposição de Soluções (Guidorizzi, 2002), que diz se φ_1 é solução da equação:

$$a\ddot{y} + b\dot{y} + cy = g_1(t) \quad (210)$$

e φ_2 é solução da equação

$$a\ddot{y} + b\dot{y} + cy = g_2(t) \quad (211)$$

e α_1 e α_2 são constantes, então a função $\varphi = \alpha_1\varphi_1(t) + \alpha_2\varphi_2(t)$ é solução da equação

$$a\ddot{y} + b\dot{y} + cy = \alpha_1g_1(t) + \alpha_2g_2(t) \quad (212)$$

B.4 Transformada de Laplace

Esta teoria é de grande utilidade na resolução de equações diferenciais ordinárias de qualquer ordem.

Definição B.4.1: Seja $f(t)$ uma função definida para todo $t \geq 0$. A função:

$$F(s) = \int_0^{\infty} f(t)dt \quad (213)$$

é chamada Transformada de Laplace de $f(t)$, e denotada por $\mathcal{L}[f(t)]$.

Exemplo B.4.1: Transformada de Laplace de algumas funções especiais:

1. $f(t) = 1$ para $s > 0$.

$$\mathcal{L}[1] = \int_0^{\infty} e^{-st} dt = \frac{1}{s} \quad (214)$$

2. $f(t) = e^{kt}$ para $s > k$.

$$\mathcal{L}[e^{kt}] = \lim_{b \rightarrow \infty} \int_0^b e^{-st} e^{kt} dt = \lim_{b \rightarrow \infty} \left[\frac{e^{(k-s)t}}{k-s} \right]_0^b = \frac{1}{s-k} \quad (215)$$

3. $f(t) = \cos(wt)$ para $s > 0$.

$$\mathcal{L}[\cos(wt)] = \lim_{b \rightarrow \infty} \int_0^b e^{-st} \cos(wt) dt = \lim_{b \rightarrow \infty} \left[\frac{we^{-st} \sin(wt) - se^{-st} \cos(wt)}{s^2 + w^2} \right]_0^b = \frac{s}{s^2 + w^2} \quad (216)$$

4. $f(t) = \sin(wt)$ para $s > 0$.

$$\mathcal{L}[\sin(wt)] = \lim_{b \rightarrow \infty} \int_0^b e^{-st} \sin(wt) dt = \lim_{b \rightarrow \infty} \left[\frac{we^{-st} \cos(wt) - se^{-st} \sin(wt)}{s^2 + w^2} \right]_0^b = \frac{w}{s^2 + w^2} \quad (217)$$

5. $f(t) = t^n$ para n inteiro $s > 0$.

$$\mathcal{L}[t^n] = \lim_{b \rightarrow \infty} \int_0^b e^{-st} t^n dt = \lim_{b \rightarrow \infty} \left[\left(-\frac{t^n e^{-st}}{s} \right)_0^b + \frac{n}{s} \int_0^b e^{-st} t^{n-1} dt \right] = \quad (218)$$

$$= \frac{n}{s} \int_0^{\infty} e^{-st} t^{n-1} dt = \frac{n}{s} \mathcal{L}[t^{n-1}] \quad (219)$$

Então, se $n = 1$, tem-se

$$\mathcal{L}[t] = \frac{1}{s} \mathcal{L}[1] = \frac{1}{s^2} \quad (220)$$

Para $n \geq 2$, tem-se:

$$\mathcal{L}[t^n] = \frac{n}{s} \mathcal{L}[t^{n-1}] = \frac{n(n-1)}{s^2} \mathcal{L}[t^{n-2}] = \dots = \frac{n!}{s^{n+1}} \quad (221)$$

Propriedade B.4.1: Linearidade, se $\mathcal{L}[f(t)] = F(s)$, $\mathcal{L}[g(t)] = G(s)$ e a e b são constantes, então:

$$\mathcal{L}[af(t) + bg(t)] = aF(s) + bG(s) = a\mathcal{L}[f(t)] + b\mathcal{L}[g(t)] \quad (222)$$

Propriedade B.4.2: Se $\mathcal{L}[f(t)] = F(s)$, para $s > s_0$, então

$$\mathcal{L}[e^{at}f(t)] = F(s-a), s > s_0 + a \quad (223)$$

Propriedade B.4.3: Se $\mathcal{L}[f(t)] = F(s)$, então:

$$\mathcal{L}[t^n f(t)] = (-1)^n \frac{d^n}{ds^n} F(s) \quad (224)$$

Propriedade B.4.4: Suponha que f e f' sejam integráveis em $[0, b]$, para todo $b > 0$. Se f for de ordem exponencial, então existe $\mathcal{L}[f'(t)]$ e

$$\mathcal{L}[f'(t)] = s\mathcal{L}[f(t)] - f(0) \quad (225)$$

Se for integrado por partes tem-se:

$$\int_0^b e^{-st} f'(t) dt = e^{-sb} f(b) - f(0) + s \int_0^b e^{-st} f(t) dt \quad (226)$$

$$\lim_{b \rightarrow \infty} \left[\int_0^b e^{-st} f'(t) dt \right] = \lim_{b \rightarrow \infty} \left[e^{-sb} f(b) - f(0) + s \int_0^b e^{-st} f(t) dt \right] \quad (227)$$

$$\mathcal{L}[f'(t)] = 0 - f(0) + \mathcal{L}[f(t)] \quad (228)$$

$$\mathcal{L}[f'(t)] = s\mathcal{L}[f(t)] - f(0) \quad (229)$$

Esta propriedade aplica-se às derivadas de ordem superior. Para uma derivada de segunda ordem tem-se:

$$\mathcal{L}[f''(t)] = s\mathcal{L}[f'(t)] - f'(0) \quad (230)$$

$$\mathcal{L}[f''(t)] = s\{s\mathcal{L}[f(t)] - f(0)\} - f'(0) \quad (231)$$

$$\mathcal{L}[f''(t)] = s^2\mathcal{L}[f(t)] - sf(0) - f'(0) \quad (232)$$

Portanto,

$$\mathcal{L}[f''(t)] = s^2\mathcal{L}[f(t)] - sf(0) - f'(0) \quad (233)$$

Esta equação é de grande utilidade na resolução de equações diferenciais de segunda ordem.

B.5 Transformada Inversa de Laplace

Definição B.5.1: Seja a função $F(s)$ definida em um intervalo (a, ∞) , o problema agora é encontrar uma função $f(t)$ tal que $\mathcal{L}[f(t)] = F(s)$. Esta tal f é chamada Transformada Inversa de F e será indicada por $\mathcal{L}^{-1}[F(s)]$

Exemplo B.5.1: As transformadas inversas das funções especiais são:

1. $\mathcal{L}^{-1}[\frac{1}{s}] = 1,$
2. $\mathcal{L}^{-1}[\frac{1}{s-k}] = e^{kt},$
3. $\mathcal{L}^{-1}[\frac{1}{s^{n+1}}] = \frac{t^n}{n!},$
4. $\mathcal{L}^{-1}[\frac{s}{s^2+w^2}] = \cos(wt),$
5. $\mathcal{L}^{-1}[\frac{w}{s^2+w^2}] = \text{sen}(wt).$

Exemplo B.5.2: Resolver $\mathcal{L}^{-1}[\frac{1}{s^2-4s+5}]$.

Solução: Reescrevendo $s^2 - 4s + 5 = (s - 2)^2 + 1$, tem-se:

$$\frac{1}{s^2 - 4s + 5} = \frac{1}{(s - 2)^2 + 1} = \mathcal{L}[e^{2t}\text{sen}(t)] \quad (234)$$

Portanto,

$$\mathcal{L}^{-1}[\frac{1}{s^2 - 4s + 5}] = e^{2t}\text{sen}(t) \quad (235)$$

Exemplo B.5.3: Resolver $\mathcal{L}^{-1}[\frac{s+2}{s^2+2s+10}]$.

Solução: Reescrevendo $s^2 + 2s + 10 = (s + 1)^2 + 9$, tem-se

$$\frac{s + 2}{s^2 + 2s + 10} = \frac{s + 1 - 1}{(s + 1)^2 + 9} = \frac{s + 1}{(s + 1)^2 + 9} + \frac{1}{3} \frac{3}{(s + 1)^2 + 3^2} \quad (236)$$

Note-se que

$$\mathcal{L}^{-1}[\frac{s + 1}{(s + 1)^2 + 3^2}] = e^{-t} \cos(3t) \quad (237)$$

$$\mathcal{L}^{-1}[\frac{3}{(s + 1)^2 + 3^2}] = e^{-t} \operatorname{sen}(3t) \quad (238)$$

Portanto,

$$\mathcal{L}^{-1}[\frac{s + 2}{s^2 + 2s + 10}] = e^{-t} \cos(3t) + \frac{1}{3} e^{-t} \operatorname{sen}(3t) \quad (239)$$

Exemplo B.5.5: Resolver $\mathcal{L}^{-1}[\frac{3s^2-7s+12}{(s-2)(s-3)(s+2)}]$.

Solução: Reescrevendo $\frac{3s^2-7s+12}{(s-2)(s-3)(s+2)} = \frac{A}{s-2} + \frac{B}{s-3} + \frac{C}{s+2}$ e determinando-se A , B e C , tem-se:

$$\frac{3s^2 - 7s + 12}{(s - 2)(s - 3)(s + 2)} = -\frac{5}{2} \frac{1}{s - 2} + \frac{18}{5} \frac{1}{s - 3} + \frac{19}{10} \frac{1}{s + 2} \quad (240)$$

Portanto,

$$\mathcal{L}^{-1}[\frac{3s^2 - 7s + 12}{(s - 2)(s - 3)(s + 2)}] = -\frac{5}{2} e^{2t} + \frac{18}{5} e^{3t} + \frac{19}{10} e^{-2t} \quad (241)$$

B.6 Sistema de Equações Diferenciais Lineares de Primeira Ordem

Definição B.6.1: Um sistema de equações diferenciais lineares de primeira ordem pode ser representado por:

$$\begin{cases} x'_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ x'_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ x'_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{cases} \quad (242)$$

onde as incógnitas são as funções deriváveis $x_1 = x_1(t)$, $x_2 = x_2(t)$, ..., $x_n = x_n(t)$ e $x'_1 = \frac{dx_1}{dt}$, $x'_2 = \frac{dx_2}{dt}$, ..., $x'_n = \frac{dx_n}{dt}$.

Assim o sistema pode ser reescrito como:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (243)$$

Definindo $\mathbf{x} = \mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$, e $A = (a_{ij})$, pode-se escrever este sistema de maneira mais sucinta como:

$$\mathbf{x}' = A\mathbf{x} \quad (244)$$

A solução desta equação é qualquer função vetorial $\mathbf{x} = \mathbf{x}(t)$ que a torne verdadeira. Em geral essa equação possui uma infinidade de soluções, mas o Problema de Valor Inicial (P.V.I.):

$$\begin{cases} \mathbf{x}' = A\mathbf{x} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases} \quad (245)$$

possui uma única solução.

Teorema B.6.1: Seja A uma matriz $n \times n$. Então o sistema de equações diferenciais $\mathbf{x}' = A\mathbf{x}$ possui um conjunto de n soluções linearmente independentes

$$[\mathbf{x}_1(t), \mathbf{x}_2(t), \dots, \mathbf{x}_n(t)] \quad (246)$$

e toda solução $\mathbf{x} = \mathbf{x}(t)$ se expressa unicamente como combinação linear dessas soluções fundamentais, ou seja,

$$\mathbf{x}(t) = c_1\mathbf{x}_1(t) + c_2\mathbf{x}_2(t) + \cdots + c_n\mathbf{x}_n(t) \quad (247)$$

Teorema 3.6.2: Se λ é um autovalor de A e \mathbf{v} é um autovetor associado ao autovalor λ , então $\mathbf{x} = e^{\lambda t}\mathbf{v}$ é uma solução do sistema $\mathbf{x}' = A\mathbf{x}$.

Demonstração: Seja $\mathbf{x} = e^{\lambda t}\mathbf{v}$ como no enunciado do teorema. Então pelo lado direito da equação tem-se:

$$A\mathbf{x} = A(e^{\lambda t}\mathbf{v}) \quad (248)$$

$$A\mathbf{x} = e^{\lambda t} A\mathbf{v} \quad (249)$$

$$A\mathbf{x} = \lambda e^{\lambda t} \mathbf{v} \quad (250)$$

e pelo lado esquerdo:

$$\mathbf{x}'(t) = \lim_{h \rightarrow 0} \frac{\mathbf{x}(t+h) - \mathbf{x}(t)}{h} \quad (251)$$

$$\mathbf{x}'(t) = \lim_{h \rightarrow 0} \frac{e^{\lambda(t+h)} \mathbf{v} - e^{\lambda t} \mathbf{v}}{h} \quad (252)$$

$$\mathbf{x}'(t) = \lim_{h \rightarrow 0} \frac{e^{\lambda(t+h)} - e^{\lambda t}}{h} \mathbf{v} \quad (253)$$

$$\mathbf{x}'(t) = (e^{\lambda t} \mathbf{v})' = \lambda e^{\lambda t} \mathbf{v} \quad (254)$$

Portanto, tem-se que $\mathbf{x}' = A\mathbf{x}$ e conclui-se que $\mathbf{x} = e^{\lambda t} \mathbf{v}$ é solução da equação diferencial.

Teorema B.6.3: Seja A uma matriz quadrada $n \times n$, e sejam $\lambda_1, \lambda_2, \dots, \lambda_k$ autovalores (não necessariamente distintos) de A . Se $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ são autovetores linearmente independentes de A associados a cada um desses respectivos autovalores, então

$$\mathbf{x}_1 = e^{\lambda_1 t} \mathbf{v}_1$$

$$\mathbf{x}_2 = e^{\lambda_2 t} \mathbf{v}_2$$

$$\vdots$$

$$\mathbf{x}_k = e^{\lambda_k t} \mathbf{v}_k$$

são soluções linearmente independentes do sistema $\mathbf{x}' = A\mathbf{x}$.

Demonstração: em virtude do Teorema 3.6.2 sabe-se que $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ são soluções do sistema $\mathbf{x}' = A\mathbf{x}$. Considerando a equação:

$$c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \dots + c_k \mathbf{x}_k = \mathbf{0} \quad (255)$$

Substituindo as soluções nessa equação, tem-se:

$$c_1 e^{\lambda_1 t} \mathbf{v}_1 + c_2 e^{\lambda_2 t} \mathbf{v}_2 + \dots + c_k e^{\lambda_k t} \mathbf{v}_k = \mathbf{0} \quad (256)$$

Esta equação deve ser válida para todo $t \in R$, então em particular deve ser válida para $t = 0$. Assim, fazendo-se $t = 0$, tem-se:

$$c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \cdots + c_k \mathbf{v}_k = 0 \quad (257)$$

Porém, por hipótese, os autovetores $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ são linearmente independentes. Assim, a única solução possível para essa equação é a solução trivial $c_1 = c_2 = \cdots = c_k = 0$. Isso mostra que a única solução para a equação

$$c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \cdots + c_k \mathbf{x}_k = 0 \quad (258)$$

é a solução trivial, logo as soluções $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ de $\mathbf{x}' = A\mathbf{x}$ também são linearmente independentes.

Teorema B.6.4: Se a matriz A é diagonalizável, então a solução geral do sistema de equações diferenciais lineares de primeira ordem $\mathbf{x}' = A\mathbf{x}$ é

$$\mathbf{x}(t) = c_1 e^{\lambda_1 t} \mathbf{v}_1 + c_2 e^{\lambda_2 t} \mathbf{v}_2 + \cdots + c_n e^{\lambda_n t} \mathbf{v}_n \quad (259)$$

onde $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ são os autovetores linearmente independentes de A associados aos respectivos autovalores $\lambda_1, \lambda_2, \dots, \lambda_k$ (não necessariamente distintos) de A .

Demonstração: Se A é diagonalizável, então existe uma base $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ de R^n formada por autovetores de A . Em particular, tem-se exatamente n autovetores linearmente independentes de A . O resultado segue dos Teoremas 3.6.1 e 3.6.3.

Exemplo B.6.1: Encontrar a solução geral do sistema:

$$\begin{cases} \mathbf{x}'_1 = -7\mathbf{x}_1 + 5\mathbf{x}_2, \\ \mathbf{x}'_2 = -10\mathbf{x}_1 + 8\mathbf{x}_2. \end{cases} \quad (260)$$

Solução: O sistema pode ser reescrito como

$$\begin{bmatrix} \mathbf{x}'_1 \\ \mathbf{x}'_2 \end{bmatrix} = \begin{bmatrix} -7 & 5 \\ -10 & 8 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (261)$$

onde A é:

$$A = \begin{bmatrix} -7 & 5 \\ -10 & 8 \end{bmatrix} \quad (262)$$

O polinômio característico dessa matriz é

$$p_A(\lambda) = \det(A - \lambda I) = \begin{vmatrix} -7 & 5 \\ -10 & 8 \end{vmatrix} = \lambda^2 - \lambda - 6 = (-2 - \lambda)(3 - \lambda) \quad (263)$$

Portanto, os autovalores são $\lambda_1 = -2$ e $\lambda_2 = 3$. O vetor $\mathbf{v}_1 = (1, 1)$ é o autovetor associado ao autovalor λ_1 e que $\mathbf{v}_2 = (1, 2)$ é o autovetor associado ao autovalor λ_2 .

Conclui-se que $x_1 = e^{-2t}(1, 1)$ e $x_2 = e^{3t}(1, 2)$ são as duas soluções para $\mathbf{x}' = A\mathbf{x}$. Pelo Teorema 3.6.1, essas são as soluções fundamentais e toda solução será combinação linear dessas duas. Assim,

$$x(t) = c_1 e^{-2t}(1, 1) + c_2 e^{3t}(1, 2) \quad (264)$$

ou

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = c_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{-2t} + c_2 \begin{bmatrix} 1 \\ 2 \end{bmatrix} e^{3t} \quad (265)$$

onde c_1 e c_2 são constantes arbitrárias que podem ser obtidas a partir de uma condição inicial.

C Sistemas de Controle

Este capítulo informará conceitos básicos referentes aos sistemas de controle que são de fundamental importância para esta pesquisa.

C.1 Sistemas de Controle

Há uma grande variedade de sistemas em que teorias de controle são aplicáveis tais como sistemas de controle de velocidade, sistemas de controle de temperatura, sistemas de controle robusto, sistemas empresariais, dentre outros. Definições básicas (Nise, 2002 e Ogata, 2010) serão descritas a seguir.

Definição C.1.1: *Variável controlada* é a grandeza que será medida e controlada. Geralmente, a variável controlada é a saída do sistema.

Definição C.1.2: *Variável manipulada* é a grandeza modificada pelo controlador, de modo que afete o valor da variável controlada.

Definição C.1.3: *Planta* é qualquer equipamento físico a ser controlado com o propósito de realizar determinada operação. Exemplos são: componente mecânico, reator químico, forno, dentre outros.

Definição C.1.4: *Processo* é toda operação a ser controlada.

Definição C.1.5: *Sistema* é uma combinação de componentes que agem em conjunto para atingir determinado objetivo. Tais sistemas podem ser físicos, químicos, biológicos, econômicos, dentre outros.

Definição C.1.6: *Distúrbio* é um sinal que afeta de maneira indesejável a variável de saída do sistema. Se gerado dentro de um sistema é denominado distúrbio interno, enquanto distúrbio externo é aquele que é gerado fora do sistema e comporta-se como um sinal de entrada do sistema.

Definição C.1.7: *Controle com realimentação* é uma operação que na presença de distúrbios tende a diminuir a diferença entre a saída de um sistema e alguma entrada de referência e atua com base nesta diferença.

Sistema de controle de malha fechada é um sistema de controle com realimentação de saída a qual o sinal de erro atuante, diferença dos sinais de entrada e saída, realimenta o controlador, minimizando o erro e acertando a saída do sistema ao valor desejado. A Figura 12 ilustra um sistema com realimentação de saída com entrada $u(t)$ e saída $y(t)$. Os blocos C , G e H referem-se à modelagem matemática do sistema.

Há outro tipo de sistema de realimentação na literatura denominado sistema com realimentação de estados. Tal sistema está descrito na Figura 13. Tal sistema é representado em espaço de estados cujo vetor de entrada $u(t)$ é uma combinação linear $(-K)$ do vetor de estados $x(t)$.

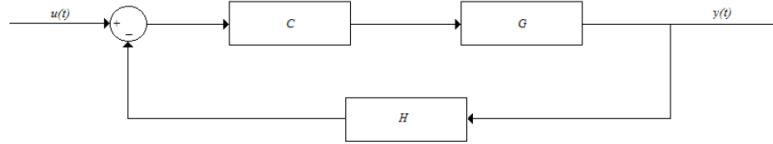


Figura 12: Modelo de um Sistema com Realimentação de Saída

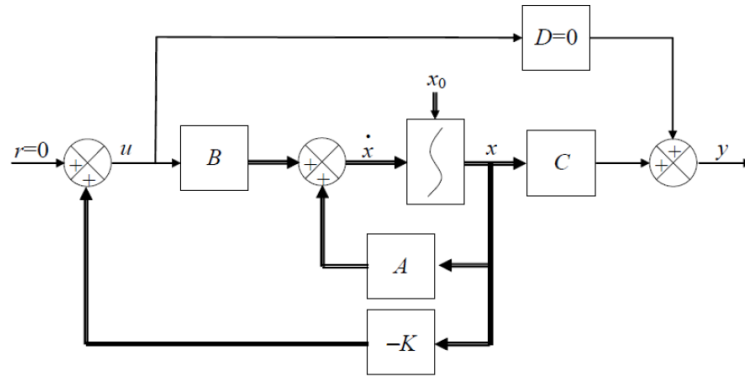


Figura 13: Modelo de um Sistema com Realimentação de Estados

Tal sistema é diferente do sistema com realimentação de saída. Neste último o vetor de entrada $u(t)$ é uma combinação linear (L) da saída $y(t)$, Figura 14.

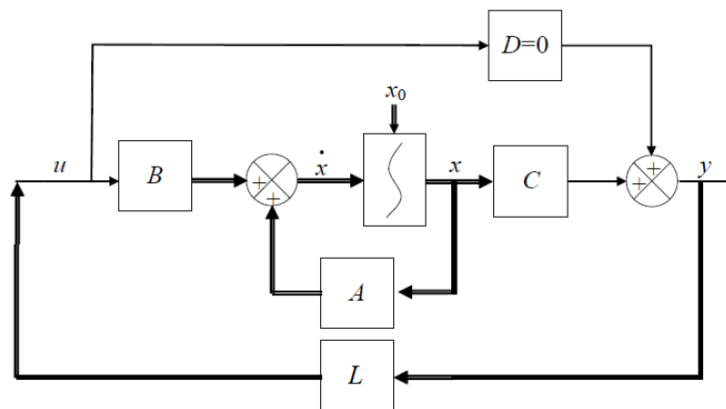


Figura 14: Modelo de um Sistema em Espaço de Estados com Realimentação de Saída

Sistemas de controle de malha aberta são aqueles em que o sinal de saída não exerce nenhuma ação de controle no sistema. Assim sendo, o sinal de saída não é realimentado nem comparado com o sinal de entrada. A Figura 15 ilustra um modelo de sistema em malha aberta, onde $u(t)$ é o sinal de entrada e $y(t)$ é o sinal de saída. Os blocos C e G referem-se à modelagem matemática do sistema.

Sistemas de controle de malha fechada versus de malha aberta

O uso da realimentação no sistema de malha fechada permite que a resposta do sistema



Figura 15: Modelo de um Sistema em Malha Aberta

seja praticamente insensível a distúrbios externos e a variação internas nos parâmetros do sistema. Desta maneira, é possível a utilização de componentes imprecisos para se montar o projeto o obter o controle preciso do sistema.

Do ponto de vista da estabilidade, o sistema de malha aberta é mais fácil de ser montado uma vez que os problemas de estabilidade são menos significativos. Entretanto, estudos de estabilidade são importantes em sistemas de controle de malha fechada, pois permite a correção do erro entre entrada e saída.

É conveniente a utilização dos sistemas de malha aberta quando o sinal de entrada é conhecido e o sistema está imune a distúrbios. No entanto, quando o sinal é desconhecido e o sistema está submetido a distúrbios, o sistema em malha fechada é a melhor solução. Quanto ao custo e potência, o sistema em malha fechada apresenta valores maiores por utilizar maior número de equipamentos.

O sistema adotado neste projeto de pesquisa é o de malha fechada pois será necessário fazer estudos de estabilidade.

C.2 Modelagem Matemática de Sistemas de Controle

A fim de se estudar sistemas de controle será necessário modelar os sistemas dinâmicos em termos matemáticos e analisar suas características dinâmicas. Modelar matematicamente um sistema é definir um conjunto de equações que represente a dinâmica do sistema com precisão. Tal modelagem não é única, diversos modelos matemáticos podem descrever um sistema.

A dinâmica de sistemas mecânicos, elétricos, biológicos, químicos, econômicos, dentre outros sistemas pode ser descrito em termos de equações diferenciais. Equações diferenciais são obtidas pelas leis físicas que regem o sistema - as leis de Kirchhoff para os sistemas elétricos por exemplo.

Quando se modela um sistema dinâmico, certas simplificações são tomadas e afetam consequentemente a precisão dos resultados da análise. Na maioria das vezes desprezam-se as não linearidades do sistema pelo fato do modelo linear ser mais simples e fácil de trabalhar. No entanto, se estas não linearidades são importantes utiliza-se a representação em espaço de estados e aplica-se a análise.

Definição C.2.1: *Sistemas lineares* são todos os sistemas em que pode-se aplicar o princípio da superposição, que afirma que se a resposta produzida pela aplicação simultânea de duas funções é a soma das duas respostas individuais.

Definição C.2.2: *Sistemas lineares invariantes ao tempo* são sistemas dinâmicos compostos por componentes lineares de parâmetros concentrados invariantes no tempo que podem ser descritos por equações diferenciais lineares invariantes no tempo, isto é, coeficientes constantes.

Definição C.2.3: *Sistemas lineares variantes no tempo* são sistemas cujos coeficientes das equações diferenciais são funções do tempo.

Definição C.2.4: *Função de transferência* é a relação entre a transformada de Laplace da saída e a transformada de Laplace da entrada, admitindo-se todas as condições iniciais nulas. Ela só pode ser empregada em sistemas lineares invariantes no tempo.

Seja um sistema linear invariante no tempo descrito pela seguinte equação diferencial:

$$a_0 y^{(n)} + a_1 y^{(n-1)} + \dots + a_{n-1} \dot{y} + a_n y = b_0 x^{(n)} + b_1 x^{(n-1)} + \dots + b_{n-1} \dot{x} + b_n x \quad (266)$$

onde x é a entrada do sistema e y é a saída do sistema. A função de transferência $G(s)$ é a relação entre a transformada de Laplace da saída e a transformada de Laplace da entrada com todas as condições iniciais nulas. Assim sendo,

$$G(s) = \frac{\mathcal{L}[saida]}{\mathcal{L}[entrada]} = \frac{Y(s)}{X(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^m + a_1 s^{m-1} + \dots + a_{m-1} s + a_m} \quad (267)$$

Se a maior potência de s no denominador for n , então o sistema é denominado sistema de ordem n .

Assim sendo, a partir da função de transferência $G(s)$ é possível obter a saída $Y(s)$ em função da entrada $X(s)$:

$$Y(s) = G(s)X(s) \quad (268)$$

No entanto, esta multiplicação só é possível no domínio da frequência. No domínio do tempo a saída é obtida através da integral de convolução.

Definição C.2.5: *Integral de convolução* é a relação entre o sinal de entrada $x(t)$ e o sinal de saída $y(t)$ no domínio do tempo e é definido por:

$$y(t) = \int_0^t x(\tau)g(t - \tau)d\tau \quad (269)$$

onde $x(t)$ e $g(t)$ são 0 para $t < 0$.

C.3 Diagrama de Blocos

Definição C.3.1: *Diagrama de blocos* é uma representação gráfica das funções desempenhadas por cada componente de um sistema e descreve o fluxo de sinais entre eles. Esses diagramas mostram o inter-reacionamento entre os diversos componentes do sistema.

Num diagrama de blocos, as funções modeladas são representadas por blocos funcionais que contêm as funções matemáticas que descrevem o sistema por completo. A utilidade desta representação é a facilidade na compreensão do funcionamento do sistema do ponto de vista matemático e no estudo do mesmo através das equações de cada bloco.

Na maioria dos estudos em diagramas de blocos, o modelo matemático representado está como uma função de transferência. Portanto, é comum o estudo no domínio da frequência e não na do tempo. Determinados blocos recebem uma determinada nomenclatura em função de seus objetivos tais como: controlador, amplificador, atuador, sensor, planta, dentre outros. Em função de seu uso na pesquisa de projeto será estudado os tipos de cada controlador existentes.

C.4 Controladores Automáticos

Controladores automáticos comparam o sinal de saída do sistema com o sinal de entrada. Esta comparação fornecerá o erro deste sinais e ações de controle serão tomadas a fim de reduzir este erro a zero. Com isso, diversas ações de controle foram criadas a fim de reduzir o erro entre os sinais, cada um diferenciando-se na complexidade de construção e operação dos controladores. A maioria dos controladores industriais pode ser classificada em:

1. Controladores de duas posições ou *on-off*;
2. Controladores proporcionais;
3. Controladores integrais;
4. Controladores proporcional-integrais;
5. Controladores proporcional-derivativos;
6. Controladores proporcional-integral-derivativos.

C.4.1 Controladores de duas posições ou *on-off*

Neste tipo de controlador, a ação de controle só pode assumir duas posições: *on* (ativo) ou *off* (inativo). Este tipo de ação de controle é muito utilizado devido à simplicidade e ao baixo custo de construção.

Assim, considere o sinal de erro na entrada do controlador de $e(t)$ e o sinal de saída do mesmo de $u(t)$. Na ação de controle *on – off* o sinal $u(t)$ pode assumir somente dois valores, um máximo ou um mínimo. Portanto,

$$u(t) = U_1, e(t) > 0 \quad (270)$$

$$u(t) = U_2, e(t) < 0 \quad (271)$$

onde U_1 e U_2 são constantes. Geralmente, controladores de duas posições são dispositivos elétricos.

C.4.2 Controladores proporcionais

Neste controlador, a relação entre o sinal de saída do controlador $u(t)$ e o sinal de erro atuante $e(t)$ é:

$$u(t) = K_p e(t) \quad (272)$$

,
ou pela transformada de Laplace:

$$\frac{U(s)}{E(s)} = K_p \quad (273)$$

.

Assim, como pode ser visto, este controlador é um amplificador com ganho ajustável.

C.4.3 Controladores integrais

Neste controlador, o sinal de saída do controlador $u(t)$ varia a uma taxa proporcional ao sinal de erro atuante $e(t)$. Assim a relação é:

$$\frac{du(t)}{dt} = K_i e(t) \quad (274)$$

ou

$$u(t) = K_i \int_0^t e(t) dt \quad (275)$$

onde K_i é uma constante ajustável. Pela transformada de Laplace tem-se:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (276)$$

.

C.4.4 Controladores proporcional-integrais

Neste controlador, a relação entre o sinal de saída do controlador $u(t)$ e o sinal de erro atuante $e(t)$ é:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (277)$$

,
e pela transformada de Laplace tem-se:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (278)$$

,
onde T_i é chamado *tempo integrativo*.

C.4.5 Controladores proporcional-derivativos

Neste controlador, a relação entre o sinal de saída do controlador $u(t)$ e o sinal de erro atuante $e(t)$ é:

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (279)$$

,
ou pela transformada de Laplace tem-se:

$$\frac{U(s)}{E(s)} = K_p (1 + T_d s) \quad (280)$$

,
onde T_d é denominado *tempo derivativo*.

C.4.6 Controladores proporcional-integral-derivativos

Neste controlador, a relação entre o sinal de saída do controlador $u(t)$ e o sinal de erro atuante $e(t)$ é:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (281)$$

,
e pela transformada de Laplace tem-se:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (282)$$

onde K_p é o ganho proporcional, T_i é o tempo integrativo e T_d é o tempo derivativo.

C.5 Modelagem em Espaço de Estados

Antes de falar sobre a modelagem é fundamental conhecer os conceitos relacionados análise de sistemas de controle em espaço de estados.

Definição C.5.1: *Teoria de controle moderno* tem como base o conceito de estado e foi desenvolvida a fim de realizar tarefas de alta precisão na análise de sistemas de engenharia.

Definição C.5.2: *Estado* de um sistema dinâmico é o menor conjunto de variáveis, denominadas variáveis de estado, tais que o conhecimento dessas variáveis em $t = t_0$ determina completamente o comportamento do sistema para qualquer instante $t \geq t_0$.

Definição C.5.3: *Variáveis de estado* de um sistema dinâmico são aquelas que constituem o menor conjunto de variáveis capaz de determinar o estado desse sistema. Se n variáveis de estado $x_1, x_2, x_3, \dots, x_n$ são necessárias para descrever completamente o sistema dinâmico então n variáveis formam um conjunto de variáveis de estado.

Definição C.5.4: *Vetor de estados* é um vetor \mathbf{x} com n variáveis de estado. Portanto, um vetor de estado é aquele que determina univocamente o estado do sistema $x(t)$ para qualquer instante $t \geq t_0$, uma vez que é dado o estado em $t = t_0$ e a entrada $u(t)$ para $t \geq t_0$ é especificada.

Definição C.5.5: *Espaço de estados* é o espaço n -dimensional, cujos eixos coordenados são formados pelos eixos x_1, x_2, \dots, x_n , as variáveis de estado.

Equações no espaço de estados

A modelagem em espaço de estados envolve três tipos de variáveis: variáveis de entrada, variáveis de saída e variáveis de estado. A modelagem de sistemas dinâmicos não é única, mas o número de variáveis de estado é o mesmo para qualquer representação em espaço de estados.

Seja um sistema dinâmico com múltiplas entradas e múltiplas saídas e que envolva n variáveis de estado. Considere que existam r entradas $u_1(t), u_2(t), \dots, u_r(t)$ e m saídas $y_1(t), y_2(t), \dots, y_m(t)$. Então o sistema pode ser descrito como:

$$\begin{cases} \dot{x}_1(t) = f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \dot{x}_2(t) = f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ \dot{x}_n(t) = f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t). \end{cases} \quad (283)$$

As saídas do sistema podem ser descritas por:

$$\begin{cases} y_1(t) = g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ y_2(t) = g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ y_m(t) = g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t). \end{cases} \quad (284)$$

Portanto, se for definido que:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \quad (285)$$

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix} \quad (286)$$

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix} \quad (287)$$

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix} \quad (288)$$

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, t) = \begin{bmatrix} g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix} \quad (289)$$

então

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (290)$$

$$\dot{\mathbf{y}}(t) = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \quad (291)$$

onde estas equações são definidas como equação de estado e equação de saída respectivamente. Se as funções \mathbf{f} e \mathbf{g} forem dependentes do tempo, o sistema é chamado sistema variante no tempo.

Se estas equações forem linearizadas em torno de um ponto de operação, então tem-se as seguintes equações de estado e saída linearizadas:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (292)$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \quad (293)$$

onde $\mathbf{A}(t)$ é denominada matriz de estado, $\mathbf{B}(t)$, matriz de entrada, $\mathbf{C}(t)$, matriz de saída, e $\mathbf{D}(t)$, matriz de transmissão direta.

Correlação entre funções de transferência e equações no espaço de estados

Considere o sistema cuja função de transferência é dada por:

$$\frac{Y(s)}{U(s)} = G(s) \quad (294)$$

Esse sistema pode ser representado no espaço de estados pelas seguintes equações:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (295)$$

$$y = \mathbf{C}\mathbf{x} + Du \quad (296)$$

onde \mathbf{x} é o vetor de estado, u é a entrada e y é a saída. A transformada de Laplace destas equações é dada por:

$$s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}U(s) \quad (297)$$

$$Y(s) = \mathbf{C}\mathbf{X}(s) + DU(s) \quad (298)$$

Estabelecendo $\mathbf{x}(0) = \mathbf{0}$, ou seja, quando as condições iniciais são nulas, tem-se:

$$s\mathbf{X}(s) + \mathbf{A}\mathbf{X}(s) = \mathbf{B}U(s) \quad (299)$$

$$(s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{B}U(s) \quad (300)$$

Multiplicando ambos os lados esquerdos por $(s\mathbf{I} - \mathbf{A})^{-1}$ tem-se:

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s) \quad (301)$$

Assim,

$$Y(s) = \mathbf{C}\mathbf{X}(s) + DU(s) \quad (302)$$

$$Y(s) = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + D]U(s) \quad (303)$$

$$\frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} + D \quad (304)$$

Portanto, a função de transferência em termos de $\mathbf{A}, \mathbf{B}, \mathbf{C}$ e D é:

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} + D \quad (305)$$

O conhecimento do modelo matemático do sistema é o primeiro passo para a análise de um sistema de controle. A partir deste modelo, é possível analisar o desempenho do sistema através de diversos modelos disponíveis.

Resposta Transitória e resposta estacionária

Em um sistema de controle a resposta temporal é composta por duas partes: a resposta transitória e a resposta estacionária. Resposta transitória é aquela que vai do estado inicial ao estado final. Resposta estacionária é o comportamento do sistema quando o tempo t tende ao infinito. Assim, a resposta temporal do sistema $c(t)$ pode ser definida como:

$$c(t) = c_{tr}(t) + c_{ss}(t) \quad (306)$$

onde $c_{tr}(t)$ é a resposta transitória e $c_{ss}(t)$ é a resposta estacionária.

Estabilidade absoluta, estabilidade relativa e erro estacionário

A característica mais importante no comportamento dinâmico de um sistema dinâmico é sua estabilidade absoluta, ou seja, se ele é estável ou instável. Um sistema de controle está

em equilíbrio se a saída permanece no mesmo estado na ausência de qualquer distúrbio ou sinal de entrada. Um sistema de controle linear e invariante no tempo é estável se a saída retorna para o estado de equilíbrio quando o mesmo é submetido a uma condição inicial. O mesmo sistema é criticamente estável se as oscilações do sinal de saída se repetirem de maneira contínua.

Em um sistema físico de controle há energia armazenada de maneira que quando é aplicado um sinal na entrada do sistema a saída não acompanha a entrada imediatamente, ela apresenta uma resposta transitória antes que o regime permanente seja alcançado. A resposta transitória de um sistema de controle apresenta oscilações amortecidas antes de atingir o estado permanente. Se a saída do sistema em regime permanente não coincidir com a entrada, diz-se que o sistema apresenta um erro estacionário.

Sistema de primeira ordem

Um sistema de primeira ordem pode ser descrito pela seguinte relação matemática entrada-saída:

$$\frac{C(s)}{R(s)} = \frac{1}{Ts + 1} \quad (307)$$

Aplicando um degrau unitário na entrada do sistema, ou seja, $R(s) = \frac{1}{s}$, tem-se:

$$C(s) = \frac{1}{Ts + 1} \frac{1}{s} \quad (308)$$

Expandindo o segundo membro em frações parciais tem-se:

$$C(s) = \frac{1}{s} - \frac{T}{Ts + 1} = \frac{1}{s} - \frac{1}{s + (\frac{1}{T})} \quad (309)$$

Assim sendo, aplicando a transformada inversa de Laplace obtém-se a resposta do sistema:

$$c(t) = 1 - e^{-\frac{t}{T}}, t \geq 0 \quad (310)$$

A partir desta resposta é possível verificar que a saída não alcança o valor 1 imediatamente, há um termo exponencial que atrasa este objetivo, correspondendo à resposta transitória.

Sistema de segunda ordem

A função de transferência de malha fechada de um sistema de segunda ordem é descrito

por:

$$\frac{C(s)}{R(s)} = \frac{K}{Js^2 + Bs + K} \quad (311)$$

,

podendo ser reescrita como:

$$\frac{C(s)}{R(s)} = \frac{\frac{K}{J}}{\left[s + \frac{B}{2J} + \sqrt{\left(\frac{B}{2J}\right)^2 - \frac{K}{J}}\right]\left[s + \frac{B}{2J} - \sqrt{\left(\frac{B}{2J}\right)^2 - \frac{K}{J}}\right]} \quad (312)$$

.

Os polos de malha fechada são complexos conjugados se $B^2 - 4JK < 0$ e são reais se $B^2 - 4JK \geq 0$. Para a análise da resposta transitória é conveniente escrever:

$$\frac{K}{J} = \omega_n^2 \quad (313)$$

,

$$\frac{B}{J} = 2\zeta\omega_n = 2\sigma \quad (314)$$

,

onde σ é chamado atenuação, ω_n é a frequência angular não amortecida e ζ é o coeficiente de amortecimento do sistema. O coeficiente de amortecimento é a relação entre amortecimento real B e o amortecimento crítico $B_c = 2\sqrt{JK}$:

$$\zeta = \frac{B}{B_c} = \frac{B}{2\sqrt{JK}} \quad (315)$$

.

Assim sendo, a função de transferência do sistema de malha fechada de segunda ordem pode ser reescrita como:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (316)$$

.

A partir desta função de transferência é possível analisar o comportamento dinâmico do sistema através dos parâmetros ζ e ω_n . Se $0 < \zeta < 1$, os polos de malha fechada são complexos conjugados e estão no semiplano esquerdo do plano complexo s . Nesta situação, o sistema é denominado subamortecido e a resposta é oscilatória. Se $\zeta = 0$, a resposta transitória não decai, ela é puramente oscilatória. Se $\zeta = 1$, o sistema é definido como criticamente amortecido. Os sistemas superamortecidos apresentam $\zeta > 1$.

Será analisada a resposta do sistema quando é aplicado o degrau unitário na entrada:

1. *Sistema subamortecido* ($0 < \zeta < 1$):

A função de transferência pode ser reescrita como:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s + \zeta\omega_n + j\omega_d)(s + \zeta\omega_n - j\omega_d)} \quad (317)$$

onde $\omega_d = \omega_n \sqrt{1 - \zeta^2}$. A frequência ω_d é denominada frequência natural amortecida do sistema. A resposta ao degrau unitário é:

$$C(s) = \frac{\omega_n^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)s} \quad (318)$$

$$, C(s) = \frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - \frac{\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} \quad (319)$$

Aplicando a transformada inversa de Laplace obtém a resposta do sistema:

$$c(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1 - \zeta^2}} \text{sen}(\omega_d t + tg^{-1} \frac{\sqrt{1 - \zeta^2}}{\zeta}) \quad (320)$$

para $t \geq 0$.

2. *Sistema criticamente amortecido* ($\zeta = 1$):

Neste sistema a função de transferência que descreve o sistema apresenta dois polos iguais. A resposta ao degrau unitário é:

$$C(s) = \frac{\omega_n^2}{(s + \omega_n)^2 s} \quad (321)$$

Aplicando a transformada inversa de Laplace obtém-se a resposta do sistema:

$$c(t) = 1 - e^{-\omega_n t}(1 + \omega_n t) \quad (322)$$

para $t \geq 0$.

3. *Sistema superamortecido* ($\zeta > 1$):

Neste sistema a função de transferência que descreve o sistema apresenta dois polos reais, negativos e desiguais. A resposta ao degrau unitário é:

$$C(s) = \frac{\omega_n^2}{(s + \zeta\omega_n + \omega_n \sqrt{\zeta^2 - 1})(s + \zeta\omega_n - \omega_n \sqrt{\zeta^2 - 1})s} \quad (323)$$

Aplicando a transformada de Laplace obtém-se a resposta do sistema:

$$c(t) = 1 + \frac{\omega_n}{2\sqrt{\zeta^2 - 1}} \left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2} \right) \quad (324)$$

para $s_1 = (\zeta + \sqrt{\zeta^2 - 1})\omega_n$, $s_2 = (\zeta - \sqrt{\zeta^2 - 1})\omega_n$ e $t \geq 0$.

D Desigualdade Matricial Linear

Este capítulo fará uma breve descrição da teoria básica de Desigualdades (Inequações) Matriciais Lineares ou LMI, do inglês *Linear Matrix Inequalities* utilizadas neste projeto de pesquisa.

D.1 Definição

Desigualdades (Inequações) Matriciais Lineares são definidas como:

$$F(\xi) = F_0 + F_1\xi_1 + F_2\xi_2 + \cdots + F_n\xi_n \succ 0 \quad (325)$$

onde F_i , $i = 1, 2, \dots, n$, são matrizes constantes, reais e simétricas ($F(\xi) = F'(\xi)$, $F(\xi) \in \mathbb{R}^{n \times n}$), e $\xi = [\xi_1, \xi_2, \dots, \xi_n]'$ é um vetor de variáveis a determinar tais que a desigualdade acima seja satisfeita.

Se a desigualdade é satisfeita, para um determinado ξ , então $F(\xi)$ é definida positiva. Assim, as seguintes relações são equivalentes:

- O polinômio $p'F(\xi)p$ é positivo, para qualquer $p \in \mathbb{R}^n$, $p \neq 0$;
- Todos os autovalores de $F(\xi)$ são positivos, ou seja, $\lambda(F(\xi)) > 0$;
- O menor autovalor de $F(\xi)$ é positivo, ou seja, $\min\{\lambda(F(\xi))\} > 0$

Em aplicações de controle, as desigualdades têm matrizes como variáveis de decisão. No entanto, sempre é possível reescrever uma LMI na forma 312. Seja a seguinte desigualdade conhecida como desigualdade de Lyapunov:

$$A'P + PA + Q \prec 0 \quad (326)$$

com

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \quad (327)$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (328)$$

onde P é uma matriz a ser determinada, $Q = Q'$ e A são matrizes dadas. Assim sendo a desigualdade pode ser reescrita na forma 312 como:

$$-(F_0 + p_1F_1 + p_2F_2 + p_3F_3) \succ 0 \quad (329)$$

onde

$$F_0 = Q, F_1 = \begin{bmatrix} 2a_{11} & a_{12} \\ a_{21} & 0 \end{bmatrix}, F_2 = \begin{bmatrix} 2a_{11} & a_{11} + a_{22} \\ a_{11} + a_{22} & 2a_{12} \end{bmatrix}, F_3 = \begin{bmatrix} 0 & 2a_{21} \\ a_{21} & 2a_{22} \end{bmatrix}. \quad (330)$$

Exemplo D.1.1: Dado um sistema dinâmico linear na forma $\dot{x} = Ax$, encontrar uma matriz P que satisfaça as desigualdades matriciais:

$$P \succ 0 \quad (331)$$

$$A'P + PA \prec 0 \quad (332)$$

onde a matriz A é:

$$A = \begin{bmatrix} -1 & 2 & 0 \\ -3 & -4 & 1 \\ 0 & 0 & -2 \end{bmatrix} \quad (333)$$

Uma possível solução para a matriz P que respeite as duas desigualdades matriciais impostas é:

$$P = \begin{bmatrix} 0,5603 & 0,0452 & -0,0070 \\ 0,0452 & 0,2490 & 0,0348 \\ -0,0070 & 0,0348 & 0,4116 \end{bmatrix} \quad (334)$$

Observa-se que $P = P'$. Os autovalores de P são:

$$\lambda(P) = \begin{bmatrix} 0,2354 \\ 0,4187 \\ 0,5667 \end{bmatrix} \quad (335)$$

Como todos os autovalores da matriz P são positivos então a desigualdade em que P seja simétrica e definida positiva foi alcançada.

Para a segunda inequação:

$$A'P + PA = \begin{bmatrix} -1 & -3 & 0 \\ -2 & -4 & 0 \\ 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} 0,5603 & 0,0452 & -0,0070 \\ 0,0452 & 0,2490 & 0,0348 \\ -0,0070 & 0,0348 & 0,4116 \end{bmatrix} + \quad (336)$$

$$+ \begin{bmatrix} 0,5603 & 0,0452 & -0,0070 \\ 0,0452 & 0,2490 & 0,0348 \\ -0,0070 & 0,348 & 0,4116 \end{bmatrix} \begin{bmatrix} -1 & -2 & 0 \\ -3 & -4 & 1 \\ 0 & 0 & -2 \end{bmatrix} \quad (337)$$

$$A'P + PA = \begin{bmatrix} -1,3918 & 0,1475 & -0,0384 \\ 0,1475 & -1,8108 & 0,0260 \\ -0,0384 & 0,0260 & -1,5766 \end{bmatrix} \quad (338)$$

Obtendo-se os autovalores:

$$\lambda(A'P + PA) = \begin{bmatrix} -1,8622 \\ -1,5755 \\ -1,3415 \end{bmatrix} \quad (339)$$

como todos os autovalores de $A'P + PA$ são negativos, então a inequação está validada.

D.2 Propriedades

Definição D.2.1: O conjunto das possíveis soluções da LMI (312), $S := \xi \in \mathbb{R}^n : F(\xi) > 0$, é convexo, ou seja,

$$\xi_a, \xi_b \in S, \exists \alpha \in [0, 1] : \alpha F(\xi_a) + (1 - \alpha)F(\xi_b) > 0 \quad (340)$$

Definição D.2.2: O sistema de LMI's também é uma LMI:

$$F_1(\xi, \delta_1) > 0, \dots, F_k(\xi, \delta_k) > 0 \Rightarrow \mathbf{F}(\xi, \delta) := \text{diag}F_1(\xi, \delta_1), \dots, F_k(\xi, \delta_k) > 0 \quad (341)$$

Definição D.2.3: Uma LMI sujeita a uma restrição de igualdade pode ser reescrita como uma LMI somente:

$$\begin{cases} F(\xi) > 0 \\ G\xi - a = 0 \end{cases} \rightarrow \tilde{F}(\tilde{\xi}) > 0 \quad (342)$$

Definição D.2.4: O Complemento de *Schur* é definido como:

$$\begin{bmatrix} F_a(\xi) & F_b(\xi) \\ F_b(\xi)' & F_c(\xi) \end{bmatrix} \succ 0 \equiv \begin{cases} F_a(\xi) \succ 0 \\ F_c(\xi) - F_b(\xi)F_c(\xi)^{-1}F_b(\xi)' \succ 0 \end{cases} \quad (343)$$

$$\equiv \begin{cases} F_c(\xi) \succ 0 \\ F_a(\xi) - F_b(\xi)'F_c(\xi)^{-1}F_b(\xi) \succ 0 \end{cases} \quad (344)$$

Definição D.2.5: Restrições de Desigualdade (Procedimento S):

Sejam F_0, F_1, \dots, F_q formas quadráticas na variável $z \in \mathbb{R}^{n_z}$ com a seguinte estrutura:

$$F_i(z) = z'T_i z + 2u_i'z + v_i \quad (345)$$

$$i = 0, 1, \dots, q \quad (346)$$

onde $T_i = T_i' \in \mathbb{R}^{n_z \times n_z}$, $u_i \in \mathbb{R}^{n_z}$ e $v_i \in \mathbb{R}$.

A seguinte condição é satisfeita:

$$F_0(z) > 0, \forall z | F_i(z) \geq 0, i = 1, \dots, q \quad (347)$$

se existirem escalares positivos τ_i tais que

$$F_0(z) - \sum_{i=1}^q \tau_i F_i(z) > 0, \forall z \quad (348)$$

Definição D.2.6: Restrição de Igualdade (Lema de *Finsler*):

Seja $\sigma \in \mathbb{R}^{n_\sigma}$, $S = S' \in \mathbb{R}^{n_\sigma \times n_\sigma}$ e $Z \in \mathbb{R}^{n_\sigma \times n_\sigma}$, $m_\sigma < n_\sigma$. Então as seguintes sentenças são verdadeiras:

- $\sigma'S\sigma > 0$ para todo σ tal que $Z\sigma = 0$;
- $(Z^\perp)'SZ^\perp > 0$, onde Z^\perp é uma base para o espaço nulo de Z ;
- $\exists \mu$ tal que $S + \mu Z'Z > 0$;
- $\exists L \in \mathbb{R}^{n_\sigma \times m_\sigma}$ tal que $S + LZ + Z'L' > 0$.

D.3 Otimização

Em aplicações práticas de controle, não há interesse apenas na estabilidade de um sistema (autovalores de $F(\xi)$ negativos), deseja-se também outros requisitos de controle, tais como:

- A energia dos estados e do sinal de controle é mínima;
- Sinais externos indesejados influenciam o mínimo possível o sinal de saída de um sistema;
- Convergência rápida para o ponto de equilíbrio.

Assim sendo, não é suficiente encontrar uma solução para o conjunto de LMI's, esta solução deve otimizar alguma característica.

Definição D.3.1: Seja S o conjunto de soluções de uma LMI, uma das possíveis soluções minimiza uma determinada função escalar $f(\xi)$:

$$f_{otima} = \inf_{\xi \in S} f(\xi) = \inf\{f(\xi) : \xi \in S\} \quad (349)$$

Definição D.3.2: A função custo $f : S \rightarrow \mathbb{R}$ é dita ser convexa se S é convexo e para $\xi_1, \xi_2 \in S$ a seguinte relação é satisfeita:

$$f(\alpha\xi_1 + (1 - \alpha)\xi_2) \leq \alpha f \quad (350)$$

Definição D.3.2: A função custo $f : S \rightarrow \mathbb{R}$ é dita ser convexa se S é convexo e para $\xi_1, \xi_2 \in S$ a seguinte relação é satisfeita:

$$f(\alpha\xi_1 + (1 - \alpha)\xi_2) \leq \alpha f(\xi_1) + (1 - \alpha)f(\xi_2) \quad (351)$$

onde $\alpha \in (0, 1)$, $\xi_1 \neq \xi_2$.

Definição D.3.3: A função $f(\xi)$ é dita ser côncava se $-f(\xi)$ é convexa.

Definição D.3.4: Um problema de otimização convexa é da seguinte forma:

$$\min_{\xi \in S} f(\xi) : F(\xi) < 0 \quad (352)$$

Definição D.3.5: Seja $\gamma \in \mathbb{R}$, o objetivo é procurar o valor mínimo da função $f(\xi)$ sujeita a um limitante superior:

$$f(\xi) \leq \gamma, \forall \xi \in S \quad (353)$$

Definição D.3.6: Conjunto de nível de S :

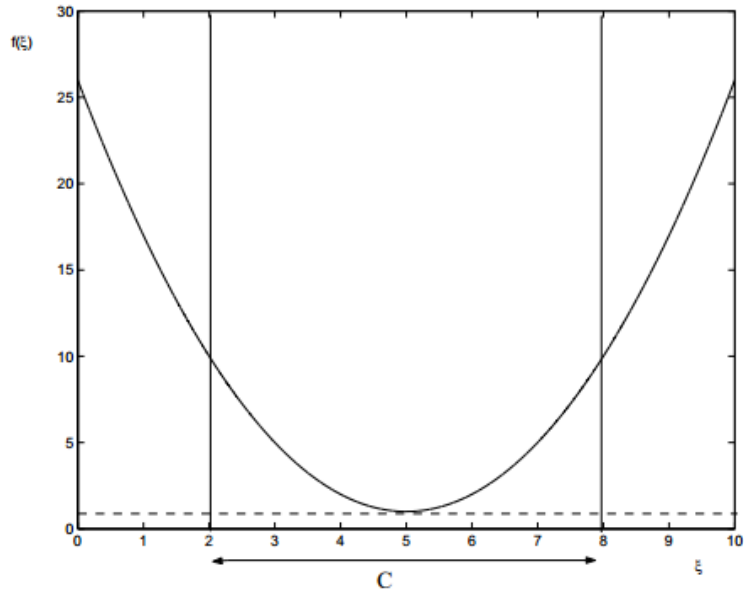


Figura 16: Função convexa $f(\xi)$

$$S_\gamma := \{\xi \in S : f(\xi) \leq \gamma\} \quad (354)$$

Definição D.3.7: Se $f : S \rightarrow \mathbb{R}$ é convexa então S_γ é convexo para todo $\gamma \in \mathbb{R}$.

D.4 Resolução Numérica de LMI's

A resolução de problemas de LMI's é obtida de forma numérica utilizando pacotes computacionais específicos.

A solução alcançada pode ser caracterizada de duas maneiras:

- Problema de factibilidade de LMI: busca por uma solução qualquer que satisfaça um conjunto de restrições na forma de LMI;
- Problema de otimização: determinar a solução ótima de uma função custo linear sujeita a um conjunto de restrições na forma de LMI.

Este projeto de pesquisa tratou-se de um problema de factibilidade de LMI: buscou-se as variáveis de decisão que definem o controlador PSS numa região factível.

E Solvers em Estudo

Este apêndice descreverá as funções básicas dos três solvers de LMI's utilizados neste projeto de pesquisa: **mincx**, **SeDuMi** e **SDPT3**. Antes disso, será informado os comandos básicos do *toolbox* LMILAB e da plataforma YALMIP (Löfberg, 2004) utilizadas para os três *solvers* utilizados.

E.1 LMILAB

LMILAB é um *toolbox* do MatLab que requer licença de uso e é aplicado na resolução de Desigualdades Matriciais Lineares (LMI's). Os *solvers* mais populares que utilizam o LMILAB são o **mincx**, **feasp** e **gevp**, diferenciando apenas em suas sintaxes. A seguir serão descritos seus comandos básicos.

E.1.1 Setlmis

Função utilizada para inicializar a criação de um novo sistema de LMI's. Ela elimina todas as LMI's definidas anteriormente. Ela sempre é utilizada para inicializar uma implementação de LMI's.

E.1.2 Lmivar

Comando responsável por definir uma variável matriz no sistema de LMI's a qual deverá ser determinada respeitando as desigualdades impostas. Sua sintaxe é:

$$[x, ndec, xdec] = \text{lmivar}(\text{tipo}, \text{struct}) \quad (355)$$

,
onde

tipo: estrutura de x :

- 1: bloco diagonal simétrico;
- 2: matriz retangular completa;
- 3: outra.

struct: dados adicionais da estrutura de x :

- $\text{tipo} = 1$: a linha de posição i da *struct* define o bloco diagonal de posição i de x .
 $\text{struct}(i,1)$: tamanho do bloco;
 $\text{struct}(i,2)$: tipo do bloco (0: bloco escalar, 1: bloco completo e -1: bloco de zeros).

- tipo = 2: `struct = [M,N]` se x é uma matriz $M \times N$
- tipo = 3: `struct` é uma matriz de mesma dimensão de x onde
`struct(i,j)=0` se $x(i,j) = 0$;
`struct(i,j)=+n` se $x(i,j) =$ variável de decisão de ordem n ;
`struct(i,j)=-n` se $x(i,j) =$ variável de decisão é (-1) vezes a ordem n .

x: identificador para a nova variável matriz;

ndec: número total de variáveis de decisão;

xdec: entradas dependentes de x nas variáveis de decisão.

Exemplo E.2.1:

a) Criar uma variável matriz simétrica completa de dimensão 2×2 :

$$x = \text{lmivar}(1, [2 \ 1]) \quad (356)$$

b) Criar uma variável matriz 2×3 :

$$x = \text{lmivar}(2, [2 \ 3]) \quad (357)$$

c) Criar uma variável matriz diagonal 3×3 :

$$x = \text{lmivar}(1, [3 \ 0]) \quad (358)$$

E.1.3 Lmiterm

Comando responsável por definir as desigualdades lineares do sistema de LMI's. Sua sintaxe é:

$$\text{lmiterm}(\text{termid}, A, B, \text{flag}) \quad (359)$$

termid: vetor com quatro entradas especificando a natureza e a localização da variável

- `termid(1) = +n`: lado esquerdo da n -ésima LMI é menor que o direito.
- `termid(1) = -n`: lado esquerdo da n -ésima LMI é maior que o direito.
- `termid(2:3)`: informa a posição (linha e coluna) na LMI no sistema de LMI's.

- `termid(4) = 0`: variável é um termo constante.
- `termid(4) = X`: variável (X) é da forma $A * X * B$.
- `termid(4) = -X`: variável (X) é da forma $A * X' * B$.

A: matriz com valores constantes que pré-multiplica a variável a se determinar.

B: matriz com valores constantes que pós-multiplica a variável a se determinar.

flag: especifica se $A * X * B + A * X' * B$ é um bloco diagonal, se sim, `flag = 's'`.

Exemplo E.3.1: Especificar a desigualdade: $A' * P + P * A < 0$ com P sendo a variável.

```
setlmis([])
P = lmivar(1,[2 1])
lmiterm([1 1 1 P],A',1,'s')
```

E.1.4 Getlmis

Comando responsável por obter todas as desigualdades matriciais definidas pelo `lmiterm()`.

Exemplo 6.4.1: Especificar a desigualdade: $A' * X + X * A < 0$ com P sendo a variável a se determinar.

```
setlmis([])
P = lmivar(1,[2 1])
lmiterm([1 1 1 P],A',1,'s')
lmisys = getlmis
```

E.1.5 Decnbr

Comando responsável por determinar o número de variáveis de decisão do sistema de LMI's.

E.1.6 Defcx

Comando que especifica a função objetivo do solver `mincx`. Sua sintaxe é

$$[v1, \dots, vk] = defcx(lmisys, j, x1, \dots, xk) \quad (360)$$

Há o retorno dos valores $v1, \dots, vk$ da matriz de variáveis com os identificadores $x1, \dots, xk$ quando as variáveis de decisão valem 1 e as outras 0.

E.1.7 Dec2mat

Comando que extrai o valor da variável de decisão após a resolução da desigualdade matricial. Sua sintaxe é:

$$X = \text{dec2mat}(l\text{misys}, \text{decvars}, \text{xid}) \quad (361)$$

O comando retorna o valor da matriz de variáveis de decisão com identificador *xid* e vetor de variáveis de decisão *decvars*.

E.2 YALMIP

YALMIP é um *toolbox* de distribuição gratuita para uso no MatLab na solução e modelagem de problemas avançados de otimização. Uma das características que o tornam vantajoso é a capacidade de usá-lo em conjunto com *solvers* externos. Atualmente o YALMIP trabalha com 20 *solvers* para diversos tipos de problema. Seus comandos básicos são:

E.2.1 SDPVAR

É utilizado para definir variáveis de decisão simbólicas do YALMIP.

- $x = \text{sdpvar}(n,m,'tipo','campo')$
- $x = \text{sdpvar}(\text{dim1},\text{dim2},\text{dim3},\dots,\text{dimn},'tipo','campo')$

Matriz quadrada: $P = \text{sdpvar}(n,n);$

Matriz simétrica: $P = \text{sdpvar}(n,n,'symetric');$

Matriz completa: $P = \text{sdpvar}(n,m,'full');$

Matriz com valores complexos: $P = \text{sdpvar}(n,m,'full','complex');$

Matriz diagonal: $X = \text{diag}(\text{sdpvar}(n,m));$

Hankel: $P = \text{hankel}(\text{sdpvar}(n,n));$

Toeplitz: $P = \text{toeplitz}(\text{sdpvar}(n,n))$

E.2.2 SET

É utilizado para definir as equações, inequações e restrições contidas no problema

$F = \text{set}([])$

$F = \text{set}(X)$

$F = \text{set}(X,'tag')$

Exemplo E.2.1:


```
P = sdpvar(n,n);
F = set(P>0) + set(P(:)>0);
```

E.2.3 SDPSETTINGS

É utilizado para configurar parâmetros dos solvers.

```
options = sdpsettings('campo',valor,'campo',valor,...);
```

Exemplo E.2.2:

```
options = sdpsettings('solver','sedumi','sedumi.eps',1e-12);
```

E.2.4 SOLVESDP

É a função mais comum para solucionar problemas de otimização.

$$diagnostics = solvesdp(F, h, options) \quad (362)$$

,

onde F são as equações, inequações e restrições do problema, h é a variável que deseja-se minimizar e $options$ são os parâmetros do solver que deseja-se alterar.

Exemplo E.2.3:

```
x = sdpvar(n,1);
```

```
F = set(A*x<b);
```

```
h = c'*x;
```

```
solvesdp(F,h);
```

Exemplo E.2.4: Dado um sistema dinâmico linear na forma $x' = Ax$, onde o objetivo é provar a estabilidade encontrando uma matriz simétrica P que satisfaça as seguintes inequações:

$$P > 0 \quad (363)$$

$$A'P + PA < 0 \quad (364)$$

,

onde a matriz A é dada. O seguinte código do MatLab foi implementado:

```
A = [-1 2 0;-3 -4 1;0 0 -2];
```

```
P = sdpvar(3,3);
```

```
F = set(P>0) + set(A'*P+P*A<0);
```

```
solvesdp(F)
```

Executado o código implementado obtém-se a seguinte matriz P :

$$P = \begin{bmatrix} 0,5603 & 0,0452 & -0,0070 \\ 0,0452 & 0,2490 & 0,0348 \\ -0,0070 & 0,348 & 0,4116 \end{bmatrix} \quad (365)$$

A fim de verificar os resultados, serão obtidos os autovalores da matriz P :

$$\lambda(P) = \begin{bmatrix} 0,2354 \\ 0,4187 \\ 0,5667 \end{bmatrix} \quad (366)$$

,
como todos autovalores de P são positivos, então a inequação (341) está validada. Para a inequação (342) deve-se obter o primeiro membro:

$$A'P + PA = \begin{bmatrix} -1 & -3 & 0 \\ -2 & -4 & 0 \\ 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} 0,5603 & 0,0452 & -0,0070 \\ 0,0452 & 0,2490 & 0,0348 \\ -0,0070 & 0,0348 & 0,4116 \end{bmatrix} + \quad (367)$$

$$+ \begin{bmatrix} 0,5603 & 0,0452 & -0,0070 \\ 0,0452 & 0,2490 & 0,0348 \\ -0,0070 & 0,348 & 0,4116 \end{bmatrix} \begin{bmatrix} -1 & -2 & 0 \\ -3 & -4 & 1 \\ 0 & 0 & -2 \end{bmatrix} \quad (368)$$

$$A'P + PA = \begin{bmatrix} -1,3918 & 0,1475 & -0,0384 \\ 0,1475 & -1,8108 & 0,0260 \\ -0,0384 & 0,0260 & -1,5766 \end{bmatrix} \quad (369)$$

Obtendo-se os autovalores:

$$\lambda(A'P + PA) = \begin{bmatrix} -1,8622 \\ -1,5755 \\ -1,3415 \end{bmatrix} \quad (370)$$

,
como todos os autovalores de $A'P + PA$ são negativos, então a inequação (342) está validada.

E.3 Mincx

Mincx é um *solver* de Desigualdades Matriciais Lineares cujo propósito é minimizar uma função objetivo linear. Este *solver* faz parte do pacote LMILAB.

Sintaxe

A sintaxe deste solver é:

$$[COPT, XOPT] = mincx(LMISYS, C, OPTIONS, XINIT, TARGET).$$

Este *solver* resolve o seguinte problema de otimização:

$$\begin{aligned} & \text{Minimizar } c' * x \\ & \text{Sujeito a } L(x) < R(x) \end{aligned}$$

onde x é o vetor de variáveis de decisão.

Entradas:

- *LMISYS*: descreve as restrições do sistema de LMI;
- *C*: vetor com o mesmo tamanho de x e utilizado na função objetivo;
- *OPTIONS*: entrada opcional que define a relação de exatidão requerida para a função objetivo, o número máximo de iterações, dentre outras funções;
- *XINIT*: entrada opcional que define os valores iniciais para x ;
- *TARGET*: entrada opcional que define o valor objetivo ($c' * x < TARGET$).

Saídas:

- *COPT*: mínimo global da função objetivo $c' * x$;
- *XOPT*: valor de minimização do vetor x das variáveis de decisão.

E.4 SeDuMi

SeDuMi é um *solver* de Desigualdades Matriciais Lineares disponibilizado pela plataforma YALMIP. É um *solver* de minimização/otimização de uma região definida por um cone homogêneo.

Sintaxe

A sintaxe deste *solver* é:

$$[X, Y, INFO] = sedumi(A, b, c, K, pars).$$

onde

- $[X, Y, INFO] = sedumi(A, b, 0)$ ou $sedumi(A, b)$ o objetivo do problema é encontrar $x \geq 0$ tal que $A * x = b$;

- $[X, Y, INFO] = sedumi(A, 0, c)$ ou $sedumi(A, c)$ o objetivo do problema é encontrar y tal que $A' * y \leq c$;
- K : define o valor mínimo para x , caso $K = 0, x \geq 0$;
- $pars$: define um conjunto de parâmetros para resolução do problema de minimização como a região de operação, a exatidão requerida, número máximo de interações, dentre outros.

E.5 SDPT3

SDPT3 é um *solver* de Desigualdades Matriciais Lineares disponibilizado pela plataforma YALMIP. É um *solver* de otimização de uma região linear quadrática (SQL do inglês, *Semidefinite-quadratic-linear*).

Sintaxe

A sintaxe deste *solver* é:

$$[obj, x, y, z, info, runhist] = sdpt3(blk, At, C, b, OPTIONS, x0, y0, z0).$$

onde

- blk : vetor que descreve a estrutura do bloco diagonal dos dados do SQL;
- At : vetor com $Atp = [svec(Ap1)...svec(Apm)]$;
- b, C : dados da instância SQL;
- $x0, y0, z0$: comando opcional para a inicialização dos vetores;
- $OPTIONS$: comando opcional para especificação de parâmetros de operação do *solver*;
- obj : $[< C, x >, < b, y >]$;
- x, y, z : solução ótima aproximada ou primal ou dual;
- $info$: fornece informações técnicas da solução do problema;
- $runhist$: fornece um histórico do problema.

Exemplo E.1: O exemplo a seguir envolve a resolução de sistema dinâmico linear na forma $x' = Ax$, onde o objetivo é provar a estabilidade encontrando uma matriz simétrica P (similar ao exemplo 6.2.4), mas fazendo uso dos três *solvers* individualmente. Primeiramente será mostrado o código fazendo uso do *solver* mincx na plataforma LMILAB e posteriormente os *solvers* SeDuMi e SDPT3 na plataforma YALMIP.

mincx na plataforma LMILAB

```

% =====
% Avaliando a estabilidade de Lyapunov do sistema de inequações.
%
% Encontrando X>0 tal que X*A+A'*X<0
%
% Sistema apresenta solução desde que seja estável
% =====
% Sistema e sua dimensão
A = [-1 5; 0 -2];
n = size(A,1);
% Criando um novo sistema de desigualdades matriciais setlmis([]);
% Criando uma matriz simétrica e simbólica X = lmvivar(1,[2 1]);
% Definindo as inequações de Lyapunov lmiterm([-1 1 1 X],1,1);
lmiterm([1 2 2 X],A',1,'s');
% Obtendo as desigualdades definidas lmisys = getlmis;
h = decnbr(lmisys);
c = zeros(h,1);
for j=1:h
[zj] = defcx(lmisys,j,X);
c(j) = trace(zj);
end
% Resolvendo as LMI's
tstart = cputime;
[copt,xopt] = mincx(lmisys,c,[]);
tfinal = cputime;
deltat = tfinal-tstart;
% Extraíndo o resultado da variável de decisão
X = dec2mat(lmisys,xopt,X);
A solução apresentada pelo solver mincx foi:

```

$$deltat = 0,1716s \tag{371}$$

$$X = \begin{bmatrix} 0,0700 & 0,0056 \\ 0,0056 & 0,3088 \end{bmatrix} \times 10^{-11} \tag{372}$$

os autovalores são:

$$\lambda(X) = \begin{bmatrix} 0,0699 \\ 0,3090 \end{bmatrix} \times 10^{-11} \tag{373}$$

$$\lambda(A'X + XA) = \begin{bmatrix} -0,1277 \\ -0,0042 \end{bmatrix} \times 10^{-10} \quad (374)$$

percebe-se que a teoria de Lyapunov foi cumprida: $X > 0$ e $A'X + XA < 0$.

SeDuMi na plataforma YALMIP

```

yalmip('clear')
clear all
close all
clc
% =====
% Avaliando a estabilidade de Lyapunov do sistema de inequações.
%
% Encontrando X>0 tal que X*A+A'*X<0
%
% Sistema apresenta solução desde que seja estável
% =====
% Sistema e sua dimensão
A = [-1 5; 0 -2];
n = size(A,1);
% Criando uma matriz simétrica e simbólica
X = sdpvar(n);
% Definindo as inequações de Lyapunov
ineqs = set(X*A+A'*X<0) + set(X>0);
% Escolhendo o solver e seus parâmetros de operação
disp('— solver SeDuMi do YALMIP —')
opts=sdpsettings;
opts.solver='sedumi';
opts.sedumi.eps=1e-5;
opts.sedumi.bigeeps=1e-5;
opts.verbose=0;
tstart = cputime;
% Encontrando a matriz X
solvesdp(ineqs,[],opts);
tfinal = cputime;
deltat = tfinal-tstart
% Avaliando a solução de X
Xsol = double(X)
Xsol_eig = eig(Xsol)
XAsol_eig = eig(double(X*A+A'*X))
A solução apresentada pelo solver SeDuMi foi:

```

$$\text{deltat} = 2,2776s \quad (375)$$

$$X = \begin{bmatrix} 0,3883 & 0,5267 \\ 0,5267 & 1,6203 \end{bmatrix} \quad (376)$$

os autovalores são:

$$\lambda(X) = \begin{bmatrix} 0,1939 \\ 1,8148 \end{bmatrix} \quad (377)$$

$$\lambda(A'X + XA) = \begin{bmatrix} -1,4185 \\ -0,5729 \end{bmatrix} \quad (378)$$

percebe-se que a teoria de Lyapunov foi cumprida: $X > 0$ e $A'X + XA < 0$.

SDPT3 na plataforma YALMIP

```

yalmip('clear')
clear all
close all
clc
% =====
% Avaliando a estabilidade de Lyapunov do sistema de inequações.
%
% Encontrando X>0 tal que X*A+A'*X<0
%
% Sistema apresenta solução desde que seja estável
% =====
% Sistema e sua dimensão
A = [-1 5; 0 -2];
n = size(A,1);
% Criando uma matriz simétrica e simbólica
X = sdpvar(n);
% Definindo as inequações de Lyapunov
ineqs = set(X*A+A'*X<0) + set(X>0);
% Escolhendo o solver e seus parâmetros de operação
%disp('— solver SDPT3 do YALMIP —')
%opts=sdpsettings;
%opts.solver='sdpt3';
%opts.sdpt3.eps=1e-5;
%opts.sdpt3.bigeps=1e-5;
%opts.verbose=0;
tstart = cputime;

```

```

% Encontrando a matriz X
solvesdp(ineqs,[],opts);
tfinal = cputime;
deltat = tfinal-tstart
% Avaliando a solução de X
Xsol = double(X)
Xsol_eig = eig(Xsol)
XAsol_eig = eig(double(X*A+A'*X))
A solução apresentada pelo solver SDPT3 foi:

```

$$deltat = 2,5428s \quad (379)$$

$$X = \begin{bmatrix} 3,2346 & 4,5811 \\ 4,5811 & 14,5162 \end{bmatrix} \quad (380)$$

os autovalores são:

$$\lambda(X) = \begin{bmatrix} 1,6087 \\ 16,1421 \end{bmatrix} \quad (381)$$

$$\lambda(A'X + XA) = \begin{bmatrix} -13,1384 \\ -5,5841 \end{bmatrix} \quad (382)$$

percebe-se que a teoria de Lyapunov foi cumprida: $X > 0$ e $A'X + XA < 0$.

Conclui-se pelo exemplo que os três solvers são capazes de resolver um sistema de inequações lineares (LMF's) diferenciando-se apenas nos parâmetros de operação o que afeta diretamente em seu desempenho.

F Código Implementado com o mincx

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Código mincx
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
format long
setlmis([])
clc

tstart = fix(clock);

npo = 11*5*8*2; % Numero de pontos de operacao

dados = open('modelos_mbinf.mat');

alfa = 0.1;
beta = 7.2079;
gama = 26.3157;
K = 0.000104;
zeta = 0.05;
teta = acos(zeta);

for k=1:1:npo
Aj(1,k).aj = dados.matrizes_ss.A{1,k};
end

Bj = [0 0 0 0 866.6667 0 0]';
Cj = [0 1 0 0 0 0 0];

Ac = [-alfa 0 0 0; gama-alfa*beta, -gama, 0, 0; beta*gama-alfa*beta*beta, gama-beta*gama,
-gama, 0; beta*beta*gama-alfa*beta*beta*beta, beta*gama-beta*beta*gama, gama-beta*gama,
-gama];
Bc = [1; beta; beta*beta; beta*beta*beta];
Cc = [0 0 0 K];

for k = 1:1:npo
A(1,k).a = [Aj(1,k).aj, Bj*Cc; Bc*Cj*Aj(1,k).aj, Ac];
end
```

```

P = lmivar(1,[11 1]);

lmiterm([-1 1 1 P],1,1);
for k=1:1:npo
lmiterm([k+1 1 1 P],A(1,k).a',1,'s');
end

lmisys = getlmis;
h = decnbr(lmisys);
c = zeros(h,1);
options=[1e-2 500 1e1 20 1];

for j=1:h
[zj] = defcx(lmisys,j,P);
c(j) = trace(zj);
end

[copt,xopt] = mincx(lmisys,c,options,[],1e-4);

P = dec2mat(lmisys,xopt,P);

iter = 0;

while (iter < 1)
iter = iter + 1
setlmis([]);

X = P(1:7,1:7);
Y = P(8:11,8:11);
Z = P(1:7,8:11);

sigma = lmivar(1,[1 1]);
I1 = eye(7);
I2 = eye(4);

[g1,n,sg1] = lmivar(2,[1 1]); % Variável gama

Ax1 = [0,0,0,0; sg1,-sg1,0,0; 0,sg1,-sg1,0; 0,0,sg1,-sg1];

[Ac1,n,sAc1] = lmivar(3,[Ax1]);
Ac2 = [1,0,0,0; 0,1,0,0; 0,beta,1,0; 0,beta*beta,beta,1];

```

Ac3 = [-alfa,0,0,0; -alfa*beta,0,0,0; -alfa*beta*beta,0,0,0; -alfa*beta*beta*beta,0,0,0];

[c1,n,sc1] = lmivar(2,[1 1]);

[Cc,n,sCc] = lmivar(3,[0,0,0,sc1]);

for k=1:1:npo

lmiterm([k 1 1 0],sin(teta)*(X*Aj(1,k).aj+Aj(1,k).aj'*X));
lmiterm([k 1 1 0],sin(teta)*(Z*Bc*Cj*Aj(1,k).aj+Aj(1,k).aj'*Cj'*Bc'*Z'));
lmiterm([k 1 1 sigma],1,-I1);
lmiterm([k 1 2 0],sin(teta)*(Aj(1,k).aj'*Z));
lmiterm([k 1 2 Ac1],sin(teta)*Z*Ac2,1);
lmiterm([k 1 2 0],sin(teta)*(Z*Ac3));
lmiterm([k 1 2 0],sin(teta)*(Aj(1,k).aj'*Cj'*Bc'*Y));
lmiterm([k 1 2 Cc],sin(teta)*X*Bj,1);
lmiterm([k 2 2 -Ac1],sin(teta),Ac2'*Y,'s');
lmiterm([k 2 2 0],sin(teta)*(Ac3'*Y+Y*Ac3));
lmiterm([k 2 2 -Cc],sin(teta),Bj'*Z,'s');
lmiterm([k 2 2 sigma],1,-I2);

lmiterm([k 1 3 0],cos(teta)*(Aj(1,k).aj'*X-X*Aj(1,k).aj));
lmiterm([k 1 3 0],cos(teta)*(Aj(1,k).aj'*Cj'*Bc'*Z'-Z*Bc*Cj*Aj(1,k).aj));
lmiterm([k 1 4 0],cos(teta)*(Aj(1,k).aj'*Z));
lmiterm([k 1 4 Ac1],(-1)*cos(teta)*Z*Ac2,1);
lmiterm([k 1 4 0],(-1)*cos(teta)*(Z*Ac3));
lmiterm([k 1 4 0],cos(teta)*(Aj(1,k).aj'*Cj'*Bc'*Y));
lmiterm([k 1 4 Cc],(-1)*cos(teta)*X*Bj,1);
lmiterm([k 2 3 0],(-1)*cos(teta)*(Z'*Aj(1,k).aj));
lmiterm([k 2 3 -Ac1],cos(teta),Ac2'*Z');
lmiterm([k 2 3 0],cos(teta)*(Ac3'*Z'));
lmiterm([k 2 3 0],(-1)*cos(teta)*(Y*Bc*Cj*Aj(1,k).aj));
lmiterm([k 2 3 -Cc],cos(teta),Bj'*X);
lmiterm([k 2 4 -Ac1],cos(teta),Ac2'*Y);
lmiterm([k 2 4 Ac1],(-1)*cos(teta)*Y*Ac2,1);
lmiterm([k 2 4 0],cos(teta)*(Ac3'*Y-Y*Ac3));
lmiterm([k 2 4 Cc],(-1)*cos(teta)*Z'*Bj,1);
lmiterm([k 2 4 -Cc],cos(teta),Bj'*Z);

lmiterm([k 3 3 0],sin(teta)*(X*Aj(1,k).aj+Aj(1,k).aj'*X));
lmiterm([k 3 3 0],sin(teta)*(Z*Bc*Cj*Aj(1,k).aj+Aj(1,k).aj'*Cj'*Bc'*Z'));
lmiterm([k 3 3 sigma],1,-I1);
lmiterm([k 3 4 0],sin(teta)*(Aj(1,k).aj'*Z));
lmiterm([k 3 4 Ac1],sin(teta)*Z*Ac2,1);

```

lmiterm([k 3 4 0],sin(teta)*(Z*Ac3));
lmiterm([k 3 4 0],sin(teta)*(Aj(1,k).aj'*Cj'*Bc'*Y));
lmiterm([k 3 4 Cc],sin(teta)*X*Bj,1);
lmiterm([k 4 4 -Ac1],sin(teta),Ac2'*Y,'s');
lmiterm([k 4 4 0],sin(teta)*(Ac3'*Y+Y*Ac3));
lmiterm([k 4 4 -Cc],sin(teta),Bj'*Z,'s');
lmiterm([k 4 4 sigma],1,-I2);

end

lmisys2 = getlmis;

h = decnbr(lmisys2);
c = zeros(h,1);
x0 = zeros(h,1);

c(1) = 1;

x0(1) = -2e-4;

options=[1e-3 500 1e3 10 1];

[copt,xopt] = mincx(lmisys2,c,options,x0,-1e-4);

sigma = dec2mat(lmisys2,xopt,sigma);
Ac1 = dec2mat(lmisys2,xopt,Ac1);
Cc = dec2mat(lmisys2,xopt,Cc);
Ac = Ac2*Ac1 + Ac3;
gama = Ac1(2,1);
K = Cc(4);

menor_damp = 10;
for k=1:1:np0
k;
A(1,k).a = [Aj(1,k).aj, Bj*Cc; Bc*Cj*Aj(1,k).aj, Ac];
[ww,zz,pp] = damp(A(1,k).a);
zz;
menorz = min(zz);
if menor_damp > menorz
menor_damp = menorz;
end

```

```

end

[sigma,menor_damp]

if sigma < 0
tfinal = fix(clock);
t1 = tstart(3)*24*3600 + tstart(4)*3600 + tstart(5)*60 + tstart(6);
t2 = tfinal(3)*24*3600 + tfinal(4)*3600 + tfinal(5)*60 + tfinal(6);
deltaT = t2 - t1;
break;
end
setlmis([]);

sigma = lmivar(1,[1 1]);

P = lmivar(1,[11 1]);

I = eye(11);

lmiterm([-1 1 1 P],1,1);
for k=1:1:npo
lmiterm([k+1 1 1 P],sin(teta)*A(1,k).a',1,'s');
lmiterm([k+1 1 1 sigma],1,-I);
lmiterm([k+1 1 2 P],cos(teta)*A(1,k).a',1);
lmiterm([k+1 1 2 P],(-1)*cos(teta),A(1,k).a);
lmiterm([k+1 2 2 P],sin(teta)*A(1,k).a',1,'s');
lmiterm([k+1 2 2 sigma],1,-I);
end

lmisys3 = getlmis;
h = decnbr(lmisys3);
c = zeros(h,1);
x0 = zeros(h,1);
c(1) = 1;
x0(1) = -4e-4;

options=[1e-3 500 1e3 10 1];

[copt,xopt] = mincx(lmisys3,c,options,x0,-1e-4);

sigma = dec2mat(lmisys3,xopt,sigma);

```

```

P = dec2mat(lmisys3,xopt,P);

menor_damp = 10;
for k=1:1:npo
[ww,zz,pp] = damp(A(1,k).a);
menorz = min(zz);
if menor_damp > menorz
menor_damp = menorz;
end
end

[sigma,menor_damp]

if sigma < 0
tfinal = fix(clock);
t1 = tstart(3)*24*3600 + tstart(4)*3600 + tstart(5)*60 + tstart(6);
t2 = tfinal(3)*24*3600 + tfinal(4)*3600 + tfinal(5)*60 + tfinal(6);
deltaT = t2 - t1;
break;
end

end

deltaT

```

G Código Implementado com o SeDuMi

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Código SeDuMi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
format long
clc

tstart = fix(clock);
npo = 11*5*8*2;

dados = open('modelos_mbinf.mat');

alfa = 0.1;
beta = 7.2079;
gama = 26.3157;
K = 0.000104;
zeta = 0.05;
teta = acos(zeta);

for k=1:1:npo
Aj(1,k).aj = dados.matrizes_ss.A{1,k};
end

Bj = [0 0 0 0 866.6667 0 0];
Cj = [0 1 0 0 0 0];

Ac = [-alfa 0 0 0; gama-alfa*beta, -gama, 0, 0; beta*gama-alfa*beta*beta, gama-beta*gama,
-gama, 0; beta*beta*gama-alfa*beta*beta*beta, beta*gama-beta*beta*gama, gama-beta*gama,
-gama];
Bc = [1; beta; beta*beta; beta*beta*beta];
Cc = [0 0 0 K];

for k = 1:1:npo
A(1,k).a = [Aj(1,k).aj, Bj*Cc; Bc*Cj*Aj(1,k).aj, Ac];
end

P = sdpvar(11,11);

ineq1 = set(P >= 0);
```

```

for k=1:1:npo
ineq1 = ineq1 + set(A(1,k).a'*P + P*A(1,k).a <= 0);
end

opts=sdpssettings;
opts.solver='sedumi';
opts.sedumi.eps=1e-4;
opts.sedumi.maxiter=500;
opts.radius=1e1;
opts.maxiter=500;
opts.verbose=0;

diag1=solvesdp(ineq1,[],opts);
P = double(P);

iter = 0;

while (iter < 1)
iter = iter + 1

P = double(P);
yalmip('clear')
gama = sdpvar(1);
K = sdpvar(1);

Ac = [-alfa 0 0 0; gama-alfa*beta, -gama, 0, 0; beta*gama-alfa*beta*beta, gama-beta*gama,
-gama, 0; beta*beta*gama-alfa*beta*beta*beta, beta*gama-beta*beta*gama, gama-beta*gama,
-gama];
Bc = [1; beta; beta*beta; beta*beta*beta];
Cc = [0 0 0 K];

for k = 1:1:npo
A(1,k).a = [Aj(1,k).aj, Bj*Cc; Bc*Cj*Aj(1,k).aj, Ac];
end

for k = 1:1:npo
M(1,k).m = [sin(teta)*(A(1,k).a'*P + P*A(1,k).a), cos(teta)*(A(1,k).a'*P - P*A(1,k).a);...
cos(teta)*((A(1,k).a'*P - P*A(1,k).a))', sin(teta)*(A(1,k).a'*P + P*A(1,k).a)];
end

sigma = sdpvar(1);

```



```

I = eye(22);

ineq2 = set(M(1,1).m - sigma*I <= 0);
for k=2:1:npo
ineq2 = ineq2 + set(M(1,k).m - sigma*I <= 0);
end

opts=sdpssettings;
opts.solver='sedumi';
opts.sedumi.eps=1e-4;
opts.sedumi.maxiter=500;
opts.radius=1e3;
opts.maxiter=500;
opts.verbose=0;

diag2=solvesdp(ineq2,sigma,opts);
sigma = double(sigma);
sol(iter,1) = sigma;

menor_damp = 10;
for k=1:1:npo
A(1,k).a = double(A(1,k).a);
[ww,zz,pp] = damp(A(1,k).a);
menorz = min(zz);
if menor_damp > menorz
menor_damp = menorz;
end
end

[sigma,menor_damp]

if sigma < 0
tfinal = fix(clock);
t1 = tstart(3)*24*3600 + tstart(4)*3600 + tstart(5)*60 + tstart(6);
t2 = tfinal(3)*24*3600 + tfinal(4)*3600 + tfinal(5)*60 + tfinal(6);
deltaT = t2 - t1;
break;
end

% Passo 2

gama = double(gama);

```

```

K = double(K);

yalmip('clear')
P = sdpvar(11,11);
sigma = sdpvar(1);

for k = 1:1:npo
M(1,k).m = [sin(teta)*(A(1,k).a'*P + P*A(1,k).a), cos(teta)*(A(1,k).a'*P - P*A(1,k).a);...
cos(teta)*((A(1,k).a'*P - P*A(1,k).a))', sin(teta)*(A(1,k).a'*P + P*A(1,k).a)];
end

ineq3 = set(P >= 0);
for k=1:1:npo
ineq3 = ineq3 + set(M(1,k).m - sigma*I <= 0);
end

opts=sdpssettings;
opts.solver='sedumi';
opts.sedumi.eps=1e-4;
opts.sedumi.maxiter=500;
opts.radius=1e3;
opts.maxiter=500;
opts.verbose=0;

diag3=solvesdp(ineq3,sigma,opts);
P = double(P);
sigma = double(sigma);
sol(iter,2) = sigma;

menor_damp = 10;
for k=1:1:npo
[ww,zz,pp] = damp(A(1,k).a);
menorz = min(zz);
if menor_damp > menorz
menor_damp = menorz;
end
end

[sigma,menor_damp]

if sigma < 0
tfinal = fix(clock);

```

```
t1 = tstart(3)*24*3600 + tstart(4)*3600 + tstart(5)*60 + tstart(6);
t2 = tfinal(3)*24*3600 + tfinal(4)*3600 + tfinal(5)*60 + tfinal(6);
deltaT = t2 - t1;
break;
end
end

deltaT
```


H Código Implementado com o SDPT3

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Código SDPT3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
format long
clc

tstart = fix(clock);
npo = 11*8*5*2;

dados = open('modelos_mbinf.mat');

alfa = 0.1;
beta = 7.2079;
gama = 26.3157;
K = 0.000104;
zeta = 0.05;
teta = acos(zeta);

for k=1:1:npo
Aj(1,k).aj = dados.matrizes_ss.A{1,k};
end

Bj = [0 0 0 0 866.6667 0 0];
Cj = [0 1 0 0 0 0 0];

Ac = [-alfa 0 0 0; gama-alfa*beta, -gama, 0, 0; beta*gama-alfa*beta*beta, gama-beta*gama,
-gama, 0; beta*beta*gama-alfa*beta*beta*beta, beta*gama-beta*beta*gama, gama-beta*gama,
-gama];
Bc = [1; beta; beta*beta; beta*beta*beta];
Cc = [0 0 0 K];

for k = 1:1:npo
A(1,k).a = [Aj(1,k).aj, Bj*Cc; Bc*Cj*Aj(1,k).aj, Ac];
end

P = sdpvar(11,11);

ineq1 = set(P >= 0);
```

```

for k=1:1:npo
ineq1 = ineq1 + set(A(1,k).a'*P + P*A(1,k).a <= 0);
end

opts=sdpssettings;
opts.solver='sdpt3';
opts.sdpt3.gaptol = 1e-7;
opts.sdpt3.maxit = 100;
opts.radius=1e1;
opts.maxiter=100;
opts.verbose=0;

diag1=solvesdp(ineq1,[],opts);
P = double(P);

iter = 0;

while (iter < 1)
iter = iter + 1

P = double(P);
yalmip('clear')
gama = sdpvar(1);
K = sdpvar(1);

Ac = [-alfa 0 0 0; gama-alfa*beta, -gama, 0, 0; beta*gama-alfa*beta*beta, gama-beta*gama,
-gama, 0; beta*beta*gama-alfa*beta*beta*beta, beta*gama-beta*beta*gama, gama-beta*gama,
-gama];
Bc = [1; beta; beta*beta; beta*beta*beta];
Cc = [0 0 0 K];

for k = 1:1:npo
A(1,k).a = [Aj(1,k).aj, Bj*Cc; Bc*Cj*Aj(1,k).aj, Ac];
end

for k = 1:1:npo
M(1,k).m = [sin(teta)*(A(1,k).a'*P + P*A(1,k).a), cos(teta)*(A(1,k).a'*P - P*A(1,k).a);...
cos(teta)*((A(1,k).a'*P - P*A(1,k).a))', sin(teta)*(A(1,k).a'*P + P*A(1,k).a)];
end

sigma = sdpvar(1);

```

```

I = eye(22);

ineq2 = set(M(1,1).m - sigma*I <= 0);
for k=2:1:npo
ineq2 = ineq2 + set(M(1,k).m - sigma*I <= 0);
end

opts=sdpssettings;
opts.solver='sdpt3';
opts.usex0 = 1;
opts.sdpt3.gaptol = 1e-5;
opts.sdpt3.maxit = 100;
opts.radius=1e3;
opts.maxiter=100;
opts.verbose=0;

assign(sigma,-1e-8);

diag2=solvedp(ineq2,sigma,opts);
sigma = double(sigma);
sol(iter,1) = sigma;

menor_damp = 10;
for k=1:1:npo
A(1,k).a = double(A(1,k).a);
[ww,zz,pp] = damp(A(1,k).a);
menorz = min(zz);
if menor_damp > menorz
menor_damp = menorz;
end
end

[sigma,menor_damp]

if sigma < 0
tfinal = fix(clock);
t1 = tstart(3)*24*3600 + tstart(4)*3600 + tstart(5)*60 + tstart(6);
t2 = tfinal(3)*24*3600 + tfinal(4)*3600 + tfinal(5)*60 + tfinal(6);
deltaT = t2 - t1;
break;
end

```

```

% Passo 2

gama = double(gama);
K = double(K);

yalmip('clear')
P = sdpvar(11,11);
sigma = sdpvar(1);

for k = 1:1:npo
M(1,k).m = [sin(teta)*(A(1,k).a'*P + P*A(1,k).a), cos(teta)*(A(1,k).a'*P - P*A(1,k).a);...
cos(teta)*((A(1,k).a'*P - P*A(1,k).a))', sin(teta)*(A(1,k).a'*P + P*A(1,k).a)];
end

ineq3 = set(P >= 0);
for k=1:1:npo
ineq3 = ineq3 + set(M(1,k).m - sigma*I <= 0);
end

opts=sdpssettings;
opts.solver='sdpt3';
opts.usex0 = 1;
opts.sdpt3.gaptol = 1e-5;
opts.sdpt3.maxit = 100;
opts.radius=1e3;
opts.maxiter=100;
opts.verbose=0;

assign(sigma,-1e-8);

diag3=solvedp(ineq3,sigma,opts);
P = double(P);
sigma = double(sigma);
sol(iter,2) = sigma;

menor_damp = 10;
for k=1:1:npo
[ww,zz,pp] = damp(A(1,k).a);
menorz = min(zz);
if menor_damp > menorz
menor_damp = menorz;
end
end

```



```
end
```

```
[sigma,menor_damp]
```

```
if sigma < 0
```

```
tfinal = fix(clock);
```

```
t1 = tstart(3)*24*3600 + tstart(4)*3600 + tstart(5)*60 + tstart(6);
```

```
t2 = tfinal(3)*24*3600 + tfinal(4)*3600 + tfinal(5)*60 + tfinal(6);
```

```
deltaT = t2 - t1;
```

```
break;
```

```
end
```

```
end
```

```
deltaT
```