

Análise e Síntese de
Controladores e Filtros Robustos
para Sistemas com Domínios Politópicos de Incerteza

Tese de Doutorado submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica.

por

Eduardo Nunes Gonçalves

Engenheiro Eletricista – UFMG

Mestre em Engenharia Elétrica – UFMG

Orientador: Prof. Dr. Reinaldo Martinez Palhares

Co-orientador: Prof. Dr. Ricardo Hiroshi Caldeira Takahashi

Setembro – 2006

Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica
Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica

Análise e Síntese de Controladores e Filtros Robustos para Sistemas com Domínios Politópicos de Incerteza

Autor: Eduardo Nunes Gonçalves

Orientador: Prof. Dr. Reinaldo M. Palhares

Co-orientador: Prof. Dr. Ricardo H. C. Takahashi

Tese de Doutorado submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Engenharia de Computação e Telecomunicações.**

Banca Examinadora

Alexandre Trofino Neto, Dr. DAS/UFSC
Leonardo Antônio Borges Tôrres, Dr. DELT/EE/UFMG
Pedro Luis Dias Peres, Dr. DT/FEEC/Unicamp
Reinaldo Martinez Palhares, Dr. DELT/EE/UFMG
Renato Cardoso Mesquita, Dr. DEE/EE/UFMG
Ricardo Hiroshi Caldeira Takahashi, Dr. Matemática/ICEx/UFMG

Belo Horizonte, MG

Setembro/2006

“All truths are easy to understand once they are discovered; the point is to discover them.”
- Galileo Galilei (1564-1642)

Agradecimentos

Gostaria de agradecer à toda minha família e aos colegas do CEFET/MG que proporcionaram todo o apoio, motivação e condições necessárias para o pleno desenvolvimento e finalização desta tese.

Sou infinitamente grato ao meu orientador, Prof. Reinaldo Martinez Palhares, por me aceitar, escolher o tema desta tese, orientar em todos os sentidos da palavra, ensinar e compartilhar seu imenso conhecimento na área de Controle Robusto e Otimização, sem o qual certamente este trabalho não poderia ser finalizado com o nível de qualidade desejado.

Não menos importante para a conclusão desta tese é a significativa participação do professor co-orientador, Prof. Ricardo Hiroshi Caldeira Takahashi, com comprovado conhecimento tanto na área de Otimização como na área de Controle Robusto, que junto com o Prof. Reinaldo, me concederam o privilégio de desenvolver e colocar em práticas suas brilhantes idéias.

Reconheço também a fundamental importância da contribuição do Prof. Renato Cardoso Mesquita na área de Geometria Computacional, que junto com as áreas de Controle Robusto e de Otimização compõem os três alicerces deste trabalho. Não tenho a menor dúvida de que os resultados obtidos neste trabalho são frutos da oportunidade e felicidade de poder reunir e integrar os conhecimentos dos três professores, profundos conhecedores de cada uma destas áreas.

Obrigado a todos os membros docentes, técnicos-administrativos e discentes do Programa de Pós-Graduação em Engenharia Elétrica da UFMG, em especial os professores para os quais tive o privilégio de ser aluno no Curso de Doutorado, seja regularmente matriculado ou como ouvinte: Antônio de Pádua Braga, Benjamin Rodrigues de Menezes, Fábio Gonçalves Jota, Hani Camille Yehia, João Antônio Vasconcelos, Luis Antonio Aguirre, Reinaldo Martinez Palhares, Ricardo Hiroshi Caldeira Takahashi e Walmir Matos Caminhas. Agradeço também ao Prof. Eduardo Mazoni Andrade Marçal Mendes, colega de graduação e mestrado, que, sempre que solicitado, prontamente prestou o apoio necessário.

Agradeço também aos demais membros das bancas examinadoras da Qualificação e da Tese de Doutorado, Prof. Alexandre Trofino Neto, Prof. João Manoel Gomes da Silva Jr., Prof. Leonardo Antônio Borges Tôres, Prof. Pedro Luis Dias Peres e Prof. Rodney Rezende Saldanha, por suas valiosas contribuições para o aprimoramento do trabalho realizado e do texto final desta tese.

Registro meus agradecimentos à professores de outras instituições, colegas de pós-graduação e de alunos da graduação em programas de Iniciação Científica, com os quais partilhei a elaboração de artigos com resultados apresentados nesta tese e algumas extensões dos mesmos: Adrian Nelson Vaisman Chasin, Prof. Cláudio Dias Campos, Saulo Benchimol Bastos e Prof. Petr Ya. Ekel.

Por fim, gostaria de agradecer em particular ao Prof. Benjamin Rodrigues de Menezes, que, na qualidade de coordenador do Programa de Pós-Graduação em Engenharia Elétrica da UFMG e meu orientador no Mestrado, orientou e intermediou o meu contato com o Prof. Reinaldo Martinez Palhares.

PARA MEUS PAIS, JOSÉ GONÇALVES (IN MEMORIAN)
E MARIA DE LOURDES,
MINHA ESPOSA MÁRCIA
E MINHAS FILHAS MARINA E TATIANA

Resumo

Esta tese propõe novas estratégias de análise e projeto $\mathcal{H}_2/\mathcal{H}_\infty$ de sistemas robustos, com posicionamento regional de pólos, aplicadas a sistemas lineares invariantes no tempo com domínios de incerteza politópicos. O método de análise de estabilidade robusta proposto combina condições suficientes por desigualdade matricial linear (LMI) e uma estratégia de partição de politopos. O método de análise de desempenho proposto é baseado na combinação do algoritmo branch-and-bound com formulações de análise LMI e permite o cálculo do custo \mathcal{H}_2 ou \mathcal{H}_∞ com qualquer precisão desejada. O método de projeto proposto é baseado em um problema de otimização em que os elementos do controlador, filtro ou modelo reduzido são os parâmetros de otimização e os objetivos e restrições de projeto são verificados em um conjunto finito de pontos, sendo o conjunto inicial formado pelos vértices do politopo, com a inclusão iterativa, quando necessário, de pontos interiores. O projeto é validado para todo o politopo através dos métodos de análise propostos. A motivação deste trabalho é a busca por um método de projeto que proporcione soluções menos conservadoras ou que atenda às restrições de estrutura ou desempenho para as quais não existem formulações LMI ou as formulações existentes não resultam em soluções factíveis. O procedimento de projeto proposto é testado em vários problemas de projeto de controladores, filtros e redução de modelos, resultando sempre em desempenho igual ou superior aos obtidos por outras estratégias publicadas na literatura.

Palavras-chave: Análise de Estabilidade Robusta, Cálculo de Custos \mathcal{H}_2 e \mathcal{H}_∞ , Controle Robusto Multiobjetivo, Filtragem Robusta Multiobjetivo, Redução de Modelos, Incertezas Politópicas, Otimização.

Abstract

This thesis proposes new strategies for analysis and synthesis of robust $\mathcal{H}_2/\mathcal{H}_\infty$ systems with regional pole placement, considering linear time-invariant systems with polytopic uncertainty domains. The proposed robust stability analysis approach combines linear matrix inequality (LMI) sufficient conditions and a polytope partition strategy. The proposed performance analysis approach is based on the combination of a branch-and-bound algorithm and LMI analysis formulations allowing the computation of the \mathcal{H}_2 or \mathcal{H}_∞ costs with any required accuracy. The proposed synthesis approach is based on an optimization problem where the controller, filter, or reduced model parameters are the optimization variables and the design objectives as well the constraints are verified in a finite set of points initialized with the polytope vertex set, with the iterative inclusion, when necessary, of interior points. The design is validated for the whole polytope by means of the proposed analysis approaches. The aim of this work is to seek for a design approach that provides less conservative solutions or even consider performance/structural constraints that are not handled by means of LMI formulations or are not feasible. The synthesis procedure is applied to several controller, filter, and model reduction problems achieving improved or the same performance than other strategies.

Keywords: Robust Stability Analysis, \mathcal{H}_2 and \mathcal{H}_∞ Cost Computation, Robust Multiobjective Control, Robust Multiobjective Filtering, Model Reduction, Polytopic Uncertainty, Optimization.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Acrônimos	xiii
Lista de Símbolos	xiv
1 Introdução	1
1.1 LMIs na teoria de controle robusto	1
1.2 Motivação	6
1.3 Objetivo	7
1.4 Escopo	7
1.5 Metodologia	9
1.6 Estrutura da tese	11
2 Partição de Politopos	13
2.1 Introdução	13
2.2 Definições básicas	15
2.3 Triangularização de politopos	16
2.4 Subdivisão de simplexos	17
2.5 Subdivisão de simplexo orientada pelas arestas	20
2.6 Conclusões	24
3 Análise de \mathcal{D}-Estabilidade Robusta	26
3.1 Introdução	26
3.2 Formulação do problema	28
3.3 Novo método de análise de \mathcal{D} -estabilidade robusta	32
3.3.1 Descrição do procedimento proposto	32
3.3.2 Complexidade do procedimento de análise proposto	34
3.4 Exemplos ilustrativos	35
3.4.1 Exemplos de análise de estabilidade robusta	35
3.4.2 Exemplos de análise de \mathcal{D} -estabilidade	47
3.5 Conclusões	55

4	Cômputo dos Custos \mathcal{H}_2 e \mathcal{H}_∞	57
4.1	Normas de sinais e sistemas	57
4.2	Cálculos de normas e custos	58
4.3	O Algoritmo Branch-and-Bound	61
4.4	O Algoritmo BnB Aplicado ao Cálculo dos Custos \mathcal{H}_2 e \mathcal{H}_∞ com a Precisão Requerida	63
4.4.1	Escolha das funções limitantes	63
4.4.2	Técnica de partição do politopo	64
4.5	Complexidade do Procedimento de Análise Proposto	65
4.6	Exemplos Ilustrativos de Análise de Sistemas Contínuos no Tempo	68
4.6.1	Cálculo do custo \mathcal{H}_2	68
4.6.2	Cálculo do custo \mathcal{H}_∞	70
4.7	Exemplos Ilustrativos de Análise de Sistemas Discretos no Tempo	75
4.7.1	Cálculo do custo \mathcal{H}_2	75
4.7.2	Cálculo do custo \mathcal{H}_∞	80
4.8	Conclusões	87
5	Procedimento Geral de Projeto	90
5.1	Introdução	90
5.2	Formulação do Problema	93
5.3	Procedimento de Projeto Proposto	94
5.3.1	Solução do Problema Auxiliar	96
5.3.2	Cálculo dos Custos	104
5.3.3	Verificação do Posicionamento Regional de Pólos	106
5.4	Conclusões	107
6	Projeto de Controladores	109
6.1	Introdução	109
6.2	Formulação do Problema	111
6.3	Exemplos Ilustrativos	115
6.3.1	Exemplos Ilustrativos de Controle por Realimentação de Estado	116
6.3.2	Exemplos Ilustrativos de Controle por Realimentação de Saída	122
6.4	Síntese de Controladores PID	134
6.4.1	Introdução	134
6.4.2	Formulação do Problema	135
6.4.3	Exemplos Ilustrativos	137
6.5	Conclusões	141
7	Projeto de Filtros	143
7.1	Introdução	143
7.2	Formulação do Problema	144
7.3	Exemplos Ilustrativos	147
7.4	Conclusões	156

8	Redução de Modelos	157
8.1	Introdução	157
8.2	Redução de Modelos Balanceada	159
8.3	Redução de Modelos Incertos	161
8.4	Procedimento de Projeto	162
8.5	Exemplos Ilustrativos	164
8.6	Conclusões	170
9	Conclusões Finais	172
9.1	Sumário das contribuições da tese	172
9.2	Comentários finais e perspectivas futuras	174
	Referências bibliográficas	177

Lista de Figuras

2.1	Triangularização de Delaunay.	17
2.2	Triangularizações que não são Delaunay.	17
2.3	Efeito do refinamento da partição de um triângulo pelo esquema de subdivisão por 2.	18
2.4	Efeito do refinamento da partição de um triângulo pelo esquema de subdivisão por 3.	19
2.5	Efeito do refinamento da partição de um triângulo em 4.	19
2.6	Exemplo de sub-tetraedro gerado pela partição de um tetraedro em 2^3 partes.	23
2.7	Exemplo de partição de um triângulo em 3^2 partes.	24
3.1	Regiões LMI.	30
3.2	Partição do domínio incerto (2-simplexo) para localizar um sistema instável.	43
3.3	Distribuição dos autovalores do sistema politópico aleatório que não é robustamente estável.	44
3.4	Partição de um domínio de incerteza 3-d (Ex. 3.3).	46
3.5	Lugar das raízes para $\gamma = 1,6029$ (Ex. 3.3).	47
3.6	Região de análise de D-estabilidade (Ex. 3.5).	48
3.7	Exemplo de distribuição dos pólos para $(n, N) = (5, 5)$ (Ex. 3.5).	49
3.8	Posicionamento de pólos para $A \in \mathcal{P}$ (Ex. 3.7).	54
4.1	Simulação do algoritmo BnB - iteração 0.	66
4.2	Simulação do algoritmo BnB - iteração 1.	66
4.3	Simulação do algoritmo BnB - iteração 2.	66
4.4	Simulação do algoritmo BnB - iteração 3.	67
4.5	Simulação do algoritmo BnB - iteração 4.	67
4.6	Simulação do algoritmo BnB - iteração 5.	67
4.7	Evolução das funções Φ_{li} e Φ_{ls} no cálculo do custo \mathcal{H}_2 (dO ₁₂ -P) (Ex. 4.1).	70
4.8	Partição do espaço dos parâmetros incertos no cálculo do custo \mathcal{H}_2 (dO ₁₂ -P) (Ex. 4.1).	71
4.9	Norma \mathcal{H}_2 para $p \in \Omega_p$ (Ex. 4.1).	71
4.10	Partição do espaço dos parâmetros incertos no cálculo do custo \mathcal{H}_∞ (dO ₁) (Ex. 4.2).	73
4.11	Norma \mathcal{H}_∞ para $p \in \Omega_p$ (Ex. 4.2).	73
4.12	Norma \mathcal{H}_∞ para $\alpha \in \Omega$ (Ex. 4.6).	84
4.13	Norma \mathcal{H}_∞ para $0,32 \leq \alpha_1 \leq 0,38$ e $0,34 \leq \alpha_2 \leq 0,40$ (Ex. 4.6).	84
4.14	Evolução das funções limitantes no cálculo do custo ϵ -garantido \mathcal{H}_∞ (Ex. 4.6).	85
4.15	Partição do politopo no cálculo do custo \mathcal{H}_∞ (dO _{L5}) para $\rho = 3,3$ (Ex. 4.7).	87

4.16	Evolução das funções limitantes no cálculo do custo \mathcal{H}_∞ (dO_{L5}) para $\rho = 3,3$ (Ex. 4.7). 87	
5.1	Simulação do procedimento de otimização pela inclusão progressiva de pontos de fixação.	92
5.2	Descrição do algoritmo elipsoidal.	99
5.3	Comparação do comportamento da função objetivo sem e com a fatoração UDU^T . . .	103
5.4	Comparação do comportamento da função objetivo sem e com a fatoração UDU^T . . .	103
6.1	Diagrama de blocos geral do sistema de controle em malha-fechada.	113
6.2	Curva tipo Pareto obtida com a função msf_{syn} e o procedimento proposto.	118
6.3	Posicionamento dos pólos em malha-fechada com \mathcal{K}_{pd} para k e f variando entre os valores limites.	119
6.4	Evolução da função objetivo no processo de otimização para o cálculo do controlador \mathcal{K}_p	120
6.5	Evolução dos parâmetros de otimização no cálculo do controlador \mathcal{K}_p	120
6.6	Subdivisão do espaço de incerteza após a terceira iteração (Ex. 6.2).	122
6.7	Evolução da norma \mathcal{H}_2 sobre a aresta $b_1 + b_2 = 1$ para cada iteração (Ex. 6.2). . . .	122
6.8	Curvas tipo Pareto para os controladores por realimentação de saída (Ex. 6.3).	125
6.9	Posicionamento regional de pólos (Ex. 6.6).	132
6.10	Diagrama de blocos da configuração PID-ISA.	135
6.11	Diagrama de blocos modificado do PID-ISA.	136
6.12	Resposta transitória de $y(t)$ nos oito vértices do politopo (Ex. 6.8).	138
6.13	Resposta transitória de $y(t)$ para os três vértices (Ex. 6.9).	140
6.14	Respostas transitórias de $y(t)$ para os três vértices do politopo para os sistemas instáveis em malha-aberta (Ex. 6.9).	141
7.1	Diagrama de blocos do problema de filtragem.	146
7.2	Evolução da superfície da norma \mathcal{H}_2 ao quadrado (Ex. 7.2).	151
7.3	Evolução do custo \mathcal{H}_2 (Ex. 7.2).	152
7.4	Partição do domínio de incerteza e localização das coordenadas de pior caso de norma \mathcal{H}_2 (Ex. 7.2).	152
7.5	Partição final do domínio de incerteza no cálculo do custo \mathcal{H}_2 (Ex. 7.3).	155
8.1	Diagramas de valores singulares máximos de $E(s, p)$ do modelo reduzido fixo para os vértices do politopo (Ex. 8.1).	165
8.2	Diagramas de valores singulares máximos de $E(s, p)$ do modelo reduzido dependente de parâmetros para os vértices do politopo (Ex. 8.1).	166
8.3	Diagramas de valores singulares de $E(z, p)$ com o modelo reduzido fixo (tracejado) e modelo reduzido incerto (linha sólida) para os vértices do politopo (Ex. 8.2).	168
8.4	Respostas ao impulso do modelo original (linha pontilhada), modelo fixo de ordem reduzida (linha tracejada), e modelo incerto de ordem reduzida (linha sólida) para $p = [0,2 \ 0,1]^T$ (Ex. 8.2).	169
8.5	Norma \mathcal{H}_∞ do erro de redução de $G_r(z, \alpha)$ na primeira (tracejada) e segunda (sólida) iterações do procedimento proposto (Ex. 8.3).	171

Lista de Tabelas

3.1	Número de simplexos gerados pela triangularização de Delaunay de um hipercubo. . .	34
3.2	Número de novos vértices e de sub-simplexos obtidos com a subdivisão de simplexo orientada pelas arestas.	35
3.3	Taxas de sucesso (%) para identificar a estabilidade robusta (LMIC) (Ex. 3.1)	37
3.4	Tempos médios (s) para identificar estabilidade robusta (LMIC) (Ex. 3.1)	38
3.5	Tempos médios (s) para identificar estabilidade robusta (SeDuMi) (Ex. 3.1)	38
3.6	Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.1)	39
3.7	Taxas de sucesso (%) para identificar a estabilidade robusta (LMIC) (Ex. 3.2)	40
3.8	Tempos médios (s) para identificar estabilidade robusta (LMIC) (Ex. 3.2)	41
3.9	Tempos médios (s) para identificar estabilidade robusta (SeDuMi) (Ex. 3.2)	42
3.10	Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.2)	42
3.11	Taxas de sucesso (%) para identificar a \mathcal{D} -estabilidade robusta (Ex. 3.5).	50
3.12	Tempos de processamento médio (s) (Ex. 3.5).	51
3.13	Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.6)	52
3.14	Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.6)	52
3.15	Desempenho do procedimento de análise de \mathcal{D} -estabilidade robusta (Ex. 3.7).	55
4.1	Custos ε -garantido \mathcal{H}_2 com precisão $\varepsilon = 0,001$ (Ex. 4.1).	70
4.2	Custos ε -garantido \mathcal{H}_∞ com precisão $\varepsilon = 0,001$ (Ex. 4.2).	72
4.3	Custo \mathcal{H}_∞ com precisão $\varepsilon = 0,1$ (Ex. 4.3).	75
4.4	Taxas de sucesso (%) para o cálculo do custo garantido \mathcal{H}_2 (Ex. 4.4).	77
4.5	“Erros” relativos médios (%) do custo garantido \mathcal{H}_2 (Ex. 4.4).	77
4.6	Tempos computacionais médios (s) para cálculo do custo garantido \mathcal{H}_2 (Ex. 4.4). . .	78
4.7	Custos \mathcal{H}_2 para $\varepsilon = 0,02$ (Ex. 4.5).	79
4.8	Taxas de sucesso (%) para calcular o custo garantido \mathcal{H}_∞ (Ex. 4.6).	81
4.9	“Erros” relativos (%) do custo garantido \mathcal{H}_∞ (Ex. 4.6).	81
4.10	Tempos computacionais médios (s) (Ex. 4.6).	82
4.11	Custos \mathcal{H}_∞ com precisão $\varepsilon = 0,01$ para $\rho = 3$ (Ex. 4.7).	85
4.12	Custos \mathcal{H}_∞ com precisão $\varepsilon = 0,01$ para $\rho = 3,3$ (Ex. 4.7).	86
4.13	Custo \mathcal{H}_∞ calculado pela formulação (OL $_{T_2}$) para $\rho = 3,3$ (Ex. 4.7).	86
6.1	Resultados de projeto do procedimento proposto (Ex. 6.1).	117
6.2	Resultados de projeto para os controladores por realimentação da saída (Ex. 6.3). . .	124
6.3	Resultados de projeto (Ex. 6.4).	128

6.4	Resultados de projeto para o problema misto (Ex. 6.5).	129
6.5	Valores dos parâmetros incertos nos quatro pontos de operação distintos (Ex. 6.6).	130
6.6	Resultados do projeto \mathcal{H}_2 (Ex. 6.6).	131
6.7	Resultados do projeto \mathcal{H}_∞ (Ex. 6.6).	131
7.1	Custo $\mathcal{H}_2, \delta_c^2$ (Ex. 7.1).	149
7.2	Custos $\mathcal{H}_2, \delta_c^2$ (Ex. 7.3).	154
7.3	Custos H_2, δ_c^2 (Ex. 7.4).	156

Acrônimos

BMI	- “Bilinear Matrix Inequality” - Desigualdade Matricial Bilinear
BnB	- “Branch-and-Bound”
CEA	- “Cone Ellipsoidal Algorithm” - Algoritmo Cone-Elipsoidal
FIR	- “Finite Impulse Response” - Resposta ao Impulso Finita
LFT	- “Linear Fractional Transformation” - Transformação Fracional Linear
LQG	- “Linear-Quadratic-Gaussian”
LMI	- “Linear Matrix Inequality” - Desigualdade Matricial Linear
LMIs	- “Linear Matrix Inequalities” - Desigualdades Matriciais Lineares
LMIC	- “LMI Control Toolbox”
LIT	- Sistema Linear Invariante no Tempo
MIMO	- “Multiple Input, Multiple Output” - Múltiplas Entradas, Múltiplas Saídas
PID	- Controlador Proporcional-Integral-Derivativo
QS	- “Quadratic Stability” - Estabilidade Quadrática
RMS	- “Root Mean Squared” - Raiz quadrada do valor médio ao quadrado
SDP	- “Semidefinite Programming” - Programação semidefinida
SeDuMi	- “Self-Dual-Minimization”
SISO	- “Single Input, Single Output” - Uma Entrada, Uma Saída

Lista de Símbolos

\triangleq	- igual por definição
\exists	- existe
\in	- pertence a
\subset	- está contido em
\cup	- união
\mathbb{N}	- conjunto dos números naturais, $\mathbb{N} = \{0, 1, 2, \dots, \infty\}$
\mathbb{Z}	- conjunto dos números inteiros, $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
\mathbb{R}	- corpo dos números reais
\mathbb{C}	- corpo dos números complexos
$\angle(\cdot)$	- ângulo do argumento
∇f	- gradiente (ou subgradiente) da função f
$ \lambda $	- valor absoluto do escalar λ
δ	- restrição de projeto para norma \mathcal{H}_2
δ_c	- custo ε -garantido \mathcal{H}_2
$\delta_{c.g.}$	- custo garantido \mathcal{H}_2
$\delta_{p.c.}$	- pior caso da norma \mathcal{H}_2 no conjunto Ω
$\tilde{\delta}_{p.c.}$	- pior caso da norma \mathcal{H}_2 no conjunto $\tilde{\Omega}$
ε	- precisão relativa usada como critério de parada no algoritmo elipsoidal
ε_δ	- precisão relativa usada no critério de parada do procedimento geral de projeto
\otimes	- produto Kronecker, $C = A \otimes B$ é uma matriz de blocos em que $C_{ij} = A_{ij}B$
I, I_d	- matriz identidade, matriz identidade de ordem $d \times d$
$\det(A)$	- determinante da matriz A
$\rho(A)$	- posto da matriz A
$\lambda(A)$	- espectro da matriz A , ou seja, conjunto dos autovalores da matriz A
$\lambda_i(A)$	- i -ésimo autovalor da matriz A
$\bar{\lambda}$	- valor máximo da variável escalar λ
$\underline{\lambda}$	- valor mínimo da variável escalar λ
γ	- restrição de projeto para norma \mathcal{H}_∞
γ_c	- custo ε - garantido \mathcal{H}_∞
$\gamma_{c.g.}$	- custo garantido \mathcal{H}_∞
$\gamma_{p.c.}$	- pior caso da norma \mathcal{H}_∞ no conjunto Ω
$\tilde{\gamma}_{p.c.}$	- pior caso da norma \mathcal{H}_∞ no conjunto $\tilde{\Omega}$
$\bar{\sigma}(A)$	- valor singular máximo da matriz A

α	-	vetor de coeficientes da combinação convexa ou de parâmetros incertos
$\alpha_{(2)}$	-	vetor de coordenadas correspondente ao valor $\delta_{p.c.}$
$\alpha_{(\infty)}$	-	vetor de coordenadas correspondente ao valor $\gamma_{p.c.}$
Ω_M	-	domínio de incerteza caracterizado pela combinação convexa de seus vértices
Ω_p	-	domínio de incerteza definido pelas restrições sobre os parâmetros incertos
Ω	-	domínio politópico de incerteza que pode representar Ω_M ou Ω_p
$\tilde{\Omega}$	-	conjunto finito de pontos do domínio politópico de incerteza
A^*	-	transposta complexa conjugada da matriz A
$A \succ 0$	-	matriz A é definida positiva ou $\lambda_i(A) > 0, \forall i$
$A \succeq 0$	-	matriz A é semi-definida positiva ou $\lambda_i(A) \geq 0, \forall i$
$A \prec 0$	-	matriz A é definida negativa ou $\lambda_i(A) < 0, \forall i$
$A \preceq 0$	-	matriz A é semi-definida negativa ou $\lambda_i(A) \leq 0, \forall i$
*	-	em matrizes simétricas, corresponde a termos simétricos em relação à diagonal
$\text{Co}\{\cdot\}$	-	envelope ou casca convexa (“convex-hull”) do argumento que lista os vértices do politopo
$\partial(\cdot)$	-	fronteira de uma região
$\text{int}(\cdot)$	-	interior de uma região
$\text{diag}(\cdot)$	-	matriz bloco-diagonal, sendo a seqüência de blocos dada pelo argumento
$E(\cdot)$	-	esperança matemática do argumento
$\text{Real}(\cdot)$	-	parte real do argumento
$\text{Imag}(\cdot)$	-	parte imaginária do argumento
$\mathcal{L}_2[0, \infty)$	-	espaço dos sinais contínuos de energia limitada, i.e., $\int_0^\infty f^T(t)f(t)dt < \infty$
$l_2[0, \infty)$	-	espaço dos sinais discretos de energia limitada, i.e., $\sum_0^\infty f^T(k)f(k) < \infty$
N_ϵ	-	número de iterações observado no critério de parada do algoritmo elipsoidal
$\mathcal{O}(\cdot)$	-	notação “Big O” utilizada para descrever complexidade computacional
\mathcal{P}	-	politopo de matrizes
$\left[\begin{array}{c c} A & B \\ \hline C & D \end{array} \right]$	-	realização da matriz de transferência $T = C(sI - A)^{-1}B + D$
T_2	-	matriz de transferência relacionada com o desempenho \mathcal{H}_2 do sistema
T_∞	-	matriz de transferência relacionada com o desempenho \mathcal{H}_∞ do sistema
$\ T\ _2$	-	norma \mathcal{H}_2 da matriz de transferência T
$\ T\ _\infty$	-	norma \mathcal{H}_∞ da matriz de transferência T
$\text{Tr}(A)$	-	traço da matriz A
$\text{Vert}(\cdot)$	-	conjunto de vértices do argumento (politopo)
$\text{Vol}(\cdot)$	-	hiper-volume do argumento
$\ w\ _2$	-	norma \mathcal{L}_2 do sinal contínuo w ou norma l_2 do sinal discreto w

Capítulo 1

Introdução

1.1 LMIs na teoria de controle robusto

Um dos grandes desafios do projeto de sistemas de controle e filtragem é garantir a estabilidade e o desempenho dos sistemas frente às incertezas inerentes aos modelos utilizados para análise e síntese. Tais incertezas podem ser decorrentes, por exemplo, de dinâmicas negligenciadas da planta, de não linearidades e de incertezas sobre parâmetros do sistema, que não são precisamente conhecidos ou que podem sofrer variações aleatórias. Existem várias formas de se modelar sistemas incertos. Esta tese trata especificamente de problemas de controle robusto considerando sistemas lineares invariantes no tempo representados por modelos no espaço de estados, nos quais o domínio de incerteza é um politopo, o que engloba modelos politópicos e modelos com dependência afim de parâmetros incertos. Tais modelos de incerteza são bastante abrangentes e interessantes por serem associados a aspectos físicos do sistema, podendo ser obtidos do conhecimento das equações dinâmicas do sistema ou por métodos de identificação. Esta tese aborda os seguintes problemas da teoria de controle robusto:

- análise de estabilidade robusta: verificar se todos os sistemas pertencentes ao domínio de incerteza são assintoticamente estáveis ou se todos os autovalores (ou pólos) estão localizados em uma determinada região convexa do plano complexo;
- análise de desempenho robusto: verificar se todos os sistemas pertencentes ao domínio de incerteza atendem às especificações de desempenho em termos de normas \mathcal{H}_2 e \mathcal{H}_∞ de matrizes de transferência, obtendo limitantes superiores das mesmas (custo garantido);
- síntese de controladores robustos: projetar controladores por realimentação de estado ou saída tal que o sistema em malha-fechada seja estável e atenda às especificações de desempenho para todos os sistemas no domínio de incerteza;

- síntese de filtros robustos: projetar filtros de tal forma que o erro entre as variáveis estimadas e seus valores reais seja minimizado ou atenda à uma determinada especificação para todos os sistemas no domínio de incerteza;
- aproximação robusta por modelos reduzidos: obter aproximações de ordem reduzida de sistemas incertos de modo que os erros entre as saídas do modelo reduzido e do sistema real sejam minimizados, ou atendam a uma determinada especificação de desempenho, para todos os sistemas no domínio de incerteza.

Uma das formas de caracterizar o desempenho de sistemas de controle em malha-fechada, de sistemas de filtragem e de aproximações de sistemas por modelos reduzidos é através de normas matriciais de certas matrizes de transferência do sistema. Normas matriciais, como as normas \mathcal{H}_2 e \mathcal{H}_∞ , proporcionam uma medida da influência das entradas exógenas (distúrbios de carga, ruídos, sinais de referência etc.) sobre as saídas controladas do sistema (erros de rastreamento, sinais de controle etc.). Controladores LQG (do inglês, "Linear-Quadratic-Gaussian"), bastante difundidos na década de 60, podem ser vistos como um caso especial do controle ótimo \mathcal{H}_2 . Devido à característica do projeto LQG de não assegurar margens de estabilidade satisfatórias (Doyle, 1978), além da dificuldade de se conhecer as propriedades estatísticas das perturbações, cresceu o interesse pelo controle ótimo \mathcal{H}_∞ a partir do trabalho de Zames (1981). Em Doyle et al. (1989), soluções para os problemas de controle \mathcal{H}_2 e \mathcal{H}_∞ foram formuladas para sistemas no espaço de estados através de equações de Riccati. Entretanto, formulações em termos de equações de Riccati podem se tornar difíceis de serem aplicadas aos problemas de controle robusto.

A teoria de controle robusto foi bastante impulsionada a partir do final dos anos 80 quando os problemas de controle robusto passaram a ser caracterizados por problemas de otimização convexa. Um problema de otimização convexa é um problema na forma:

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{sujeito a} && f_i(x) \leq b_i, \quad i = 1, \dots, m, \end{aligned} \tag{1.1}$$

sendo as funções $f_0, \dots, f_m : \mathbb{R}^d \rightarrow \mathbb{R}$ convexas, isto é, satisfazem a condição

$$f_i(\lambda x + (1 - \lambda)y) \leq \lambda f_i(x) + (1 - \lambda)f_i(y) \tag{1.2}$$

para todo $x, y \in \mathbb{R}^d$ e para todo $0 \leq \lambda \leq 1$. Os problemas de otimização convexa possuem a propriedade de que um ótimo local corresponde ao ótimo global. Na teoria de controle robusto, os problemas são formulados como problemas de programação semi-definida (SDP, do inglês "Semidefinite Programming"), uma classe dos problemas de otimização convexa, na qual a função objetivo

é linear e as restrições são na forma de desigualdades matriciais lineares (LMIs, do inglês “Linear Matrix Inequalities”) (Vendenbergue e Boyd, 1996):

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{sujeito a} && F(x) \succ 0 \end{aligned} \quad (1.3)$$

sendo a desigualdade matricial linear, ou LMI, definida como

$$F(x) \triangleq F_0 + \sum_{i=1}^d x_i F_i \succ 0 \quad (1.4)$$

Os dados do problema são o vetor $c \in \mathbb{R}^d$ e $d + 1$ matrizes simétricas fixas $F_0, \dots, F_d \in \mathbb{R}^{n \times n}$, sendo $x = [x_1, \dots, x_d]^T$ o vetor de variáveis de decisão. O sinal da desigualdade em $F(x) \succ 0$ significa que $F(x)$ é definida positiva, isto é, $z^T F(x) z > 0$, para todo $z \in \mathbb{R}^n$, com $z \neq 0$. A função $F(x)$ é uma função afim dos elementos em x . Uma propriedade importante das LMIs é que elas definem uma restrição convexa, ou seja, o conjunto de soluções x que atende à restrição é convexo. Um conjunto \mathcal{C} é convexo se a linha entre quaisquer dois pontos do conjunto está completamente contida no conjunto: $x_1, x_2 \in \mathcal{C} \Rightarrow \lambda x_1 + (1 - \lambda)x_2 \in \mathcal{C}$ para $0 \leq \lambda \leq 1$.

Uma das vantagens imediatas da abordagem por LMIs é a disponibilidade de softwares, comerciais ou gratuitos, para a solução em tempo polinomial de tal classe de problemas de otimização como, por exemplo, o LMI Control Toolbox (Gahinet et al., 1995) e SeDuMi Interface (Peaucelle, Henrion, Labit e Taitz, 2002; Sturm, 1999), ambos para uso com o MATLAB[®] (ver outras referências em Vandenberghe e Balakrishnan (1997)). A princípio, caracterizar um problema de controle ou filtragem na forma de um problema baseado em LMIs significa na prática a sua solução, mesmo que não existam soluções analíticas disponíveis (Boyd et al., 1994). Pode-se dizer que as formulações por LMIs possuem a mesma importância na teoria de controle robusto que as equações de Lyapunov e Riccati na teoria de controle moderna e as ferramentas gráficas, como os diagramas de Bode, Nyquist e Nichols, na teoria de controle clássica (Doyle et al., 1991). Representar todas as especificações do sistema sendo projetado através de um único objetivo de controle é bastante restritivo. Deste modo, outra vantagem importante da abordagem por LMIs é que, como elas aparecem na forma de restrições no problema de projeto, elas oferecem maior flexibilidade para combinar várias especificações sobre o sistema a ser projetado (Boyd et al., 1994). A terceira característica de destaque dos problemas formulados em termos de LMIs, fundamental quando os sistemas considerados possuem domínio politópico de incerteza, é que é suficiente verificar o conjunto de vértices para garantir as especificações de estabilidade e desempenho para todos os sistemas pertencentes ao politopo.

A história das LMIs na análise de sistemas dinâmicos começou há mais de cem anos, em 1890, quando Lyapunov publicou o trabalho introduzindo o que é conhecido atualmente como teoria de Lyapunov (Boyd et al., 1994). Pela teoria de Lyapunov, a equação diferencial

$$\frac{d}{dt}x(t) = Ax(t) \quad (1.5)$$

é estável (isto é, todas as trajetória tendem a zero) se e somente se existe uma matriz definida positiva $P \succ 0$ tal que

$$A^T P + P A \prec 0 \quad (1.6)$$

Esta condição é conhecida como desigualdade de Lyapunov, sendo uma LMI. Na teoria de controle robusto, o usual é encontrar as variáveis na forma de matrizes, como no exemplo da desigualdade de Lyapunov (1.6). O termo LMI foi introduzido em Willems (1971) para referir-se a

$$F(K) = \begin{bmatrix} A^T K + K A + Q & K B + C^T \\ B^T K + C & R \end{bmatrix} \succeq 0$$

Neste trabalho, Willems refere-se à equação de Riccati como um gargalo da teoria de sistemas lineares e antevê que, apesar da importância da LMI parecer não ser apreciada àquela época, seria interessante explorar sua capacidade por meio de algoritmos computacionais.

Existe uma gama razoável de especificações que podem ser caracterizadas por LMIs (Boyd et al., 1994; Scherer et al., 1997; Skelton et al., 1998). Existem diferentes caracterizações LMI de estabilidade robusta de sistemas incertos lineares invariantes no tempo a tempo contínuo (de Oliveira, Geromel e Hsu, 1999; Ebihara e Hagiwara, 2002a; Ramos e Peres, 2002; Ebihara et al., 2005; Oliveira e Peres, 2005c; Oliveira e Peres, 2006) ou a tempo discreto (de Oliveira, Bernussou e Geromel, 1999; de Oliveira, Geromel e Hsu, 1999; Ramos e Peres, 2001; Kau et al., 2005; Oliveira e Peres, 2005c; Oliveira e Peres, 2006) ou, de forma mais geral, de posicionamento de pólos em regiões convexas do plano complexo, denominada \mathcal{D} -estabilidade (Chilali e Gahinet, 1996; Peaucelle et al., 2000; Leite e Peres, 2003; Gao e Xue, 2004). As normas \mathcal{H}_2 e \mathcal{H}_∞ e os correspondentes custos garantidos \mathcal{H}_2 e \mathcal{H}_∞ (limitantes superiores das normas no domínio de incerteza) podem ser caracterizadas por LMIs (Palhares et al., 1997; de Oliveira et al., 2002; de Oliveira et al., 2004a; de Oliveira et al., 2004b; Ebihara e Hagiwara, 2004a; Trofino et al., 2005; Xie et al., 2004; He et al., 2005; Ebihara et al., 2005; Oliveira e Peres, 2005b; Yang et al., 2005).

Exemplos de problemas de controle robusto formulados em termos de LMI incluem a síntese de controladores robustos por realimentação de estado (Bernussou et al., 1989; Chilali e Gahinet, 1996; Peaucelle et al., 2000; Apkarian et al., 2001; Ebihara e Hagiwara, 2002a; Ebihara e Hagiwara, 2002b; de Oliveira et al., 2002; Wang e Wilson, 2003; Ebihara et al., 2004; Coutinho et al., 2005), síntese de

controladores dinâmicos de ordem completa para sistemas precisamente conhecidos (Scherer, 1995; Chilali e Gahinet, 1996; Scherer et al., 1997; Masubuchi et al., 1998; Apkarian et al., 2001; Trofino, 2002; de Oliveira et al., 2002; Ebihara et al., 2004; Ebihara e Hagiwara, 2004a) e síntese de filtros de ordem completa para sistemas incertos (Geromel et al., 1998; Geromel, 1999; Geromel e de Oliveira, 2001; Palhares e Peres, 1999; Palhares e Peres, 2000b; Palhares e Peres, 2000a; Palhares e Peres, 2001; Tuan et al., 2001; Geromel et al., 2002; Liu et al., 2003; Xie et al., 2004; Barbosa et al., 2005). Porém, um importante grupo de problemas de controle não possui formulações convexas baseadas em LMIs, não sendo possível a obtenção da solução diretamente a partir de um programa para solução de LMIs. Tais problemas de controle podem recair em descrições na forma de desigualdade matricial bilinear (BMI, do inglês “Bilinear Matrix Inequality”). Este é o caso do projeto de controladores por realimentação de saída estáticos, controladores por realimentação de saída dinâmicos de ordem reduzida, projeto de controladores dinâmicos robustos, com ou sem restrições de estrutura e projeto de filtros de ordem reduzida. As BMIs foram introduzidas na área de controle por Safonov et al. (1994). Uma BMI é da forma:

$$F(x, y) = F_0 + \sum_{i=1}^d x_i F_i + \sum_{j=1}^h y_j G_j + \sum_{i=1}^d \sum_{j=1}^h x_i y_j H_{ij} \succ 0 \quad (1.7)$$

sendo $x \in \mathbb{R}^d$ e $y \in \mathbb{R}^h$ as variáveis e $F_i \in \mathbb{R}^{n \times n}$, $G_j \in \mathbb{R}^{n \times n}$ e $H_{ij} \in \mathbb{R}^{n \times n}$, $i = 1, \dots, d$, $j = 1, \dots, h$, as matrizes simétricas fixas. Observe que, fixando x , a BMI passa a ser uma LMI da variável y e vice-versa. Apesar de serem convexas para x ou para y , as BMIs não são convexas para x e y simultaneamente. Deste modo, não são conhecidos algoritmos com tempo polinomial para solução dos problemas BMI. Os algoritmos para solução de problemas na área de controle robusto na forma de BMIs são, em geral, ou métodos de otimização local alternando entre minimizar x e y (Grigoriadis e Skelton, 1994; El Ghaoui e Balakrishnan, 1994; Iwasaki e Skelton, 1995; Grigoriadis e Skelton, 1996; El Ghaoui et al., 1997; Iwasaki, 1999; Ebihara et al., 2004; Kanev et al., 2004; Yang et al., 2005), ou métodos globais (branch-and-bound) (Goh et al., 1995; Goh, 1995; Beran et al., 1997; Beran, 1997; Tuan et al., 2000), ambos baseados na solução de uma seqüência de problemas LMI. O problema de redução de modelos também resulta em formulações não convexas em termos de BMIs tanto para sistemas precisamente conhecidos (Helmersson, 1994; Grigoriadis, 1995; Geromel et al., 2004; Ebihara e Hagiwara, 2004b; Kanno, 2005; Geromel et al., 2005) como para sistemas incertos (Assunção e Peres, 1999; Wu, 1996; Wu e Jaramillo, 2002; Trofino e Coutinho, 2004; Halevi e Shaked, 2004).

1.2 Motivação

Ao se caracterizar problemas da teoria de controle robusto na forma de problemas de otimização convexos, baseados em LMIs, em geral se introduz algum grau de conservadorismo na formulação, de modo que a solução obtida pode ser um sub-ótimo do problema original. Deste modo, a motivação inicial dessa tese foi: desenvolver um procedimento geral de projeto que proporcione resultados menos conservadores que as formulações LMI, ou que obtenha resultados em casos nos quais formulações LMI não apresentem soluções factíveis, que possa ser aplicado ao projeto de controladores robustos multiobjetivo por realimentação de estado, realimentação de saída estática ou dinâmica; ao projeto de filtros robustos multiobjetivo e ao cálculo de aproximações por modelos reduzidos, aplicados a sistemas lineares, invariantes no tempo, a tempo contínuo ou discreto, com domínios politópicos de incerteza. O termo multiobjetivo é empregado aqui para referir-se aos projetos que devem atender às especificações de desempenho \mathcal{H}_2 e \mathcal{H}_∞ e às restrições de posicionamento regional de pólos. A motivação para o projeto multiobjetivo pode ser colocada da seguinte forma (Scherer et al., 1997):

- O projeto considerando desempenho \mathcal{H}_∞ é conveniente para garantir estabilidade robusta de sistemas incertos, para rejeição de distúrbios na forma de sinais com energia limitada e para expressar especificações no domínio da frequência tais como faixa de passagem e ganho em baixas frequências.
- O projeto considerando desempenho \mathcal{H}_2 é equivalente ao controle ótimo LQG sendo útil para tratar de rejeição a distúrbios com características estocásticas, como ruídos de medição ou perturbações aleatórias.
- O posicionamento regional de pólos é útil para garantir determinadas características da resposta transitória, como o tempo de decaimento ou o amortecimento.

Como discutido na seção anterior, o controle LQG, desenvolvido nos anos 60, não consegue lidar de forma adequada com sistemas com modelos incertos. Pequenas variações do modelo baseado no espaço de estados podem levar a respostas transitórias totalmente diferentes, como mostrado em Skelton et al. (1998, pág. 34). Por outro lado, as técnicas de projeto \mathcal{H}_∞ são bastante adequadas para tratar de aspectos no domínio da frequência e de questões de robustez, porém, proporcionam pouco controle sobre o comportamento transitório e sobre a localização dos pólos em malha-fechada (Chilali e Gahinet, 1996). Fica claro, a necessidade de técnicas de controle multiobjetivo que combinem os aspectos vantajosos de cada critério.

O procedimento geral de projeto desenvolvido nesta tese, inicialmente direcionado ao projeto de controladores robustos, baseado em otimização não convexa diretamente no espaço de parâmetros do controlador, proporciona melhores resultados que justificam o esforço computacional demandado.

O procedimento de projeto desenvolvido requer ferramentas de análise robusta com determinadas características que não são proporcionadas pelas caracterizações LMI devido ao conservadorismo das mesmas. Deste modo, a segunda motivação desta tese foi a de desenvolver novos procedimentos de análise de estabilidade robusta e de desempenho robusto \mathcal{H}_2 e \mathcal{H}_∞ , menos conservadores, para serem aplicados ao procedimento geral de projeto proposto nesta tese. Devido aos bons resultados obtidos para os problemas de controle robusto, nesta tese, o procedimento geral de projeto também é aplicado ao problema de filtragem robusta e ao problema de aproximação robusta por modelos reduzidos.

1.3 Objetivo

O objetivo desta tese é propor novas estratégias de análise e síntese de sistemas de controle e filtragem robustos multiobjetivos, considerando critérios de desempenho \mathcal{H}_2 e \mathcal{H}_∞ e restrições de posicionamento regional de pólos. Através do procedimento geral de projeto proposto, busca-se, não só obter resultados menos conservadores do que os obtidos por formulações LMI baseadas na teoria de estabilidade de Lyapunov, como também obter soluções para os casos em que as formulações LMI não são factíveis ou, principalmente, para os casos em que ainda não existem formulações LMI. A estratégia de projeto proposta pode ser aplicada para síntese de controladores por realimentação de estado, realimentação estática ou dinâmica de saída, projeto de filtros e aproximação por modelos reduzidos. Uma importante propriedade do procedimento de projeto proposto é sua flexibilidade para incluir restrições de ordem e estrutura.

1.4 Escopo

As estratégias de análise e de projeto desenvolvidas nesta tese são destinadas a sistemas incertos lineares invariantes no tempo, a tempo contínuo ou discreto, com domínio politópico de incerteza. Considere o sistema linear invariante no tempo representado pelo seguinte modelo no espaço de estados:

$$\begin{aligned}\delta[x(\tau)] &= Ax(\tau) + Bw(\tau) \\ z(\tau) &= Cx(\tau) + Dw(\tau)\end{aligned}\tag{1.8}$$

com $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n_w}$, $C \in \mathbb{R}^{n_z \times n}$ e $D \in \mathbb{R}^{n_z \times n_w}$, sendo

$$\begin{aligned}\delta[x(\tau)] &\triangleq \frac{dx(t)}{dt}, \quad \tau \triangleq t \in \mathbb{R} \quad \text{para sistemas contínuos no tempo} \\ \delta[x(\tau)] &\triangleq x(k+1), \quad \tau \triangleq k \in \mathbb{N} \quad \text{para sistemas discretos no tempo}\end{aligned}$$

Seja S a matriz sistema definida como

$$S \triangleq \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (1.9)$$

Considere que a matriz sistema não é precisamente conhecida, mas pertence a um conjunto poliédrico convexo fechado, ou politopo: $S \in \mathcal{P} \subset \mathbb{R}^{(n+n_z) \times (n+n_w)}$.

No caso de modelos politópicos, o conjunto \mathcal{P} é um politopo no espaço de matrizes definido pelo conjunto de todas matrizes obtidas pela combinação convexa de seus N vértices:

$$\mathcal{P} \triangleq Co\{S_1, S_2, \dots, S_N\} = \left\{ S(\theta) : S(\theta) = \sum_{i=1}^N \theta_i S_i, \theta \in \Omega_M \right\} \quad (1.10)$$

sendo $Co\{\cdot\}$ a casca convexa do argumento e

$$S_i \triangleq \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \quad i = 1, \dots, N \quad (1.11)$$

os vértices do politopo. O vetor de coordenadas do politopo, $\theta = [\theta_1 \dots \theta_N]^T$, pertence ao conjunto definido como

$$\Omega_M \triangleq \left\{ \theta \in \mathbb{R}^N : \theta_i \geq 0, i = 1, \dots, N, \sum_{i=1}^N \theta_i = 1 \right\} \quad (1.12)$$

O conjunto Ω_M também pode ser representado como um simplexo no espaço de dimensão $N - 1$:

$$\Omega_M \triangleq \left\{ \hat{\theta} \in \mathbb{R}^{N-1} : \theta_i \geq 0, i = 1, \dots, N-1, \sum_{i=1}^{N-1} \theta_i \leq 1 \right\} \quad (1.13)$$

desde que $\theta_N = 1 - \sum_{i=1}^{N-1} \theta_i$.

No caso de sistemas dependentes de parâmetros, a matriz sistema é dependente afim do vetor de parâmetros incertos $p = [p_1, p_2, \dots, p_d]^T \in \mathbb{R}^d$:

$$\mathcal{P} \triangleq \{S(p) : S(p) = S_0 + p_1 S_1 + \dots + p_d S_d, p \in \Omega_p\} \quad (1.14)$$

Se os parâmetros incertos variam entre valores limites, ou seja, $p_i \in [\underline{p}_i, \bar{p}_i]$, sendo \underline{p}_i e \bar{p}_i os valores mínimo e máximo do i -ésimo parâmetro incerto, o vetor p pertence a um hiper-retângulo no espaço d -dimensional:

$$\Omega_p \triangleq \left\{ p \in \mathbb{R}^d : \underline{p}_i \leq p_i \leq \bar{p}_i, i = 1, \dots, d \right\} \quad (1.15)$$

O domínio Ω_p também pode ser um politopo de formato qualquer se existirem restrições lineares adicionais sobre os parâmetros, como, por exemplo, no estudo de falhas de sensores e atuadores ou em aproximações de incertezas limitadas em norma por incertezas politópicas:

$$\Omega_p \triangleq \{p \in \mathbb{R}^d : a_i^T p \leq b_i, i = 1, \dots, m\} \quad (1.16)$$

sendo a_i^T as linhas de uma matriz $\mathcal{A} \in \mathbb{R}^{m \times d}$ e b_i os elementos de um vetor $b \in \mathbb{R}^m$.

1.5 Metodologia

A grande maioria das pesquisas na área de projeto de sistemas de controle e filtragem robustos é baseada em problemas de otimização convexos com função objetivo linear e restrições em termos de LMIs, derivadas das formulações de estabilidade de Lyapunov. Estas abordagens de projeto são obtidas através de formulações de análise de estabilidade ou de cálculo de custos garantidos \mathcal{H}_2 e \mathcal{H}_∞ . Como uma forma alternativa de projeto, nesta tese é proposto formatar os problemas de controle, filtragem e aproximação em termos de um problema de otimização multiobjetivo, cujos parâmetros de otimização são os próprios parâmetros do controlador, filtro ou modelo reduzido. No caso de sistemas incertos, este problema requer a otimização do pior caso de um número infinito de sistemas pertencentes ao domínio de incerteza. Para viabilizar a solução deste problema, propõe-se que as funções objetivo e as restrições sejam verificadas em um conjunto finito de pontos, definido inicialmente pelos vértices do politopo que caracteriza o espaço de incertezas do modelo. A dificuldade de tal estratégia é que, uma vez que a formulação não é convexa, considerando apenas os vértices do politopo não se garante a efetiva minimização das funções objetivo e o atendimento das restrições em todo o politopo. Deste modo, é necessária uma validação do resultado obtido no processo de otimização para todo o politopo. Se na validação for verificado que o pior caso das funções objetivo ocorre fora dos vértices ou que alguma restrição não é atendida em todo o politopo, os pontos de pior caso são incluídos no conjunto finito de pontos avaliados e o processo de otimização é repetido. Os passos de otimização e validação são repetidos até que todas as restrições sejam atendidas e que as funções objetivo converjam para um valor com determinada precisão relativa. Deste modo, na estratégia proposta, busca-se minimizar o pior caso das normas \mathcal{H}_2 e \mathcal{H}_∞ no interior do politopo, ao invés de se minimizar os custos garantidos \mathcal{H}_2 e \mathcal{H}_∞ , como ocorre nas formulações LMI. Para implementação do procedimento de projeto proposto são necessários um algoritmo de otimização adequado para tratar de problemas não convexos e não diferenciáveis e um procedimento de análise que possa identificar os pontos de pior caso no domínio de incerteza com uma determinada precisão.

Poderiam ser utilizadas formulações de análise LMI para validar o projeto para todo o politopo,

entretanto, tais formulações provaram ser conservadoras para o caso de sistemas invariantes no tempo e, em muitos casos, não eram úteis para determinar o custo garantido por não obterem soluções factíveis. Deste modo, um novo problema teve que ser considerado, o de desenvolver uma ferramenta de análise eficiente que permitisse a determinação dos custos \mathcal{H}_2 e \mathcal{H}_∞ , sem conservadorismo, e que fornecesse o valor e a coordenada do pior caso das normas \mathcal{H}_2 e \mathcal{H}_∞ no politopo, atendendo a uma precisão pré-determinada. Para isso, a primeira tentativa foi utilizar um algoritmo genético para determinar o pior caso das normas \mathcal{H}_2 e \mathcal{H}_∞ em todo o politopo. Como este método não garante a convergência para o pior caso, foi necessário substituí-lo por um procedimento de cálculo com garantia de convergência para o máximo global. A solução escolhida para este problema foi adotar uma combinação do algoritmo branch-and-bound com formulações LMI de análise. A idéia básica deste algoritmo é a de “dividir para conquistar”. Ao se dividir o politopo, as formulações LMI produzem resultados menos conservadores. A possibilidade de determinação da precisão do cálculo é devida à característica do algoritmo branch-and-bound de utilizar duas funções que convergem para o valor ótimo, uma aproximando por valores inferiores e outra por valores superiores ao do máximo global. No cálculo dos custos \mathcal{H}_2 e \mathcal{H}_∞ , a função limitante inferior é escolhida como a norma calculada nos vértices do politopo e de suas partições e a função limitante superior é o custo garantido, calculado com formulações LMI, para o politopo e suas partições geradas por sucessivas divisões. Se o politopo é dividido sucessivamente até que o mesmo tenda a um ponto, o cálculo do custo garantido no politopo se iguala ao valor da norma no ponto, o que garante a convergência do algoritmo para o máximo global. O número de iterações necessárias do algoritmo é limitado pela precisão requerida. Esta estratégia de análise, que combina a redução do conservadorismo das formulações LMI pela partição do domínio de incerteza com uma discretização direcionada do domínio, também é utilizada na verificação do posicionamento regional de pólos.

Para implementar o procedimento de análise proposto, baseado no algoritmo branch-and-bound, foi necessário desenvolver uma técnica adequada de partição de politopos de qualquer formato, em qualquer dimensão, que garantisse a convergência do algoritmo branch-and-bound de forma eficiente. A técnica de partição adotada nesta tese considera malhas simpliciais, ou seja, a decomposição do politopo em um conjunto de simplexes, que torna o procedimento de análise proposto não só eficaz como também eficiente. Conhecendo o esforço computacional requerido por procedimentos de otimização global, em especial o algoritmo branch-and-bound, a eficiência obtida pela implementação adotada é bastante surpreendente, podendo os procedimentos de análise propostos serem considerados como ferramentas importantes para a teoria de controle robusto.

1.6 Estrutura da tese

O **Capítulo 2** apresenta a técnica proposta de partição de polítopos que é fundamental para a implementação eficiente dos procedimentos de análise de estabilidade robusta e de desempenho robusto propostos nesta tese. A partição de polítopos irá combinar a triangularização de Delaunay, utilizada para decompor polítopos de forma geral em um conjunto de simplexos (malha simplicial), com uma técnica de divisão de simplexo orientada pelas arestas, cujo algoritmo, uma importante contribuição desta tese, apresentado em detalhes na Seção 2.5, foi especialmente desenvolvido para a implementação dos procedimentos de análise propostos.

No **Capítulo 3** é apresentada a contribuição desta tese na área de análise de \mathcal{D} -estabilidade robusta. O procedimento de análise proposto, com a capacidade de identificar tanto a \mathcal{D} -estabilidade robusta do sistema incerto, como um caso de sistema que não é \mathcal{D} -estável pertencente ao domínio politópico de incerteza, é baseado na redução de conservadorismo de condições suficientes LMI de \mathcal{D} -estabilidade pela partição do domínio politópico de incerteza e em um método “inteligente” de grade. Exemplos ilustrativos, incluindo testes exaustivos, são apresentados para demonstrar a eficiência e a eficácia do procedimento de análise proposto.

O **Capítulo 4** apresenta a contribuição desta tese na área de análise de desempenho de sistemas com incertezas politópicas. É proposta uma nova estratégia de cálculo dos custos \mathcal{H}_2 e \mathcal{H}_∞ com uma precisão especificada, denominados custos ε -garantidos, baseada na combinação de formulações de análise LMI e no algoritmo branch-and-bound. Por esta estratégia é possível calcular o custo com qualquer precisão requerida, o pior caso da norma no espaço de incerteza e a coordenada do ponto de pior caso. Vários exemplos serão apresentados para demonstrar a eficácia do método de cálculo proposto.

O **Capítulo 5** trata da contribuição desta tese na área de síntese de controladores, filtros e modelos reduzidos. Neste capítulo é apresentada, em detalhes, a proposta de procedimento geral de projeto multiobjetivo $\mathcal{H}_2/\mathcal{H}_\infty$, com posicionamento regional de pólos, que será aplicada a síntese de controladores por realimentação de estado, controladores por realimentação estática de saída, controladores por realimentação dinâmica de saída (de ordem completa ou reduzida e com estrutura fixa ou não); síntese de filtros (de dimensão completa ou reduzida) e aproximação de modelos por outros de ordem reduzida, para sistemas lineares invariantes no tempo, contínuos ou discretos no tempo, com domínios politópicos de incerteza. A estratégia geral de projeto é baseada em um procedimento iterativo de dois passos: primeiro, o cálculo do controlador, filtro ou modelo reduzido, por algoritmo de otimização, diretamente no espaço dos parâmetros do controlador, filtro ou modelo reduzido, considerando um conjunto finito de pontos do polítopo e, segundo, validação do projeto para todo o polítopo utilizando os procedimentos de análise apresentados nos Capítulos 3 e 4, que determinam se existe a necessidade de acrescentar novos pontos no conjunto finito para nova rodada do procedimento.

Nos **Capítulos 6 e 7**, o procedimento geral de projeto é aplicado a diferentes problemas de controle e filtragem, sendo que cada capítulo apresenta vários exemplos ilustrativos, com comparação com as estratégias existentes, e conclusões específicas do capítulo.

O **Capítulo 8** descreve a aplicação do procedimento geral de projeto para obtenção de modelos reduzidos fixos ou dependentes de parâmetros. Exemplos extraídos da literatura na área são considerados para demonstrar a utilidade do procedimento proposto.

Finalmente, o último capítulo apresenta as conclusões gerais do trabalho até o atual estágio de pesquisa e propostas de desenvolvimento futuro.

Em cada capítulo é apresentada uma revisão bibliográfica específica sobre o assunto sendo tratado.

Capítulo 2

Partição de Politopos

2.1 Introdução

Algoritmos tipo branch-and-bound (BnB) podem ser aplicados a problemas de otimização global. A idéia básica destes algoritmos é considerar duas funções limitantes inferior e superior que convergem para o máximo (ou mínimo) global do problema à medida que o domínio é sucessivamente subdividido. Algoritmos BnB já foram utilizados em diferentes aplicações na área de controle robusto, como, por exemplo: análise de estabilidade de sistemas politópicos (DeMarco et al., 1990), cálculo do grau de estabilidade mínimo de sistemas lineares dependente de parâmetros (Balakrishnan et al., 1991), cálculo do valor máximo da norma \mathcal{H}_∞ de sistemas lineares dependente de parâmetros (Balemi et al., 1991), cálculo do pior caso da covariância do estado de sistemas lineares com parâmetros incertos (Balakrishnan e Boyd, 1991), cálculo dos valores máximo e mínimo do ganho RMS de sistemas lineares discretos no tempo incertos (Balemi e Balakrishnan, 1992), síntese de controladores via BMI (Goh et al., 1995; Goh, 1995; Beran et al., 1997; Beran, 1997), projeto de estruturas flexíveis com objetivo de minimizar norma \mathcal{H}_∞ (Sipila et al., 1999), cálculo da norma mínima de modelos LFT em um conjunto estruturado (McCloskey et al., 2000) e para redução de modelos de sistemas incertos a tempo contínuo e discreto (Assunção e Peres, 1999). Um dos aspectos mais importantes na implementação de algoritmos BnB é a técnica de partição do domínio da função avaliada. Nos trabalhos citados, os modelos de sistema analisados possuem parâmetros incertos que variam dentro de determinados intervalos conhecidos. Em todos os casos, o algoritmo BnB é desenvolvido para tratar especificamente da partição de domínios na forma de hiper-retângulos. A estratégia utilizada é a de dividir o hiper-retângulo pela metade, com o corte sendo feito nas arestas de maior dimensão (considerando variáveis normalizadas). Divisão de hiper-retângulos são fáceis de implementar com bom desempenho desde que elas sempre resultam em subdomínios com o mesmo formato. Entretanto, um problema muito mais difícil de ser tratado ocorre quando o domínio a ser particionado não

possui forma retangular. Além disso, em um espaço de dimensão d , um hiper-retângulo possui 2^d vértices, o que causa o rápido crescimento da complexidade computacional do problema de avaliar uma função nos vértices de tal objeto geométrico, à medida que cresce a dimensão do problema.

Como discutido na Seção 1.4, esta tese irá apresentar nos próximos capítulos técnicas de análise de estabilidade e desempenho baseadas em algoritmos tipo BnB para serem aplicadas a sistemas incertos representados tanto por modelos por dependência afim de parâmetros como por modelos politópicos. Deste modo, o método de partição de politopos, a ser empregado na implementação dos algoritmos tipo BnB, deve ser desenvolvido para tratar de três situações possíveis:

1. Modelos politópicos: simplexo com N vértices,

$$\Omega_M \triangleq \left\{ \hat{\theta} = [\theta_1 \dots \theta_{N-1}]^T \in \mathbb{R}^{N-1} : \theta_i \geq 0, i = 1, \dots, N-1, \sum_{i=1}^{N-1} \theta_i \leq 1 \right\}$$

2. Modelos dependentes de parâmetros nos quais os parâmetros podem variar entre limites conhecidos: hiper-retângulo com 2^d vértices que correspondem às combinações dos valores extremos dos d parâmetros incertos,

$$\Omega_p \triangleq \left\{ p = [p_1 \dots p_d]^T \in \mathbb{R}^d : \underline{p}_i \leq p_i \leq \bar{p}_i, i = 1, \dots, d \right\}$$

3. Modelos dependentes de parâmetros considerando restrições lineares adicionais sobre os d parâmetros incertos: politopo com formato qualquer:

$$\Omega_p \triangleq \left\{ p = [p_1 \dots p_d]^T \in \mathbb{R}^d : a_i^T p \leq b_i, i = 1, \dots, m \right\}$$

sendo a_i^T as linhas de uma matriz $\mathcal{A} \in \mathbb{R}^{m \times d}$ e b_i os elementos de um vetor $b \in \mathbb{R}^m$.

Em aplicações de engenharia, nas quais é necessária a partição de espaços, é comum trabalhar com dois tipos de formas geométricas: hiper-retângulos ou simplexos (Moore, 1992). Para poder tratar de politopos com qualquer formato, englobando os três casos citados, nesta tese, o algoritmo BnB é implementado considerando malhas simpliciais. Neste caso a estratégia empregada é a de decompor o domínio na forma de politopo em um conjunto de simplexos com os refinamentos posteriores sendo realizados por uma técnica de subdivisão de simplexos especialmente desenvolvida. Tal escolha é justificada pelas seguintes vantagens em considerar simplexos ao invés de hiper-retângulos:

1. qualquer politopo poder ser decomposto exatamente em um conjunto de simplexos, o que não é possível com hiper-retângulos;

2. um simplexo é o politopo mais simples existente, com o menor número de vértices necessário, isto é, em um espaço d -dimensional um simplexo possui $d + 1$ vértices ao passo que o hiper-retângulo possui 2^d vértices,
3. um d -simplexo é definido somente pela sua lista de $d+1$ vértices, não sendo necessária nenhuma informação sobre sua topologia, uma vez que qualquer subconjunto de d vértices forma uma face do simplexo.

A segunda vantagem será de especial interesse para aplicação do algoritmo BnB neste trabalho pois serão realizados cálculos por formulações de análise baseadas em desigualdade matriciais lineares (LMIs) cujos números de variáveis escalares de decisão e de linhas de restrições LMIs serão funções do número de vértices do politopo.

Para convergência eficiente de algoritmos tipo BnB, não basta que o hiper-volume seja reduzido pelo procedimento de subdivisão, uma vez que o volume pode tender a zero quando as arestas do politopo em apenas uma das dimensões tende a zero. O importante é que a máxima distância entre os vértices do politopo tenda a zero de modo que o politopo tenda para um ponto com a seqüência de subdivisões. Mesmo na bisseção de hiper-retângulos é necessário adotar estratégias para garantir este comportamento, como é o caso na bisseção da maior aresta em cada passo. Deste modo, a escolha da técnica de refinamento da malha simplicial deverá levar em consideração esta característica fundamental.

2.2 Definições básicas

As seguintes definições serão úteis na apresentação do método de partição de politopos proposto nesta tese.

Definição 2.1 Um conjunto $\mathcal{P} = \{x \in \mathbb{R}^n : \mathcal{A}x \leq b\}$ é um **poliedro** definido pela matriz $\mathcal{A} \in \mathbb{R}^{m \times n} = [a_1, \dots, a_m]^T$ e pelo vetor $b \in \mathbb{R}^m$. Se o poliedro é fechado então \mathcal{P} é denominado **politopo**. Um politopo também pode ser definido a partir de seu conjunto de vértices $\{v_1, \dots, v_N\}$: $\mathcal{P} = \text{Co}(v_1, \dots, v_N) = \{x \in \mathbb{R}^n : x = \sum_{i=1}^N \theta_i v_i, \theta_i \geq 0, i = 1, \dots, N, \sum_{i=1}^N \theta_i = 1\}$, sendo $\text{Co}(\cdot)$ a casca convexa do argumento.

Definição 2.2 Um conjunto de pontos $\{p_0, p_1, \dots, p_k\}$ é denominado **independente afim** quando, para um subscrito $i \in \{0, \dots, k\}$ arbitrário, porém fixo, o conjunto de vetores $\{p_j - p_i : j \in \{0, \dots, k\}, j \neq i\}$ é linearmente independente.

Definição 2.3 Um d -**simplexo** é um politopo no espaço d -dimensional definido por $d+1$ vértices que são afim independentes. Para cada dimensão d , um d -simplexo é o objeto geométrico mais simples

nessa dimensão. Desde modo, um 0-simplexo é um ponto, um 1-simplexo é um segmento de reta, um 2-simplexo é um triângulo, um 3-simplexo é um tetraedro etc.

Definição 2.4 Dois simplexos $\sigma, \sigma' \in \mathbb{R}^d$ são **congruentes** se existir um vetor de translação $v \in \mathbb{R}^d$, um escalar $c > 0$, e uma matriz ortogonal $Q \in \mathbb{R}^{d \times d}$ tal que $\sigma' \cong v + cQ\sigma$ (Bey, 2000).

Definição 2.5 Seja $\mathcal{P} \in \mathbb{R}^d$ um politopo, o conjunto $\{\mathcal{P}_1, \dots, \mathcal{P}_k\}$ de subpolitopos de \mathcal{P} é denominado uma **partição** de \mathcal{P} se $\bigcup_{i \in \{1, \dots, k\}} \mathcal{P}_i = \mathcal{P}$ e para todo $i, j, i \neq j$, $\text{int}(\mathcal{P}_i) \cap \text{int}(\mathcal{P}_j) = \emptyset$.

Definição 2.6 Uma **triangularização** de um politopo $\mathcal{P} \in \mathbb{R}^d$ é uma partição de \mathcal{P} em conjunto de d -simplexos cuja união é o politopo e a intersecção de qualquer dois simplexos do conjunto é uma face comum (possivelmente vazia).

2.3 Triangularização de politopos

Politopos de qualquer formato podem ser divididos exatamente em um conjunto de simplexos pelo processo conhecido como triangularização. Existem diferentes possibilidades de triangularização de um politopo, sendo interessante que os simplexos gerados sejam bem formatados: os ângulos entre as arestas não sejam muito pequenos ou muito grandes. Neste trabalho, no caso de modelos dependentes de parâmetros, a decomposição de Ω_p em um conjunto de simplexos é realizada pelo método de triangularização de Delaunay. No espaço bidimensional, o método de triangularização de Delaunay busca maximizar o ângulo mínimo entre as arestas para todas as triangulações possíveis. Um simplexo da triangularização de Delaunay tem a seguinte propriedade: ele determina uma hipersfera cujo interior não contém nenhum outro ponto do conjunto de pontos a não ser os $d + 1$ pontos que determinam o simplexo. Esta propriedade é ilustrada pela Figura 2.1 que apresenta uma triangularização de Delaunay de um politopo definido por 5 vértices no espaço bi-dimensional, na qual pode ser visto que os 3 vértices de cada triângulo definem um círculo que não contém nenhum dos demais pontos do conjunto. A Figura 2.2 apresenta outras possibilidades de triangularização do mesmo politopo para as quais a propriedade da triangularização de Delaunay não é atendida. Existe uma relação estreita entre a triangularização de Delaunay de um conjunto de pontos e a casca convexa da projeção em um parabolóide destes pontos em uma dimensão superior (“lifting transformation”) (de Berg et al., 2000). Deste modo, algoritmos para cálculo da casca convexa em espaços $(d + 1)$ -dimensional podem ser usados para calcular a triangularização de Delaunay no espaço d -dimensional de forma eficiente. Isto é utilizado, por exemplo, pela função `delaunayn(.)` do MATLAB[®] que é baseada no algoritmo Quickhull (Barber, 1996).

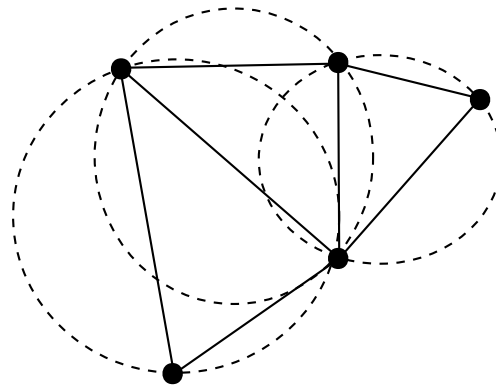


Figura 2.1: Triangulação de Delaunay.

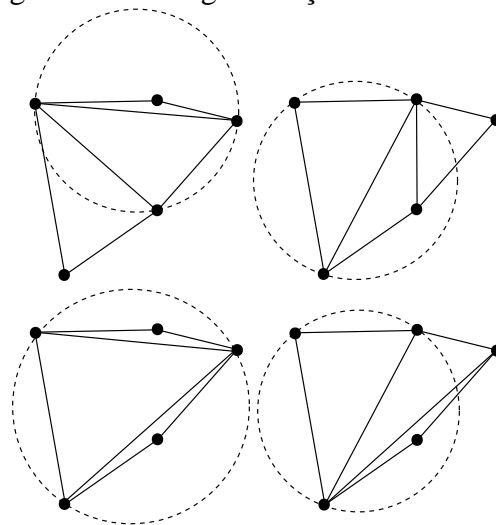


Figura 2.2: Triangulações que não são Delaunay.

2.4 Subdivisão de simplexos

Sendo Ω um simplexo ou tendo sido decomposto em um conjunto de simplexos pela triangulação de Delaunay, para implementar os refinamentos posteriores da malha simplicial é necessário escolher uma técnica apropriada de subdivisão de simplexo. Para que algoritmos tipo branch-and-bound converjam eficientemente, é necessário que a técnica de refinamento garanta que a maior distância entre dois vértices de cada simplexo seja minimizada a medida que os volumes dos simplexos sejam reduzidos pelas sucessivas subdivisões, sem gerar simplexos mal formatados. Para isso é necessário que a técnica de refinamento seja estável, o que significa que o número de classes congruentes de simplexos (Definição 2.4) obtidas pelas sucessivas subdivisões deva ser limitado.

Para se dividir um triângulo no espaço bi-dimensional podem ser consideradas, entre outras, três estratégias diferentes: (a) bisseção do triângulo ao meio, pela inclusão de um novo vértice sobre o ponto médio da aresta de maior comprimento; (b) dividir o triângulo em três, pela inclusão de um

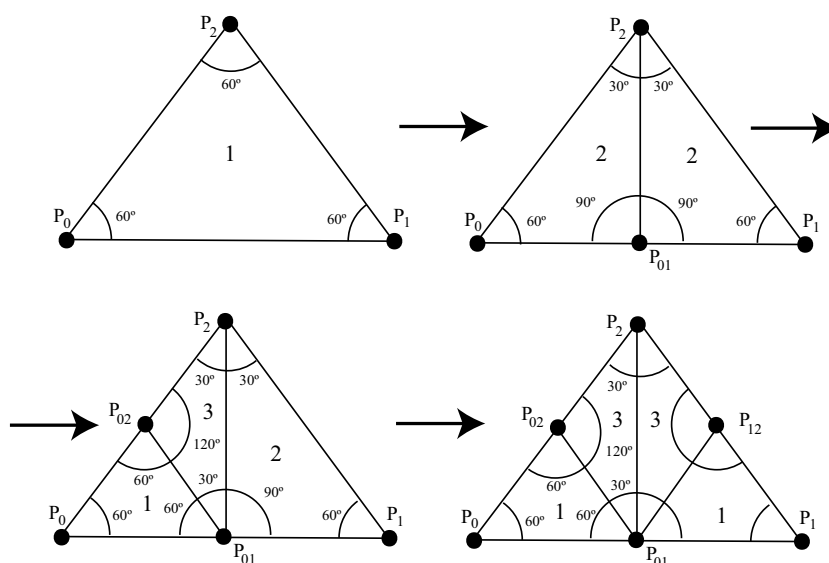


Figura 2.3: Efeito do refinamento da partição de um triângulo pelo esquema de subdivisão por 2.

novo vértice no centro de gravidade do triângulo; ou (c) dividir o triângulo em quatro pela inclusão de três novos vértices sobre os pontos médios de cada aresta.

A divisão pela bissetão é o procedimento mais simples sendo apresentada na Figura 2.3. Esta figura mostra a seqüência de partição de um triângulo equilátero (ângulos internos iguais a 60°), pertencente a classe de congruência numerada como 1. Na primeira subdivisão são gerados dois triângulos de uma segunda classe de congruência (ângulos internos iguais a 30° , 60° e 90°). O refinamento desta divisão gera um triângulo da classe 1 e um triângulo de uma terceira classe (um ângulo interno igual a 120° e dois ângulos de 30°). O último refinamento resulta em quatro triângulos que pertencem às classes 1 e 3.

A segunda opção de divisão, baseada na introdução de um vértice no centro de gravidade do triângulo, é apresentada na Figura 2.4. Com um refinamento são obtidos três triângulos de uma classe de congruência diferente da classe original. É possível perceber que, com refinamentos sucessivos, são obtidos triângulos cada vez menos bem formatados, sem redução da distância entre os vértices mais distantes do triângulo sendo refinado. Esta técnica se mostra totalmente inadequada para ser utilizada com o algoritmo do tipo branch-and-bound.

A última opção de divisão, com a inclusão de novos vértices sobre cada aresta do triângulo, denominada divisão orientada pelas arestas (do inglês “edgewise subdivision”) (Edelsbrunner e Grayson, 2000), gera triângulos da mesma classe do triângulo original, independentemente do número de refinamentos realizados, como pode ser visto na Figura 2.5. Observe pela Figura 2.3 que, se for necessário dividir o triângulo em quatro pela técnica de bissetão, são necessários três subdivisões com a geração de dois triângulos na partição inicial além dos quatro triângulos no refinamento final.

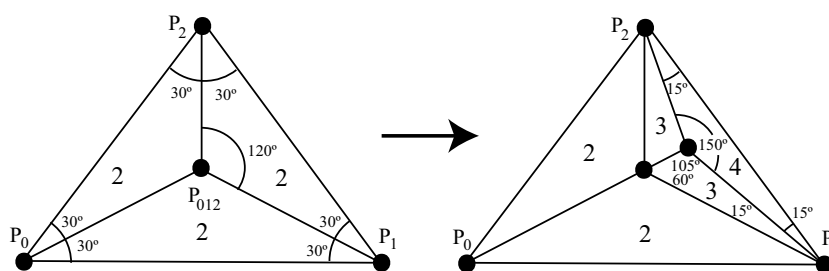


Figura 2.4: Efeito do refinamento da partição de um triângulo pelo esquema de subdivisão por 3.

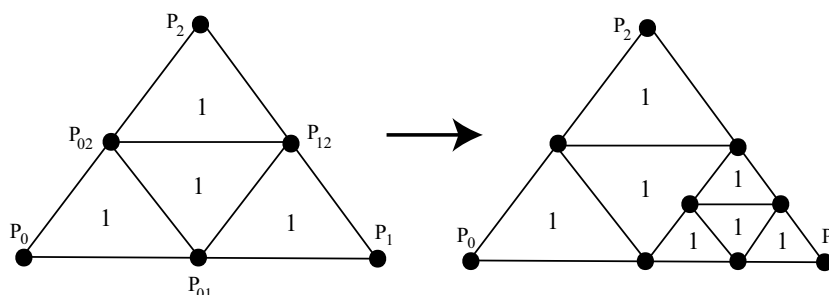


Figura 2.5: Efeito do refinamento da partição de um triângulo em 4.

É provado em Bey (2000) que o número de classes diferentes de simplexes gerados pela divisão orientada pelas arestas para espaços d -dimensional é limitada em $d!/2$, valor ótimo, ao passo que, pela bisseção, não existe prova que este número seja limitado no caso de dimensões maiores que $d = 2$. Deste modo, se o simplexo original é bem formatado, a divisão orientada pelas arestas irá gerar simplexes bem formatados, o que pode não ocorrer pela técnica da bisseção, mesmo com a partição sobre a aresta de maior dimensão. Devido a esta característica, esta técnica pode ser considerada a mais apropriada para o refinamento da malha simplicial requerido pelos procedimentos de análise a serem apresentados nos próximos dois capítulos. Uma desvantagem da técnica de divisão de simplexo orientada pelas arestas seria a sua implementação em dimensões superiores à $d = 3$ devido à inexistência na literatura de algoritmos para esse fim. Para resolver este problema, foi desenvolvido nesta tese um algoritmo de fácil implementação computacional que será descrito na próxima seção.

Com base nestes fatos, a escolha adotada nesta tese para refinamento da malha simplicial é a técnica de subdivisão de simplexo orientada pelas arestas. Além do número ótimo de classes congruentes, existem outras vantagens em se aplicar a subdivisão orientada pelas arestas, sendo importante para algoritmos tipo branch-and-bound a característica de que os subsimplexes obtidos terão o mesmo volume d -dimensional, sendo $1/2^d$ do volume original (para o caso da subdivisão orientada pelas arestas em 2^d) (Edelsbrunner e Grayson, 2000). Observe que, diferentemente de outras aplicações em Engenharia, tais como elementos finitos, não existe a preocupação em se garantir a consistência do

particionamento pela não utilização de vértices de um simplexo sobre a aresta de outro.

2.5 Subdivisão de simplexo orientada pelas arestas

Nesta seção é proposto um algoritmo para implementar a técnica de subdivisão orientada pelas arestas de um simplexo d -dimensional em k^d simplexos. A implementação desta técnica de subdivisão é baseada em um modelo de simplexo na forma de ábaco, denominado esquema de cores (“color scheme”), apresentado em Edelsbrunner e Grayson (2000) que é relacionado com o trabalho de Goodman e Peters (1995). A mesma notação utilizada por Edelsbrunner e Grayson (2000) será adotada aqui.

Considere um d -simplexo σ definido como uma seqüência de $d + 1$ pontos (vértices), $P_i \in \mathbb{R}^d$, $i = 0, \dots, d$, que são independentes afins em \mathbb{R}^d . A subdivisão orientada pelas arestas de σ em k^d simplexos será obtida a partir dos pontos P_0, P_1, \dots, P_d e de novos pontos obtidos a partir de uma matriz $M \in \mathbb{N}^{k \times (d+1)}$, denominada esquema de cores, cujos elementos, $\chi_{i,j}$, são números inteiros na faixa $[0, d]$, denominados cores, que representam os subscritos dos pontos P_0, P_1, \dots, P_d (Edelsbrunner e Grayson, 2000). A j -ésima coluna de M , $M_j = [\chi_{0,j} \ \chi_{1,j} \ \dots \ \chi_{k-1,j}]^T$, definirá o j -ésimo vértice $P_{\chi_{0,j}\chi_{1,j}\dots\chi_{k-1,j}}$ do novo simplexo definido como

$$P_{\chi_{0,j}\chi_{1,j}\dots\chi_{k-1,j}} \triangleq \frac{1}{k}(P_{\chi_{0,j}} + P_{\chi_{1,j}} + \dots + P_{\chi_{k-1,j}}) \quad (2.1)$$

Para se adequar ao algoritmo que será proposto, os índices das linhas de M iniciam com 0 ao invés de 1 como definido por Edelsbrunner e Grayson (2000). As principais características do esquema de cores são que os elementos aparecem em ordem não decrescente quando lidos como texto (ordenação lexicográfica), iniciando com $\chi_{0,0} = 0$ e terminando com $\chi_{k-1,d} = d$, com o primeiro elemento de uma linha igual ao último elemento da linha anterior, $\chi_{i,0} = \chi_{i-1,d}$ (Edelsbrunner e Grayson, 2000) :

$$\chi_{0,0} \leq \chi_{0,1} \leq \dots \leq \chi_{0,d} = \chi_{1,0} \leq \chi_{1,1} \leq \dots \leq \chi_{k-1,d}$$

e suas colunas M_j , $j = 0, \dots, d$, são organizadas de tal forma que, da coluna M_{j-1} para a coluna seguinte M_j , apenas uma das cores muda por um incremento unitário, isto é, $M_j = M_{j-1} + \xi^{l_j}$, sendo ξ^{l_j} a l_j -ésima coluna da matriz identidade I_k , ou seja, os elementos de ξ^{l_j} são definidos como

$$\xi_i^{l_j} \triangleq \begin{cases} 1, & i = l_j \\ 0, & i \neq l_j \end{cases} \quad (2.2)$$

O problema tratado aqui é como obter os k^d esquemas de cores para gerar a subdivisão completa

do simplexo. O algoritmo proposto a seguir irá realizar a tarefa de gerar automaticamente os esquemas de cores, tornando viável o uso do método descrito.

Seja $\chi_{i,j}^n$ o elemento da i -ésima linha, j -ésima coluna do n -ésimo esquema de cores. No algoritmo proposto, o n -ésimo esquema de cores, M^n , $n = 0, 1, \dots, k^d - 1$, será criado linha por linha iniciando com $\chi_{0,0}^n = 0$. Para saber se o próximo elemento da matriz será mantido ou incrementado em um é necessário representar o índice n do simplexo M^n no sistema numérico com base k :

$$n = x_{d-1} \times k^{d-1} + x_{d-2} \times k^{d-2} + \dots + x_0 \times k^0 \quad (2.3)$$

Os valores dos dígitos x_{d-j} , $j = 1, 2, \dots, d$, determinarão qual linha da coluna $j - 1$ será incrementada em um para gerar a coluna j , ou seja $M_j = M_{j-1} + \xi^{l_j}$ com $l_j = x_{d-j}$. Ao terminar uma linha, a próxima linha inicia com a última cor da linha anterior, ou seja, $\chi_{i,0}^n = \chi_{i-1,d}^n$. O procedimento descrito é implementado pelo algoritmo descrito a seguir (Gonçalves, Palhares, Takahashi e Mesquita, 2006a).

Algoritmo: Esquema de cores

```

para  $n = 0, 1, \dots, k^d - 1$ 
   $x_{d-1} \dots x_0 \leftarrow$  converta  $n$  para base  $k$ ;
   $cor \leftarrow 0$ ;
  para  $i = 0, 1, \dots, k - 1$ 
     $\chi_{i,0}^n \leftarrow cor$ ;
    para  $j = 1, \dots, d$ 
      se  $x_{d-j} = i$  então
         $cor \leftarrow cor + 1$ ;
      fim se
       $\chi_{i,j}^n \leftarrow cor$ ;
    fim para
  fim para
fim para
fim algoritmo

```

Considere, por exemplo, a subdivisão de um tetraedro em $k^d = 2^3 = 8$ sub-tetraedros. O esquema de cores é formatado com k linhas e $d + 1$ colunas:

$$M = \begin{bmatrix} \chi_{0,0} & \chi_{0,1} & \chi_{0,2} & \chi_{0,3} \\ \chi_{1,0} & \chi_{1,1} & \chi_{1,2} & \chi_{1,3} \end{bmatrix}$$

Utilizando o algoritmo apresentado, os 8 sub-tetraedros são obtidos como sendo

$$0_{10} \equiv 000_2 \Rightarrow M^0 = \begin{bmatrix} 0 \rightsquigarrow & 1 \rightsquigarrow & 2 \rightsquigarrow & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix} \Rightarrow \sigma_0 = \{P_{03}, P_{13}, P_{23}, P_{33}\}$$

$$1_{10} \equiv 001_2 \Rightarrow M^1 = \begin{bmatrix} 0 \rightsquigarrow & 1 \rightsquigarrow & 2 & 2 \\ 2 & 2 & 2 \rightsquigarrow & 3 \end{bmatrix} \Rightarrow \sigma_1 = \{P_{02}, P_{12}, P_{22}, P_{23}\}$$

$$2_{10} \equiv 010_2 \Rightarrow M^2 = \begin{bmatrix} 0 \rightsquigarrow & 1 & 1 \rightsquigarrow & 2 \\ 2 & 2 \rightsquigarrow & 3 & 3 \end{bmatrix} \Rightarrow \sigma_2 = \{P_{02}, P_{12}, P_{13}, P_{23}\}$$

$$3_{10} \equiv 011_2 \Rightarrow M^3 = \begin{bmatrix} 0 \rightsquigarrow & 1 & 1 & 1 \\ 1 & 1 \rightsquigarrow & 2 \rightsquigarrow & 3 \end{bmatrix} \Rightarrow \sigma_3 = \{P_{01}, P_{11}, P_{12}, P_{13}\}$$

$$4_{10} \equiv 100_2 \Rightarrow M^4 = \begin{bmatrix} 0 & 0 \rightsquigarrow & 1 \rightsquigarrow & 2 \\ 2 \rightsquigarrow & 3 & 3 & 3 \end{bmatrix} \Rightarrow \sigma_4 = \{P_{02}, P_{03}, P_{13}, P_{23}\}$$

$$5_{10} \equiv 101_2 \Rightarrow M^5 = \begin{bmatrix} 0 & 0 \rightsquigarrow & 1 & 1 \\ 1 \rightsquigarrow & 2 & 2 \rightsquigarrow & 3 \end{bmatrix} \Rightarrow \sigma_5 = \{P_{01}, P_{02}, P_{12}, P_{13}\}$$

$$6_{10} \equiv 110_2 \Rightarrow M^6 = \begin{bmatrix} 0 & 0 & 0 \rightsquigarrow & 1 \\ 1 \rightsquigarrow & 2 \rightsquigarrow & 3 & 3 \end{bmatrix} \Rightarrow \sigma_6 = \{P_{01}, P_{02}, P_{03}, P_{13}\}$$

$$7_{10} \equiv 111_2 \Rightarrow M^7 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 \rightsquigarrow & 1 \rightsquigarrow & 2 \rightsquigarrow & 3 \end{bmatrix} \Rightarrow \sigma_7 = \{P_{00}, P_{01}, P_{02}, P_{03}\}$$

Observe que, para calcular, por exemplo, o sub-tetraedro com índice 6, a mudança das cores no esquema de cores será especificada escrevendo 6 na base 2, ou seja, $x = 110_2$, o que significa que as duas primeiras mudanças de cores ocorrem na linha 1 e a última mudança de cor ocorre na linha 0. Este esquema de cores mostra que o sub-tetraedro é definido pelo conjunto de pontos $\{P_{01}, P_{02}, P_{03}, P_{13}\}$ como destacado na Figura 2.6, sendo o ponto $P_{\chi_{0,j}\chi_{1,j}}$ calculado por (2.1).

Esta subdivisão do tetraedro corresponde ao mesmo resultado do algoritmo *RedRefinement3D* apresentado em Bey (2000), com cada sub-tetraedro definido pela mesma seqüência de vértices obtidas pelo algoritmo proposto. Em Bey (2000) é apresentada a conjectura de que a ordem dos vértices que define cada sub-tetraedro é fundamental para que o algoritmo seja estável em refinamentos sucessivos, garantindo que apenas três classes congruentes serão geradas.

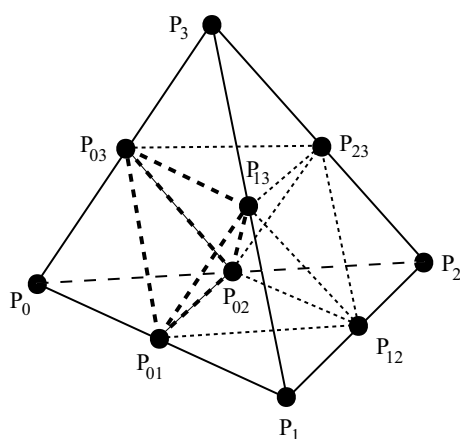


Figura 2.6: Exemplo de sub-tetraedro gerado pela partição de um tetraedro em 2^3 partes.

No caso de uma divisão no espaço bi-dimensional com $k = 3$, apresentada na Figura 2.7, dois exemplos de esquemas de cores seriam:

$$2_{10} \equiv 02_3 \Rightarrow M^2 = \begin{bmatrix} 0 & \rightsquigarrow 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & \rightsquigarrow 2 \end{bmatrix} \Rightarrow \sigma_2 = \{P_{011}, P_{111}, P_{112}\}$$

sendo

$$P_{011} = \frac{1}{3}(P_0 + 2P_1), \quad P_{111} = P_1, \quad P_{112} = \frac{1}{3}(2P_1 + P_2)$$

e

$$4_{10} = 11_3 \Rightarrow M^4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \rightsquigarrow 1 & \rightsquigarrow 2 \\ 2 & 2 & 2 \end{bmatrix} \Rightarrow \sigma_4 = \{P_{002}, P_{012}, P_{022}\}$$

sendo

$$P_{002} = \frac{1}{3}(2P_0 + P_2), \quad P_{012} = \frac{1}{3}(P_0 + P_1 + P_2), \quad P_{022} = \frac{1}{3}(P_0 + 2P_2)$$

O ábaco apresentado na Figura 3 em Edelsbrunner e Grayson (2000), tratando da subdivisão em 3^7 , pode ser gerado pelo algoritmo proposto:

$$1371_{10} = 1 \times 3^6 + 2 \times 3^5 + 1 \times 3^4 + 2 \times 3^3 + 2 \times 3^2 + 1 \times 3^1 + 0 \times 3^0 \equiv 1212210_3$$

$$\Rightarrow M^{1371} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \rightsquigarrow 1 \\ 1 & \rightsquigarrow 2 & 2 & \rightsquigarrow 3 & 3 & 3 & \rightsquigarrow 4 & 4 \\ 4 & 4 & \rightsquigarrow 5 & 5 & \rightsquigarrow 6 & \rightsquigarrow 7 & 7 & 7 \end{bmatrix}$$

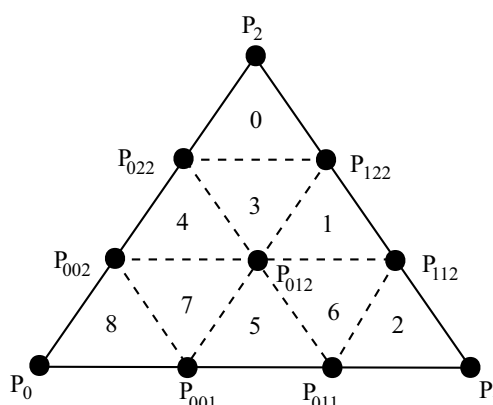


Figura 2.7: Exemplo de partição de um triângulo em 3^2 partes.

Para espaços de dimensão superiores a quatro, fica difícil comprovar visualmente que o algoritmo proposto para implementar a divisão de simplexo orientada pelas arestas funciona corretamente. Um teste fácil de ser realizado é verificar a propriedade deste algoritmo de divisão de que o volume d -dimensional dos k^d simplexos gerados são iguais ao volume do simplexo original dividido por k^d . O hiper-volume de um d -simplexo σ pode ser calculado a partir do seguinte determinante:

$$\text{vol}(\sigma) = \frac{1}{d!} |\det [P_1 - P_0 \dots P_i - P_0 \dots P_d - P_0]| \quad (2.4)$$

Por exemplo, considere o simplexo no espaço 6-dimensional definido pelo conjunto de vértices: $\{[1 \ 0,1 \ 0 \ 0 \ 0 \ 0]^T; [0 \ 1 \ 0 \ 0,3 \ 0 \ 0]^T; [0,2 \ 0 \ 1 \ 0 \ 0 \ 0]^T; [0 \ 0 \ 0 \ 1 \ 0,1 \ 0]^T; [0 \ 0,2 \ 0 \ 0 \ 1 \ 0]^T; [0 \ 0 \ 0 \ 0 \ 0,1 \ 1]^T; [0,8 \ 1 \ 0,9 \ 1 \ 1 \ 1]^T\}$. O volume deste simplexo é calculado como sendo $5,4276 \times 10^{-3}$. Para $k = 4$, os 4096 simplexos gerados pelo algoritmo de divisão proposto possuem o mesmo volume igual a $1,3251 \times 10^{-6}$, sendo que a soma dos volumes é igual ao volume do simplexo original. Outros testes semelhantes envolvendo volumes foram realizados com sucesso.

2.6 Conclusões

Neste capítulo foi apresentada uma nova estratégia de partição de polítopos que será útil para o emprego de algoritmos tipo branch-and-bound nos procedimentos de análise de estabilidade e de desempenho de sistemas incertos lineares invariantes no tempo que serão apresentados nos próximos dois capítulos. A estratégia proposta é baseada em malhas simpliciais. No caso de modelos dependentes de parâmetros, caso o polítopo já não seja um simplexo, é utilizada a triangularização de Delaunay para decompor o polítopo exatamente em um conjunto de simplexos. Os refinamentos posteriores são realizados por uma técnica de divisão de simplexo orientada pelas arestas. Como contribuição desta

tese é apresentado um algoritmo de simples implementação para a técnica de divisão de simplexo em qualquer dimensão. Tal algoritmo permite o emprego do algoritmo branch-and-bound para politopos de qualquer formato. Deste modo é possível tratar de modelos politópicos e modelos por dependência afim de parâmetros. Lidar com malhas simpliciais ao invés de hiper-retângulos também irá resultar em um algoritmo branch-and-bound muito mais eficiente para a aplicação que será considerada nos capítulos seguintes.

Capítulo 3

Análise de \mathcal{D} -Estabilidade Robusta

3.1 Introdução

A análise de estabilidade de sistemas incertos é, a princípio, um problema de difícil tratamento uma vez que é necessária a verificação de infinitos sistemas pertencentes ao domínio de incerteza. A Teoria de Lyapunov tem sido empregada intensivamente para a análise de estabilidade robusta uma vez que o problema de dimensão infinita é reduzido a um problema de dimensão finita sendo que apenas os vértices do domínio politópico de incerteza necessitam ser verificados. A análise de estabilidade pela Teoria de Lyapunov é caracterizada por um problema de factibilidade, formulado em termos de LMIs, que pode ser facilmente resolvido por um dos programas “LMI-solvers” disponíveis (ver Peaucelle, Henrion, Labit e Taitz (2002) para uma lista de “solvers” e interfaces). A condição de estabilidade quadrática, baseada em uma única função de Lyapunov, é a formulação mais simples, porém a mais conservadora. Para reduzir o conservadorismo, podem-se utilizar funções de Lyapunov dependentes de parâmetros (de Oliveira, Bernussou e Geromel, 1999; de Oliveira, Geromel e Hsu, 1999; de Oliveira e Skelton, 2002; Ramos e Peres, 2001; Ramos e Peres, 2002; de Oliveira, 2004; Kau et al., 2005; Oliveira e Peres, 2005c) e funções de Lyapunov com dependência polinomial de parâmetros (Henrion et al., 2004; Chesi et al., 2005b; Oliveira e Peres, 2005a; Oliveira e Peres, 2006). A vantagem destas últimas formulações é que o conservadorismo da condição suficiente pode ser reduzido com o aumento do grau da função de Lyapunov polinomial, porém, a complexidade aumenta rapidamente tanto com o grau do polinômio como com o número de vértices do domínio incerto politópico. Em Chesi (2005) é apresentada uma condição suficiente e “assintoticamente” necessária para análise de estabilidade robusta de sistemas lineares contínuos no tempo, com domínio de incerteza na forma de um hiper-cubo, que pode ser verificada através de problemas convexos de otimização LMI. Em Ebihara et al. (2005) são propostas condições LMI para análise robusta de estabilidade de sistemas contínuos no tempo considerando funções de Lyapunov associadas

com derivadas de alta ordem do vetor de estado. No caso das formulações LMI de análise de estabilidade que são apenas condições suficientes, quando não é encontrada uma solução para o problema de factibilidade, nada pode ser afirmado a respeito da estabilidade do sistema. Como verificado em Leite e Peres (2003) e Kau et al. (2005), a eficiência da formulação LMI de análise diminui com o aumento do número de vértices do politopo e da ordem do sistema. É possível obter formulações menos conservadoras aumentando o número de variáveis de decisão ao custo de um maior tempo de processamento. A Teoria de Lyapunov pode ser estendida para tratar do problema de \mathcal{D} -estabilidade robusta em que se deseja verificar se todos os pólos estão robustamente localizados em regiões convexas do plano complexo, denominadas regiões \mathcal{LMI} (Chilali e Gahinet, 1996; Peaucelle et al., 2000). A \mathcal{D} -estabilidade robusta pode ser caracterizada baseada em estabilidade quadrática ou baseada em funções de Lyapunov dependentes de parâmetros (Peaucelle et al., 2000; Leite e Peres, 2003; Gao e Xue, 2004).

Neste capítulo é proposto um novo método de análise que permite determinar se um sistema linear invariante no tempo é robustamente \mathcal{D} -estável ou não. Para obter este resultado, o método de análise proposto é baseado em um procedimento que combina formulações LMI de análise, derivadas da Teoria de Lyapunov, e uma técnica de partição de politopos. A idéia básica do procedimento proposto é particionar o politopo iterativamente até que todos os subpolitopos obtidos atendam à condição suficiente de \mathcal{D} -estabilidade robusta, ou seja encontrado um sistema pertencente ao politopo que não é \mathcal{D} -estável. O método de análise proposto combina a redução do conservadorismo das condições suficientes baseadas em LMIs, pela partição do politopo, com um método de grade (ou discretização do domínio) para verificação pontual dos autovalores dos sistemas pertencentes ao politopo. A estratégia de dividir o politopo até que possa ser afirmado se um sistema incerto é instável ou robustamente estável não é nova, tendo sido utilizada por DeMarco et al. (1990) para o caso de um politopo na forma de hiper-retângulo, sendo mais restrita que o novo método proposto. A contribuição deste trabalho é apresentar uma implementação completamente diferente da apresentada no início dos anos 90, considerando formulações de análise baseadas em LMIs desenvolvidas recentemente e a nova técnica de partição de politopos proposta nesta tese, apresentada no Capítulo 2. O grande diferencial da estratégia de partição de politopos proposta é o fato de que a mesma considera malha simplicial permitindo a aplicação do método tanto para modelos de incerteza politópicos como para modelos com dependência afim de parâmetros. Trabalhar com malha simplicial é mais eficiente uma vez que os simplexes são os politopos com menor número de vértices possíveis em uma determinada dimensão. O número de vértices do politopo possui influência direta na complexidade (número de variáveis de decisão e número de linhas das LMIs) das condições suficientes de estabilidade baseadas em LMIs. Parte dos resultados apresentados neste capítulo também podem ser vistos em Gonçalves, Palhares, Takahashi e Mesquita (2006d).

3.2 Formulação do problema

Considere o sistema linear invariante no tempo representado pelo seguinte modelo no espaço de estados:

$$\delta[x(\tau)] = Ax(\tau), \quad A \in \mathcal{P} \quad (3.1)$$

com $A \in \mathbb{R}^{n \times n}$, sendo

$$\delta[x(\tau)] \triangleq \frac{dx(t)}{dt}, \quad \tau \triangleq t \in \mathbb{R} \quad \text{para sistemas contínuos no tempo}$$

$$\delta[x(\tau)] \triangleq x(k+1), \quad \tau \triangleq k \in \mathbb{N} \quad \text{para sistemas discretos no tempo}$$

Considere que a matriz A não é precisamente conhecida, mas pertence a um domínio incerto: $A \in \mathcal{P}$. No caso de modelos politópicos, o conjunto \mathcal{P} é um politopo no espaço de matrizes definido pelos N vértices, A_i :

$$\mathcal{P} \triangleq \left\{ A(\theta) : A(\theta) = \sum_{i=1}^N \theta_i A_i, \quad \theta \in \Omega_M \right\} \quad (3.2)$$

sendo que o conjunto Ω_M pode ser representado como um simplexo no espaço de dimensão $N - 1$:

$$\Omega_M \triangleq \left\{ \hat{\theta} \in \mathbb{R}^{N-1} : \theta_i \geq 0, \quad i = 1, \dots, N-1, \quad \sum_{i=1}^{N-1} \theta_i \leq 1 \right\} \quad (3.3)$$

com $\theta_N = 1 - \sum_{i=1}^{N-1} \theta_i$.

No caso de modelos dependentes de parâmetros, a matriz A é dependente afim do vetor de parâmetros incertos $p = [p_1, p_2, \dots, p_d]^T \in \mathbb{R}^d$:

$$\mathcal{P} \triangleq \{A(p) : A(p) = A_0 + p_1 A_1 + \dots + p_d A_d, \quad p \in \Omega_p\} \quad (3.4)$$

podendo Ω_p ser um hiper-retângulo no espaço d -dimensional:

$$\Omega_p \triangleq \left\{ p \in \mathbb{R}^d : \underline{p}_i \leq p_i \leq \bar{p}_i, \quad i = 1, \dots, d \right\} \quad (3.5)$$

ou um politopo de formato qualquer:

$$\Omega_p \triangleq \{p \in \mathbb{R}^d : a_i^T p \leq b_i, \quad i = 1, \dots, m\} \quad (3.6)$$

sendo a_i^T as linhas de uma matriz $\mathcal{A} \in \mathbb{R}^{m \times d}$ e b_i os elementos de um vetor $b \in \mathbb{R}^m$.

O modelo por dependência afim de parâmetros pode ser convertido em um modelo politópico. Neste caso, os vértices do politopo de matrizes, A_1, \dots, A_N , são obtidos a partir das combinações dos valores extremos de p_i, \dots, p_d e de outras possíveis restrições. Por exemplo, no caso em que Ω_p é um hiper-retângulo, o conjunto Ω_M correspondente terá 2^d vértices.

O problema de análise de \mathcal{D} -estabilidade robusta é determinar se os autovalores das infinitas matrizes A do domínio de incerteza pertencem a um determinada região \mathcal{D} do plano complexo, isto é, $\lambda(A) \triangleq \{\lambda_1(A), \dots, \lambda_n(A)\} \subset \mathcal{D}, \forall A \in \mathcal{P}$. Serão consideradas regiões \mathcal{D} convexas e simétricas em relação ao eixo real conhecidas como regiões \mathcal{LMI} . Neste trabalho é considerada a definição de região \mathcal{LMI} apresentada em Peaucelle et al. (2000) que é uma versão ligeiramente modificada da região \mathcal{LMI} apresentada em Chilali e Gahinet (1996).

Definição 3.1 *Seja $R \in \mathbb{R}^{2m \times 2m}$ uma matriz simétrica que pode ser particionada como*

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^T & R_{22} \end{bmatrix} : \begin{array}{l} R_{11} = R_{11}^T \in \mathbb{R}^{m \times m}, \\ R_{22} = R_{22}^T \in \mathbb{R}^{m \times m}, R_{22} \succeq 0. \end{array} \quad (3.7)$$

A região \mathcal{D} do plano complexo é definida como

$$\mathcal{D} \triangleq \{z \in \mathbb{C} : R_{11} + R_{12}z + R_{12}^T z^* + R_{22}zz^* \prec 0\}. \quad (3.8)$$

As seguintes regiões \mathcal{LMI} são consideradas neste trabalho (ver Figura 3.1):

- Semi-plano esquerdo, $\text{Real}(z) < h_1$:

$$R_{ve} = \left[\begin{array}{c|c} -2h_1 & 1 \\ \hline 1 & 0 \end{array} \right]$$

- Semi-plano direito, $\text{Real}(z) > h_2$:

$$R_{vd} = \left[\begin{array}{c|c} 2h_2 & -1 \\ \hline -1 & 0 \end{array} \right]$$

- Disco com raio r com centro em $(q, 0)$, $|z - q| < r$:

$$R_d = \left[\begin{array}{c|c} -r^2 + q^2 & -q \\ \hline -q & 1 \end{array} \right]$$

- Setor cônico com vértice na origem e ângulo interno $2\theta < \pi$, $\text{tg}(\theta)\text{Real}(z) < |\text{Imag}(z)|$:

$$R_s = \left[\begin{array}{cc|cc} 0 & 0 & \text{tg}(\theta) & 1 \\ 0 & 0 & -1 & \text{tg}(\theta) \\ \hline \text{tg}(\theta) & -1 & 0 & 0 \\ 1 & \text{tg}(\theta) & 0 & 0 \end{array} \right]$$

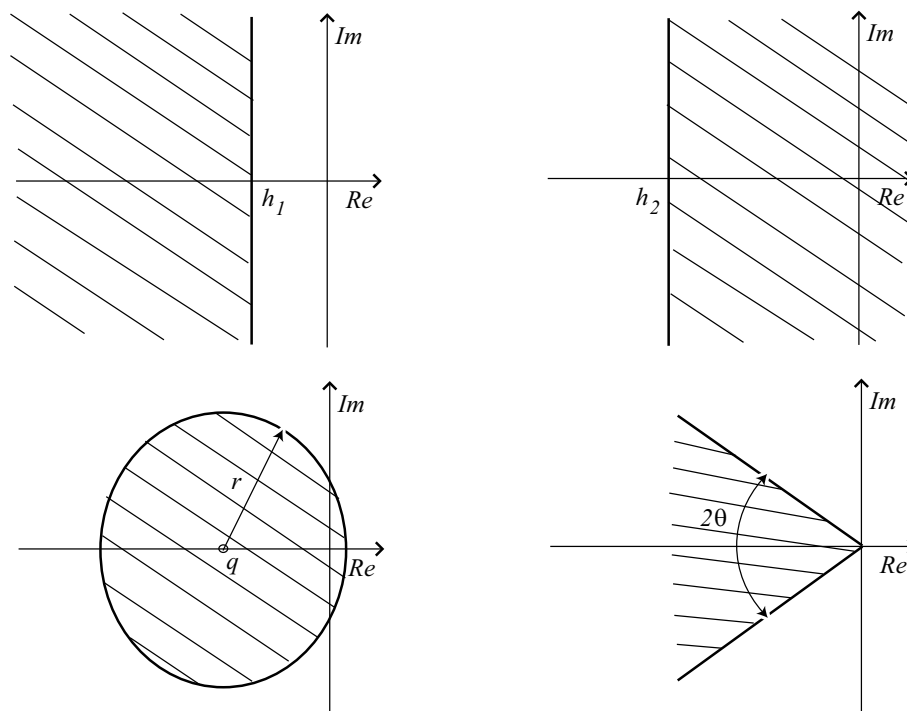


Figura 3.1: Regiões LMI.

Definição 3.2 (Peaucelle et al., 2000). A matriz $A \in \mathbb{R}^{n \times n}$ é dita ser \mathcal{D} -estável se e somente se todos os seus autovalores estão localizados na região \mathcal{D} definida por (3.8).

Definição 3.3 (Peaucelle et al., 2000). O sistema descrito pela eq. (3.1) é robustamente \mathcal{D} -estável se e somente se A é \mathcal{D} -estável para toda $A \in \mathcal{P}$.

Definição 3.4 (Peaucelle et al., 2000). O sistema descrito pela eq. (3.1) é quadraticamente \mathcal{D} -estável se e somente se existe uma matriz simétrica definida positiva P , tal que para todo $A \in \mathcal{P}$:

$$R_{11} \otimes P + R_{12} \otimes (PA) + R_{12}^T \otimes (A^T P) + R_{22} \otimes (A^T P A) \succ 0. \quad (3.9)$$

O operador \otimes representa o produto de Kronecker de duas matrizes. $C = A \otimes B$ é uma matriz em blocos com cada bloco dado por $C_{ij} = A_{ij}B$. O produto de Kronecker possui as seguintes propriedades (Brewer, 1978; Chilali e Gahinet, 1999):

$$\begin{aligned} 1 \otimes A &= A \\ (A + B) \otimes C &= A \otimes C + B \otimes C \\ (A \otimes B)(C \otimes D) &= AC \otimes BD \\ (A \otimes B)^T &= A^T \otimes B^T \\ (A \otimes B)^{-1} &= A^{-1} \otimes B^{-1} \end{aligned}$$

Os autovalores de $A \otimes B$ são os produtos dos pares $\lambda_i(A)\lambda_j(B)$ dos autovalores de A e B . Os valores singulares de $A \otimes B$ são os produtos $\sigma_i(A)\sigma_j(B)$ dos valores singulares de A e B .

Teorema 3.1 (Teorema 2 em Peaucelle et al. (2000)). *O sistema descrito pela eq. (3.1) é quadraticamente \mathcal{D} -estável se somente se existe uma mesma matriz simétrica definida positiva $P \in \mathbb{R}^{n \times n}$ tal que, para todo $i = 1, \dots, N$:*

$$\begin{bmatrix} R_{11} \otimes P + R_{12} \otimes (PA_i) + R_{12}^T \otimes (A_i^T P) & * \\ (L^T \otimes P)(I_d \otimes A_i) & -I_d \otimes P \end{bmatrix} \prec 0 \quad (3.10)$$

sendo $R_{22} = LL^T$.

Prova: Ver (Peaucelle et al., 2000).

O resultado principal apresentado em Peaucelle et al. (2000), relacionado com o Teorema 1 apresentado em de Oliveira, Geromel e Hsu (1999), é o teorema reproduzido em seguida.

Teorema 3.2 (Teorema 4 em Peaucelle et al. (2000)). *Se existe $F \in \mathbb{R}^{mn \times mn}$, $G \in \mathbb{R}^{mn \times n}$ e N matrizes simétricas definidas positivas P_i tal que, para todo $i = 1, \dots, N$*

$$\begin{bmatrix} R_{11} \otimes P_i + F(I_m \otimes A_i) + (I_m \otimes A_i^T)F^T & R_{12} \otimes P_i + (I_m \otimes A_i^T)G - F \\ R_{12}^T \otimes P_i + G^T(I_m \otimes A_i) - F^T & R_{22} \otimes P_i - G - G^T \end{bmatrix} \prec 0, \quad (3.11)$$

então (3.1) é robustamente \mathcal{D} -estável.

Prova: Ver (Peaucelle et al., 2000).

As formulações LMIs mais recentes buscam a redução do conservadorismo ao custo de formulações mais complexas com maior número de variáveis de decisão e/ou restrições, requerendo cada vez maior esforço computacional. Recentemente foi apresentado em Oliveira e Peres (2006) uma formulação LMI baseada em função de Lyapunov com dependência polinomial homogênea dos parâmetros

na qual o conservadorismo pode ser reduzido pelo aumento do grau do polinômio com correspondente aumento do esforço computacional. A idéia do procedimento a ser apresentado na próxima seção é o de reduzir o conservadorismo da condição suficiente LMI pela partição do politopo. Com isso, será mostrado que é possível resolver o problema de determinação da \mathcal{D} -estabilidade robusta aplicando formulações LMI menos complexas de forma mais eficiente que formulações LMI mais complexas.

3.3 Novo método de análise de \mathcal{D} -estabilidade robusta

3.3.1 Descrição do procedimento proposto

Seja $\mathcal{LMI}(\mathcal{P})$ a função que implementa o problema de factibilidade com restrições LMI que caracteriza a \mathcal{D} -estabilidade robusta, na qual o politopo \mathcal{P} é representado por seu conjunto de vértices $\{A_1, \dots, A_N\}$. Dado um problema de factibilidade na forma $L(x) \prec 0$, no qual x é o vetor de variáveis de decisão, $\mathcal{LMI}(\mathcal{P})$ soluciona o problema convexo auxiliar: minimize t sujeito à $L(x) \prec tI$. Esta função retorna o escalar t_{min} obtido pelo problema de minimização. O problema é factível se e somente se $t_{min} \leq 0$.

No procedimento de análise de \mathcal{D} -estabilidade robusta apresentado a seguir, a idéia básica é dividir o politopo e subpolitopos até que uma das seguintes condições de parada seja atendida:

- o politopo ou todos os subpolitopos obtidos pelas subdivisões atendam à condição suficiente de \mathcal{D} -estabilidade robusta;
- ou seja identificado um sistema correspondente a um vértice do politopo ou subpolitopos que não seja \mathcal{D} -estável;
- ou seja atingido o número máximo de iterações permitido.

Para escolher qual subpolitopo deva ser subdividido, pode ser utilizada a informação a respeito dos pólos do sistema nos vértices do politopo ou de forma mais simples, o valor de t_{min} retornado pela função $\mathcal{LMI}(\mathcal{P})$. Foi adotada a segunda opção pela simplicidade e por ter o mesmo efeito da primeira opção na maioria dos testes realizados.

No procedimento apresentado na seqüência, α pode representar $\hat{\theta}$ ou p , Ω pode representar Ω_M ou Ω_p , o conjunto \mathcal{L} contém os simplexes cuja condição suficiente de estabilidade robusta não é atendida (problema de factibilidade não possui solução), $\alpha_{(k)}$ representa as coordenadas ($\hat{\theta}$ ou p) do k -ésimo vértice do politopo \mathcal{P} ou dos subpolitopos \mathcal{S} e $\alpha_{(u)}$, se não for vazio, contém as coordenadas ($\hat{\theta}$ ou p) do primeiro sistema encontrado que não é \mathcal{D} -estável.

Procedimento de Análise de \mathcal{D} -Estabilidade Robusta

Passo 1. Inicialize $\mathcal{L} \leftarrow \emptyset$, $\alpha_{(u)} \leftarrow \emptyset$.

Passo 2. Compute $t_{min} = \mathcal{LMI}(\mathcal{P})$. Se $t_{min} \leq 0$, então vá para o passo 9.

Passo 3. Verifique se todos os vértices de \mathcal{P} , $A(\alpha_{(k)})$, são \mathcal{D} -estáveis. Se existe i, k tal que $\lambda_i(A(\alpha_{(k)})) \notin \mathcal{D}$, então faça $\alpha_{(u)} \leftarrow \alpha_{(k)}$ e vá para o passo 9.

Passo 4. Se \mathcal{P} é um simplexo, então faça $\mathcal{L} \leftarrow \mathcal{P}$, senão aplique a triangularização de Delaunay para decompor \mathcal{P} em um conjunto de simplexos $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_r\}$ e vá para o passo 8.

Passo 5. Encontre o simplexo $\mathcal{S}_m \in \mathcal{L}$ com maior valor de t_{min} .

Passo 6. Gere os novos vértices $A(\alpha_{(j)})$ sobre as arestas de \mathcal{S}_m requeridos pela técnica de divisão orientada pelas arestas. Se existe i, j tal que $\lambda_i(A(\alpha_{(j)})) \notin \mathcal{D}$, então faça $\alpha_{(u)} \leftarrow \alpha_{(j)}$ e vá para o passo 9.

Passo 7. Particione \mathcal{S}_m no conjunto de simplexos $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_r\}$ usando a técnica de divisão orientada pelas arestas e exclua \mathcal{S}_m de \mathcal{L} .

Passo 8. Para todo $\mathcal{S}_i \in \mathcal{S}$, se $t_{min} = \mathcal{LMI}(\mathcal{S}_i) > 0$, então $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{S}_i$.

Passo 9. Se $\mathcal{L} \neq \emptyset$ e $\alpha_{(u)} = \emptyset$, então vá para o passo 5, senão finalize.

Ao fim do procedimento de análise proposto, se \mathcal{L} e $\alpha_{(u)}$ são vazios, então o sistema incerto é robustamente \mathcal{D} -estável, senão $\alpha_{(u)}$ contém a coordenada do primeiro caso encontrado de sistema no polítipo \mathcal{P} que não é \mathcal{D} -estável.

Para que o procedimento proposto seja eficiente, é necessário uma escolha adequada da condição suficiente LMI para implementar a função $\mathcal{LMI}(\mathcal{P})$. Como será visto nos exemplos ilustrativos que serão apresentados, tal escolha deverá levar em consideração o compromisso entre complexidade e conservadorismo.

Comentário 3.1 *Ao invés de verificar todas as restrições de posicionamento de pólos simultaneamente, nesta tese é adotada a estratégia de se verificar o posicionamento de pólos em cada região individualmente. Nota-se que é mais eficiente verificar uma região de cada vez. Os problemas de factibilidade envolvendo todas as restrições simultaneamente são muito mais complexos de serem resolvidos, envolvendo um número maior de variáveis de otimização e de restrições. Além disso, tratando de várias regiões simultaneamente, as formulações LMI consideradas são tais que as mesmas funções de Lyapunov devem atender às diferentes restrições, o que é uma restrição adicional desnecessária. A verificação individual de cada região também possibilita a escolha da formulação mais adequada para cada tipo de região.*

3.3.2 Complexidade do procedimento de análise proposto

A partição do espaço de incerteza Ω é realizada pelo procedimento descrito no Capítulo. 2. Considere um domínio de incerteza Ω no espaço d -dimensional com N vértices. O passo inicial do procedimento proposto requer a verificação de uma condição suficiente LMI de \mathcal{D} -estabilidade robusta de um politopo \mathcal{P} com N vértices e a verificação da localização de n autovalores de N sistemas para testar se existe $\lambda_i(A(\alpha_{(j)})) \notin \mathcal{D}$, para $i = 1, \dots, n$ e $j = 1, \dots, N$. Caso o politopo inicial não seja um simplexo, na primeira iteração é realizada a decomposição do politopo em um conjunto de simplexos. A Tabela 3.1 apresenta o número de simplexos gerados pela triangularização de Delaunay considerando um hipercubo no espaço d -dimensional, que corresponde ao número de condições LMI que devem ser testadas de politopos com $d+1$ vértices. Neste caso não é necessária a verificação de autovalores desde que não são incluídos novos vértices para a formação do conjunto de simplexos. No caso em que o politopo inicial já é um simplexo e nas iterações subseqüentes à triangularização, é empregada a divisão de simplexo orientada pelas arestas descrita na Seção 2.5, para $k = 2$. Com esta técnica de subdivisão de simplexo, a cada iteração é necessário verificar os n autovalores de $\frac{1}{2}(d+1)d$ sistemas, correspondentes aos novos vértices sobre cada aresta, e 2^d condições LMI de politopos com $d+1$ vértices, conforme apresentado na Tabela 3.2.

No caso de modelos por dependência afim de parâmetros, uma vantagem significativa da decomposição do hiper-retângulo em simplexos, que diferencia o presente trabalho de resultados anteriores de aplicação de algoritmos tipo branch-and-bound, é a significativa redução do número de vértices dos politopos a serem testados. Considere, por exemplo, um sistema com $d = 5$ parâmetros incertos que podem assumir valores em um intervalo conhecido. Neste caso $\Omega \in \mathbb{R}^5$ é um hiper-retângulo com $N = 2^d = 32$ vértices. Trabalhando com simplexos, os subpolitopos passam a ter $N = d + 1 = 6$ vértices. Esta diferença no número de vértices significa uma redução considerável no tempo computacional requerido pela função $\mathcal{LMI}(\mathcal{P})$ que testa a condição suficiente LMI.

Tabela 3.1: Número de simplexos gerados pela triangularização de Delaunay de um hipercubo.

Dimensão d	2	3	4	5	6	7
Número de simplexos	2	6	24	103	648	3642

Como discutido na Seção 2.4, uma opção mais simples de divisão do simplexo seria a técnica de bisseção com a subdivisão do simplexo ao meio com a inclusão de um novo vértice sobre a maior aresta. Com esta técnica, independente da dimensão do espaço de incerteza, a cada iteração, são necessários verificar os autovalores de apenas um sistema e verificar a condição LMI para apenas dois politopos. Apesar da simplicidade, tal técnica pode resultar em um tempo de processamento total maior sem garantia de convergência para espaços de maiores dimensões.

Tabela 3.2: Número de novos vértices e de sub-simplexos obtidos com a subdivisão de simplexo orientada pelas arestas.

Dimensão d	1	2	3	4	5	6	7	d
Novos vértices	1	3	6	10	15	21	28	$(d+1)d/2$
Número de sub-simplexos	2	4	8	16	32	64	128	2^d

Mais detalhes sobre o algoritmo proposto de divisão de simplexo, com exemplo de implementação pelo MATLAB[®], são apresentados em Gonçalves, Palhares, Takahashi e Mesquita (2006a).

3.4 Exemplos ilustrativos

A seguir serão apresentados exemplos ilustrativos que comprovam a eficiência do método de análise de \mathcal{D} -estabilidade robusta proposto. As condições suficientes baseadas em LMIs, utilizadas nas comparações e no procedimento de análise proposto foram implementadas no LMI Control Toolbox (LMIC) (Gahinet et al., 1995) e em alguns exemplos também na SeDuMi Interface (SeDuMi) (Peaucelle, Henrion, Labit e Taitz, 2002; Sturm, 1999), ambos para uso com o MATLAB[®]. A complexidade computacional do LMIC é $\mathcal{O}(K^3L)$ enquanto o SeDuMi possui complexidade $\mathcal{O}(K^2L^{2.5} + L^{3.5})$, sendo K o número total de variáveis de decisão escalares e L o número total de linhas das LMIs. Em ambos os casos, o raio de factibilidade é fixado igual a 10^9 e o número máximo de iterações é igual a 500. A implementação permite a partição do domínio de incerteza representado pelas coordenadas do politopo de matrizes (modelo politópico) ou pelos parâmetros incertos (modelo com dependência afim de parâmetros). Os resultados foram obtidos utilizando um computador com processador Pentium IV 2.8GHz, 1GB de RAM e 4.096MB de memória virtual (dimensão máxima aceita pelo sistema). Parte dos resultados apresentados e outros exemplos também podem ser encontrados em Gonçalves, Palhares, Takahashi e Mesquita (2006d) e Gonçalves, Palhares, Takahashi e Mesquita (2006b).

3.4.1 Exemplos de análise de estabilidade robusta

Exemplo 3.1 Para avaliar a eficiência do procedimento de análise proposto aplicado a análise de estabilidade robusta de sistemas contínuos no tempo, é considerada neste exemplo uma comparação numérica exaustiva com condições suficientes baseadas em LMIs. Para cada par (n, N) , com $n \in [2, 5]$ e $N \in [2, 5]$, foram gerados de forma aleatória 250 sistemas politópicos robustamente estáveis por meio de um procedimento similar ao adotado em Leite e Peres (2003): 1) os vértices do politopo são gerados por matrizes cujos elementos são números reais uniformemente distribuídos no intervalo

[-1,1]; 2) para cada vértice, a matriz A é “deslocada” de tal modo que $\max_i \text{Real}(\lambda_i(A)) = -0,0001$; 3) é determinado $\max_{\alpha,i} \text{Real}(\lambda_i(A(\alpha)))$, $\alpha \in \Omega$, considerando uma grade para α_i , $i = 1, \dots, N$, variando com passo igual a 0,1; 4) todos os vértices do politopo são “deslocados” de tal modo que seja garantido pelo menos um autovalor com parte real igual a -0,0001 no politopo e 5) o sistema politópico aleatório resultante é verificado com o procedimento de análise proposto e o mesmo é incluído entre os 250 testes se o mesmo for identificado como robustamente estável ou se a condição de estabilidade não for identificada considerando o número limitado de 500 iterações (hipótese que não ocorreu nenhuma vez).

O procedimento de análise proposto, implementado com a condição suficiente baseada no Lema 1 apresentado em Ramos e Peres (2002), representado por BB, é comparado com as seguintes condições suficientes baseadas em LMIs: (QS) estabilidade quadrática, conforme Teorema 1 apresentado em Peaucelle et al. (2000); (EB₁) Teorema 1 apresentado em Ebihara e Hagiwara (2002a); (PE) Teorema 1 apresentado em de Oliveira, Geromel e Hsu (1999) ou Teorema 4 apresentado em Peaucelle et al. (2000); (RA) Lema 1 apresentado em Ramos e Peres (2002); (LE) Teorema 1 apresentado em Leite e Peres (2003); (EB₂) Teorema 2 apresentado em Ebihara et al. (2005); (GA) Teorema 4 apresentado em Gao e Xue (2004), (OL₁) Teorema 1 apresentado em Oliveira e Peres (2006) com grau $g = 3$ e (OL₂) Teorema 2 apresentado em Oliveira e Peres (2006) com grau $g = 2$. A Tabela 3.3 apresenta as taxas percentuais de sucesso para identificação da estabilidade robusta dos sistemas politópicos. Para o conjunto de 4000 testes realizados, o procedimento proposto é o único que apresenta taxa de sucesso igual a 100%. As condições suficientes LE, OL₁ e OL₂ também apresentam boas taxas de sucesso (aproximadamente igual ou superior a 90%). As Tabelas 3.4 e 3.5 apresentam os tempos computacionais médios requeridos, calculados para cada par (n, N) , considerando a implementação no MATLAB[®] pelo LMI Control Toolbox (LMIC) e pela SeDuMi Interface (SeDuMi), respectivamente. Considerando a implementação no LMIC, o procedimento de análise proposto apresenta tempos computacionais médios inferiores a quase todas as condições suficientes LMI, em quase todos os pares n, N , com exceção da condição baseada no conceito de estabilidade quadrática e da baseada no Lema 1 em Ramos e Peres (2002) (utilizado na implementação do procedimento proposto). Porém, estas duas últimas falham em várias situações, principalmente com o aumento de n e N . Considerando a implementação no SeDuMi, o procedimento proposto apresenta tempos médios inferiores e superiores comparados com as formulações LMI com melhores taxas de sucesso dependendo de (n, N) . Isto ocorre devido ao fato de que as formulações mais simples, incluindo a utilizada na implementação do procedimento proposto, requerem mais tempo computacional ao passo que as formulações mais complexas requerem menos tempo computacional em relação à implementação no LMIC. Considerando ambas as implementações, o procedimento de análise proposto na versão do LMIC apresenta menores tempos computacionais que as condições LMI com melhores taxa de

Tabela 3.3: Taxas de sucesso (%) para identificar a estabilidade robusta (LMIC) (Ex. 3.1)

n	N	QS	EB ₁	PE	RA	LE	GA	EB ₂	OL ₁	OL ₂	BB
2	2	51,2	60,4	100	100	100	100	100	100	100	100
	3	18,8	48,8	89,6	100	100	100	94,4	100	100	100
	4	21,2	22,8	52,4	99,6	100	99,6	54,4	100	100	100
	5	18,0	21,2	49,6	98,8	100	98,8	50,4	100	100	100
3	2	29,6	55,2	96,4	94,0	96,4	94,0	100	99,6	99,6	100
	3	8,8	28,0	70,8	89,6	96,0	89,6	76,8	99,6	100	100
	4	8,8	11,6	35,2	90,8	96,4	90,8	38,8	98,8	99,6	100
	5	4,4	7,2	26,4	84,4	97,2	84,4	28,4	100	100	100
4	2	27,2	54,4	96,8	91,6	96,8	91,6	100	99,6	100	100
	3	3,6	20,4	63,6	86,8	93,6	86,8	73,6	100	100	100
	4	1,2	7,2	28,8	84,4	93,6	84,4	34,4	100	100	100
	5	1,2	1,2	12,8	74,0	89,6	74,0	14,8	99,2	99,2	100
5	2	23,2	51,2	98,4	94,0	98,4	94,0	100	99,6	99,6	100
	3	1,2	18,4	59,6	91,2	96,0	91,2	71,2	99,6	99,2	100
	4	0	3,6	20,4	84,4	94,4	84,4	29,2	99,6	99,6	100
	5	0,4	1,2	11,6	80,4	93,6	80,4	13,6	98,8	99,2	100

sucesso em todos os casos.

Além da eficiência para identificação de sistemas incertos robustamente estáveis, também é interessante analisar a capacidade do procedimento proposto para identificar os sistemas politópicos que não são robustamente estáveis. Na geração dos sistemas incertos aleatórios pelo procedimento descrito, o procedimento proposto identificou a condição de estabilidade de todos os sistemas obtidos, até se atingir o número de 250 robustamente estáveis, com tempos computacionais e números de iterações médios bastantes satisfatórios conforme resultados apresentados na Tabela 3.6. Nesta tabela, para cada par (n, N) , T_{re} e T_{ne} representam os números de sistemas politópicos identificados como robustamente estáveis ou não, respectivamente, T_a representa o tempo computacional médio, S_a e S_m representam os números médio e máximo de iterações (ou subdivisões), respectivamente. No caso da identificação de sistemas não robustamente estáveis, devido ao método de geração de sistemas aleatórios, são necessárias pelo menos duas subdivisões para que sejam verificadas matrizes A já não testadas pela técnica de grade. O número máximo e médio de iterações não parece ser fortemente influenciado pela ordem do sistema ou pelo número de vértices. Esta característica pode ser explicada em parte pelo fato de que a técnica de subdivisão de simplexo orientada pelas arestas produz uma bisseção de todas as bordas simultaneamente. Esta pode ser considerada uma outra vantagem da técnica de partição de politopo proposta. Para este conjunto de testes, o número de divisões requerido foi surpreendentemente baixo.

Tabela 3.4: Tempos médios (s) para identificar estabilidade robusta (LMIC) (Ex. 3.1)

n	N	QS	EB ₁	PE	RA	LE	GA	EB ₂	OL ₁	OL ₂	BB	
2	2	0,03	0,06	0,03	0,03	0,08	0,03	0,12	0,04	0,07	0,03	
	3	0,03	0,10	0,05	0,04	0,21	0,05	0,19	0,10	0,20	0,04	
	4	0,03	0,14	0,07	0,06	0,46	0,08	0,35	0,26	0,58	0,07	
	5	0,04	0,17	0,09	0,09	0,90	0,11	0,45	0,69	1,61	0,10	
	3	2	0,03	0,08	0,04	0,03	0,12	0,04	0,19	0,05	0,12	0,04
3	3	0,04	0,14	0,07	0,05	0,39	0,08	0,46	0,14	0,61	0,07	
	4	0,04	0,20	0,12	0,08	1,04	0,15	0,98	0,46	2,86	0,17	
	5	0,05	0,25	0,16	0,12	2,40	0,25	1,37	1,43	10,4	0,34	
	4	2	0,04	0,12	0,06	0,04	0,27	0,07	0,39	0,06	0,31	0,05
	4	3	0,05	0,23	0,14	0,07	1,18	0,18	1,18	0,24	2,49	0,10
4		0,05	0,34	0,26	0,11	3,68	0,36	2,67	0,97	13,0	0,21	
5		0,05	0,46	0,37	0,18	9,86	0,71	4,14	3,97	56,0	0,78	
5		2	0,05	0,21	0,11	0,05	0,69	0,13	0,84	0,09	0,90	0,06
5		3	0,06	0,40	0,28	0,10	3,41	0,36	3,04	0,44	8,45	0,12
	4	0,06	0,60	0,54	0,18	12,2	0,85	6,84	2,24	48,0	0,34	
	5	0,07	0,81	0,76	0,30	34,3	1,67	10,5	9,95	205,0	0,79	

Tabela 3.5: Tempos médios (s) para identificar estabilidade robusta (SeDuMi) (Ex. 3.1)

n	N	QS	EB ₁	PE	RA	LE	GA	EB ₂	OL ₁	OL ₂	BB	
2	2	0,13	0,17	0,17	0,16	0,27	0,19	0,26	0,18	0,22	0,16	
	3	0,12	0,18	0,20	0,18	0,43	0,27	0,29	0,28	0,38	0,19	
	4	0,13	0,20	0,21	0,20	0,75	0,38	0,38	0,50	0,70	0,21	
	5	0,13	0,21	0,23	0,24	1,37	0,41	0,41	1,01	1,19	0,31	
	3	2	0,13	0,18	0,19	0,15	0,31	0,21	0,28	0,19	0,25	0,19
3	3	0,12	0,20	0,21	0,19	0,49	0,27	0,40	0,30	0,48	0,35	
	4	0,13	0,22	0,24	0,22	0,93	0,35	0,52	0,62	1,02	0,85	
	5	0,13	0,23	0,26	0,25	1,82	0,46	0,58	1,32	2,00	1,64	
	4	2	0,14	0,20	0,21	0,17	0,34	0,23	0,41	0,20	0,31	0,21
	4	3	0,13	0,23	0,24	0,20	0,62	0,30	0,62	0,37	0,77	0,40
4		0,13	0,25	0,28	0,24	1,36	0,41	0,83	0,85	2,01	1,01	
5		0,14	0,26	0,29	0,31	2,82	0,59	1,09	2,15	5,40	4,17	
5		2	0,15	0,23	0,24	0,18	0,45	0,25	0,68	0,23	0,44	0,22
5		3	0,14	0,26	0,29	0,23	0,90	0,37	1,11	0,46	1,47	0,38
	4	0,14	0,29	0,35	0,28	2,16	0,56	1,64	1,37	5,55	1,24	
	5	0,14	0,31	0,37	0,34	4,74	0,81	2,16	3,93	15,59	3,38	

Tabela 3.6: Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.1)

n	N	Sist. rob. estáveis				Sist. não rob. est.				
		T_{re}	T_a (s)	S_a	S_m	T_{ne}	T_a (s)	S_a	S_m	
2	2	250	0,03	0	0	87	0,11	4,0	7	
	3	250	0,04	0	0	235	0,36	3,9	8	
	4	250	0,07	0	0	432	1,05	3,9	9	
	5	250	0,10	0,0	1	878	2,84	3,8	26	
	3	2	250	0,04	0,1	3	88	0,15	3,9	7
3	3	250	0,07	0,2	9	290	0,47	4,0	8	
	4	250	0,17	0,1	2	533	1,29	3,9	9	
	5	250	0,34	0,3	8	1079	3,58	3,9	15	
	4	2	250	0,05	0,1	6	70	0,23	4,2	7
	3	250	0,10	0,2	4	265	0,65	3,9	8	
4	4	250	0,21	0,2	3	589	1,68	3,7	8	
	5	250	0,78	0,2	4	1205	4,54	3,7	17	
	5	2	250	0,06	0,0	2	67	0,33	4,0	8
	3	250	0,12	0,1	2	214	1,00	3,9	10	
	4	250	0,34	0,2	3	523	2,67	3,8	10	
5	250	0,79	0,2	5	937	7,16	3,9	18		

Exemplo 3.2 Neste exemplo é considerado o mesmo teste exaustivo do Ex. 3.1 para o caso de análise de estabilidade robusta de sistemas incertos a tempo discreto. A diferença com relação à geração aleatória de modelos politópicos é que no passo (2), para cada vértice, a matriz A é multiplicada por um escalar de tal modo que pelo menos um autovalor tenha módulo igual a 0,95 e no passo (4) todos os vértices do politopo são multiplicados por um escalar de tal modo que seja garantido pelo menos um autovalor com módulo igual a 0,999.

O procedimento de análise proposto, implementado com a condição suficiente baseada no Teorema 4 apresentado em Peaucelle et al. (2000), representado por BB, é comparado com as seguintes condições suficientes baseadas em LMIs: (QS) estabilidade quadrática, conforme Teorema 1 apresentado em Peaucelle et al. (2000); (dO) Teorema 2 apresentado em de Oliveira, Bernussou e Geromel (1999); (PE) Teorema 1 apresentado em de Oliveira, Geromel e Hsu (1999) ou Teorema 4 apresentado em Peaucelle et al. (2000); (RA) Lema 1 apresentado em Ramos e Peres (2001); (LE) Teorema 1 apresentado em Leite e Peres (2003); (GA) Teorema 4 apresentado em Gao e Xue (2004); (KA) Teorema 1 apresentado em Kau et al. (2005); (OL1) Teorema 4 apresentado em (Oliveira e Peres, 2005c) com grau $g = 3$; (OL2) Teorema 3 apresentado em Oliveira e Peres (2006) com grau $g = 3$ e (OL3) Teorema 4 apresentado em Oliveira e Peres (2006) com grau $g = 2$. A Tabela 3.7 apresenta as taxas percentuais de sucesso para identificação da estabilidade robusta dos sistemas politópicos e a Tabela 3.8 apresenta os tempos computacionais médios requeridos calculados para cada par (n, N)

Tabela 3.7: Taxas de sucesso (%) para identificar a estabilidade robusta (LMIC) (Ex. 3.2)

n	N	QS	dO	PE	RA	LE	GA	KA	OL1	OL2	OL3	BB
2	2	37,2	51,6	99,2	87,6	99,2	88,4	99,2	91,6	82,8	100	100
	3	2,0	28,8	86,4	74,4	98,4	77,6	99,2	83,2	58,4	100	100
	4	0,4	12,4	60,8	58,0	96,8	58,4	97,6	66,0	38,0	100	100
	5	0	14,4	47,2	38,8	95,6	52,0	97,6	51,2	24,8	100	100
3	2	11,2	47,6	96,8	87,6	96,8	88,4	96,8	90,8	84,4	100	100
	3	0,4	10,4	64,0	62,4	85,2	63,6	85,2	69,6	65,2	99,6	100
	4	0	6,4	40,0	49,2	84,0	52,8	85,2	60,0	44,4	100	100
	5	0	4,0	21,2	30,8	76,0	42,8	79,6	45,2	29,6	100	100
4	2	4,8	38,4	93,6	84,4	93,2	85,6	93,6	88,0	88,4	100	100
	3	0,4	10,4	60,4	60,4	85,6	65,6	85,6	70,4	66,0	100	100
	4	0	3,2	24,0	42,4	76,8	45,6	78,0	58,0	57,6	99,6	100
	5	0	1,2	16,8	30,4	70,4	39,2	71,6	48,0	34,8	99,2	100
5	2	4,0	34,0	92,0	79,6	92,0	80,8	92,0	84,4	88,0	100	100
	3	0	4,0	55,2	62,8	83,6	64,0	84,8	68,0	74,0	99,6	100
	4	0	1,2	24,8	50,4	76,8	51,2	78,8	60,8	63,2	100	100
	5	0	0,4	7,2	32,0	63,2	33,2	66,4	47,2	51,2	99,6	100

considerando a implementação no LMI Control Toolbox (LMIC). Considerando estes resultados, é possível dividir as condições suficientes baseadas em LMIs em dois grupos: as primeiras quatro pertencendo ao grupo das formulações mais simples (baixo tempo computacional) mas com baixa taxa de sucesso para sistemas politópicos de ordem mais alta e maior número de vértices e as últimas seis pertencendo ao grupo de formulações com melhores taxas de sucesso, porém mais complexas (maior número de variáveis de decisão e maior número de linhas de LMIs) e com maiores tempos computacionais. O método proposto apresenta 100% de taxa de sucesso independentemente de (n, N) , com menor tempo computacional do que as seis últimas condições LMI mais complexas, como mostrado pelos resultados nas Tabelas 3.7 e 3.8. O Teorema 4 apresentado em Oliveira e Peres (2006), com grau $g = 2$, é a única condição suficiente baseada em LMIs com taxa de sucesso similar a do procedimento proposto. Porém, o tempo computacional médio requerido pelo Teorema 4 em Oliveira e Peres (2006) cresce muito mais rapidamente com o aumento de n e N em comparação com o procedimento proposto, sendo cerca de 49 vezes maior para $(n, N) = (5, 5)$.

Considerando as implementações das condições suficientes LMI e do procedimento proposto para o “solver” SeDuMi, os tempos computacionais médios resultantes são consideravelmente diferentes dos obtidos com o LMI Control Toolbox conforme pode ser verificado pela Tabela 3.9. O SeDuMi reduz a diferença entre os tempos médios das formulações mais simples e das formulações mais complexas tornando as últimas muito mais atrativas. Neste caso o procedimento proposto requer mais tempo computacional do que na implementação anterior devido a um acréscimo no tempo requerido

Tabela 3.8: Tempos médios (s) para identificar estabilidade robusta (LMIC) (Ex. 3.2)

n	N	QS	dO	PE	RA	LE	GA	KA	OL1	OL2	OL3	BB
2	2	0,03	0,02	0,03	0,03	0,07	0,07	0,05	0,09	0,06	0,07	0,03
	3	0,03	0,03	0,05	0,08	0,21	0,14	0,18	0,99	0,24	0,21	0,08
	4	0,03	0,05	0,07	0,16	0,49	0,28	0,64	8,13	0,84	0,60	0,25
	5	0,04	0,05	0,09	0,31	0,98	0,47	2,07	45,3	2,59	1,58	0,61
3	2	0,03	0,03	0,04	0,04	0,11	0,12	0,09	0,13	0,08	0,11	0,05
	3	0,03	0,06	0,08	0,10	0,39	0,36	0,60	2,28	0,43	0,60	0,17
	4	0,05	0,07	0,13	0,23	1,07	0,75	3,28	20,4	1,96	2,62	0,45
	5	0,05	0,10	0,19	0,47	2,69	1,37	16,8	127,	8,09	10,1	1,19
4	2	0,04	0,06	0,07	0,05	0,24	0,27	0,18	0,22	0,12	0,28	0,08
	3	0,05	0,10	0,16	0,16	1,09	0,82	1,80	4,63	0,88	2,22	0,30
	4	0,06	0,15	0,29	0,39	3,73	1,96	14,3	43,3	4,34	12,4	0,89
	5	0,06	0,20	0,40	0,86	10,7	3,70	83,5	298,	22,8	51,8	1,99
5	2	0,05	0,10	0,12	0,08	0,62	0,57	0,40	0,41	0,21	0,78	0,14
	3	0,06	0,20	0,32	0,26	3,40	1,87	5,17	8,84	1,67	7,89	0,55
	4	0,08	0,29	0,58	0,67	12,7	4,32	46,0	84,5	9,87	43,7	1,49
	5	0,10	0,40	0,85	1,52	39,4	8,82	295,	591,	49,6	189,	3,86

pelas formulações mais simples. A divisão em dois grupos observada no caso do LMI Control Toolbox não se repete com a mesma clareza no caso do SeDuMi. Observe na Tabela 3.9 que, com exceção dos casos para $N = 2$ ou $n = 3$, o procedimento de análise proposto requer menos tempo computacional do que as condições suficiente LMI com taxa de sucesso superiores a 90%. O Teorema 4 apresentado em Oliveira e Peres (2006), com grau $g = 2$, único com taxa de sucesso próxima a 100%, requer menos tempo computacional do que o procedimento proposto apenas nos casos (n, N) iguais à $(3, 4)$ e $(3, 5)$.

Como no exemplo de sistemas aleatórios a tempo contínuo, na geração dos sistemas incertos aleatórios pelo procedimento descrito, o procedimento de análise proposto identificou a condição de estabilidade de todos os sistemas aleatórios gerados, até se atingir o número de 250 robustamente estáveis. A Tabela 3.10 apresenta os tempos computacionais médios, os números médios e máximos de iterações usando a mesma simbologia empregada na Tabela 3.6. Novamente é verificado que o número médio de iterações (partições do politopo) para identificação de sistemas instáveis não é diretamente influenciado pela dimensão $N - 1$ do domínio de incerteza. Tal característica não é apresentada pela técnica de bisseção de simplexes ou hiper-retângulos.

Para ilustrar o comportamento do procedimento de análise de estabilidade proposto como uma técnica de grade “inteligente” para localizar um caso de sistema instável pertencente ao politopo,

Tabela 3.9: Tempos médios (s) para identificar estabilidade robusta (SeDuMi) (Ex. 3.2)

n	N	QS	dO	PE	RA	LE	GA	KA	OL1	OL2	OL3	BB
2	2	0,12	0,15	0,15	0,16	0,38	0,21	0,24	0,22	0,18	0,23	0,16
	3	0,11	0,14	0,19	0,22	0,39	0,30	0,38	0,78	0,44	0,40	0,31
	4	0,11	0,14	0,19	0,31	0,74	0,45	0,71	3,54	1,09	0,80	0,63
	5	0,11	0,15	0,22	0,46	1,37	0,66	1,33	16,4	2,57	1,29	1,14
3	2	0,12	0,14	0,22	0,17	0,35	0,25	0,26	0,25	0,20	0,24	0,24
	3	0,11	0,15	0,20	0,23	0,47	0,36	0,49	0,98	0,54	0,48	0,48
	4	0,12	0,16	0,23	0,39	0,93	0,55	1,31	6,13	1,44	1,07	1,10
	5	0,12	0,17	0,25	0,58	1,83	0,80	4,50	40,7	3,59	2,16	2,85
4	2	0,12	0,16	0,20	0,17	0,40	0,32	0,27	0,27	0,23	0,34	0,23
	3	0,12	0,16	0,24	0,28	0,60	0,45	0,83	1,52	0,68	0,87	0,55
	4	0,13	0,17	0,28	0,48	1,33	0,74	3,59	13,5	2,00	2,08	2,06
	5	0,13	0,19	0,29	0,91	2,78	1,10	20,6	113,8	5,70	5,36	3,83
5	2	0,12	0,19	0,23	0,19	0,57	0,40	0,34	0,32	0,27	0,41	0,29
	3	0,13	0,18	0,29	0,35	0,87	0,63	1,58	2,67	0,90	1,57	0,77
	4	0,14	0,20	0,32	0,71	1,99	1,01	11,8	30,3	2,85	6,58	2,06
	5	0,15	0,23	0,38	1,64	4,63	1,71	72,3	277,	9,45	15,01	5,55

Tabela 3.10: Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.2)

n	N	Sist. rob. estáveis				Sist. não rob. est.			
		T_{re}	T_a (s)	S_a	S_m	T_{ne}	T_a (s)	S_a	S_m
2	2	250	0,033	0,008	1	10	0,189	3,100	4
	3	250	0,080	0,184	6	44	0,560	3,909	7
	4	250	0,254	0,472	6	65	1,104	3,508	6
	5	250	0,612	0,552	3	116	2,168	3,155	6
3	2	250	0,049	0,036	2	21	0,397	3,905	6
	3	250	0,173	0,424	5	35	0,911	4,029	6
	4	250	0,448	0,652	3	104	1,567	3,490	8
	5	250	1,187	0,856	3	138	3,274	3,420	8
4	2	250	0,078	0,068	2	15	0,691	3,533	5
	3	250	0,297	0,444	3	42	1,451	3,714	8
	4	250	0,888	0,820	4	80	2,629	3,475	7
	5	250	1,987	0,916	4	114	5,459	3,483	8
5	2	250	0,137	0,080	1	20	1,588	3,950	6
	3	250	0,554	0,472	2	33	2,740	3,697	7
	4	250	1,489	0,776	2	53	5,328	3,698	6
	5	250	3,859	1,044	3	115	9,314	3,400	7

considere o seguinte sistema politópico aleatório:

$$A(\alpha) = \alpha_1 \begin{bmatrix} 0,0699 & -0,3305 & 0,8855 \\ 1,0222 & -0,8474 & 0,3503 \\ 0,5042 & -0,2338 & 0,7260 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0,2845 & 0,9694 & 0,5347 \\ 0,5607 & -0,0698 & 0,6782 \\ -0,9108 & -0,3930 & 0,0280 \end{bmatrix} \\ + \alpha_3 \begin{bmatrix} 0,4737 & -0,3082 & 0,2987 \\ 0,9060 & -1,6707 & -0,7147 \\ -1,2058 & 1,1457 & 0,3485 \end{bmatrix}$$

A Figura 3.2 apresenta a partição do domínio incerto para localizar um caso de sistema instável no politopo para provar que o sistema politópico não é robustamente estável. Os números nos centros dos 2-simplexos (triângulos) informam em que iteração o simplexo é verificado ser robustamente estável sendo deste modo eliminado da região de busca. Observe na Figura 3.2 que, após a 4^a subdivisão, a região de busca é reduzida para apenas 1/256 da área do politopo inicial. Isto explica a razão da eficiência do procedimento proposto para localizar um caso de sistema instável dentre o número infinito de sistemas pertencentes ao politopo. Após a 6^a subdivisão, o sistema correspondente às coordenadas $\alpha = [0,0781 \ 0,1406 \ 0,7813]^T$, indicado pela seta na Figura 3.2, é verificado ser instável e o procedimento é finalizado. A distribuição dos autovalores deste sistema politópico é apresentada na Figura 3.3 onde as setas indicam a localização dos autovalores do sistema identificado com instável.

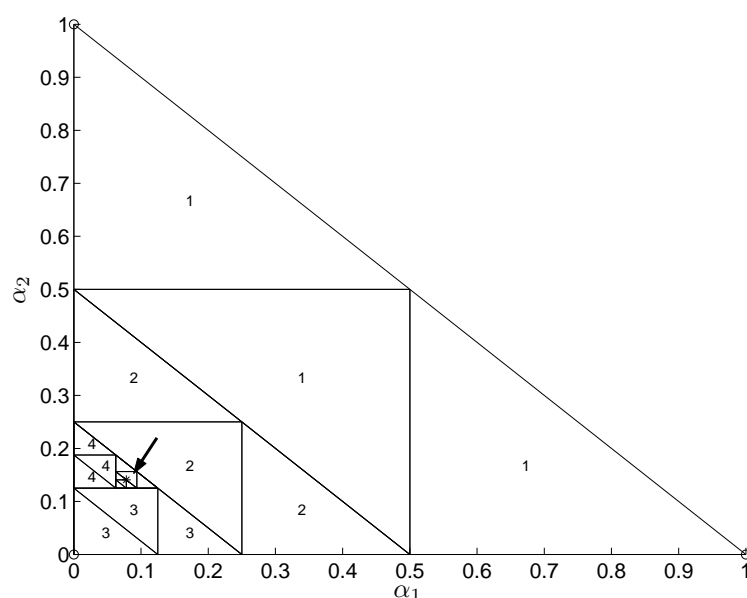


Figura 3.2: Partição do domínio incerto (2-simplexo) para localizar um sistema instável.

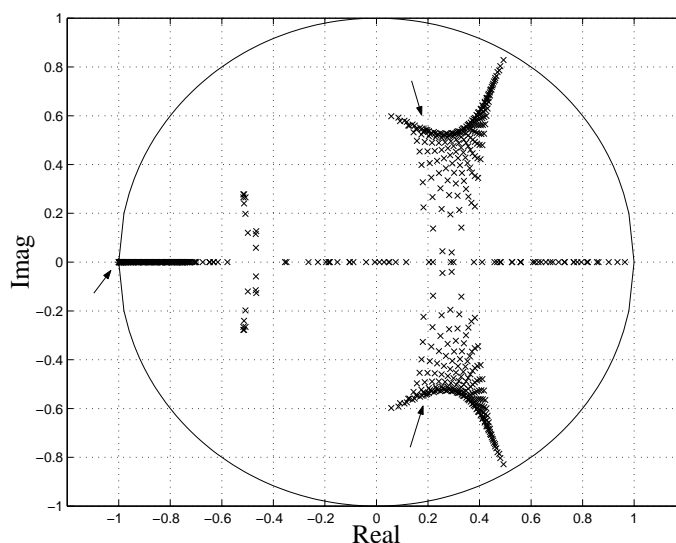


Figura 3.3: Distribuição dos autovalores do sistema politópico aleatório que não é robustamente estável.

Exemplo 3.3 Para avaliar o desempenho do procedimento de análise proposto para o caso de um domínio de incerteza na forma de um cubo, considere o seguinte sistema linear invariante no tempo, a tempo contínuo, com dependência afim de parâmetros modelado por:

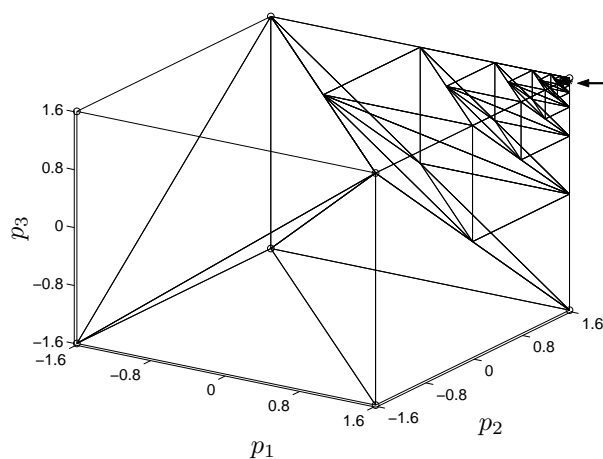
$$\begin{aligned}
 A = & \begin{bmatrix} -4,02 & -0,06 & -0,63 & -0,33 \\ -0,72 & -3,72 & 0,70 & -0,50 \\ -0,81 & -0,68 & -3,58 & 0,39 \\ -0,97 & 0,81 & 0,34 & -3,86 \end{bmatrix} + p_1 \begin{bmatrix} 0,09 & -0,17 & -0,27 & -0,28 \\ -0,56 & 0,66 & 0,47 & -0,87 \\ -0,92 & 0,28 & 0,65 & -0,99 \\ -0,74 & 0,95 & 0,13 & 0,68 \end{bmatrix} \\
 & + p_2 \begin{bmatrix} 0,26 & -0,16 & -0,82 & -0,49 \\ 0,13 & 0,66 & 0,61 & 0,61 \\ 0,09 & -0,70 & 0,80 & -0,05 \\ -0,11 & -0,46 & -0,12 & -0,67 \end{bmatrix} + p_3 \begin{bmatrix} -0,04 & -0,18 & -0,24 & 0,07 \\ -0,12 & 0,90 & 0,74 & 0,02 \\ 0,73 & 0,92 & 0,60 & -0,94 \\ -0,99 & -0,99 & 0,99 & 0,42 \end{bmatrix}
 \end{aligned}$$

Este sistema possui três parâmetros incertos que pertencem ao intervalo $[-\gamma, \gamma]$, isto é, $\Omega = \{p \in \mathbb{R}^3 : -\gamma \leq p_i \leq \gamma, \forall i\}$. O domínio de incerteza Ω é um cubo de 8 vértices no espaço de 3ª dimensão. Este sistema não é robustamente estável para $\gamma = 1,6029$. O método de análise proposto, implementado com o Lema 1 apresentado em Ramos e Peres (2002), identifica um sistema instável para $p = [1,6029 \ 1,6029 \ 1,5528]^T$ após 7 subdivisões (triangularização de Delaunay mais 6 divisões de simplexo) e 4,297s de tempo de processamento. Neste caso, nas estratégias que lidam com a divisão de hiper-retângulos, os politopos possuem 8 vértices no espaço de 3ª dimensão, ao passo que,

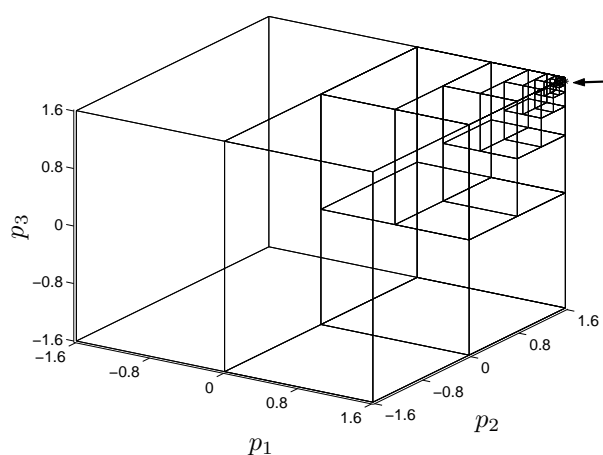
pela técnica de partição de politopo proposta, após a decomposição do hiper-retângulo em simplexes pela triangularização de Delaunay, os simplexes possuem apenas 4 vértices, contribuindo para eficiência do procedimento proposto. O método de análise proposto, implementado com o Lema 1 apresentado em Ramos e Peres (2002) e a bissecção de hiper-retângulos, identifica o mesmo sistema instável após 18 subdivisões e 29,359s de tempo de processamento, o que corrobora a maior eficiência da malha simplicial. As Figuras 3.4(a) e 3.4(b) apresentam a partição requerida do domínio de incerteza e a indicação do sistema instável identificado considerando as malhas simplicial e de hiper-retângulos, respectivamente. O lugar das raízes para este caso é apresentado na Figura 3.5. Utilizando uma técnica de grade para localizar sistemas instáveis no domínio de incerteza, é necessário utilizar um passo de variação para os parâmetros incertos igual a $\frac{2}{100}1,6029$ (com uma variação de $\frac{25}{1000}1,6029$ não se localiza nenhum ponto), o que resulta em um único vetor de coordenadas, igual ao obtido pelo procedimento proposto, após 96,875s de processamento, sem considerar o tempo necessário para identificar qual é a variação da grade adequada. Este tempo é bem maior que o tempo requerido pelo procedimento proposto que atua como uma espécie de técnica de grade “inteligente” considerando que os refinamentos são orientados. Observe que a região de instabilidade no domínio de incerteza é realmente pequena, sendo que o vértice muito próximo a ela corresponde a um sistema estável. Com uma variação na grade de $\frac{1}{100}1,6029$, localiza-se apenas mais um sistema instável nas coordenadas $p = [1,6029 \ 1,6029 \ 1,5548]^T$. A pequena região de sistemas instáveis, no caso $\{p \in \mathbb{R}^3 : p_1 = 1,6029; p_2 = 1,6029; 1,5381 \leq p_3 \leq 1,5614\}$ (0,7% do comprimento da aresta), torna a verificação bastante difícil, demonstrando como o procedimento proposto é realmente eficiente.

Para $\gamma = 1,6028$, o procedimento proposto implementado com malha simplicial identifica a estabilidade do sistema após 4 subdivisões e 3,0s de processamento ao passo que adotando malha de hiper-retângulos são necessárias 8 subdivisões e 11,344s de processamento. Para este valor de γ , apenas o Teorema 1 apresentado em Oliveira e Peres (2006), com grau $g = 2$ ($g = 3$), e o Teorema 2 apresentado em Oliveira e Peres (2006), com grau $g = 1$ ($g = 2$), entre os métodos listados no Exemplo 3.1, identificam a estabilidade robusta, porém com tempos computacionais de 22,516s (353,781s) e 61,734s ($3,1426 \times 10^3$ s), respectivamente.

Exemplo 3.4 Para avaliar o desempenho do procedimento proposto no caso de um domínio de incerteza de quinta dimensão, considere um sistema linear invariante no tempo, a tempo discreto, com



(a) Partição em uma malha simplicial.



(b) Partição em uma malha de hiper-retângulos.

Figura 3.4: Partição de um domínio de incerteza 3-d (Ex. 3.3).

dependência afim de parâmetros modelado pelas seguintes equações:

$$x(k+1) = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ 0,8456 & -0,7385 & -0,9547 & -0,7196 \\ 0,4689 & 0,3435 & -0,3114 & 0,2833 \\ -0,0763 & -0,7671 & -0,4479 & -0,5089 \end{bmatrix} x(k) + \begin{bmatrix} p_5 \\ -0,4419 \\ 0,4661 \\ 0,8099 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x(k)$$

Este sistema possui cinco parâmetros incertos variando entre os seguintes limites: $p_1 \in [0,7755, 1,0493]$, $p_2 \in [0,8149, 1,1025]$, $p_3 \in [0,6207, 0,8397]$, $p_4 \in [0,8044, 1,0884]$ e

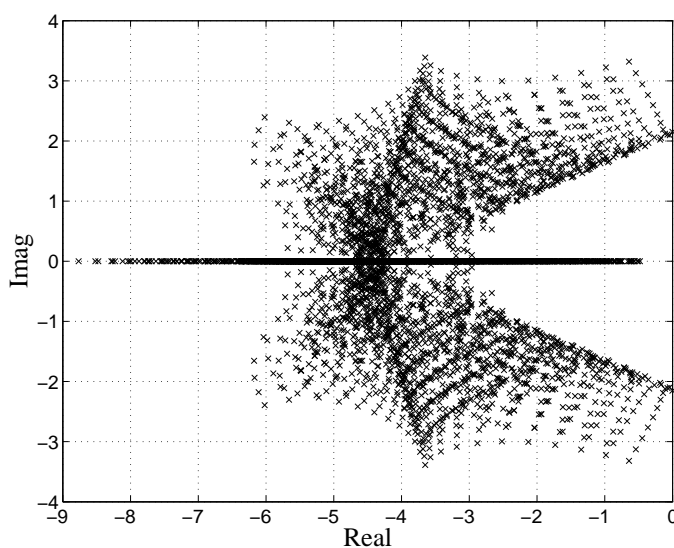


Figura 3.5: Lugar das raízes para $\gamma = 1,6029$ (Ex. 3.3).

$p_5 \in [0,2056, 0,2782]$. O domínio de incerteza é um hiper-retângulo de 32 vértices no espaço de 5ª dimensão. Com o procedimento de projeto a ser apresentado no capítulo 5, é projetado o seguinte controlador estático por realimentação de saída para estabilizar o sistema: $K = [-1,5229 \quad -0,2722]$.

Considerando a implementação no LMI Control Toolbox, o método de análise proposto, implementado com a condição suficiente baseada no Teorema 4 apresentado em Peaucelle et al. (2000), identifica a estabilidade robusta após uma iteração (decomposição do hiper-retângulo em 103 5-simplexos) e 97,672s de tempo de processamento. As formulações baseadas em estabilidade quadrática e no teorema 2 apresentado em de Oliveira, Bernussou e Geromel (1999) não são capazes de identificar a estabilidade robusta deste sistema. As formulações apresentadas em Ramos e Peres (2001) e Gao e Xue (2004) requerem 548s e 1352s, respectivamente, para identificar a estabilidade robusta deste sistema. Para o computador com a configuração citada, ambas as formulações apresentadas em Leite e Peres (2003) e Kau et al. (2005) resultaram em um erro de falta de memória na execução do “LMI solver”. As formulações baseadas em polinômios requerem tempos computacionais proibitivos para a montagem do problema.

3.4.2 Exemplos de análise de \mathcal{D} -estabilidade

Serão consideradas nos exemplos as seguintes formulações LMI de análise de \mathcal{D} -estabilidade: (QS) formulação baseada em estabilidade quadrática na forma do Teorema 2 apresentado em Peaucelle et al. (2000), com função de Lyapunov comum; (PE) formulação baseada no Teorema 4 apresentado em Peaucelle et al. (2000), (LE) formulação baseada no Teorema 1 apresentado em Leite e Peres (2003) e (GA) formulação baseada no Teorema 4 apresentado em Gao e Xue (2004), sendo

que as três últimas adotam funções de Lyapunov dependentes de parâmetros.

Exemplo 3.5 Para avaliar a eficiência do procedimento de análise proposto aplicado à análise de \mathcal{D} -estabilidade robusta, é considerada novamente uma comparação numérica exaustiva com formulações de análise baseadas em LMI. Considere a região \mathcal{D} como sendo a intersecção das seguintes regiões apresentadas na Figura 3.6: $\text{Real}(\lambda_i(A(\alpha))) < h_1 = c_r + r_r$, $|\lambda_i(A(\alpha))| < r = |c_r| + r_r$ e $\pi - |\angle \lambda_i(A(\alpha))| < \pi/4, \forall i$.

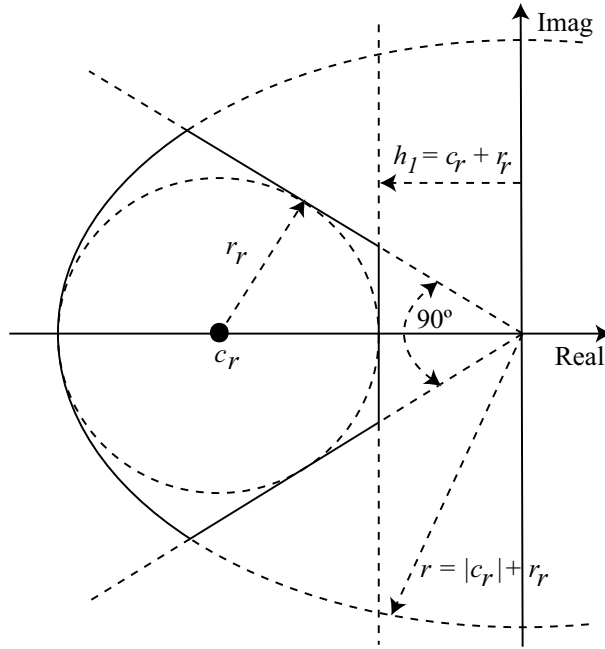


Figura 3.6: Região de análise de \mathcal{D} -estabilidade (Ex. 3.5).

Para cada par (n, N) , com $n \in [2, 5]$ e $N \in [2, 5]$, foram gerados de forma aleatória 100 sistemas politópicos, $A(\alpha) = \alpha_1 A_1 + \dots + \alpha_N A_N$, $\alpha \in \Omega_M$, robustamente \mathcal{D} -estáveis. Este caso exige um procedimento de geração de sistemas politópicos aleatórios um pouco mais complexo que o considerado em exemplos anteriores. O disco de raio r_r e centro em $(c_r, 0)$ (ver Figura 3.6) é utilizado como referência para os autovalores das matrizes A_j correspondentes aos primeiros $N - 1$ vértices do politopo. Para garantir autovalores próximos a fronteira do setor cônico, o N -ésimo vértice do politopo terá dois autovalores fixos, iguais a $\lambda_i(A_N) = -\frac{1}{\sqrt{2}}r \pm (\frac{1}{\sqrt{2}}r - 0,001)$, $i = 1, 2$, (próximo a intersecção das fronteiras das regiões disco e setor cônico) e os demais autovalores (para $n > 2$) sobre o eixo real na faixa $(-r, h_1)$. Neste exemplo é adotado $c_r = -2$ e $r_r = \frac{\sqrt{2}}{2}|c_r|$. Considere $\sigma_m \triangleq \frac{1}{n} \sum_i \text{Real}(\lambda_i(A_j))$ e $\omega_m \triangleq \max_i |\lambda_i(A_j)|$. Os sistemas aleatórios com esta característica foram gerados por meio do seguinte procedimento: 1) os primeiros $N - 1$ vértices do politopo são gerados por matrizes cujos elementos são números reais uniformemente distribuídos no intervalo $[-1, 1]$; 2) para cada vértice, a matriz A_j , $j = 1, \dots, N - 1$, é “deslocada” de modo que o valor

médio da parte real dos autovalores seja nulo, i.e $A_j = A_j - \sigma_m I_n$; 3) a matriz A_j é multiplicada por um escalar de tal modo que o módulo máximo do autovalor seja igual a $r_r - 0,001$, $A_j = (r_r - 0,001)A_j/\omega_m$; 4) a matriz A_j é novamente “deslocada” para que os autovalores se localizem dentro do disco de referência com $\sigma_m = c_r$, $A_j = A_j - c_r I_n$; 5) o N -ésimo vértice é calculado como $A_N = T^{-1}A_n T$, sendo A_n uma matriz bloco diagonal com os dois autovalores fixos próximos à intersecção do disco e do setor cônico e T uma matriz não singular com elementos aleatórios uniformemente distribuídos na faixa $[-1, 1]$; 6) o sistema politópico aleatório resultante é verificado por uma técnica de grade, com os elementos do vetor de coordenadas α variando com passo de 0,1, sendo descartados os casos com ocorrência de autovalores fora da região \mathcal{D} e 7) o sistema politópico aleatório resultante é verificado com o procedimento de análise proposto e o mesmo é incluído entre os 100 testes se for identificado como robustamente estável ou se a condição de estabilidade não for identificada considerando o número limitado de 500 iterações (hipótese que, como nos exemplos anteriores, não ocorreu nenhuma vez). Um exemplo da distribuição dos pólos de $A(\alpha)$, $\forall \alpha \in \Omega$, para $(n, N) = (5, 5)$, é mostrado na Figura 3.7.

O método de análise proposto, denominado BB nas tabelas a seguir, é implementado considerando o Teorema 4 apresentado em Peaucelle et al. (2000) para regiões disco e setor cônico e o Lema 1 apresentado em Ramos e Peres (2002) para região semi-plano (com a substituição de A_j por $(A_j - h_1 I_n)$, $j = 1, \dots, N$). A formulação baseada no Teorema 1 apresentado em Leite e Peres (2003) não é considerada neste exemplo devido ao alto custo computacional requerido e por provocar o erro “Out of memory” para $(n, N) = (5, 5)$ quando utilizado o SeDuMi.

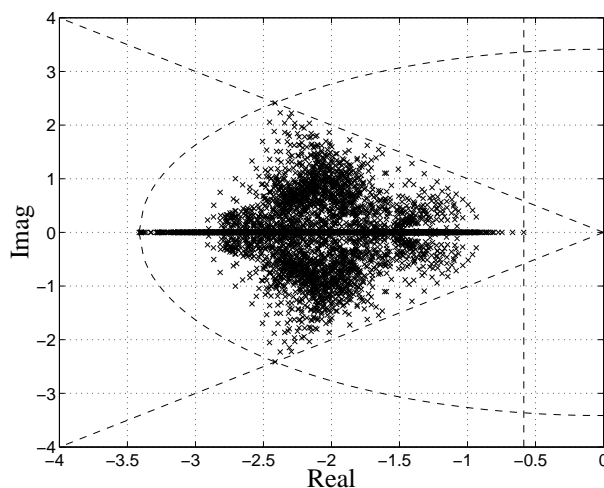


Figura 3.7: Exemplo de distribuição dos pólos para $(n, N) = (5, 5)$ (Ex. 3.5).

A Tabela 3.11 apresenta as taxas de sucesso percentuais para identificação da \mathcal{D} -estabilidade robusta para cada par (n, N) . Como nos testes anteriores, o procedimento proposto permite que con-

dições suficientes LMI identifiquem todos os sistemas analisados. A Tabela 3.12 apresenta os tempos computacionais médios, no caso das implementações no LMI Control Toolbox (LMIC) e no SeDuMi. No caso do LMIC, é verificado que o procedimento proposto requer sempre menos tempo computacional do que as condições suficientes PE e GA. Este resultado é devido à estratégia de analisar cada região individualmente ao invés de tratar todas as regiões simultaneamente em um único problema de otimização LMI. Com o SeDuMi, o procedimento de análise proposto nem sempre apresenta tempos computacionais médios inferiores aos das condições suficientes PE e GA, uma vez que as condições suficientes LMI apresentam tempos médios menores, ao passo que o procedimento proposto apresenta tempos médios maiores que os verificados nas implementações no LMI Control Toolbox para a maioria dos casos. Ainda assim, a taxa de sucesso de 100% e a capacidade de identificar sistemas não robustamente \mathcal{D} -estáveis tornam o procedimento proposto uma boa opção de ferramenta de análise.

Tabela 3.11: Taxas de sucesso (%) para identificar a \mathcal{D} -estabilidade robusta (Ex. 3.5).

n	N	QS	PE	GA	BB
2	2	15	100	90	100
	3	3	96	85	100
	4	0	81	80	100
	5	0	50	74	100
3	2	3	97	81	100
	3	1	78	73	100
	4	0	65	70	100
	5	0	40	62	100
4	2	0	94	80	100
	3	0	81	71	100
	4	0	51	61	100
	5	0	34	56	100
5	2	0	99	80	100
	3	0	72	64	100
	4	0	47	49	100
	5	0	28	51	100

Exemplo 3.6 O objetivo deste exemplo é comparar a implementação do procedimento proposto de análise de \mathcal{D} -estabilidade implementado no LMI Control Toolbox de três formas diferentes: (BB_1) utiliza o Teorema 4 apresentado em Peaucelle et al. (2000) para regiões disco e setor cônico e o Lema 1 apresentado em Ramos e Peres (2002) adaptado para região semi-plano e subdivisão de simplexo orientada pelas arestas; (BB_2) difere da anterior apenas por usar para análise da região setor cônico o Teorema 4 apresentado em Gao e Xue (2004) com a modificação de que a variável matricial Q é fixada como em Leite e Peres (2003) e (BB_3) difere da anterior apenas pelo uso do método

Tabela 3.12: Tempos de processamento médio (s) (Ex. 3.5).

n	N	LMIC				SeDuMi				
		QS	PE	GA	BB	QS	PE	GA	BB	
2	2	0,18	1,72	0,61	0,18	0,12	0,33	0,35	0,55	
	3	0,28	3,49	1,43	0,29	0,15	0,40	0,61	0,69	
	4	0,37	5,73	2,70	0,46	0,18	0,48	0,96	0,99	
	5	0,46	8,98	4,59	1,06	0,21	0,54	1,52	3,10	
	3	2	0,35	5,15	1,62	0,30	0,16	0,86	0,56	0,68
3	3	0,53	11,48	4,06	0,52	0,20	1,04	0,90	0,90	
	4	0,72	18,52	7,68	1,05	0,26	1,27	1,70	1,95	
	5	0,92	28,00	13,54	1,83	0,30	1,48	2,63	4,50	
	4	2	0,67	15,42	4,19	0,58	0,23	2,78	1,06	0,87
	3	1,02	29,48	10,00	1,10	0,31	3,25	1,78	1,31	
4	4	1,39	51,14	20,01	2,10	0,38	3,88	3,14	2,79	
	5	1,78	67,61	34,14	4,50	0,45	4,34	5,11	7,62	
	5	2	1,19	36,89	9,88	1,14	0,35	9,57	2,06	1,19
	3	1,83	76,18	23,48	2,42	0,50	11,49	3,31	1,91	
	4	2,50	122,83	47,81	4,97	0,58	13,24	5,81	4,70	
5	3,21	157,83	77,33	9,23	0,74	14,40	10,05	10,77		

da bisseção na divisão de simplexo. Foi gerado um novo conjunto de testes utilizando o mesmo procedimento do Ex. 3.5, para $n = [2, 5]$ e $N = [3, 5]$.

A Tabela 3.13 apresenta os comportamentos do procedimento de análise proposto para identificação dos sistemas robustamente \mathcal{D} -estáveis, com as três diferentes implementações, sendo T_{re} o número de sistemas politópicos identificados como robustamente estáveis, T_a o tempo computacional médio, S_a e S_m os números médio e máximo de iterações (ou subdivisões), respectivamente. A Tabela 3.14 apresenta os comportamentos do procedimento de análise proposto para identificação dos sistemas não robustamente \mathcal{D} -estáveis gerados, sendo T_{ne} o número de sistemas identificados. Não ocorreu nenhum caso de sistema aleatório que não pôde ser identificado. Comparando os resultados das implementações BB_1 e BB_2 é verificado que o procedimento proposto é mais eficiente, na maioria dos casos, considerando as condições suficientes da segunda implementação que explora totalmente a possibilidade de utilizar a formulação mais eficiente para cada tipo de região. Comparando os tempos médios das implementações BB_2 e BB_3 é verificado que a divisão de simplexo orientada pela arestas é mais eficiente do que o método da bisseção, para as dimensões consideradas, tanto para identificar sistemas robustamente estáveis como para localizar um caso de sistema instável dentro do domínio politópico de incerteza.

Tabela 3.13: Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.6)

n	N	Sistemas robustamente \mathcal{D} -estáveis									
		T_{re}	T_a (s)			S_a			S_m		
			BB ₁	BB ₂	BB ₃	BB ₁	BB ₂	BB ₃	BB ₁	BB ₂	BB ₃
2	3	100	0,30	0,26	0,26	0,03	0,01	0,01	1	1	1
	4	100	0,51	0,49	0,45	0,22	0,20	0,28	4	4	8
	5	100	1,14	0,97	0,81	0,39	0,36	1,05	2	2	16
3	3	100	0,56	0,42	0,43	0,14	0,18	0,26	1	1	2
	4	100	0,95	0,76	0,81	0,19	0,19	0,46	2	2	10
	5	100	1,91	2,09	3,88	0,33	0,36	3,05	2	2	61
4	3	100	1,09	0,71	0,76	0,12	0,20	0,36	2	3	6
	4	100	2,06	1,80	2,44	0,27	0,35	1,15	2	2	10
	5	100	4,84	4,47	9,19	0,39	0,47	3,29	2	2	52
5	3	100	2,62	1,32	1,45	0,16	0,23	0,41	1	2	4
	4	100	5,59	3,65	6,21	0,35	0,42	1,55	3	3	15
	5	100	9,92	7,30	14,62	0,47	0,48	2,69	1	2	35

Tabela 3.14: Comportamento do procedimento de análise proposto (LMIC) (Ex. 3.6)

n	N	Sistemas não robustamente \mathcal{D} -estáveis									
		T_{ne}	T_a (s)			S_a			S_m		
			BB ₁	BB ₂	BB ₃	BB ₁	BB ₂	BB ₃	BB ₁	BB ₂	BB ₃
2	3	8	1,21	1,09	1,38	4,63	4,63	9,25	8	8	16
	4	25	2,80	2,76	3,93	4,36	4,36	15,96	7	7	29
	5	16	4,95	5,14	8,15	4,13	4,13	24,56	6	6	84
3	3	7	2,48	1,47	1,90	4,14	4,14	8,29	5	5	10
	4	11	2,34	2,20	3,60	3,82	3,82	12,82	5	5	29
	5	16	8,78	8,07	17,57	4,13	4,13	30,38	6	6	77
4	3	6	8,14	3,24	4,70	4,50	4,50	10,67	7	7	24
	4	9	5,78	4,95	9,32	4,44	4,44	14,78	6	6	27
	5	19	14,87	13,00	40,12	4,42	4,47	35,00	7	7	78
5	3	12	5,38	4,21	6,49	4,67	4,67	9,25	6	6	12
	4	6	15,40	9,58	22,21	4,00	4,00	14,00	5	5	22
	5	10	22,83	17,55	46,88	3,10	3,10	20,40	6	6	65

Exemplo 3.7 Considere o modelo linearizado da dinâmica longitudinal de um avião, apresentado em Leibfritz e Lipinski (2003) (Exemplo AC1), com a inclusão neste exemplo de cinco parâmetros incertos:

$$\frac{dx}{dt} = \begin{bmatrix} 0 & 0 & 1,132 & 0 & -1 \\ 0 & -0,0538 & -0,1712 & 0 & 0,0705 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0,0485 & 0 & -9,8556 & -1,013 \\ 0 & p_1 & 0 & p_2 & -p_3 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ -0,12 & 1 & 0 \\ 0 & 0 & 0 \\ p_4 & 0 & -1,665 \\ p_5 & 0 & -0,0732 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x$$

O vetor de parâmetros incertos $p = [p_1 \ p_2 \ p_3 \ p_4 \ p_5]^T$ varia dentro do hiper-retângulo $\Omega_p = \{p \in \mathbb{R}^5 : 0,2618 \leq p_1 \leq 0,3200; 0,9479 \leq p_2 \leq 1,1585; 0,6173 \leq p_3 \leq 0,7545; 3,9771 \leq p_4 \leq 4,8609; 1,4175 \leq p_5 \leq 1,7325\}$, com 32 vértices no espaço de quinta dimensão.

Baseado no procedimento de projeto a ser apresentado no Capítulo 5, foi projetado um controlador dinâmico por realimentação de saída, de ordem reduzida, para robustamente posicionar os pólos em malha-fechada na intersecção da região semi-plano, $\text{Real}(\lambda) < -1,2$, do disco centrado na origem com raio $r = 5$, e do setor cônico com ângulo interno $\theta = \frac{\pi}{2}$, ou seja, $\lambda_i(A(\alpha)) \in \mathcal{D}, \forall i, \forall \alpha \in \Omega$, $\mathcal{D} = \{\lambda \in \mathbb{C} : \text{Real}(\lambda) < -1,2, |\lambda| < 5, |\angle \lambda| < \pi - \frac{\pi}{4}\}$:

$$A_c = -4,3978 \quad B_c = \begin{bmatrix} 0,0027 & 1,0141 & -0,0855 \end{bmatrix}$$

$$C_c = \begin{bmatrix} 0,2744 \\ -1,4964 \\ 0,9079 \end{bmatrix} \quad D_c = \begin{bmatrix} 4,9594 & 0,0950 & -3,4659 \\ -0,1940 & -4,1934 & -0,4711 \\ -2,8859 & -1,3523 & 9,2776 \end{bmatrix}$$

A distribuição dos pólos do sistema de 6^a ordem em malha-fechada é apresentada na Figura 3.8 para os parâmetros variando nos intervalos específicos (32 vértices mais 500 pontos aleatórios).

O objetivo deste exemplo é comparar as condições suficientes LMI com diferentes implementações do procedimento proposto no LMI Control Toolbox. Neste caso, para a configuração de computador utilizada, com a formulação baseada no Teorema 1 apresentado em Leite e Peres (2003), na análise do politopo inicial, a função *feasp*(\cdot) do LMI Control Toolbox gera um erro não tratável de memória, quando a mesma muda da técnica baseada em Cholesky para a técnica QR (Gahinet

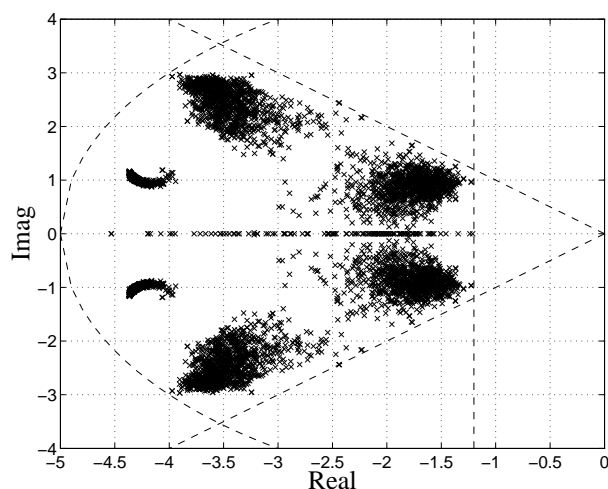


Figura 3.8: Posicionamento de pólos para $A \in \mathcal{P}$ (Ex. 3.7).

et al., 1995). Com a partição do hiper-retângulo em um conjunto de simplexos, devido à redução do número de vértices, o erro de memória deixa de ocorrer. Deste modo, o procedimento de análise de \mathcal{D} -estabilidade só pode ser usado com esta formulação se a iteração 0 não for processada.

A Tabela 3.15 compara o desempenho do método de análise proposto para identificar \mathcal{D} -estabilidade robusta implementado com três diferentes formulações LMI de análise. As duas primeiras formulações LMI não conseguem determinar se o sistema em malha-fechada com o controlador apresentado é ou não robustamente \mathcal{D} -estável. A última condição LMI é capaz de determinar a \mathcal{D} -estabilidade robusta do sistema porém com tempo computacional total bem superior ao procedimento proposto implementado com as duas primeiras formulações LMI mais simples. Este tempo total ainda seria bem maior se as três regiões fossem analisadas simultaneamente. Deste modo, os resultados da Tabela 3.15 indicam novamente que o uso do algoritmo tipo “branch-and-bound” não significa necessariamente menos eficiência em relação às formulações LMI mais complexas, mesmo para domínios de incertezas em espaços de maior dimensão e com maior número de vértices. Observe também que uma única formulação não apresenta menores tempos computacionais para todas as regiões testadas, o que torna mais interessante a possibilidade de se implementar o procedimento de análise proposto com diferentes condições LMI para cada tipo de região. Usando o Lema 1 apresentado em Ramos e Peres (2002), adaptado para região semi-plano, são necessários 1 minuto e 12 segundos para verificar a região disco e considerando o Teorema 4 apresentado em Gao e Xue (2004), com a variável matricial Q fixada como em Leite e Peres (2003), são necessários 1 hora, 6 minutos e 47 segundos para análise da região setor cônico.

Tabela 3.15: Desempenho do procedimento de análise de \mathcal{D} -estabilidade robusta (Ex. 3.7).

Região	Formulação LMI	QS	PE	GA
semi-plano	N. de partições	229	0	0
	Tempo computacional	15min18s	3min41s	6min37s
disco	N. de partições	34	0	0
	Tempo computacional	3min48s	7min58s	2h19min15s
Setor	N. de partições	40	1	0
	Tempo computacional	19min33s	2h10min32s	1h53min23s
Tempo computacional total		38min40s	2h22min11s	4h29min15s

3.5 Conclusões

Neste capítulo foi apresentado um procedimento de análise para estabelecer se um sistema linear invariante no tempo incerto, com modelo de incerteza politópico ou dependente afim de parâmetros, é robustamente \mathcal{D} -estável ou não. Devido ao conservadorismo das formulações LMI de análise que são apenas condições suficientes, quando não é obtida uma solução para o problema de factibilidade, não é possível afirmar que o sistema não é robustamente \mathcal{D} -estável. É verificado através de testes exaustivos que as condições suficientes LMI, com matriz de Lyapunov comum ou dependente de parâmetros, perde eficácia com o aumento da ordem do sistema e do número de vértices do domínio de incerteza na forma de politopo. O método proposto particiona o politopo de forma iterativa até que todos os subpolitopos atendam à condição suficiente de \mathcal{D} -estabilidade robusta ou até que seja encontrada uma coordenada do politopo que corresponde a um sistema que possui pólos fora da região \mathcal{D} desejada. Apesar da complexidade de implementação, o método de análise proposto, combinado com formulações LMI mais simples, pode requerer menos tempo computacional e menos disponibilidade de memória que formulações LMI mais complexas, que resultem em número excessivo de variáveis de decisão e de restrições LMI no problema de factibilidade. Apesar de apresentar baixos tempos computacionais em alguns casos, em geral, formulações baseadas em estabilidade quadrática (matriz de Lyapunov comum para todas as regiões e vértices) requerem um número elevado de partições do domínio de incerteza podendo levar a interrupção do procedimento proposto devido a um critério de parada baseado em número máximo de iterações do algoritmo. Deste modo, é mais interessante implementar o procedimento de análise proposto com condições suficientes baseadas em funções de Lyapunov dependentes de parâmetros com melhor compromisso entre complexidade e conservadorismo. O procedimento de análise proposto corresponde, na prática, a uma condição necessária e suficiente apesar da possibilidade de se atingir um número limite de iterações sem a identificação da condição de \mathcal{D} -estabilidade. As condições suficientes LMI baseadas em funções de Lyapunov com dependência polinomial homogênea dos parâmetros possuem a característica de redução do conservadorismo com o aumento do grau do polinômio, porém apresentam alta complexidade computacional

com números de variáveis escalares de otimização e número de linhas de LMIs proibitivos no caso de número de vértices elevado.

Como contribuição secundária na área de análise de \mathcal{D} -estabilidade, é verificado que é mais eficiente realizar a análise de cada região individualmente do que efetuar a análise conjunta em um único problema de factibilidade baseado em LMI. Com esta estratégia, os problemas isolados são mais simples, com menos variáveis escalares de decisão e menos restrições.

A principal dificuldade na implementação do método de análise proposto é a operação de partição do polítopo com formato geral, não restrito a um hiper-retângulo, em espaços de qualquer dimensão. Este problema é solucionado usando a combinação da triangularização de Delaunay com a nova técnica de divisão de simplexo orientada pelas arestas apresentada no Capítulo 2.

Capítulo 4

Cômputo dos Custos \mathcal{H}_2 e \mathcal{H}_∞

4.1 Normas de sinais e sistemas

Uma das formas de calcular a norma de um sinal contínuo (discreto) no domínio do tempo é pela norma \mathcal{L}_2 (l_2) definida como

$$\|w(t)\|_2 \triangleq \sqrt{\int_{-\infty}^{\infty} w(t)^2 dt} \quad (\text{sistema a tempo contínuo}) \quad (4.1)$$

$$\|w(k)\|_2 \triangleq \sqrt{\sum_{k=-\infty}^{\infty} w(k)^2} \quad (\text{sistema a tempo discreto}) \quad (4.2)$$

Se esta integral (soma) é finita, o sinal é dito ser de quadrado integrável (somável), representado por $w(t) \in \mathcal{L}_2$ ($w(k) \in l_2$), o que pode ser interpretado fisicamente como um sinal com energia limitada.

Seja $T_{zw}(\lambda) = D + C(\lambda I - A)^{-1}B$, λ representando s ou z , a matriz de transferência relacionando o vetor de entradas exógenas w e o vetor de saídas controladas z (o significado duplo da letra z fica claro de acordo com o contexto). Duas normas de matrizes de transferência T_{zw} de interesse são as normas \mathcal{H}_2 e \mathcal{H}_∞ . A norma \mathcal{H}_2 de uma matriz de transferência, estritamente própria, de um sistema linear invariante no tempo estável a tempo contínuo, é definida como

$$\|T_{zw}\|_2 \triangleq \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Tr} [T_{zw}^*(j\omega)T_{zw}(j\omega)] d\omega} \quad (4.3)$$

sendo $(\cdot)^*$ a transposta conjugada e $\text{Tr}(\cdot)$ o traço do argumento.

No caso de sistemas lineares invariantes no tempo estáveis a tempo discreto, a norma é definida

como

$$\|T_{zw}\|_2 \triangleq \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} \text{Tr} [T_{zw}^*(e^{j\omega}) T_{zw}(e^{j\omega})] d\omega} \quad (4.4)$$

A norma \mathcal{H}_∞ de uma matriz de transferência de um sistema estável, corresponde ao pico do ganho da resposta em frequência, sendo definida como

$$\|T_{zw}\|_\infty \triangleq \sup_{\omega \in \mathbb{R}} \bar{\sigma}[T_{zw}(j\omega)] \quad (\text{sistema a tempo contínuo}) \quad (4.5)$$

$$\|T_{zw}\|_\infty \triangleq \sup_{\omega \in [0, 2\pi]} \bar{\sigma}[T_{zw}(e^{j\omega})] \quad (\text{sistema a tempo discreto}) \quad (4.6)$$

sendo $\bar{\sigma}(\cdot)$ o valor singular máximo do argumento.

Estas duas normas de matrizes de transferência possuem interpretações relacionando os sinais de entrada e saída no domínio do tempo:

- Para uma entrada $w(t)$ na forma de um processo ruído branco com variância e intensidade unitárias, a variância do sinal de saída $z(t)$ em regime estacionário é dada por $\|T_{zw}\|_2$:

$$\|T_{zw}\|_2 = \lim_{T \rightarrow \infty} E \left\{ \frac{1}{T} \int_0^T z(t)^T z(t) dt \right\} \quad (4.7)$$

sendo $E\{\cdot\}$ é a esperança matemática do argumento.

- O ganho \mathcal{L}_2 ou ganho RMS de um sistema estável linear invariante no tempo, correspondendo ao maior ganho entre a entrada e saída sobre todos os sinais de entrada limitados $w(t) \in \mathcal{L}_2$, é dado por

$$\|T_{zw}\|_\infty = \max_{w \in \mathcal{L}_2, w \neq 0} \frac{\|z\|_2}{\|w\|_2} \quad (4.8)$$

4.2 Cálculos de normas e custos

Existem vários métodos disponíveis para o cálculo das normas \mathcal{H}_2 e \mathcal{H}_∞ no caso de sistemas precisamente conhecidos. A norma \mathcal{H}_2 pode ser calculada exatamente através de um número finito de operações em termo dos grammianos de controlabilidade ou observabilidade: $\|T_{wz}\|_2^2 = \text{Tr}(C X_c C^T) = \text{Tr}(B^T X_o B)$, sendo X_c e X_o os grammianos de controlabilidade e observabilidade, respectivamente, que podem ser calculados resolvendo as seguintes equações de Lyapunov:

$$A X_c + X_c A^T + B B^T = 0 \quad (4.9)$$

$$X_o A + A^T X_o + C^T C = 0 \quad (4.10)$$

A norma \mathcal{H}_∞ pode ser calculada com precisão adequada por meio de procedimentos iterativos (ver por exemplo (Boyd et al., 1989)). A norma \mathcal{H}_∞ pode ser calculada pela busca linear do valor mínimo de γ tal que a matriz Hamiltoniana não possui autovalores sobre o eixo imaginário (Zhou e Doyle, 1998):

$$H = \begin{bmatrix} A + BR^{-1}D^TC & BR^{-1}B^T \\ -C^T(I + DR^{-1}D^T)C & -(A + BR^{-1}D^TC)^T \end{bmatrix}$$

sendo $R \triangleq \gamma^2 - D^TD$.

Entretanto, para sistemas incertos, não existe atualmente nenhum método para determinar o valor exato do limitante para a norma no conjunto incerto. Existem estratégias de cálculo dos limitantes, denominados custos garantidos \mathcal{H}_2 e \mathcal{H}_∞ , baseadas em formulações por desigualdades matriciais lineares (LMIs). As primeiras formulações por LMIs foram baseadas no conceito de estabilidade quadrática (Palhares et al., 1997), mas o uso de uma única função de Lyapunov para todo o domínio de incerteza resulta geralmente em resultados conservadores. Para reduzir o conservadorismo, nos últimos anos foram publicados vários trabalhos que adotaram o uso de funções de Lyapunov dependentes de parâmetros, variáveis matriciais extras e/ou parâmetros de sintonia, como por exemplo em Apkarian et al. (2001), de Oliveira et al. (2002), de Oliveira et al. (2004a), de Oliveira et al. (2004b), Ebihara e Hagiwara (2004a), Xie et al. (2004), Trofino et al. (2005) e He et al. (2005) e as referências por eles citadas. Em Ebihara et al. (2005) são propostas condições LMI para análise robusta de desempenho de sistemas contínuos no tempo considerando funções de Lyapunov associadas com derivadas de alta ordem do vetor de estado. Contudo, os valores obtidos por estas estratégias são apenas limites superiores dos custos exatos e a precisão dos resultados obtidos pode variar consideravelmente de um caso para outro. Além disso, para reduzir o conservadorismo, as novas formulações estão cada vez mais complexas, requerendo maior esforço computacional, tanto em termos de tempo de processamento quanto em requisito de memória do computador. Recentemente foram apresentadas novas formulações baseadas em funções de Lyapunov quadráticas com dependência polinomial homogênea de grau arbitrário nos parâmetros (Chesi et al., 2005a; Oliveira e Peres, 2005b). A vantagem desta abordagem é que a precisão do custo garantido pode ser melhorada com o aumento do grau da dependência polinomial de parâmetros. Entretanto, a complexidade destas formulações aumenta rapidamente com o número de vértices do domínio politópico de incerteza e com o grau do polinômio.

Neste capítulo será apresentado um procedimento de cálculo dos custos \mathcal{H}_2 e \mathcal{H}_∞ , com precisão especificada, que possa ser usado como ferramenta de análise de desempenho de sistemas dinâmicos incertos, com domínio politópico de incerteza, representados por modelos politópicos ou modelos com dependência afim de parâmetros.

Considere o sistema linear invariante no tempo representado pelo modelo no espaço de estados a

seguir:

$$\begin{aligned}\delta[x(\tau)] &= Ax(\tau) + Bw(\tau) \\ z(\tau) &= Cx(\tau) + Dw(\tau)\end{aligned}\tag{4.11}$$

com $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n_w}$, $C \in \mathbb{R}^{n_z \times n}$ e $D \in \mathbb{R}^{n_z \times n_w}$, sendo

$$\begin{aligned}\delta[x(\tau)] &\triangleq \frac{dx(t)}{dt}, \quad \tau \triangleq t \in \mathbb{R} \quad \text{para sistemas contínuos no tempo} \\ \delta[x(\tau)] &\triangleq x(k+1), \quad \tau \triangleq k \in \mathbb{N} \quad \text{para sistemas discretos no tempo}\end{aligned}$$

Seja S a matriz sistema definida como:

$$S \triangleq \begin{bmatrix} A & B \\ C & D \end{bmatrix}\tag{4.12}$$

Considere que a matriz sistema S não é precisamente conhecida, mas pertence a um conjunto poliédrico convexo fechado, ou politopo: $S \in \mathcal{P}$.

No caso de modelos politópicos, o conjunto \mathcal{P} é um politopo no espaço de matrizes definido pelo conjunto de todas matrizes obtidas pela combinação convexa de seus N vértices:

$$\mathcal{P} \triangleq \left\{ S(\theta) : S(\theta) = \sum_{i=1}^N \theta_i S_i, \theta \in \Omega_M \right\}\tag{4.13}$$

sendo

$$S_i \triangleq \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \quad i = 1, \dots, N\tag{4.14}$$

os vértices do politopo. Como discutido nos capítulos anteriores, o conjunto Ω_M pode ser representado como um simplexo no espaço de dimensão $N - 1$:

$$\Omega_M \triangleq \left\{ \hat{\theta} \in \mathbb{R}^{N-1} : \theta_i \geq 0, i = 1, \dots, N-1, \sum_{i=1}^{N-1} \theta_i \leq 1 \right\}\tag{4.15}$$

com $\theta_N = 1 - \sum_{i=1}^{N-1} \theta_i$.

No caso de sistemas dependentes de parâmetros, a matriz sistema é dependente afim do vetor de parâmetros incertos $p = [p_1, p_2, \dots, p_d]^T \in \mathbb{R}^d$:

$$\mathcal{P} \triangleq \{S(p) : S(p) = S_0 + p_1 S_1 + \dots + p_d S_d, p \in \Omega_p\}\tag{4.16}$$

Se os parâmetros incertos variam entre valores limites, i.e. $p_i \in [\underline{p}_i, \bar{p}_i]$, sendo \underline{p}_i e \bar{p}_i os valores mínimo e máximo do i -ésimo parâmetro incerto, o vetor p pertence a um hiper-retângulo no espaço d -dimensional:

$$\Omega_p \triangleq \left\{ p \in \mathbb{R}^d : \underline{p}_i \leq p_i \leq \bar{p}_i, i = 1, \dots, d \right\} \quad (4.17)$$

O domínio Ω_p também pode ser um polítopo de formato qualquer se existirem restrições lineares adicionais sobre os parâmetros.

Para a análise de sistemas representados por modelos politópicos ou por modelos com dependência afim de parâmetros, este trabalho propõe uma estratégia para o cálculo dos valores máximos das normas \mathcal{H}_2 e \mathcal{H}_∞ (ou custos \mathcal{H}_2 e \mathcal{H}_∞) com uma precisão desejada, baseada no algoritmo branch-and-bound (BnB). Como discutido na introdução do Capítulo 2, o uso de algoritmos tipo branch-and-bound na área de controle robusto não é novidade, já tendo sido utilizado inclusive para cálculo de desempenho (Balemi et al., 1991). Retomar o uso do algoritmo BnB para a aplicação em tela é justificável considerando o aumento do poder computacional em relação ao início dos anos 90 e a publicação de formulações convexas para o cálculo de limitantes superiores para os custos, eficientes com relação ao aspecto computacional. Será apresentada de forma detalhada uma implementação do algoritmo branch-and-bound cuja operação de partição do domínio será baseada na triangularização de Delaunay e na subdivisão de simplexo orientada pelas arestas, descrita em detalhes no Capítulo 2. Diferente das aplicações anteriores do algoritmo BnB, tal estratégia de partição permite estender a aplicação deste algoritmo para domínios de incerteza não restritos ao caso do hiper-retângulo. Tal extensão permite considerar os casos de modelos politópicos ou modelos com dependência afim de parâmetros com restrições adicionais.

Os primeiros resultados utilizando esta estratégia foram apresentados em Gonçalves, Palhares, Takahashi e Mesquita (2004). Parte dos resultados apresentados neste capítulo também pode ser vista em Gonçalves, Palhares, Takahashi e Mesquita (2006c). Esta estratégia também já foi utilizada para determinação do nível de atenuação de distúrbio \mathcal{H}_∞ de sistemas lineares sujeitos a incertezas politópicas e retardo no tempo (Gonçalves, Campos, Ekel, Palhares, Takahashi e Mesquita, 2004; Gonçalves, Bastos, Palhares, Takahashi, Mesquita, Campos e Ekel, 2005).

4.3 O Algoritmo Branch-and-Bound

O algoritmo branch-and-bound (BnB) pode ser utilizado para encontrar o máximo global de uma função $f(\alpha) : \mathbb{R}^d \rightarrow \mathbb{R}$, em um domínio $\alpha \in \Omega \subset \mathbb{R}^d$, garantindo uma precisão especificada desde que a taxa de variação de $f(\alpha)$ e o domínio Ω sejam limitados. O algoritmo branch-and-bound considera o princípio de que se $f(\alpha)$ é lipshitziana com constante L positiva, tal que $|f(\alpha_1) - f(\alpha_2)| \leq L\|\alpha_1 - \alpha_2\|$, $\alpha_1, \alpha_2 \in \Omega$, então se $\bigcup_{i=1}^s Sph(\alpha_i, \epsilon/L) \supseteq \Omega$, com $Sph(c, r)$ significando a hipersfera

com centro c e raio r , então $\max_{1 \leq i \leq s} f(\alpha_i)$ estima o máximo global com precisão ϵ (Clausen e Zilinskas, 2002).

A descrição do algoritmo BnB apresentada aqui é adaptada de Balakrishnan et al. (1991). Na aplicação que será considerada neste trabalho, o domínio Ω é um politopo. Para um domínio $\Omega_i \subseteq \Omega$, pode-se definir

$$\Phi_{max}(\Omega_i) \triangleq \max_{\alpha \in \Omega_i} f(\alpha) \quad (4.18)$$

O algoritmo BnB calcula $\Phi_{max}(\Omega)$ baseado em duas funções, $\Phi_{li}(\Omega_i)$ e $\Phi_{ls}(\Omega_i)$, definidas sobre $\{\Omega_i : \Omega_i \subseteq \Omega\}$. Estas duas funções devem satisfazer as seguintes condições:

$$\Phi_{li}(\Omega_i) \leq \Phi_{max}(\Omega_i) \leq \Phi_{ls}(\Omega_i) \quad (4.19)$$

$$\forall \epsilon > 0, \exists \delta > 0 \text{ tal que } \forall \Omega_i \subseteq \Omega, \dim(\Omega_i) \leq \delta \Rightarrow \Phi_{ls}(\Omega_i) - \Phi_{li}(\Omega_i) \leq \epsilon \quad (4.20)$$

A condição (4.19) estabelece que as funções $\Phi_{li}(\Omega_i)$ e $\Phi_{ls}(\Omega_i)$ calculam os limites inferior e superior de $\Phi_{max}(\Omega_i)$, respectivamente. A condição (4.20) estabelece que, quando a máxima distância entre os vértices de Ω_i , denotado por $\dim(\Omega_i)$, tende a zero, a diferença entre os limites inferior e superior converge para zero.

O algoritmo BnB inicia-se pelo cálculo de $\Phi_{li}(\Omega)$ e $\Phi_{ls}(\Omega)$. Se $(\Phi_{ls} - \Phi_{li})/\Phi_{li} \leq \epsilon$, sendo ϵ uma precisão relativa pré-especificada, então o algoritmo finaliza. Se o critério de parada não é atingido, é necessário particionar o politopo Ω em subpolitopos menores de forma que $\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_s$, e calcular $\Phi_{li}(\Omega_i)$ e $\Phi_{ls}(\Omega_i)$, $i = 1, \dots, s$. Então

$$\max_{1 \leq i \leq s} \Phi_{li}(\Omega_i) \leq \Phi_{max}(\Omega) \leq \max_{1 \leq i \leq s} \Phi_{ls}(\Omega_i)$$

fornece novos limites para $\Phi_{max}(\Omega)$. Se a diferença relativa entre os novos limites é menor ou igual a ϵ , o algoritmo finaliza. Caso contrário, a partição de Ω é refinada e novos limites são calculados. O algoritmo BnB converge uma vez que, pelas sucessivas partições, $\dim(\Omega_i)$, $i = 1, \dots, s$, tende para zero (subpolitopo tende para um ponto), fazendo com que $\Phi_{ls}(\Omega_i) - \Phi_{li}(\Omega_i)$ tenda para zero.

A versão do algoritmo BnB utilizada, adaptada de (Balakrishnan et al., 1991), é apresentada a seguir:

Algoritmo: Branch-and-Bound

$k \leftarrow 0$;
 $\mathcal{L}_0 \leftarrow \{\Omega\}$;
 $L_0 \leftarrow \Phi_{li}(\Omega)$;
 $U_0 \leftarrow \Phi_{ls}(\Omega)$;
enquanto $(U_k - L_k)/L_k > \varepsilon$
 selecione $\Omega_u \in \mathcal{L}_k$ tal que $\Phi_{ls}(\Omega_u) = U_k$;
 particione Ω_u em $\Omega_1, \dots, \Omega_s$;
 $\mathcal{L}_{k+1} \leftarrow \{\mathcal{L}_k - \Omega_u\} \cup \{\Omega_1, \dots, \Omega_s\}$;
 $L_{k+1} \leftarrow \max_{\Omega_i \in \mathcal{L}_{k+1}} \Phi_{li}(\Omega_i)$;
 $U_{k+1} \leftarrow \max_{\Omega_i \in \mathcal{L}_{k+1}} \Phi_{ls}(\Omega_i)$;
 elimine todo $\Omega_i \in \mathcal{L}_{k+1}$ tal que $\Phi_{ls}(\Omega_i) < L_{k+1}$;
 $k \leftarrow k + 1$;
fim enquanto
fim algoritmo

4.4 O Algoritmo BnB Aplicado ao Cálculo dos Custos \mathcal{H}_2 e \mathcal{H}_∞ com a Precisão Requerida

4.4.1 Escolha das funções limitantes

Seja $T_{zw}(\lambda, \alpha) = C(\alpha)(\lambda I - A(\alpha))^{-1}B(\alpha) + D(\alpha)$ a matriz de transferência de w para z , com λ representando s ou z e $\alpha \in \Omega$, sendo que α corresponde a $\hat{\theta}$ ou p , e Ω representa Ω_M ou Ω_p . Por conveniência, a dependência de T_{zw} de λ e α poderá ser omitida a partir deste ponto. O problema considerado é o cálculo do valor máximo da norma \mathcal{H}_q , $q \in \{2, \infty\}$, da matriz de transferência $T_{zw}(\lambda, \alpha)$, para todo $\alpha \in \Omega$. Seja $\delta_{p.c.}$ e $\gamma_{p.c.}$ os valores máximos das normas \mathcal{H}_2 e \mathcal{H}_∞ no domínio de incerteza:

$$\delta_{p.c.} \triangleq \max_{\alpha \in \Omega} \|T_{zw}(\lambda, \alpha)\|_2 \quad (4.21)$$

$$\gamma_{p.c.} \triangleq \max_{\alpha \in \Omega} \|T_{zw}(\lambda, \alpha)\|_\infty \quad (4.22)$$

Os custos garantidos \mathcal{H}_2 e \mathcal{H}_∞ , $\delta_{c.g.}$ e $\gamma_{c.g.}$, calculados através de formulações baseadas em LMIs, são apenas um limite superior para o valor máximo das normas no domínio de incerteza, sem nenhuma

informação a respeito da diferença entre estes valores:

$$\max_{\alpha \in \Omega} \|T_{zw}(\lambda, \alpha)\|_2 \leq \delta_{c.g.} \quad (4.23)$$

$$\max_{\alpha \in \Omega} \|T_{zw}(\lambda, \alpha)\|_\infty \leq \gamma_{c.g.} \quad (4.24)$$

Uma vez que o cálculo exato do valor máximo da norma é um problema de difícil solução, neste trabalho é apresentado um método para o cálculo dos custos ϵ -garantidos \mathcal{H}_2 e \mathcal{H}_∞ , δ_c e γ_c , que são definidos como os valores que atendem as seguintes desigualdades:

$$\delta_{p.c.} \leq \delta_c \leq (1 + \epsilon)\delta_{p.c.} \quad (4.25)$$

$$\gamma_{p.c.} \leq \gamma_c \leq (1 + \epsilon)\gamma_{p.c.} \quad (4.26)$$

Para aplicar o algoritmo BnB no cálculo dos custos \mathcal{H}_q , $q \in \{2, \infty\}$, com qualquer precisão requerida, é necessário encontrar as funções $\Phi_{li}(\Omega_i)$ e $\Phi_{ls}(\Omega_i)$ que satisfazem as condições (4.19) e (4.20). Considere $T_{zw,i} = C_i(\lambda I - A_i)^{-1}B_i + D_i$, $i = 1, \dots, N$, as matrizes de transferência dos vértices do politopo $\Omega_i \subseteq \Omega$, sendo N o número de vértices correspondente. A função limite inferior pode ser definida como o pior caso da norma \mathcal{H}_q calculada nos vértices de Ω_i :

$$\Phi_{li}(\Omega_i) = \max_{1 \leq i \leq N} \|T_{zw,i}\|_q \quad (4.27)$$

A função limite superior $\Phi_{ls}(\Omega_i)$ pode ser definida como sendo qualquer formulação LMI para cálculo dos custos garantidos \mathcal{H}_q (Palhares et al., 1997). Como no procedimento proposto de análise de \mathcal{D} -estabilidade robusta, é claro que a eficiência do algoritmo branch-and-bound dependerá da escolha da formulação LMI com melhor compromisso entre complexidade e conservadorismo.

Com esta escolha de funções limitantes, o algoritmo branch-and-bound irá combinar a redução do conservadorismo das formulações LMI pela partição do domínio de incerteza (função limitante superior) com uma técnica “inteligente” de grade (função limitante inferior) em que o refinamento da grade ocorre apenas no subpolitopo com maior valor de custo garantido. O custo computacional relacionado com o cálculo do custo \mathcal{H}_q pelas duas técnicas simultaneamente é justificado pela disponibilidade da informação da precisão do cálculo.

4.4.2 Técnica de partição do politopo

No procedimento proposto de cálculo de custo ϵ -garantido baseado no algoritmo branch-and-bound é aplicada a mesma técnica de partição de politopos utilizada no procedimento de análise de \mathcal{D} -estabilidade robusta, descrito no Capítulo 3, que combina triangularização de Delaunay (decom-

posição do politopo em um conjunto de simplexos) e subdivisão de simplexo orientada pelas arestas, descrita em detalhes no Capítulo 2.

4.5 Complexidade do Procedimento de Análise Proposto

As Figuras 4.1 a 4.6 apresentam uma simulação do algoritmo branch-and-bound para o cálculo do custo \mathcal{H}_2 ou \mathcal{H}_∞ . No passo inicial são calculadas as normas nos vértices do politopo e o custo garantido para o politopo (Figura 4.1). Neste caso o limite inferior é o maior valor de norma nos vértices, ou seja 10, e o limite superior é o custo garantido igual a 100. Na primeira partição (Figura 4.2), como o retângulo não é um simplexo, a divisão é realizada pela triangularização de Delaunay que divide o retângulo em dois triângulos sem acrescentar novos vértices. Deste modo o valor do limite inferior se mantém mas o valor do limite superior passa a ser o maior valor de custo garantido obtido como sendo 80.

A partir deste ponto, as divisões ocorrem nos triângulos com maior custo garantido. A divisão do triângulo pela técnica proposta gera quatro novos triângulos e três novos vértices (Figura 4.3). Os valores dos limites são atualizados para o novo valor máximo da norma igual a 16 e o novo valor máximo de custo garantido igual a 60. Observe que o triângulo com custo garantido 12 pode ser descartado do espaço de busca já que o maior valor de norma é superior a este valor. As divisões prosseguem até que a diferença relativa entre os limites superior e inferior atinja a precisão desejada (Figuras 4.4 a 4.6).

Como discutido anteriormente, observe o refinamento “inteligente” da grade, sendo que o maior número de pontos testados é concentrado na região onde existe a possibilidade de encontrar o valor máximo. Outro dado importante informado por este método de análise é a coordenada em que ocorre o valor máximo da norma, que será bastante útil no procedimento de projeto a ser apresentado no Capítulo 5.

No procedimento de cálculo do custo com precisão especificada, quando o politopo inicial não é um simplexo, na primeira partição do politopo, realizada pela triangularização de Delaunay, não é acrescentado nenhum novo vértice e o número de simplexos gerados dependerá, não somente da dimensão do espaço e do número de vértices, como também da distribuição espacial destes vértices. A decomposição do politopo em simplexos, que correspondem ao politopo com menor número de vértices para uma dada dimensão, facilita o cálculo das funções limitantes inferior e superior tornando o algoritmo muito mais eficiente. No caso da função limitante superior, baseada no cálculo do custo garantido, este fato é mais evidente no caso de formulações LMI baseadas em funções de Lyapunov dependentes de parâmetros, para as quais, não só o número de restrições LMI, mas também o número de variáveis escalares de otimização, dependem do número de vértices do politopo.

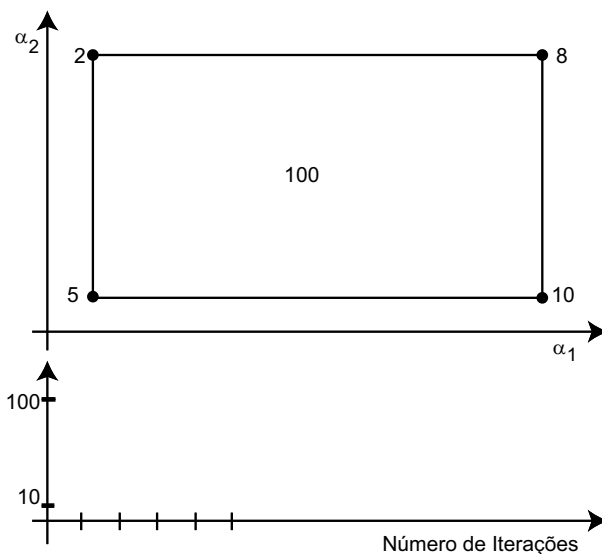


Figura 4.1: Simulação do algoritmo BnB - iteração 0.

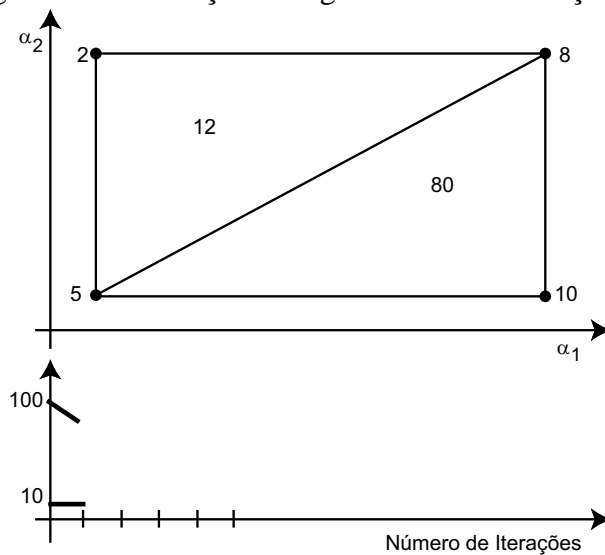


Figura 4.2: Simulação do algoritmo BnB - iteração 1.

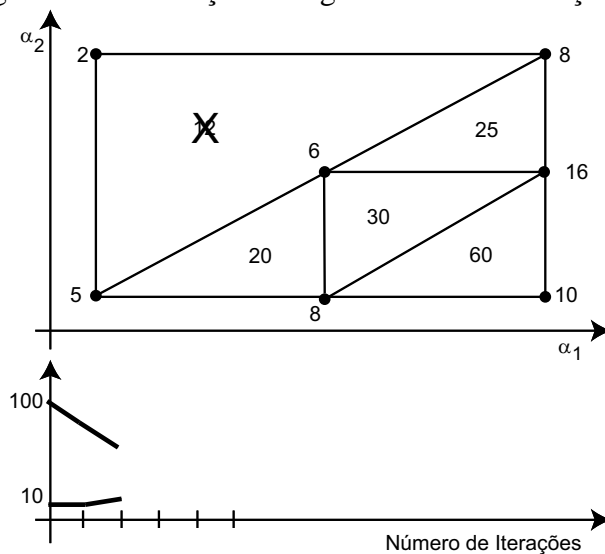


Figura 4.3: Simulação do algoritmo BnB - iteração 2.

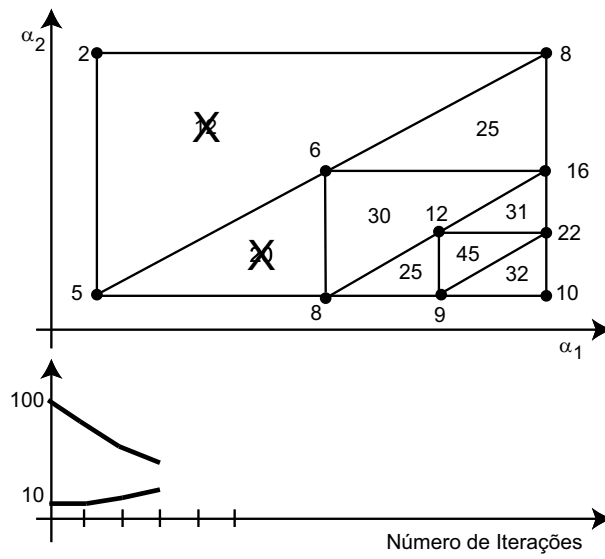


Figura 4.4: Simulação do algoritmo BnB - iteração 3.

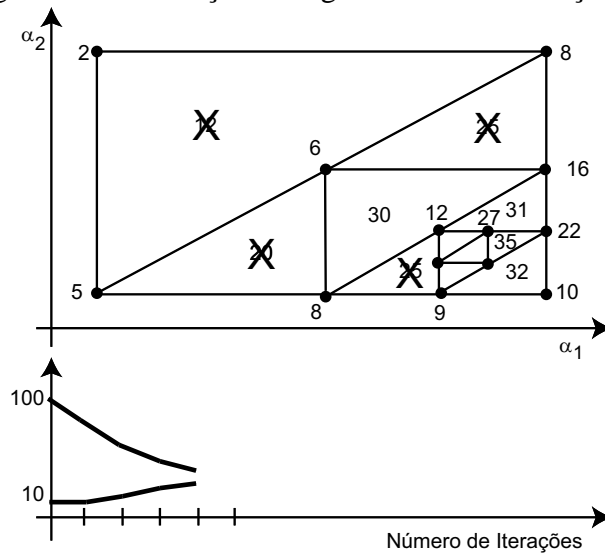


Figura 4.5: Simulação do algoritmo BnB - iteração 4.

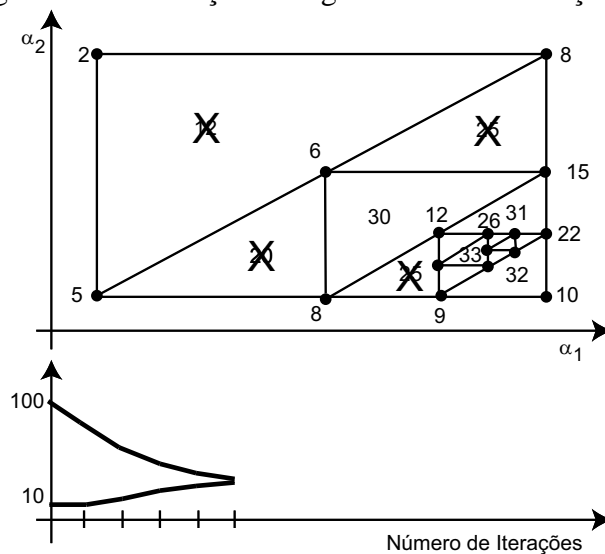


Figura 4.6: Simulação do algoritmo BnB - iteração 5.

Após a triangularização de Delaunay, com a subdivisão de simplexo orientada pelas arestas, a cada iteração é necessário calcular $\frac{1}{2}d(d+1)$ normas dos sistemas correspondentes aos novos vértices sobre cada aresta e 2^d custos garantidos. Este procedimento pode acarretar em alto custo computacional à medida que a dimensão d do espaço de incerteza aumenta.

Como discutido no Capítulo 3, uma opção mais simples de divisão do simplexo seria a técnica de bisseção com a subdivisão do simplexo ao meio com a inclusão de um novo vértice sobre a maior aresta. Com esta técnica, independente da dimensão do espaço de incerteza, é necessário calcular apenas uma norma e dois custos garantidos a cada iteração. Apesar da simplicidade, tal técnica pode resultar em um tempo de processamento total maior, com menor convergência do algoritmo branch-and-bound.

4.6 Exemplos Ilustrativos de Análise de Sistemas Contínuos no Tempo

O algoritmo branch-and-bound para cálculo dos custos com precisão ε foi implementado no LMI Control Toolbox para MATLAB[®] sendo utilizado um computador baseado no Pentium IV 2.8GHz, com 1 GBytes de RAM. No cálculo dos custos garantidos é adotado o vetor de opções $[\varepsilon/100 \ 500 \ 1e9 \ 10 \ 1]$ na função $mincx(\cdot)$.

4.6.1 Cálculo do custo \mathcal{H}_2

Para sistemas contínuos no tempo, serão analisadas as seguintes formulações para o cálculo do custo garantido \mathcal{H}_2 : (QS) estabilidade quadrática, apresentada em Palhares et al. (1997); (EB) Teorema 2, apresentado em Ebihara e Hagiwara (2004a), com $b = 1$; (TR) Teorema 2, apresentado em Trofino et al. (2005), (dO₁₂) combinação dos Lemas 1 e 2 e (dO₃₄) combinação dos Lemas 3 e 4, apresentados em de Oliveira et al. (2004a). No cálculo dos custos garantidos \mathcal{H}_2 , existem duas possibilidades de implementação da formulação LMI, denominadas em muitos trabalhos como formas primal (P) e dual (D), que podem apresentar resultados bastante diferentes para um mesmo problema. Neste trabalho é adotada a denominação primal para a formulação baseada no grammiano de controlabilidade.

Exemplo 4.1 Considere o sistema linear invariante no tempo contínuo, com dois parâmetros incertos, analisado em de Oliveira et al. (2004a), modelado por

$$\frac{dx(t)}{dt} = \begin{bmatrix} -2,02 & 0,16 + \sigma & -0,60 + \beta \\ -0,45 - 2\sigma & -1,85 + \beta & 0,59 - 3\sigma \\ -0,19 - 3\beta & 0,97 + 4\sigma & -0,57 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} w(t)$$

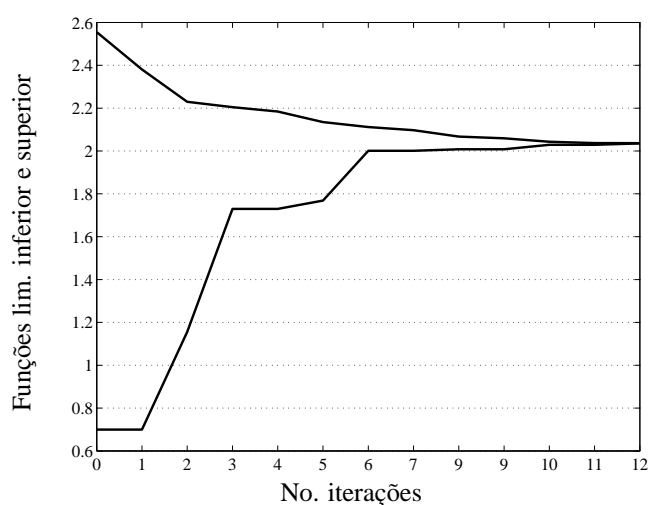
$$z(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(t)$$

sendo que os parâmetros incertos variam no politopo na forma de um quadrado: $\Omega_p = \{p = [\beta, \sigma]^T \in \mathbb{R}^2 : -1 \leq \beta \leq 1; -1 \leq \sigma \leq 1\}$.

O objetivo deste exemplo é calcular o custo ε -garantido \mathcal{H}_2 com precisão de 0,1% através do procedimento proposto de análise implementado com diferentes formulações LMI. A Tabela 4.1 apresenta os custos garantidos \mathcal{H}_2 (o símbolo ∞ significa que a formulação LMI não é factível para todo o politopo), os valores finais das funções limitantes superior e inferior, sendo o primeiro deles o valor do custo ε -garantido \mathcal{H}_2 , os erros relativos percentuais entre o custo garantido e o custo ε -garantido, os tempos computacionais requeridos pelo procedimento de análise proposto e o número total de iterações do procedimento proposto para atingir a precisão requerida. O melhor custo garantido é obtido com a forma primal da combinação dos lemas 3 e 4 em de Oliveira et al. (2004a), sendo verificado que o mesmo é 9,9% maior do que o custo ε -garantido \mathcal{H}_2 obtido pelo procedimento proposto. Através do método proposto, é sempre obtida a precisão requerida de 0,1%, mesmo no caso em que as formulações LMI não são inicialmente factíveis, mas que passam a ser factíveis após a partição do politopo. Observe que, para este exemplo, o conservadorismo das formulações LMI é reduzido consideravelmente em quase todos os casos a um custo computacional baixo. Para este exemplo, a melhor escolha da função limitante superior é o custo garantido calculado com a forma primal da combinação dos lemas 1 e 2 em de Oliveira et al. (2004a) devido ao menor tempo computacional. Para este caso, a evolução das funções limitantes no algoritmo branch-and-bound são apresentadas na Figura 4.7 e a partição do espaço dos parâmetros incertos Ω_p é apresentada na Figura 4.8. O procedimento proposto identifica o vetor $p = [\beta, \sigma]^T = [0,3438, -0,0078]^T$ como o ponto de pior caso de norma \mathcal{H}_2 . A Figura 4.9 apresenta a superfície da norma \mathcal{H}_2 para os valores admissíveis dos parâmetros incertos, na qual pode ser observada a relação entre o ponto de máximo da superfície e o maior refinamento da malha simplicial e da grade composta pelos vértices dos simplexos.

Tabela 4.1: Custos ε -garantido \mathcal{H}_2 com precisão $\varepsilon = 0,001$ (Ex. 4.1).

	QS		EB		TR		dO ₁₂		dO ₃₄	
	P	D	P	D	P	D	P	D	P	D
$\delta_{c.g.}$	∞	∞	∞	∞	∞	∞	2,555	6,281	2,236	4,763
$\Phi_{ls}(\Omega) = \delta_c$	2,038	2,037	2,037	2,037	2,037	2,037	2,037	2,037	2,035	2,036
$\Phi_{li}(\Omega) = \delta_{p.c.}$	2,035	2,035	2,035	2,035	2,035	2,035	2,035	2,035	2,035	2,035
$\frac{100}{\delta_c}(\delta_{c.g.} - \delta_c)$	∞	∞	∞	∞	∞	∞	25,5%	209%	9,9%	134%
Tempo (s)	99,92	533,5	34,2	120,3	22,6	89,89	9,97	15,06	13,86	19,58
Iterações	427	2313	33	128	33	128	12	19	11	15

Figura 4.7: Evolução das funções Φ_{li} e Φ_{ls} no cálculo do custo \mathcal{H}_2 (dO₁₂-P) (Ex. 4.1).

4.6.2 Cálculo do custo \mathcal{H}_∞

Para sistemas contínuos no tempo, serão analisadas as seguintes formulações para o cálculo do custo garantido \mathcal{H}_∞ : (QS) estabilidade quadrática, apresentada em Palhares et al. (1997); (TR) Teorema 4 (considerando a nota 2) apresentado em Trofino et al. (2005); (dO₁) Lema 1 apresentado em de Oliveira et al. (2004b); (dO₂) Lema 2 apresentado em de Oliveira et al. (2004b) ou equivalentemente Teorema 2 apresentado em He et al. (2005); (dO₃) Lema 3 apresentado em de Oliveira et al. (2004b) e (EB) Teorema 5 apresentado em Ebihara et al. (2005).

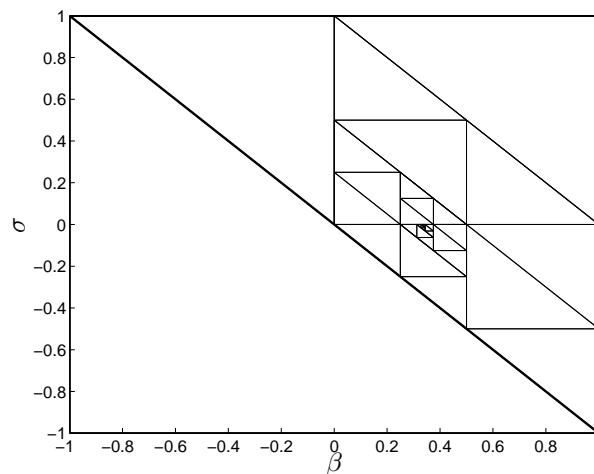


Figura 4.8: Partição do espaço dos parâmetros incertos no cálculo do custo \mathcal{H}_2 (dO₁₂-P) (Ex. 4.1).

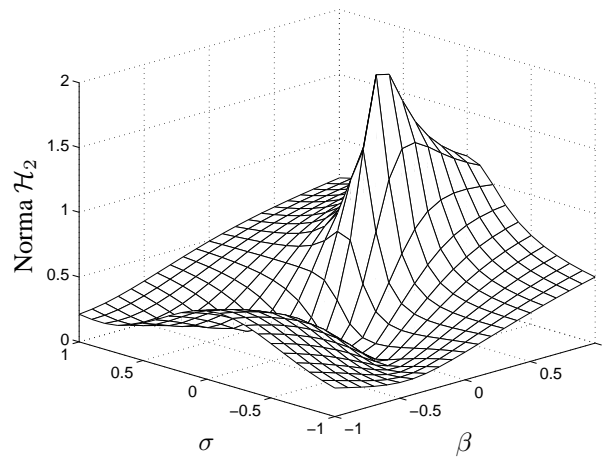


Figura 4.9: Norma \mathcal{H}_2 para $p \in \Omega_p$ (Ex. 4.1).

Exemplo 4.2 Considere o sistema linear invariante no tempo contínuo, com dois parâmetros incertos, modelado como

$$\frac{dx(t)}{dt} = \begin{bmatrix} -0,05 & -0,01 - \beta & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0,01 + \beta & -0,1 & -0,01 - \sigma & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0,01 + \sigma & -0,05 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0,1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} w(t)$$

$$z(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0,1 \end{bmatrix} x(t)$$

Para demonstrar a capacidade do método de partição baseado em malha simplicial, considere que o domínio de incerteza é um politopo com 5 vértices considerando uma restrição adicional sobre os parâmetros incertos: $\Omega_p = \{p = [\beta, \sigma]^T \in \mathbb{R}^2 : 0 \leq \beta \leq 1; 0 \leq \sigma \leq 1; \beta + \sigma \leq 1,5\}$.

O objetivo deste exemplo é demonstrar que, mesmo quando é possível calcular o custo garantido com as formulações LMI, a precisão dos mesmos pode não ser satisfatória. A vantagem do procedimento de análise de desempenho proposto é justamente garantir o cálculo com determinada precisão considerando o tempo computacional como segunda prioridade. A Tabela 4.2 apresenta os resultados obtidos com formulações LMI e com o procedimento proposto implementado apenas com as quatro primeiras formulações LMI mais simples. O símbolo ∞ indica que a condição LMI de custo garantido não é factível para todo o politopo. Observe que o procedimento de análise proposto implementado com o Lema 1 em de Oliveira et al. (2004b) calcula o custo \mathcal{H}_∞ com a precisão de 0,1% com menos tempo de processamento que as formulações LMI baseadas no Lema 3 em de Oliveira et al. (2004b) e no Teorema 5 em Ebihara et al. (2005), cujos resultados apresentam erros relativos percentuais elevados.

Tabela 4.2: Custos ε -garantido \mathcal{H}_∞ com precisão $\varepsilon = 0,001$ (Ex. 4.2).

	QS	TR	dO ₁	dO ₂	dO ₃	EB
$\delta_{c.g.}$	∞	∞	17,304	16,636	15,673	14,320
Tempo (s) para $\gamma_{c.g.}$	—	—	5,33	23,89	389,34	227,75
$\Phi_{ls}(\Omega) = \gamma_c$	5,968	5,963	5,962	5,962	—	—
$\Phi_{li}(\Omega) = \gamma_{p.c.}$	5,962	5,962	5,962	5,962	—	—
$\frac{100}{\delta_c}(\gamma_{c.g.} - \gamma_c)$	∞	∞	190%	179%	163%	140%
Tempo (s) para γ_c	1.293,0	816,7	46,4	237,2	—	—
1ª iteração factível	562	45	0	0	—	—
Iterações	1150	60	11	12	—	—

A Figura 4.10 apresenta a partição do politopo com 5 vértices no espaço bidimensional dos parâmetros incertos no cálculo do custo \mathcal{H}_∞ com o procedimento de análise proposto implementado com o Lema 1 em de Oliveira et al. (2004b). Nesta figura, a seta indica a localização do ponto de pior caso, com a precisão de 1%, identificado como sendo $p = [\beta, \sigma]^T = [0,7539, 0,7461]^T$, após 11 partições do politopo. Observe que na primeira partição, o politopo com 5 vértices é decomposto em 3 simplexes (triângulos) pela triangularização de Delaunay. A Figura 4.2 apresenta a superfície da norma \mathcal{H}_∞ para $p \in \Omega_p$ confirmando a localização do ponto de pior caso de norma.

Neste exemplo, o número de iterações do procedimento proposto, implementado com o Lema 1 ou com o Lema 2, ambos apresentados em de Oliveira et al. (2004b), é determinado pela função li-

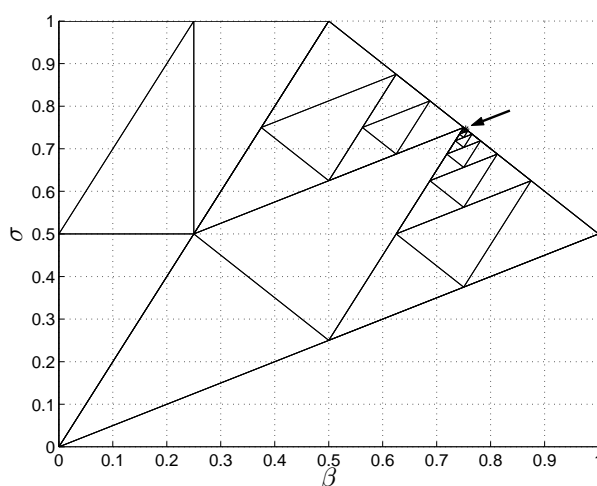


Figura 4.10: Partição do espaço dos parâmetros incertos no cálculo do custo \mathcal{H}_∞ (dO_1) (Ex. 4.2).

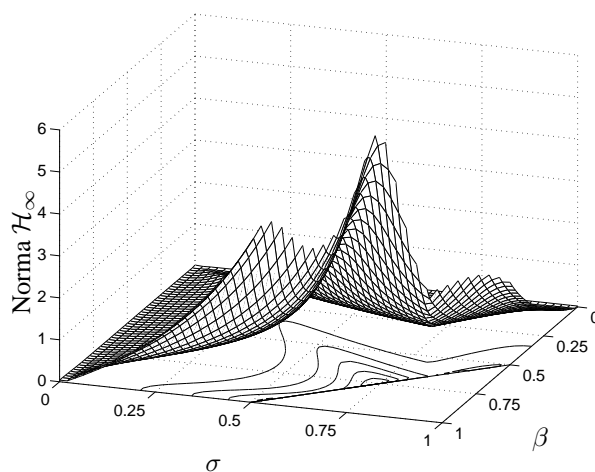


Figura 4.11: Norma \mathcal{H}_∞ para $p \in \Omega_p$ (Ex. 4.2).

mitante inferior. Após 8 e 6 iterações, no caso dos Lemas 1 e 2, respectivamente, o custo garantido já convergiu para a precisão requerida, sendo as demais iterações necessárias para a localização do ponto de pior caso de norma. Já no caso da implementação baseada em estabilidade quadrática, o conservadorismo no cálculo do custo garantido é muito alto, requerendo excessivo número de iterações e alto tempo computacional apesar da maior simplicidade da formulação.

Exemplo 4.3 Para verificar o funcionamento do método de análise proposto aplicado a um sistema de ordem mais elevada, considere o modelo de um veículo submarino apresentado em Leibfritz e

Lipinski (2003) no qual são incluídos três parâmetros incertos:

$$\frac{dx(t)}{dt} = \begin{bmatrix} 0 & p_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -p_1 & -p_2 & -4100 & 0 & 0 & 0 & 0 & 0 \\ p_3 & 0 & -p_3 & 0 & -700 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1400 & 0 & 0 & 0 \\ 0 & 0 & 1600 & -450 & -110 & 0 & 0 & 0 \\ 0 & 0 & 0 & 81 & 0 & -1 & 0 & -900 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 110 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & -1,1 & -22 \end{bmatrix} x(t)$$

$$+ \begin{bmatrix} 0 & 0 \\ 9900 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 99 \end{bmatrix} w(t) + \begin{bmatrix} 0 & 0 \\ 4,6 & 99000 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u(t)$$

$$z_\infty(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \end{bmatrix} w(t) + \begin{bmatrix} 1 & 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t)$$

com $p_1 \in [663; 1037]$, $p_2 \in [93,6; 146,4]$ e $p_3 \in [25,74; 40,26]$. Usando o procedimento geral de projeto a ser apresentado no capítulo 5, é projetado um controlador por realimentação dinâmica de saída, estruturado de 4ª ordem, com objetivo de minimizar o pior caso de norma $\|T_{z_\infty w}\|_\infty$ do sistema em malha-fechada:

$$K = \left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{cccc|cc} 0 & 0 & 0 & -9,7499 & 0,9154 & 10,7133 \\ 1 & 0 & 0 & -20,2208 & 0,8171 & 5,3653 \\ 0 & 1 & 0 & -13,0829 & -0,3397 & -18,5070 \\ 0 & 0 & 1 & -5,5692 & -1,6809 & -15,6363 \\ \hline 0 & 0 & 0 & 1 & 0,0004 & -0,9924 \\ 0 & 0 & 0 & 1 & -0,1557 & -1,8732 \end{array} \right]$$

O sistema em malha-fechada com este controlador é de 12^a ordem. Neste caso, todas as formulações consideradas de cálculo de custo garantido baseadas em LMI são infactíveis. Os custos \mathcal{H}_∞ , calculados com precisão $\varepsilon = 0,1$ através do método de análise proposto, são apresentados na Tabela 4.3. Para sistemas de ordem mais elevada, o método proposto é mais eficiente com formulações de custo garantido mais simples (menor número de variáveis de decisão em função da ordem do sistema). As formulações baseadas no Lema 2 ou no Lema 3, apresentados em de Oliveira et al. (2004b), não são consideradas devido ao elevado tempo de processamento gasto para calcular o custo garantido na iteração zero: 30,9 minutos e 17,1 horas, respectivamente. Para este exemplo, a melhor opção para a função limitante superior é o custo garantido baseado no Lema 1 apresentado em de Oliveira et al. (2004b) considerando o menor tempo de processamento.

Tabela 4.3: Custo \mathcal{H}_∞ com precisão $\varepsilon = 0,1$ (Ex. 4.3).

	QS	dO ₁
$\Phi_{ls}(\Omega) = \gamma_c$	0,994	0,938
$\Phi_{li}(\Omega) = \gamma_{p.c.}$	0,937	0,937
Tempo de processamento	1h23min8s	35min22s
Iterações (primeira factível)	178(137)	3(1)

4.7 Exemplos Ilustrativos de Análise de Sistemas Discretos no Tempo

4.7.1 Cálculo do custo \mathcal{H}_2

As seguintes caracterizações de custo garantido \mathcal{H}_2 serão consideradas nos exemplos desta seção: (QS) estabilidade quadrática, apresentado em Palhares et al. (1997); (dO_{T3}) Teorema 3, apresentado em de Oliveira et al. (2002); (dO₁₅) Lemas 1 e 5, apresentados em de Oliveira et al. (2004a); (dO₃₆) Lemas 3 e 6, também apresentados em de Oliveira et al. (2004a), (XI) Lema 2 apresentado em Xie et al. (2004) e (YA) Lema 1, nota 3, apresentado em Yang et al. (2005). São consideradas como formas primal e dual as formulações relacionadas com os gramianos de controlabilidade e observabilidade, respectivamente.

Exemplo 4.4 Considere o sistema incerto linear invariante no tempo a tempo discreto descrito por

$$\begin{aligned}x(t+1) &= Ax(t) + B_w w(t) \\z(t) &= C_z x(t)\end{aligned}\tag{4.28}$$

sendo $x(t) \in \mathbb{R}^n$, $w(t) \in \mathbb{R}^{n_w}$ e $z(t) \in \mathbb{R}^{n_z}$. Para avaliar a eficiência do procedimento de análise de desempenho proposto, é considerada uma comparação numérica com formulações de análise baseadas em LMIs. Para cada par (n, N) , com $n \in [2, 4]$ e $N \in [2, 4]$, são considerados os 100 primeiros sistemas politópicos robustamente estáveis gerados para análise de estabilidade robusta (ver Exemplo 3.2, página 37). Os elementos das matrizes $B_{w,i} \in \mathbb{R}^{n \times n_w}$ e $C_{z,i} \in \mathbb{R}^{n_z \times n}$, $i = 1, \dots, N$, são gerados de forma aleatória com distribuição uniforme no intervalo $[-1, 1]$. Para cada par (n, N) , para os 50 primeiros sistemas é considerado $n_w = n$ e $n_z = n - 1$ e para os 50 últimos sistemas é adotado $n_w = n - 1$ e $n_z = n$.

A Tabela 4.4 apresenta as taxas de sucesso para computar o custo garantido \mathcal{H}_2 com as formulações baseadas em LMIs e o custo ϵ -garantido \mathcal{H}_2 computado com o procedimento proposto, para $\epsilon = 0,01$, implementado com o custo garantido baseado no Teorema 3, apresentado em de Oliveira et al. (2002). Tanto as formas primal (P) como dual (D) são consideradas. Os resultados na Tabela 4.4 mostram que todas as formulações LMI falham para computar o custo garantido \mathcal{H}_2 em diversos casos, sendo que a taxa de sucesso decai com o aumento de n ou N . Para $(n, N) = (4, 4)$, as taxas de sucesso das formulações LMI isoladas são iguais ou inferiores a 45%. A Tabela 4.5 apresenta os erros relativos médios entre os custos garantidos e o menor valor obtido entre eles. O método proposto conseguiu calcular o custo garantido \mathcal{H}_2 com a precisão de 1% especificada em todos os 900 testes realizados. Em contraste com o método proposto, a Tabela 4.5 mostra que as formulações LMI consideradas, sem a partição do domínio de incerteza, não fornecem custos garantidos \mathcal{H}_2 com a precisão requerida, com erros médios variando de 29,7% até 590%. A Tabela 4.4 apresenta os tempos computacionais médios verificados. Neste exemplo, para todas as combinações (n, N) , o algoritmo BnB requer tempo computacional médio superior ao das formulações LMI isoladas, mas estes tempos não são proibitivos considerando a eficácia em calcular todos os custos com a precisão requerida além de identificar as coordenadas do sistema correspondente ao pior caso. Grande parte dos tempos computacionais mais elevados do procedimento proposto ocorre em situações para as quais as formulações LMI consideradas não são capazes de calcular o custo garantido \mathcal{H}_2 . Existe um compromisso entre a precisão e o tempo computacional e, deste modo, é possível reduzir o tempo computacional do procedimento proposto se for especificada uma menor precisão no cálculo.

Tabela 4.4: Taxas de sucesso (%) para o cálculo do custo garantido \mathcal{H}_2 (Ex. 4.4).

n	N	QS		dO_{T_3}		dO_{15}		dO_{36}		XI		YA		BB	
		P	D	P	D	P	D	P	D	P	D	P	D	P	D
2	2	39	39	48	48	86	86	98	98	98	98	48	48	100	100
	3	4	4	33	33	68	68	87	88	88	87	33	33	100	100
	4	0	0	14	14	57	57	60	65	65	60	14	14	100	100
3	2	13	13	55	55	89	90	97	97	97	97	55	55	100	100
	3	1	1	10	10	60	61	63	64	64	63	10	10	100	100
	4	0	0	4	4	48	48	36	36	36	36	4	4	100	100
4	2	7	7	34	34	84	83	91	92	92	90	34	34	100	100
	3	1	1	8	8	60	61	61	58	58	61	8	8	100	100
	4	0	0	3	3	45	45	25	27	27	25	3	3	100	100

Tabela 4.5: “Erros” relativos médios (%) do custo garantido \mathcal{H}_2 (Ex. 4.4).

n	N	QS		dO_{T_3}		dO_{15}		dO_{36}		XI		YA		BB	
		P	D	P	D	P	D	P	D	P	D	P	D	P	D
2	2	133	201	110	148	44,2	46,6	78,2	83,0	44,6	29,7	119	162	0,19	0,17
	3	141	130	486	524	90,3	73,3	185	195	131	93,6	503	580	0,17	0,14
	4	—	—	567	277	370	178	391	328	241	185	590	319	0,15	0,12
3	2	232	162	124	99,0	60,6	76,0	61,0	57,0	34,7	44,6	150	123	0,23	0,16
	3	280	151	292	179	103	101	166	201	137	134	375	225	0,11	0,11
	4	—	—	252	342	191	203	257	235	185	255	361	447	0,09	0,12
4	2	132	98,0	90,6	63,8	45,8	53,6	41,8	45,3	32,5	44,1	116	96,3	0,12	0,12
	3	371	288	169	248	97,7	90,9	106	103	88,4	111	227	293	0,10	0,09
	4	—	—	241	306	85,9	153	165	217	160	233	330	350	0,09	0,06

Tabela 4.6: Tempos computacionais médios (s) para cálculo do custo garantido \mathcal{H}_2 (Ex. 4.4).

n	N	QS		dO _{T3}		dO ₁₅		dO ₃₆		XI		YA		BB	
		P	D	P	D	P	D	P	D	P	D	P	D	P	D
2	2	0,06	0,05	0,08	0,08	0,08	0,07	0,12	0,13	0,17	0,17	0,07	0,07	0,71	0,79
	3	0,07	0,06	0,11	0,11	0,22	0,20	0,17	0,17	0,34	0,36	0,10	0,10	5,01	4,67
	4	0,08	0,08	0,15	0,15	0,47	0,47	0,21	0,22	0,67	0,79	0,13	0,13	20,4	17,7
3	2	0,08	0,07	0,15	0,15	0,13	0,12	0,28	0,27	0,49	0,49	0,16	0,16	1,56	1,67
	3	0,11	0,10	0,21	0,20	0,50	0,48	0,37	0,36	1,63	1,76	0,21	0,20	9,12	10,6
	4	0,12	0,12	0,28	0,27	1,02	1,14	0,50	0,45	3,44	3,44	0,30	0,30	38,9	42,0
4	2	0,12	0,11	0,29	0,29	0,25	0,25	0,64	0,64	1,48	1,52	0,32	0,31	3,13	2,95
	3	0,15	0,15	0,34	0,36	1,00	1,19	0,86	0,87	4,73	4,59	0,43	0,43	16,3	21,5
	4	0,19	0,19	0,50	0,47	3,56	3,76	1,05	1,04	8,49	9,23	0,61	0,62	58,5	64,1

Exemplo 4.5 Para avaliar o método de análise proposto, implementado com diferentes formulações LMI para o cálculo do custo garantido, aplicado a um domínio de incerteza de quinta ordem, considere o sistema linear invariante no tempo, a tempo discreto, modelado por:

$$\begin{aligned}
 x(k+1) &= \begin{bmatrix} p_1 & 0,2189 & 0,2747 \\ -2,3646 & p_2 & -0,1331 \\ 0,9901 & 1,2134 & -p_3 \end{bmatrix} x(k) \\
 &+ \begin{bmatrix} 0,0828 & 0,0037 & 0,0044 \\ 0,0037 & 0,0788 & 0,0001 \\ 0,0089 & 0,0003 & 0,1013 \end{bmatrix} w(k) + \begin{bmatrix} 0,0044 \\ 0,0001 \\ p_4 \end{bmatrix} u(k) \\
 z(k) &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} w(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\
 y(k) &= \begin{bmatrix} p_5 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(k)
 \end{aligned}$$

Este sistema possui cinco parâmetros incertos que podem variar nas faixas: $p_1 \in [1,2424; 1,5184]$, $p_2 \in [0,8655; 1,0579]$, $p_3 \in [0,5135; 0,6276]$, $p_4 \in [0,0912; 0,1114]$ e $p_5 \in [0,9; 1,1]$. O domínio de incerteza $\Omega \subset \mathbb{R}^5$ é um hiper-retângulo de 32 vértices no espaço de quinta dimensão. Usando o procedimento geral de projeto a ser apresentado no Capítulo 5, é projetado um controlador por

realimentação estática de saída para minimizar o pior caso de norma $\|T_{zw}(z, \alpha)\|_2$, $\alpha \in \Omega$, do sistema em malha-fechada: $K = [20,9620 \ -1,2958]^T$.

A Tabela 4.7 apresenta os resultados obtidos com o procedimento de análise proposto implementado com as formas primal e dual das formulações mais eficientes para o caso em estudo: (dO_{T3}) Teorema 3, apresentado em de Oliveira et al. (2002); (dO₃₆) Lemas 3 e 6, apresentados em de Oliveira et al. (2004a), e (XI) Lema 2 apresentado em Xie et al. (2004). Os custos ε -garantidos \mathcal{H}_2 são calculados com precisão $\varepsilon = 0,02$. Neste exemplo, na primeira iteração são necessários os cálculos de 103 custos garantidos com a decomposição do hiper-retângulo em um conjunto de simplexos pela triangularização de Delaunay. Nas iterações seguintes, a cada nova partição pela divisão de simplexo orientada pelas arestas são necessários os cálculos de 15 normas e 32 custos garantidos. O melhor custo garantido, não apresentado na Tabela 4.7, é calculado com a forma dual da combinação dos Lemas 1 e 5 apresentados em de Oliveira et al. (2004a), que é 7,9% maior do que o pior caso de norma verificado no politopo, $\delta_{p.c.} = 7,3939$ para $\alpha = [1,2424 \ 0,98575 \ 0,5135 \ 0,1114 \ 1,1]^T$. Porém, esta formulação requer 1 hora e 10 minutos de tempo de processamento, superior a todos os tempos computacionais das diferentes implementações do procedimento proposto, apresentados na penúltima linha da Tabela 4.7. Neste caso, a formulação dual baseada no Lemas 3 e 6, apresentados em de Oliveira et al. (2004a), proporciona o melhor compromisso entre tempo de processamento por iteração e número de iterações, resultando no menor tempo computacional. É claro que o tempo computacional pode ser reduzido se for adotada uma menor precisão para o cálculo. Por exemplo, para $\varepsilon = 0,05$, a formulação primal baseada nos Lemas 3 e 6, apresentados em de Oliveira et al. (2004a), obtém o custo ε -garantido $\delta_c = 7,71$ (4,7% de erro relativo) em apenas uma partição e 3 minutos de processamento.

Tabela 4.7: Custos \mathcal{H}_2 para $\varepsilon = 0,02$ (Ex. 4.5).

	dO _{T3}		dO ₃₆		XI	
	P	D	P	D	P	D
Custo garantido \mathcal{H}_2 , $\delta_{c.g.}$	14,0	23,0	9,27	8,32	9,17	9,66
$\frac{100}{\delta_{p.c.}}(\delta_{c.g.} - \delta_{p.c.})$	90,0%	211%	25,8%	12,9%	24,1%	30,8%
Tempo para $\delta_{c.g.}$	6s	12s	54s	72s	13s	31s
$\Phi_{ls} = \delta_c$	7,51	7,53	7,54	7,47	7,54	7,53
$\Phi_{li} = \delta_{p.c.}$	7,39	7,39	7,39	7,39	7,39	7,39
$\frac{100}{\Phi_{li}}(\Phi_{ls} - \Phi_{li})$ (%)	1,69	1,90	1,99	1,13	1,99	1,82
Tempo para δ_c	22min7s	42min44s	24min42s	5min44s	23min53s	43min47s
Iterações	87	109	33	3	51	72

4.7.2 Cálculo do custo \mathcal{H}_∞

Serão analisadas as seguintes formulações para o cálculo do custo garantido \mathcal{H}_∞ de sistemas discretos: (QS) estabilidade quadrática, apresentada em Palhares et al. (1997); (dO_{T4}) Teorema 4, apresentado em de Oliveira et al. (2002); (dO_{L4}) Lema 4, (dO_{L5}) Lema 5, (dO_{L6}) Lema 6, apresentados em de Oliveira et al. (2004b); (OL_{T2}) Teorema 2 e (OL_{T3}) Teorema 3 apresentados em Oliveira e Peres (2005b).

Exemplo 4.6 Este exemplo considera o cálculo do custo garantido \mathcal{H}_∞ de sistemas discretos no tempo. Os mesmos 900 sistemas politópicos aleatórios analisados no Exemplo 4.4 são considerados novamente. A Tabela 4.8 apresenta as taxas de sucesso para calcular os custos garantidos \mathcal{H}_∞ com as formulações baseadas em LMIs e o custo ϵ -garantido \mathcal{H}_∞ calculado com o procedimento proposto implementado com o Teorema 4 em de Oliveira et al. (2002), para $\epsilon = 0,01$, denominado BB. Nas formulações com função de Lyapunov com dependência polinomial homogênea nos parâmetros são adotados os seguintes graus: $g = 3$ para (OL_{T2}) e $g = 1$ para (OL_{T3}). A Tabela 4.9 apresenta a diferença relativa média entre cada custo garantido \mathcal{H}_∞ e o menor valor entre eles. A Tabela 4.10 lista os tempos computacionais médios. As formulações LMI isoladas produzem alguns resultados com a precisão menor que 1%, porém, com exceção do Teorema 3, com grau 1, apresentado em Oliveira e Peres (2005b), elas apresentam reduções consideráveis das taxas de sucesso com o aumento de n e N . Diferentemente do caso \mathcal{H}_2 , além da taxa de 100% de sucesso e a obtenção da precisão especificada para todos os casos, a Tabela 4.10 demonstra que o método proposto também requer, na média, menos tempo computacional do que os Lemas 4 e 6 apresentados em de Oliveira et al. (2004b) e os Teoremas 2 e 3 apresentados em Oliveira e Peres (2005b), com graus 3 e 1, respectivamente, na maioria das situações consideradas. As formulações com função de Lyapunov com dependência polinomial homogênea nos parâmetros podem apresentar taxas de sucessos e precisão melhores com o aumento do grau do polinômio, porém, resultando em tempos computacionais ainda maiores do que o procedimento proposto. Para este conjunto de testes, o método proposto de análise não é somente mais eficaz como também é mais eficiente do que as formulações LMIs mais complexas (maior número de variáveis escalares de otimização e restrições).

Para ilustrar a eficácia do procedimento proposto, considere um sistema pinçado do conjunto aleatório, gerado para $(n, N) = (4, 3)$, em que todas as formulações LMI analisadas falham para

Tabela 4.8: Taxas de sucesso (%) para calcular o custo garantido \mathcal{H}_∞ (Ex. 4.6).

n	N	QS	dO _{T4}	dO _{L4}	dO _{L5}	dO _{L6}	OL _{T2}	OL _{T3}	BB
2	2	39	48	75	98	98	83	98	100
	3	4	33	47	88	96	53	97	100
	4	0	14	27	65	77	23	97	100
3	2	13	55	76	97	97	84	97	100
	3	1	10	45	64	71	55	85	100
	4	0	4	22	36	38	2	84	100
4	2	7	34	74	92	92	88	92	100
	3	1	8	48	58	77	35	89	100
	4	0	3	5	27	8	0	79	100

Tabela 4.9: “Erros” relativos (%) do custo garantido \mathcal{H}_∞ (Ex. 4.6).

n	N	QS	dO _{T4}	dO _{L4}	dO _{L5}	dO _{L6}	OL _{T2}	OL _{T3}	BB
2	2	20,3	91,8	0,64	84,4	35,4	0,00	2,96	0,03
	3	70,1	432	2,75	170	87,0	0,37	6,80	0,07
	4	—	345	4,66	290	66,7	0,00	5,24	0,08
3	2	18,4	35,6	1,14	34,1	16,6	12,3	8,24	0,02
	3	50,6	61,5	1,73	98,9	11,9	0,01	4,72	0,02
	4	—	71,9	0,02	78,1	7,65	0,00	4,85	0,04
4	2	2,05	20,5	4,41	4,54	1,51	0,53	0,09	0,03
	3	87,6	101	0,99	16,2	8,51	0,00	0,58	0,02
	4	—	67,8	0,05	65,4	0,06	—	7,91	0,03

Tabela 4.10: Tempos computacionais médios (s) (Ex. 4.6).

n	N	QS	dO _{T4}	dO _{L4}	dO _{L5}	dO _{L6}	OL _{T2}	OL _{T3}	BB
2	2	0,09	0,11	0,45	0,18	1,11	0,66	0,58	0,51
	3	0,13	0,17	2,80	0,35	7,66	4,90	1,87	2,32
	4	0,10	0,21	9,33	0,57	33,1	18,9	4,18	5,46
3	2	0,17	0,21	1,37	0,50	3,98	1,84	2,68	0,79
	3	0,21	0,41	8,33	1,23	32,9	14,4	21,9	4,58
	4	0,20	0,62	26,4	2,47	115	57,1	86,9	14,9
4	2	0,31	0,49	3,23	1,32	11,8	4,32	16,5	1,80
	3	0,38	0,98	19,8	3,34	82,7	36,1	134	8,18
	4	0,39	1,29	57,0	6,10	273	139	602	29,4

calcular o custo garantido \mathcal{H}_∞ :

$$\begin{aligned}
 S = \alpha_1 & \left[\begin{array}{cccc|cccc}
 -0,0026 & -0,1803 & 0,1427 & 0,1380 & 0,6307 & -0,7929 & 0,6267 & 0,3591 \\
 -0,7762 & -0,9019 & -0,3102 & 0,0775 & 0,4593 & -0,6323 & 0,4852 & 0,5535 \\
 -0,3356 & 0,7966 & 0,5507 & -0,2896 & 0,8646 & -0,0969 & -0,3379 & -0,1261 \\
 0,1526 & -0,2197 & -0,0434 & -0,6678 & -0,0498 & -0,7105 & 0,9549 & 0,5262 \\
 \hline
 -0,4961 & -0,9721 & 0,6465 & 0,2621 & 0 & 0 & 0 & 0 \\
 0,0148 & -0,6241 & -0,4911 & 0,2642 & 0 & 0 & 0 & 0 \\
 0,4046 & 0,1651 & 0,3339 & -0,7759 & 0 & 0 & 0 & 0
 \end{array} \right] \\
 + \alpha_2 & \left[\begin{array}{cccc|cccc}
 0,5121 & -0,5996 & 0,8087 & 0,4287 & 0,5827 & 0,8982 & 0,7810 & -0,2250 \\
 -0,9805 & -0,6339 & 0,3673 & 0,8598 & -0,8162 & -0,5934 & -0,3344 & 0,4096 \\
 -0,4899 & -1,0445 & 0,5339 & -0,2685 & -0,3246 & 0,3057 & 0,4763 & 0,6922 \\
 1,1977 & -0,5602 & 0,5089 & -0,9118 & 0,6955 & 0,4584 & 0,5812 & 0,1726 \\
 \hline
 -0,9062 & 0,0241 & -0,6700 & -0,6685 & 0 & 0 & 0 & 0 \\
 0,6878 & -0,0432 & -0,8635 & -0,3943 & 0 & 0 & 0 & 0 \\
 0,1736 & 0,4929 & -0,4091 & 0,5975 & 0 & 0 & 0 & 0
 \end{array} \right]
 \end{aligned}$$

$$+ \alpha_3 \left[\begin{array}{cccc|cccc} -0,0308 & -0,4776 & -0,5410 & -0,2027 & 0,7197 & -0,0196 & -0,9436 & -0,7542 \\ 0,1926 & -0,6822 & -0,0383 & -0,3670 & -0,6536 & 0,6169 & 0,1629 & 0,2685 \\ 0,5401 & 0,3265 & -0,5366 & -0,4907 & 0,5306 & -0,6047 & 0,6407 & -0,7773 \\ 0,6539 & 0,4749 & -0,4374 & 0,6533 & 0,8370 & -0,7895 & 0,4991 & 0,9661 \\ \hline 0,1543 & 0,5903 & 0,1755 & -0,5349 & 0 & 0 & 0 & 0 \\ -0,7245 & -0,2375 & -0,5992 & -0,5641 & 0 & 0 & 0 & 0 \\ -0,2131 & -0,2143 & 0,9262 & 0,6814 & 0 & 0 & 0 & 0 \end{array} \right]$$

O procedimento proposto obtém o pior caso de norma igual à $\gamma_{p.c.} = 380,7$, para $\alpha_{(\infty)} = [0,3555 \ 0,3672 \ 0,2773]^T$, e o custo ϵ -garantido \mathcal{H}_{∞} igual à $\gamma_c = 382,0$. Neste caso, o cálculo é complexo devido à característica da superfície da norma \mathcal{H}_{∞} que apresenta um pico intenso no interior do domínio de incerteza, com uma variação acentuada em uma pequena região, como pode ser observado nas Figuras 4.12 e 4.13 (detalhamento próximo ao ponto de máximo global). O asterisco nas curvas de nível da Figura 4.13 indica a localização do ponto de pior caso obtido pelo procedimento proposto. A Figura 4.14 apresenta a evolução das funções limitantes no cálculo do custo ϵ -garantido \mathcal{H}_{∞} . A formulação LMI passa a ser factível apenas após a 7^a subdivisão do simplexo e o ponto de pior caso é localizado após 14 subdivisões e 34,5s de processamento.

Exemplo 4.7 Considere o sistema incerto a tempo discreto, apresentado em de Oliveira et al. (2004a), cuja matriz A varia dentro do politopo de matrizes definido pela combinação convexa $A = Co\{\rho A_1, \rho A_2, \rho A_3, \rho A_4\}$, sendo

$$A_1 = \begin{bmatrix} 0,009 & 0,027 & 0,303 & 0,066 \\ -0,093 & -0,033 & 0,021 & 0,195 \\ 0,162 & -0,048 & -0,057 & 0,057 \\ -0,06 & -0,237 & 0,138 & -0,288 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -0,315 & 0,294 & 0,252 & 0,156 \\ -0,108 & -0,249 & 0,255 & -0,093 \\ -0,186 & -0,321 & 0,105 & 0,117 \\ -0,177 & -0,216 & -0,033 & 0,129 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} -0,048 & -0,048 & 0,114 & -0,261 \\ 0,159 & 0,09 & 0,21 & -0,006 \\ 0,114 & 0,192 & 0,102 & 0,117 \\ 0,111 & 0,087 & 0 & 0,24 \end{bmatrix}$$

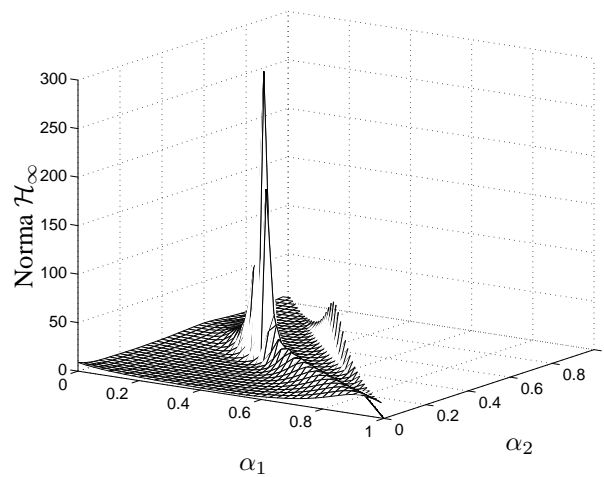


Figura 4.12: Norma \mathcal{H}_∞ para $\alpha \in \Omega$ (Ex. 4.6).

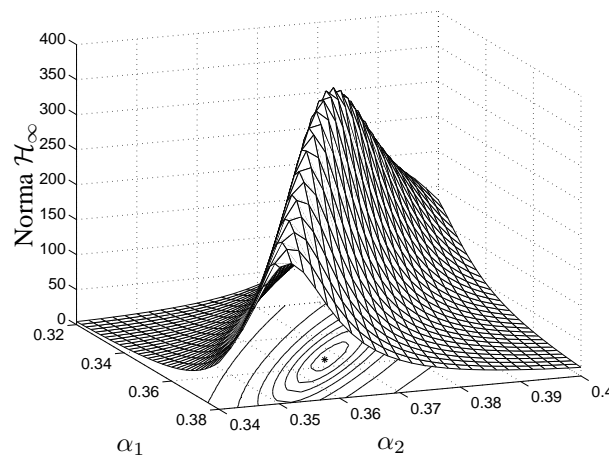


Figura 4.13: Norma \mathcal{H}_∞ para $0,32 \leq \alpha_1 \leq 0,38$ e $0,34 \leq \alpha_2 \leq 0,40$ (Ex. 4.6).

$$A_4 = \begin{bmatrix} -0,081 & 0,198 & 0,009 & 0,198 \\ 0,069 & -0,189 & 0,087 & 0,033 \\ 0,117 & -0,114 & -0,162 & 0,162 \\ 0,033 & 0,048 & 0,168 & -0,054 \end{bmatrix}$$

As demais matrizes do sistema são precisamente conhecidas:

$$B = [1 \ 0 \ 0 \ 0]^T, \quad C = [0 \ 0 \ 0 \ 1], \quad D = 0$$

A Tabela 4.11 apresenta os resultados obtidos com o procedimento proposto, implementado com diferentes formulações LMI, para $\rho = 3$, sendo $\gamma_{c.g.}$ os custos garantidos calculados para todo o po-

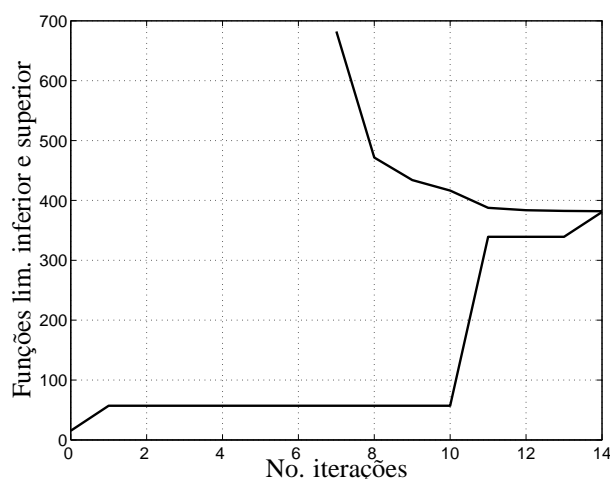


Figura 4.14: Evolução das funções limitantes no cálculo do custo ϵ -garantido \mathcal{H}_∞ (Ex. 4.6).

litopo, $\gamma_{p.c.}$ e γ_c os piores casos de norma e os custos ϵ -garantidos, respectivamente, considerando $\epsilon = 0,01$. Neste exemplo, apenas as formulações dO_{L4} e dO_{L6} fornecem valores de custos garantidos \mathcal{H}_∞ porém com valores conservadores. É verificado que todas as formulações com funções de Lyapunov dependentes de parâmetros convergem após uma única partição. No cálculo dos custos ϵ -garantidos \mathcal{H}_∞ , as formulações dO_{T4} e dO_{L5} são as mais eficientes, sendo que os tempos de processamento do procedimento proposto com estas formulações são inferiores aos tempos de processamento para o cálculo dos custos garantidos com as formulações LMI que são factíveis.

Tabela 4.11: Custos \mathcal{H}_∞ com precisão $\epsilon = 0,01$ para $\rho = 3$ (Ex. 4.7).

	QS	dO_{T4}	dO_{L4}	dO_{L5}	dO_{L6}
Custo garantido $\mathcal{H}_\infty, \gamma_{c.g.}$	não factível	não factível	14,427	não factível	4,0813
$\frac{100}{\gamma_{p.c.}}(\gamma_{c.g.} - \gamma_{p.c.})$	—	—	282%	—	8,1%
Tempo (s) para $\gamma_{c.g.}$	—	—	22,6	—	35,9
$\Phi_{ls} = \gamma_c$	3,8052	3,7739	3,7739	3,7739	3,7739
$\Phi_{li} = \gamma_{p.c.}$	3,7739	3,7739	3,7739	3,7739	3,7739
$\frac{100}{\Phi_{li}}(\Phi_{ls} - \Phi_{li})$	0,83%	0,00%	0,00%	0,00%	0,00%
Tempo (s) para γ_c	16,2	5,0	52,2	11,3	122
Iterações	9	1	1	1	1

Considerando as formulações com função de Lyapunov com dependência polinomial homogênea dos parâmetros, para o Teorema 2, apresentado em Oliveira e Peres (2005b), incrementando gradativamente o grau a partir de $g = 0$ é obtido o custo \mathcal{H}_∞ com a precisão requerida para $g = 3$ após 29,1s de processamento. Com o Teorema 3, apresentado em Oliveira e Peres (2005b), é obtido o custo \mathcal{H}_∞

com a precisão requerida para $g = 2$ após 150s de processamento. Ambos os tempos são superiores aos obtidos pelo procedimento proposto implementado com as formulações dO_{T4} e dO_{L5} .

A Tabela 4.12 apresenta os resultados obtidos para $\rho = 3,3$. Neste caso todas as formulações LMI analisadas na implementação do procedimento proposto não são factíveis para todo o politopo sendo necessária pelo menos uma partição. Neste caso, a formulação dO_{T4} é mais eficiente para ser empregada no procedimento proposto considerando o menor tempo de processamento requerido.

Tabela 4.12: Custos \mathcal{H}_∞ com precisão $\varepsilon = 0,01$ para $\rho = 3,3$ (Ex. 4.7).

	QS	dO_{T4}	dO_{L4}	dO_{L5}	dO_{L6}
$\Phi_{ls} = \gamma_c$	18,714	18,541	18,538	18,536	18,536
$\Phi_{li} = \gamma_{p.c.}$	18,536	18,536	18,536	18,536	18,536
$\frac{100}{\Phi_{li}}(\Phi_{ls} - \Phi_{li})$	0,96%	0,03%	0,01%	0,00%	0,00%
Tempo (s) para γ_c	91,5	18,5	485	38,5	1.023,9
No. iterações (primeira factível)	48(15)	5(1)	5(2)	5(1)	4(1)

A Figura 4.15 apresenta a partição do politopo de matrizes, no cálculo do custo \mathcal{H}_∞ , após 5 iterações, utilizando a formulação dO_{L5} para $\rho = 3,3$, cujo pior caso de norma é obtido para $\alpha = [0,9375 \ 0 \ 0 \ 0,0625]^T$. A Figura 4.16 apresenta as correspondentes evoluções das funções limitantes no algoritmo branch-and-bound. Neste caso, o número de iterações é função da localização do ponto de pior caso e não do cálculo do custo garantido.

Por meio dos Teoremas 2 e 3, apresentados em Oliveira e Peres (2005b), é possível calcular o custo garantido, para $\rho = 3,3$, utilizando matriz de Lyapunov com dependência polinomial homogênea de parâmetros. Sendo g o grau da dependência polinomial, como apresentado na Tabela 4.13 para o caso da formulação OL_{T2} , para $g < 3$ a condição suficiente não é atendida. Para $g \geq 3$, a exatidão do cálculo aumenta gradativamente, porém o custo computacional é consideravelmente mais elevado, considerando ou não o tempo acumulado nos testes com grau inferiores. No caso da formulação OL_{T3} , o condição é factível para $g \geq 2$, sendo que, para $g = 2$ é obtido o custo \mathcal{H}_∞ com a precisão desejada após 158s de tempo computacional e 322s de tempo acumulado (de $g = 0$ até $g = 2$).

Tabela 4.13: Custo \mathcal{H}_∞ calculado pela formulação (OL_{T2}) para $\rho = 3,3$ (Ex. 4.7).

Grau g	0	1	2	3	4	5	6
Custo garantido \mathcal{H}_∞	–	–	–	58,28	21,31	19,32	19,19
Tempo de processamento (s)	0,4	9,1	41,3	123,1	163,7	502,3	1.550,1
Tempo acumulado (s)	0,4	9,5	50,8	173,9	337,6	839,9	2.390,0

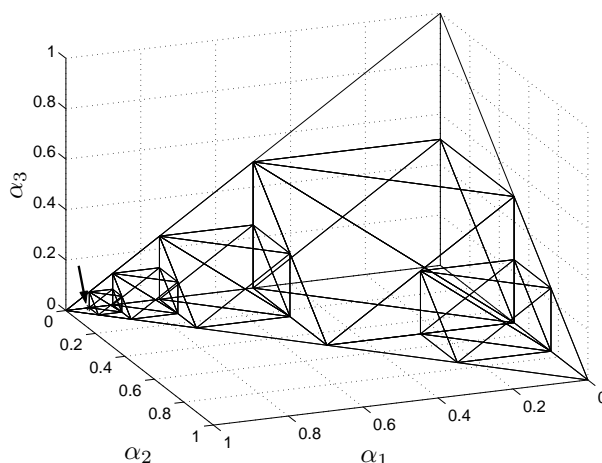


Figura 4.15: Partição do politopo no cálculo do custo \mathcal{H}_∞ (dO_{L5}) para $\rho = 3,3$ (Ex. 4.7).

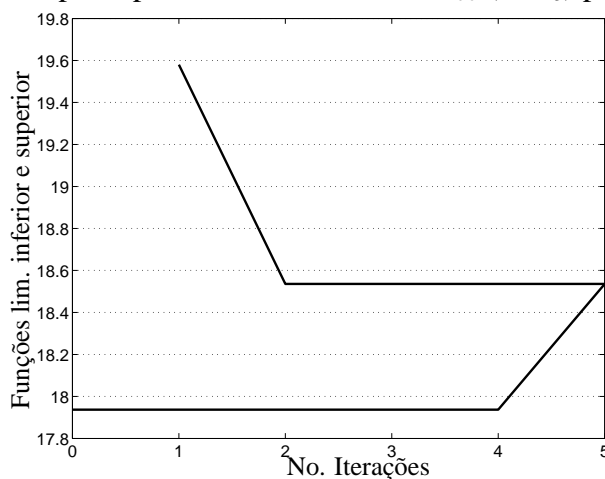


Figura 4.16: Evolução das funções limitantes no cálculo do custo \mathcal{H}_∞ (dO_{L5}) para $\rho = 3,3$ (Ex. 4.7).

4.8 Conclusões

Neste capítulo foi proposto um procedimento de cálculo dos custos \mathcal{H}_2 e \mathcal{H}_∞ , com uma precisão especificada, baseado na combinação do algoritmo branch-and-bound com formulações de cálculo de custos garantidos baseadas em LMIs, que pode ser aplicado a sistemas lineares invariantes no tempo, a tempo contínuo ou discreto, com domínio politópico de incerteza. Como demonstrado pelos vários exemplos apresentados, o procedimento de análise proposto é uma ferramenta importante na área de Controle Robusto devido à sua eficácia para o cálculo dos custos com a precisão especificada. As formulações LMI para cálculo de custos garantidos, quando factíveis, não são capazes de informar qual a precisão do resultado obtido. As novas formulações baseadas em funções de Lyapunov quadrática no estado e com dependência polinomial homogênea nos parâmetros, podem obter resultados,

quando outras formulações LMI falham, com a precisão sendo melhorada com o aumento arbitrário do grau do polinômio, porém, como foi verificado nos testes apresentados, mesmo para graus pequenos, os tempos computacionais requeridos são em geral mais elevados do que os do procedimento proposto que combina formulações mais simples com a partição do domínio de incerteza. Métodos baseados na técnica de grade (discretização do domínio de incerteza) também não informam a precisão do cálculo e também requerem tempos computacionais elevados para domínios em dimensões mais elevadas ou em casos em que existe uma variação considerável da norma em torno do ponto de máximo que exige uma grade muito fina para a sua localização. A vantagem do procedimento proposto é que, apesar da função limitante inferior também ser um método de grade, o refinamento da grade é orientado pelo cálculo dos custos garantidos utilizados como função limitante superior, que também orienta a exclusão de subdomínios nos quais comprovadamente o ponto de pior caso não está localizado.

A princípio, podem ser consideradas como desvantagens do procedimento proposto a dificuldade para sua implementação e o maior custo computacional em relação às formulações LMI menos complexas, porém menos eficazes e mais conservadoras. O problema relativo à implementação do algoritmo branch-and-bound, especialmente considerando a abrangência de domínios de incerteza pretendida, é satisfatoriamente solucionado através da escolha adequada das funções limitantes e do algoritmo inédito de divisão de simplexes, em qualquer dimensão, apresentado em detalhes no Capítulo 2, que permite a utilização do algoritmo branch-and-bound em uma classe mais ampla de problemas, não se limitando aos problemas com domínio na forma de hiper-retângulo, considerados em trabalhos anteriores nesta área. Com relação ao custo computacional, não foram observados problemas com relação a requisitos de disponibilidade de memória, entretanto, os tempos computacionais podem ser consideravelmente elevados em certas situações. É verificado que não existe uma formulação de cálculo de custo garantido única que é a mais eficiente para implementação do método proposto para qualquer tipo de sistema. É interessante que o método não seja utilizado com uma formulação de cálculo de custo garantido fixa, devendo ser testadas diferentes possibilidades, principalmente no caso do custo \mathcal{H}_2 , em que, para cada caracterização, existem as formas primal e dual com comportamentos bastante distintos. O ideal é estabelecer uma relação entre as características do sistema, como as dimensões de suas matrizes e a dimensão do espaço de incerteza, com a formulação mais eficiente para cada caso. É claro que existe a possibilidade de redução do tempo computacional em detrimento da precisão requerida. É importante ressaltar que o aspecto mais importante da contribuição na área de análise de desempenho robusto é a capacidade de cálculo dos custos com a precisão requerida, sendo que o tempo computacional será gradativamente reduzido com a evolução dos recursos computacionais.

As formulações LMI com função de Lyapunov com dependência polinomial homogênea dos parâ-

metros são bastante interessantes, havendo a possibilidade de sua utilização no procedimento proposto com o grau do polinômio fixo por iteração mas dependendo, por exemplo, do nível da partição. Isto seria interessante nos casos em que o tempo computacional para analisar o politopo inicial é elevado com formulações mais complexas, o que não seria o caso com a formulação LMI iniciando do grau zero. No estágio atual dos softwares e hardwares disponíveis, tal implementação ainda não é a mais indicada.

Além do cálculo dos custos com a precisão requerida, o método calcula simultaneamente as coordenadas do ponto de pior caso e o valor da norma correspondente. Estas informações serão úteis no procedimento geral de projeto que será apresentado no próximo capítulo. O procedimento de análise proposto também pode ser utilizado como uma ferramenta de validação de novos teoremas e implementações de formulações de análise de estabilidade ou desempenho.

Capítulo 5

Procedimento Geral de Projeto

5.1 Introdução

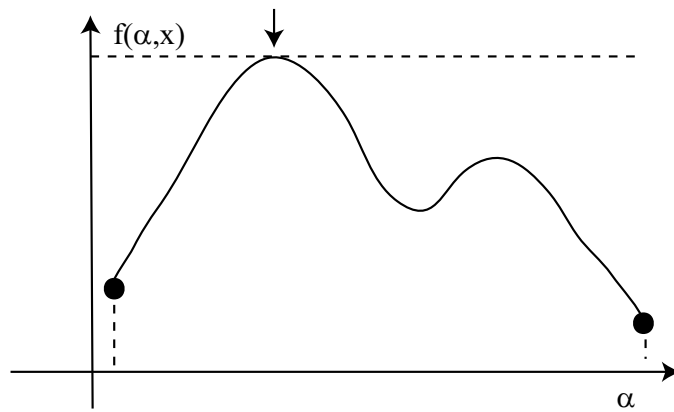
Nos últimos anos, vários trabalhos têm sido dedicados ao desenvolvimento de formulações de projeto de controladores e filtros aplicados a sistemas incertos baseados em problemas de otimização lineares com restrições na forma de desigualdades matriciais lineares (LMIs). A principal vantagem desta estratégia é que, uma vez que o problema é convexo, o problema de otimização do pior caso é tratado apenas em termos dos vértices do politopo. Neste caso, os problemas de otimização podem ser resolvidos por um dos “LMI solvers” disponíveis. O “Lyapunov shaping paradigm”, baseado no conceito de estabilidade quadrática, pode ser usado com este propósito. Entretanto, existe uma desvantagem potencial desta formulação, e de outras que empregam a mesma metodologia, que é fixar uma mesma matriz de Lyapunov para garantir o desempenho do sistema e as restrições de alocação regional de pólos. Isto significa que as únicas soluções possíveis para tais procedimentos de projeto são aquelas que admitem uma mesma matriz de Lyapunov para todo o conjunto de incertezas e para a formulação LMI de todos os objetivos de controle. Como estas soluções são apenas um subconjunto de todas as soluções possíveis relevantes, isto constitui uma fonte de conservadorismo em várias metodologias de projeto de controladores e filtros robustos para sistemas com incertezas politópicas cujos parâmetros incertos são constantes.

Vários trabalhos recentes têm abordado este problema, visando caracterizações menos conservadoras, através do uso de multiplicadores ou matrizes de Lyapunov dependentes de parâmetros. Estas caracterizações por LMI permitem o uso de matrizes de Lyapunov distintas para cada vértice do politopo e cada especificação de estabilidade e desempenho. Apesar disto, ainda é necessário fixar uma matriz para atender a todas as restrições de estabilidade e desempenho pela qual são calculadas as matrizes do controlador ou filtro.

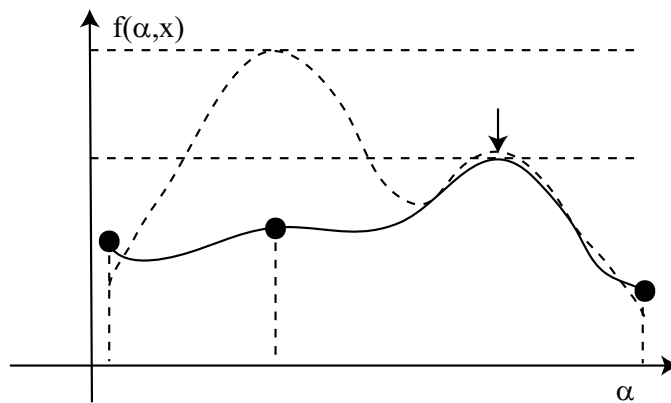
Nesta tese é proposta uma abordagem completamente diferente para tornar o problema de dimen-

são infinita (verificação de infinitos pontos) em um problema de dimensão finita de forma a viabilizar a solução do mesmo de forma menos conservadora. O procedimento iterativo de projeto, a ser apresentado em detalhes na seqüência, é dividido em duas etapas: uma etapa de síntese do controlador, filtro ou modelo reduzido e uma etapa de análise robusta dos resultados. A síntese é tratada como um problema de otimização multiobjetivo cujos objetivos de otimização (custos \mathcal{H}_2 e \mathcal{H}_∞) e as restrições (limitantes superiores para os custos \mathcal{H}_2 e \mathcal{H}_∞ e especificações de alocação regional de pólos) são considerados em um conjunto finito de pontos do domínio de incerteza politópico, que inclui inicialmente apenas o conjunto de vértices do politopo. Ao invés de construir um algoritmo de síntese baseado em LMIs, é empregado um algoritmo de otimização, adequado para tratar problemas não convexos e não diferenciáveis, cujas variáveis de otimização são os próprios parâmetros do controlador, filtro ou modelo reduzido. Na etapa de análise, a verificação dos objetivos e das restrições para todo o politopo é realizada pelos procedimentos de análise de estabilidade e de desempenho apresentados nos capítulos 3 e 4. Se for verificado que o ponto de máximo de alguma função objetivo ou que alguma restrição é violada em um ponto do politopo que não pertence ao conjunto de pontos considerado na etapa de otimização, então este ponto é incluído no conjunto finito e nova síntese é realizada. O procedimento finaliza quando for comprovado que todas as restrições são atendidas para todo o politopo e que não existe a possibilidade de minimizar ainda mais as funções objetivo pela inclusão de novos pontos no conjunto finito.

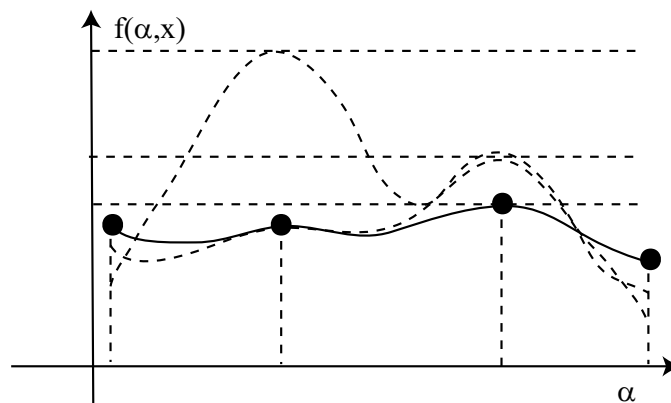
Para ilustrar a estratégia de projeto proposta, considere a Figura 5.1 que apresenta uma simulação deste método de otimização aplicado a uma função objetivo não convexa, $f(\alpha, x) : \mathbb{R} \times \mathbb{R}^d \mapsto \mathbb{R}$, sendo α uma variável escalar limitada, que caracteriza o domínio de incerteza, e x o vetor de parâmetros de otimização. Nesta figura, as curvas sólidas representam os valores de $f(\alpha, x)$ após as otimizações, considerando o conjunto finito de valores de α indicados. Na primeira iteração do procedimento de projeto, Figura 5.1(a), o vetor x é calculado para minimizar $f(\alpha, x)$ considerando apenas os dois valores extremos de α . Na etapa de análise, é verificado que o máximo global ocorre para um valor de α no interior do intervalo (indicado pela seta). Neste caso, este ponto α é incluído no conjunto finito e nova iteração do procedimento é efetuada. Na segunda iteração, Figura 5.1(b), o vetor x é determinado considerando agora os dois valores extremos de α mais o ponto de máximo identificado na iteração anterior. O efeito da inclusão do terceiro ponto é o de minimizar a curva próxima a ele forçando a mesma para o nível dos outros dois pontos. Como o máximo de $f(\alpha, x)$ ocorre novamente fora do conjunto finito, uma nova iteração é efetuada incluindo este quarto ponto. Na terceira iteração, Figura 5.1(c), o vetor x obtido resulta em uma curva suavizada de $f(\alpha, x)$ na qual o valor de máximo ocorre em um ponto já considerado e deste modo o procedimento de projeto é finalizado.



(a) 1ª iteração



(b) 2ª iteração



(c) 3ª iteração

Figura 5.1: Simulação do procedimento de otimização pela inclusão progressiva de pontos de fixação.

Observe pelas linhas tracejadas horizontais da Figura 5.1 o decaimento gradativo da função $f(\alpha, x)$ a cada iteração. Este efeito de suavização da curva da função objetivo pela inclusão progressiva de pontos de “fixação” pode ser generalizado para hiper-superfícies em qualquer dimensão. A Figura 7.2, página 151, apresenta o comportamento do procedimento de otimização em duas dimensões para um caso real de aplicação do procedimento de projeto. Não é do conhecimento do autor que tal estratégia de otimização, apesar de intuitiva, tenha sido empregada anteriormente. Será verificado por vários exemplos a serem apresentados nos capítulos subseqüentes, que o procedimento iterativo de projeto proposto fornece controladores, filtros ou modelos reduzidos com melhores desempenhos \mathcal{H}_2 e \mathcal{H}_∞ em comparação com os métodos baseados em formulações LMIs, após um número pequeno de iterações.

5.2 Formulação do Problema

Sejam $T_2(\alpha, \mathcal{K})$ e $T_\infty(\alpha, \mathcal{K})$ as funções de transferência do sistema projetado relacionadas aos desempenhos \mathcal{H}_2 e \mathcal{H}_∞ respectivamente, para um determinado ponto do domínio politópico de incerteza, Ω , especificado pelo vetor α , e um determinado controlador, filtro ou modelo reduzido, genericamente designado por \mathcal{K} , sendo a dependência em s ou z omitida para simplificar a notação. Como discutido nos capítulos anteriores, o vetor α pode representar os primeiros $N - 1$ coeficientes da combinação convexa, no caso de modelos politópicos: $\alpha = [\theta_1, \dots, \theta_{N-1}]^T \in \Omega_M$, sendo Ω_M o simplexo definido como

$$\Omega_M \triangleq \left\{ \alpha \in \mathbb{R}^{N-1} : \theta_i \geq 0, i = 1, \dots, N-1, \sum_{i=1}^{N-1} \theta_i \leq 1 \right\} \quad (5.1)$$

com $\theta_N = 1 - \sum_{i=1}^{N-1} \theta_i$. O vetor α também pode representar o vetor de parâmetros incertos, $\alpha = [p_1, \dots, p_d]^T \in \Omega_p$, no caso de modelos com dependência afim de parâmetros, sendo Ω_p um hiper-retângulo ou outro politopo qualquer. O símbolo Ω será utilizado para representar tanto Ω_M como Ω_p .

Seja Γ o conjunto de soluções \mathcal{K} que atendem as restrições de posicionamento regional de pólos:

$$\Gamma \triangleq \{ \mathcal{K} : \lambda(\bar{A}(\alpha, \mathcal{K})) \subset \mathcal{D}, \forall \alpha \in \Omega \} \quad (5.2)$$

sendo $\mathcal{D} \subset \mathbb{C}$ uma região convexa do plano complexo e simétrica em relação ao eixo das abscissas, $\lambda(\cdot)$ o espectro do argumento e \bar{A} a matriz da equação dinâmica do sistema sendo projetado.

O projeto de controladores, filtros ou modelos reduzidos robustos $\mathcal{H}_2/\mathcal{H}_\infty$ consiste em um problema de otimização multiobjetivo em que se deseja determinar os elementos de \mathcal{K} que minimizem o

seguinte vetor de objetivos:

$$J(\mathcal{K}) = \begin{bmatrix} \max_{\alpha \in \Omega} \|T_2(\alpha, \mathcal{K})\|_2 \\ \max_{\alpha \in \Omega} \|T_\infty(\alpha, \mathcal{K})\|_\infty \end{bmatrix} \quad (5.3)$$

e que pertença ao conjunto Γ das soluções \mathcal{K} que satisfazem as restrições de posicionamento de pólos.

O problema conceitual tratado aqui pode ser colocado como:

Problema Multiobjetivo do Custo Garantido $\mathcal{H}_2/\mathcal{H}_\infty$: Encontre os controladores, filtros ou modelos reduzidos \mathcal{K}^* que pertençam ao conjunto de Pareto Γ^* :

$$\Gamma^* \triangleq \{\mathcal{K}^* \in \Gamma : \nexists \mathcal{K} \in \Gamma \mid J(\mathcal{K}) \leq J(\mathcal{K}^*), J(\mathcal{K}) \neq J(\mathcal{K}^*)\} \quad (5.4)$$

Os operadores vetoriais de comparação \leq e \neq estão no contexto utilizado em otimização vetorial correspondendo a (Chankong e Haimes, 1983): $x \leq y \Rightarrow x_i \leq y_i$, para todo i até a dimensão dos vetores x e y , e $x \neq y \Rightarrow x_i \neq y_i$ para algum i . O conjunto Γ^* como definido em (5.4) pode ser interpretado como o resultado de minimizações simultâneas dos funcionais no vetor $J(\cdot)$.

5.3 Procedimento de Projeto Proposto

Considere o conjunto finito de pontos do domínio politópico de incertezas, inicializado como o conjunto dos vértices do politopo:

$$\tilde{\Omega} = \text{Vert}(\Omega) \quad (5.5)$$

Para um dado \mathcal{K} , defina os valores máximos das normas \mathcal{H}_2 e \mathcal{H}_∞ no conjunto finito $\tilde{\Omega} \subset \Omega$:

$$\begin{aligned} \tilde{\delta}_{p.c.} &\triangleq \max_{\alpha \in \tilde{\Omega}} \|T_2(\alpha, \mathcal{K})\|_2 \\ \tilde{\gamma}_{p.c.} &\triangleq \max_{\alpha \in \tilde{\Omega}} \|T_\infty(\alpha, \mathcal{K})\|_\infty \end{aligned} \quad (5.6)$$

e os valores máximos considerando todo o politopo:

$$\begin{aligned} \delta_{p.c.} &= \max_{\alpha \in \Omega} \|T_2(\alpha, \mathcal{K})\|_2 \\ \gamma_{p.c.} &= \max_{\alpha \in \Omega} \|T_\infty(\alpha, \mathcal{K})\|_\infty \end{aligned} \quad (5.7)$$

Sejam $\alpha_{(2)}$ e $\alpha_{(\infty)}$ os vetores de coordenadas correspondentes aos pontos de máximo em todo o polítopo:

$$\alpha_{(2)} = \arg \max_{\alpha \in \Omega} \|T_2(\alpha, \mathcal{K})\|_2 \quad (5.8)$$

$$\alpha_{(\infty)} = \arg \max_{\alpha \in \Omega} \|T_\infty(\alpha, \mathcal{K})\|_\infty$$

Uma vez que existem técnicas eficientes para a solução de problemas de otimização escalares, é interessante transformar o problema conceitual multiobjetivo, ou vetorial, em um problema de otimização escalar. Neste trabalho será utilizada uma combinação de duas técnicas de escalarização: o problema- λ e o problema- ϵ . No problema- λ , o vetor de funções objetivos é transformado em uma função escalar como uma soma ponderada. No problema- ϵ , uma função objetivo é considerada de cada vez, com as demais sendo tratadas como restrições. Baseado nestas técnicas de escalarização, é proposto o seguinte problema auxiliar:

Problema Auxiliar: *Dados os escalares $\lambda_2 \geq 0$, $\lambda_\infty \geq 0$, com $\lambda_2 + \lambda_\infty = 1$, $\delta > 0$ e $\gamma > 0$, encontre o controlador, filtro ou modelo reduzido $\tilde{\mathcal{K}}^*$ tal que:*

$$\begin{aligned} \tilde{\mathcal{K}}^* &= \arg \min_{\mathcal{K}} \max_{\alpha \in \tilde{\Omega}} (\lambda_2 \|T_2(\alpha, \mathcal{K})\|_2 + \lambda_\infty \|T_\infty(\alpha, \mathcal{K})\|_\infty) \\ \text{sujeito a: } &\begin{cases} \max_{\alpha \in \tilde{\Omega}} \|T_2(\alpha, \mathcal{K})\|_2 \leq \delta \\ \max_{\alpha \in \tilde{\Omega}} \|T_\infty(\alpha, \mathcal{K})\|_\infty \leq \gamma \\ \mathcal{K} \in \Gamma \end{cases} \end{aligned} \quad (5.9)$$

Através dos valores de λ_2 , λ_∞ , δ e γ , podem se configurar diferentes problemas de otimização escalar, em que $\delta = \infty$ e/ou $\gamma = \infty$ exclui a restrição correspondente.

Como discutido na introdução do capítulo, a idéia por trás do procedimento geral de projeto, que será apresentado a seguir, é simples e intuitiva. O controlador, filtro ou modelo reduzido é calculado como a solução do problema de otimização auxiliar (5.9), que considera apenas um conjunto finito de pontos, $\tilde{\Omega}$, do conjunto de infinitos pontos, Ω , e é validado “a posteriori”, para todo o polítopo, por meio de procedimentos de análise de estabilidade robusta e de desempenho robusto que devem ter a capacidade de localizar e determinar os pontos de pior caso: $\delta_{p.c.} = \|T_2(\alpha_{(2)}, \tilde{\mathcal{K}}^*)\|_2$ e $\gamma_{p.c.} = \|T_\infty(\alpha_{(\infty)}, \tilde{\mathcal{K}}^*)\|_\infty$. Se as restrições de estabilidade e desempenho não forem atendidas ou se os pontos de pior caso no polítopo não fazem parte do conjunto finito $\tilde{\Omega}$, então estes pontos são incluídos no conjunto e uma nova iteração do algoritmo é processada. Para evitar iterações desnecessárias, na

minimização da função objetivo, a inclusão de novos pontos é feita apenas quando a diferença relativa entre o pior caso de norma no politopo e o pior caso no conjunto finito $\tilde{\Omega}$ for maior que uma tolerância ε_δ .

O procedimento geral de projeto proposto neste trabalho é descrito como:

Procedimento Geral de Projeto

Passo 1. Inicialize $i \leftarrow 0$, $\tilde{\Omega}_0 \leftarrow \text{Vert}(\Omega)$.

Passo 2. $i \leftarrow i + 1$, $\tilde{\Omega}_i \leftarrow \tilde{\Omega}_{i-1}$.

Passo 3. Resolva o problema auxiliar para encontrar $\tilde{\mathcal{K}}^*$, $\tilde{\delta}_{p.c.}$ e/ou $\tilde{\gamma}_{p.c.}$.

Passo 4. Se $(\lambda_2 > 0)$ ou $(\delta < \infty)$, então calcule o custo ε -garantido \mathcal{H}_2 , δ_c , o valor $\delta_{p.c.}$ e o vetor de coordenadas correspondente $\alpha_{(2)}$ para $\tilde{\mathcal{K}}^*$.

Se $\alpha_{(2)} \notin \tilde{\Omega}_i$, então se $((\lambda_2 > 0)$ e $((\delta_{p.c.} - \tilde{\delta}_{p.c.})/\tilde{\delta}_{p.c.} > \varepsilon_\delta))$ ou $(\delta_c > \delta)$, então $\tilde{\Omega}_i \leftarrow \tilde{\Omega}_i \cup \alpha_{(2)}$.

Passo 5. Se $(\lambda_\infty > 0)$ ou $(\gamma < \infty)$, então calcule o custo ε -garantido \mathcal{H}_∞ , γ_c , o valor $\gamma_{p.c.}$ e o vetor de coordenadas correspondente $\alpha_{(\infty)}$ para $\tilde{\mathcal{K}}^*$.

Se $\alpha_{(\infty)} \notin \tilde{\Omega}_i$, então se $((\lambda_\infty > 0)$ e $((\gamma_{p.c.} - \tilde{\gamma}_{p.c.})/\tilde{\gamma}_{p.c.} > \varepsilon_\delta))$ ou $(\gamma_c > \gamma)$, então $\tilde{\Omega}_i \leftarrow \tilde{\Omega}_i \cup \alpha_{(\infty)}$.

Passo 6. Verifique se $\exists \alpha_{(u)} \in \Omega$ tal que $\lambda(\bar{A}(\alpha_{(u)}, \tilde{\mathcal{K}}^*)) \notin \mathcal{D}$, então $\tilde{\Omega}_i \leftarrow \tilde{\Omega}_i \cup \alpha_{(u)}$.

Passo 7. Se $\tilde{\Omega}_i \neq \tilde{\Omega}_{i-1}$ vá para o passo 2. Caso contrário, $\mathcal{K}^* \leftarrow \tilde{\mathcal{K}}^*$, fim.

Os detalhes dos passos do procedimento geral de projeto são apresentados na seqüência.

5.3.1 Solução do Problema Auxiliar

No passo 3 do procedimento geral de projeto proposto é necessária a solução de um problema de otimização escalar restrito não convexo definido em (5.9). Neste trabalho, o problema auxiliar de otimização escalar é solucionado através do algoritmo elipsoidal, que é um algoritmo de fácil implementação e adequado para tratar da otimização de funções não diferenciáveis (Takahashi et al., 2003).

Um elipsóide \mathcal{E} pode ser descrito como

$$\mathcal{E} = \{z \mid (z - x)^T Q^{-1} (z - x) \leq 1\} \quad (5.10)$$

sendo x um vetor que define o centro do elipsóide e $Q = Q^T > 0$ uma matriz simétrica definida positiva com autovalores $\lambda_i > 0$, que determina as dimensões e as direções dos semi-eixos do elipsóide.

A idéia básica do algoritmo elipsoidal é descrita a seguir (Boyd et al., 1994). O algoritmo inicia com um elipsóide \mathcal{E}_0 que contém a solução ótima do problema de otimização. Na seqüência, é calculado um hiper-plano de corte que passa através do centro do elipsóide, x_0 , com base em um vetor m_0 , tal que a solução ótima se localiza na metade do espaço definido por $\{z \mid m_0^T(z - x_0) < 0\}$. Deste modo, pode se garantir que a metade do elipsóide dada por

$$\mathcal{E}_0 \cap \{z \mid m_0^T(z - x_0) < 0\}$$

contém o ponto ótimo. A seguir é calculado o elipsóide \mathcal{E}_1 de mínimo volume que contém a metade do elipsóide na qual está localizada a solução ótima. O processo é repetido e, a medida que o volume do elipsóide tende a zero, o centro do elipsóide tende para a solução ótima.

Como será detalhado a seguir, o vetor m_k , que define o hiper-plano de corte, corresponde ao gradiente ou sub-gradiente da função objetivo $f(x)$ ou das restrições. O sub-gradiente de f no ponto $x \in \mathbb{R}^d$ é qualquer vetor $m \in \mathbb{R}^d$ tal que

$$f(z) \geq f(x) + m^T(z - x) \quad \forall z \in \mathbb{R}^d \quad (5.11)$$

Dado o elipsóide inicial, definido por x_0 e Q_0 , sendo Q_0 normalmente uma matriz diagonal, o algoritmo elipsoidal pode ser descrito pelas seguintes equações recursivas (Boyd et al., 1994):

$$\tilde{m} = \frac{m_k}{\sqrt{m_k^T Q_k m_k}} \quad (5.12)$$

$$x_{k+1} = x_k - \frac{1}{d+1} Q_k \tilde{m} \quad (5.13)$$

$$Q_{k+1} = \frac{d^2}{d^2 - 1} \left(Q_k - \frac{2}{d+1} Q_k \tilde{m} \tilde{m}^T Q_k \right) \quad (5.14)$$

sendo $x_k \in \mathbb{R}^d$ o vetor de parâmetros de otimização, com os elementos das matrizes do controlador, filtro ou modelo reduzido. Observe que estas fórmulas recursivas são válidas apenas para $d > 1$. No caso em que $d = 1$, a idéia básica do algoritmo elipsoidal equivale a um algoritmo de bisseção para busca unidimensional, cujo próximo elipsóide corresponde à metade do segmento de reta onde

se encontra a solução ótima:

$$x_{k+1} = \begin{cases} x_k + \frac{1}{2}\sqrt{Q_k}, & \text{se } m_k < 0 \\ x_k - \frac{1}{2}\sqrt{Q_k}, & \text{se } m_k > 0 \end{cases} \quad (5.15)$$

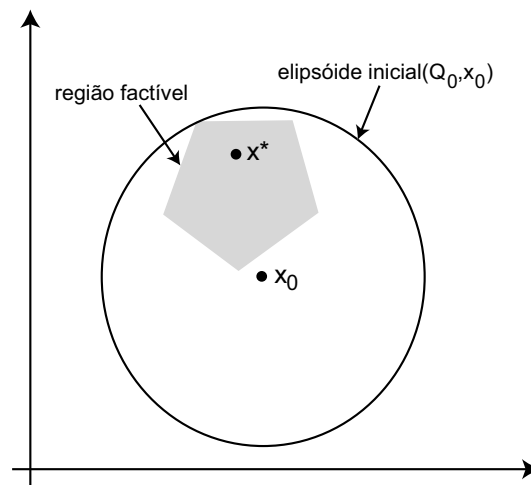
$$Q_{k+1} = \frac{1}{4}Q_k \quad (5.16)$$

Seja $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ a função objetivo e $g(x) : \mathbb{R}^d \rightarrow \mathbb{R}^r$ o vetor de restrições. Como foi apresentado, o algoritmo elipsoidal começa com um elipsóide, \mathcal{E}_0 , que contém o ponto de solução ótima, caracterizado pela matriz Q_0 e pela solução inicial x_0 , que é o centro do elipsóide, como mostrado na Figura 5.2(a). Em cada iteração, é calculado um vetor não nulo, designado m_k , que define um hiperplano de corte que passa pelo centro do elipsóide e que divide o elipsóide em dois, como apresentado na Figura 5.2(b). Quando x_k não é uma solução factível, no algoritmo elipsoidal convencional, o vetor m_k é calculado como o gradiente (ou sub-gradiente) da restrição mais violada. Neste trabalho o vetor m_k é calculado baseado no *algoritmo cone-elipsoidal* (CEA) proposto em Takahashi et al. (2003). Neste método, quando x_k não é factível, o vetor m_k é o vetor normalizado $m_k = m/\|m\|$ calculado como a soma dos gradientes (ou sub-gradientes) das restrições ativas. Baseado no vetor m_k , o vetor x_k e a matriz Q_k são atualizados pelas equações recursivas para calcular o elipsóide de volume mínimo que contém a metade do elipsóide que inclui a solução ótima. Quando x_k é uma solução factível, como mostrado na Figura 5.2(c), o vetor m_k é calculado como o gradiente (ou sub-gradiente) da função objetivo.

A cada iteração do algoritmo elipsoidal, o volume do elipsóide diminui geometricamente sendo que o volume do elipsóide \mathcal{E}_k é dado por (Boyd et al., 1994):

$$\text{vol}(\mathcal{E}_k) \leq e^{-\frac{k}{2d}} \text{vol}(\mathcal{E}_0)$$

Observe que a taxa de redução do elipsóide depende da dimensão do vetor de parâmetros de otimização. Apesar do volume ser sempre menor, o diâmetro máximo de elipsóide \mathcal{E}_{k+1} pode ser maior que o do elipsóide \mathcal{E}_k , sendo possível obter uma solução não pertencente ao elipsóide inicial.



(a) Solução inicial não factível.

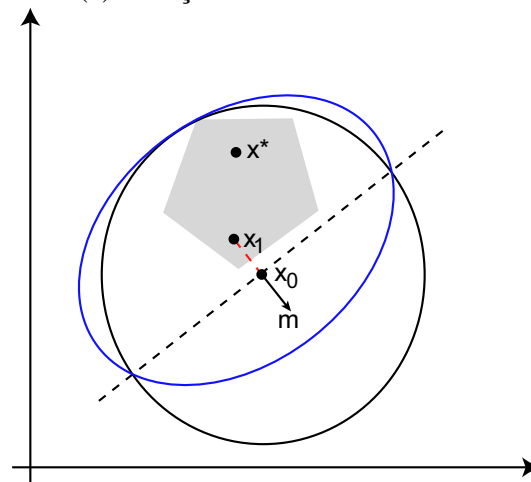
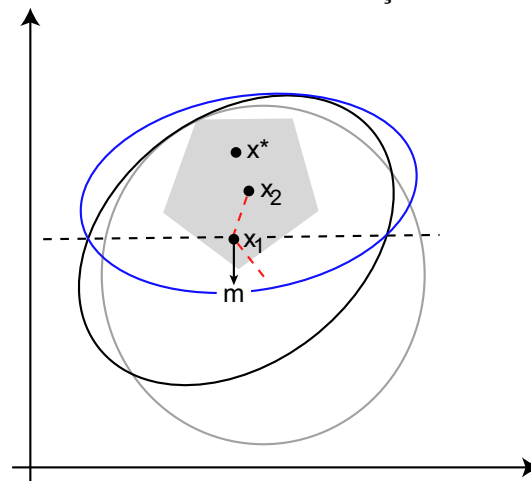
(b) Cálculo do vetor m com solução não factível.(c) Cálculo do vetor m com solução factível.

Figura 5.2: Descrição do algoritmo elipsoidal.

O cálculo do vetor m baseado no algoritmo cone-elipsoidal pode ser formulado como:

$$m = \begin{cases} \nabla f(x) & \text{se } g_j(x) < 0, \forall j = 1, \dots, r \\ \sum_{j=1}^r s_j(x) & \text{se } \exists j \mid g_j(x) \geq 0 \end{cases} \quad (5.17)$$

com

$$s_j(x) = \begin{cases} 0 & \text{se } g_j(x) < 0 \\ \nabla g_j(x) & \text{se } g_j(x) \geq 0 \end{cases} \quad (5.18)$$

sendo $\nabla(\cdot)$ a função gradiente (ou sub-gradiente).

Considerando o problema auxiliar (5.9), a função objetivo é definida como:

$$f(x) \triangleq \lambda_2 \tilde{\delta}_{p.c.}(x) + \lambda_\infty \tilde{\gamma}_{p.c.}(x)$$

sendo x os elementos das matrizes do controlador, filtro ou modelo reduzido. O vetor de funções de restrição pode ser definido de diferentes formas. Seja $\lambda_i \in \lambda(\bar{A}(\alpha, x))$ representado por $\lambda_i = \sigma_i + j\omega_{d,i}$, ou na forma polar, $\lambda_i = \omega_{n,i} \angle \theta_{p,i}$. Considerando as quatro regiões tratadas na seção 3.2 (ver Figura 3.1, pág. 30) e representando $\bar{A}(\alpha, x)$ simplesmente por \bar{A} , o vetor de funções de restrição é definido nesta tese como

$$g(x) \triangleq \begin{bmatrix} \max_{\lambda_i \in \lambda(\bar{A}), \alpha \in \tilde{\Omega}} \sigma_i - h_1 \\ \max_{\lambda_i \in \lambda(\bar{A}), \alpha \in \tilde{\Omega}} \sqrt{(\sigma_i - q)^2 + \omega_{d,i}^2} - r \\ \max_{\lambda_i \in \lambda(\bar{A}), \alpha \in \tilde{\Omega}} \pi - \theta_{p,i} - \theta \\ \max_{\lambda_i \in \lambda(\bar{A}), \alpha \in \tilde{\Omega}} h_2 - \sigma_i \\ \tilde{\delta}_{p.c.}(x) - \delta \\ \tilde{\gamma}_{p.c.}(x) - \gamma \end{bmatrix}$$

sendo que a inclusão de cada elemento do vetor $g(x)$ ocorre nos casos de $h_1 < \infty$, $r < \infty$, $\theta < \pi/2$, $h_2 > -\infty$, $\delta < \infty$ e $\gamma < \infty$, respectivamente.

Para a implementação do algoritmo de otimização cone-elipsoidal no procedimento geral de projeto proposto, estando disponíveis funções para o cálculo das normas \mathcal{H}_2 e \mathcal{H}_∞ e para a determinação de autovalores, a função objetivo e o vetor de restrições podem ser prontamente computados em termos dos parâmetros de otimização. Entretanto, não existem disponíveis funções para calcular explicitamente os gradientes (ou sub-gradientes) da função objetivo e das restrições. Nesta tese é utilizado um algoritmo simples para o cálculo numérico do gradiente por diferenças finitas.

Seja e^i a i -ésima coluna da matriz identidade $d \times d$. Considere-se um certo $\Delta > 0$, tal que $\Delta \approx 0$. O algoritmo de cálculo do vetor gradiente v da função $f(\cdot)$ no ponto $x \in \mathbb{R}^d$ pode ser definido como:

Algoritmo: Cálculo do Gradiente por Diferenças Finitas

para $i = 1, 2, \dots, d$

$$v_i \leftarrow \frac{f(x + \Delta e^i) - f(x)}{\Delta}$$

fim para

$$v \leftarrow [v_1 \ \dots \ v_d]^T$$

fim algoritmo

O valor de Δ pode ser escolhido na faixa de 10^{-5} a 10^{-10} sem influência considerável na maioria dos problemas abordados nesta tese.

O algoritmo de otimização finaliza quando $(f_{max} - f_{min})/f_{min} \leq \epsilon$, sendo f_{max} e f_{min} os valores máximo e mínimo da função objetivo nas últimas N_ϵ iterações e ϵ a precisão relativa requerida.

Para aprimorar a estabilidade numérica do cálculo da matriz quadrada simétrica definida não negativa, Q , no algoritmo de otimização elipsoidal, pode-se utilizar uma técnica de fatoração (ver, por exemplo, discussão em Goldfarb e Todd (1982, seção 4)). É considerado aqui a fatoração UDU^T :

$$Q = UDU^T \quad (5.19)$$

sendo U uma matriz quadrada triangular superior com elementos unitários na diagonal e D uma matriz diagonal. Tal fatoração é utilizada com sucesso no algoritmo de estimação recursiva de parâmetros por mínimos quadrados em estratégias de controle adaptativo. A atualização da matriz fatorada é dada por

$$U_{k+1} = U_k \bar{U} \quad (5.20)$$

$$D_{k+1} = \bar{D} \quad (5.21)$$

sendo \bar{U} e \bar{D} obtidas da fatoração UDU^T :

$$\bar{U} \bar{D} \bar{U}^T = \beta_2 \left(D_k - \frac{\beta_3 (D_k U_k m_k)(D_k U_k m_k)^T}{m_k^T Q_k m_k} \right) \quad (5.22)$$

Nos algoritmos elipsoidais existentes, quando a matriz Q perde sua característica de ser definida positiva, devido aos erros numéricos acumulados após um número elevado de iterações, ou o algo-

ritmo é finalizado ou a matriz Q é reinicializada com um valor menor que o valor inicial de Q_0 . Nesta tese, em vários testes implementados com a utilização da fatoração UDU^T , além de retardar significativamente a degeneração da matriz Q , o uso da fatoração suaviza a variação da função objetivo, característica deste método, fazendo com que o algoritmo atenda o critério de parada mais rapidamente. Deste modo, é observado que, mesmo com o custo computacional adicional para o cálculo da fatoração, o tempo total de otimização acaba sendo reduzido. Mesmo usando a fatoração UDU^T , ainda existe a possibilidade de degeneração da matriz Q . Neste trabalho, caso isto ocorra, é adotada a reinicialização da matriz Q com valor 20% menor do que o valor inicial anterior.

Considere a fatoração de uma matriz $A \in \mathbb{R}^{n \times n}$, o algoritmo de fatoração UDU^T é apresentado a seguir:

Algoritmo: Fatoração UDU^T

$D_{nn} \leftarrow A_{nn};$

$U_{nn} \leftarrow 1;$

para $i = n - 1, n - 2, \dots, 1$

$U_{in} \leftarrow A_{in}/D_{nn};$

fim para

para $j = n - 1, n - 2, \dots, 1$

$D_{jj} \leftarrow A_{jj};$

para $k = j + 1, j + 2, \dots, n$

$D_{jj} \leftarrow D_{jj} - D_{kk}U_{jk}^2;$

fim para

$U_{jj} \leftarrow 1;$

para $k = j - 1, j - 2, \dots, 1$

$U_{ij} \leftarrow A_{ij};$

para $k = j + 1, j + 2, \dots, n$

$U_{ij} \leftarrow U_{i,j} - D_{kk}U_{ik}U_{jk};$

fim para

$U_{ij} \leftarrow U_{ij}/D_{jj};$

fim para

fim para

fim algoritmo

Para ilustrar a eficiência do uso de fatoração UDU^T no algoritmo de otimização cone-elipsoidal, observe a Figura 5.3 que apresenta as evoluções da função objetivo em um processo de otimização, com mesmos critérios de parada, sem e com a fatoração. Sem a fatoração, o algoritmo convergiu com 118 iterações com o tempo de processamento de 23,444s. Com a fatoração, o algoritmo convergiu

com 98 iterações com o tempo de processamento de 19,678s.

A Figura 5.4 apresenta as evoluções finais da função objetivo, em um processo de otimização mais complexo, sem e com a fatoração e mesmos critérios de parada. Sem a fatoração, o algoritmo convergiu com 2944 iterações com o tempo de processamento de 57,5min e o menor valor da função objetivo para uma solução factível igual a 1,3083. Com a fatoração, o algoritmo convergiu com 2908 iterações com o tempo de processamento de 52,7min e o menor valor da função objetivo para uma solução factível igual a 1,3067. Mesmo não sendo tão notáveis neste caso, o uso da fatoração levou a melhores valores em todos os critérios analisados, confirmando a validade de seu uso.

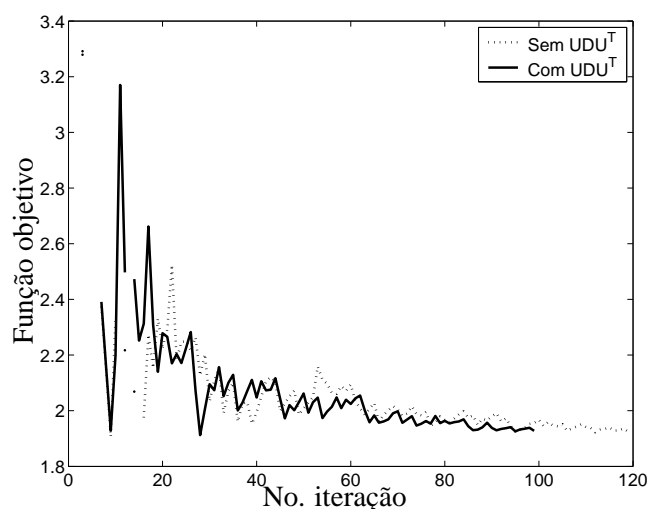


Figura 5.3: Comparação do comportamento da função objetivo sem e com a fatoração UDU^T .

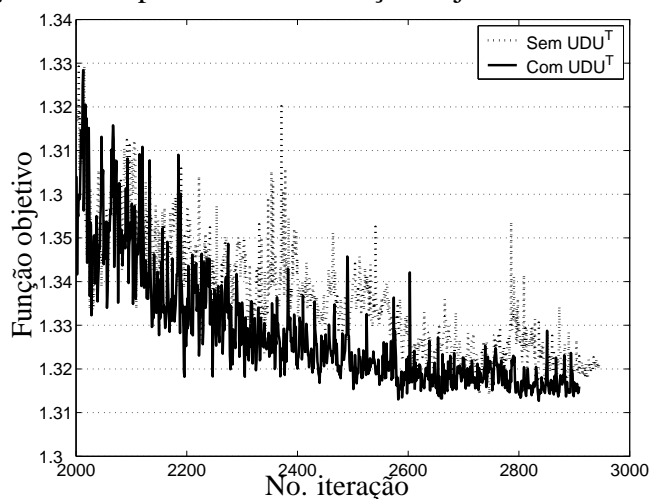


Figura 5.4: Comparação do comportamento da função objetivo sem e com a fatoração UDU^T .

Nas Figuras 5.3 e 5.4, os pontos ausentes correspondem a valores infinito da função objetivo que ocorrem quando se deseja calcular a norma de um sistema instável relacionado com uma solução não

factível.

5.3.2 Cálculo dos Custos

Os cálculos dos custos ε -garantidos \mathcal{H}_2 e \mathcal{H}_∞ , δ_c e γ_c , dos piores casos de norma \mathcal{H}_2 e \mathcal{H}_∞ no polítopo, $\delta_{p.c.}$ e $\gamma_{p.c.}$, e das coordenadas dos piores casos, $\alpha_{(2)}$ e $\alpha_{(\infty)}$, são realizados pelo procedimento de análise robusta de desempenho baseado no algoritmo branch-and-bound, apresentado no Capítulo 4. No algoritmo BnB proposto, a função limitante superior é o custo garantido \mathcal{H}_2 ou \mathcal{H}_∞ , $\delta_{c.g.}$ ou $\gamma_{c.g.}$. Como discutido no Capítulo 4, os cálculos dos custos garantidos podem ser baseados em qualquer formulação LMI existente na literatura, porém, a eficiência do algoritmo BnB depende do melhor compromisso entre custo computacional e conservadorismo.

Considere que $x(t) \in \mathbb{R}^n$, $w(t) \in \mathbb{R}^{pw}$ e $z(t) \in \mathbb{R}^{mz}$. Nas aplicações do procedimento geral de projeto, a serem apresentadas nos capítulos subseqüentes, para sistemas a tempo contínuo, optou-se por utilizar o cálculo do custo garantido \mathcal{H}_2 baseado na combinação dos Lema 1 e Lema 2 apresentados em de Oliveira et al. (2004a): seja $X \in \mathbb{R}^{m_z \times m_z}$, $W_i = W_i^T \in \mathbb{R}^{n \times n}$, para $i = 1, \dots, N$, então

$$\delta_{c.g.}^2 = \min_{X, W_i} \text{Tr}(X)$$

$$\text{sujeito a: } W_i \succ 0, \quad i = 1, \dots, N$$

$$C_i W_i C_i^T - X \preceq 0; \quad i = 1, \dots, N$$

$$C_i W_i C_i^T + C_j W_j C_j^T + C_i W_j C_i^T - 3X \preceq 0; \quad i = 1, \dots, N, \quad j \neq i, \quad j = 1, \dots, N \quad (5.23)$$

$$C_i W_j C_k^T + C_k W_j C_i^T + C_j W_i C_k^T + C_k W_i C_j^T + C_i W_k C_j^T + C_j W_k C_i^T - 6X \preceq 0; \\ i = 1, \dots, N - 2, \quad j = i + 1, \dots, N - 1, \quad k = j + 1, \dots, N$$

$$A_i W_i + W_i A_i^T \preceq -B_i B_i^T; \quad i = 1, \dots, N$$

$$A_i W_j + W_j A_i^T + A_j W_i + W_i A_j^T \preceq -(B_i B_i^T + B_j B_j^T); \\ i = 1, \dots, N - 1, \quad j = i + 1, \dots, N$$

Eventualmente, pode ser aplicada a versão dual, com a substituição de (A, B, C, D) por (A^T, C^T, B^T, D^T) , em caso de resultar em uma diferença significativa de tempo computacional.

Para o cálculo do custo garantido \mathcal{H}_∞ , optou-se pela formulação baseada no Lema 1 apresentado

em de Oliveira et al. (2004b): seja $P_i = P_i^T \in \mathbb{R}^{n \times n}$, $i = 1, \dots, N$, então

$$\gamma_{c.g.}^2 = \min_{P_i} \mu_c$$

sujeito a: $P_i \succ 0$, $i = 1, \dots, N$

$$\begin{bmatrix} A_i^T P_i + P_i A_i & P_i B_i & C_i^T \\ * & -I & D_i^T \\ * & * & -\mu_c I \end{bmatrix} \prec 0, \quad i = 1, \dots, N \quad (5.24)$$

$$\begin{bmatrix} A_i^T P_j + P_j A_i + A_j^T P_i + P_i A_j & P_i B_j + P_j B_i & C_i^T + C_j^T \\ * & -2I & D_i^T + D_j^T \\ * & * & -2\mu_c I \end{bmatrix} \prec 0$$

$i = 1, \dots, N-1, \quad j = i+1, \dots, N$

sendo que o símbolo '*' nas matrizes acima representam termos simétricos em relação à diagonal principal.

No caso de sistemas discretos no tempo, para o cálculo do custo garantido \mathcal{H}_2 é adotada a formulação baseada no Teorema 3 apresentado em de Oliveira et al. (2002), ou sua versão dual, com uma pequena modificação para tratar o caso no qual $D_i \neq 0$: seja $G \in \mathbb{R}^{n \times n}$, $W_i = W_i^T \in \mathbb{R}^{m_z \times m_z}$, $P_i = P_i^T \in \mathbb{R}^{n \times n}$, $i = 1, \dots, N$, então

$$\delta_c^2 = \min_{G, W_i, P_i} \mu$$

sujeito a: $P_i \succ 0$, $i = 1, \dots, N$

$$\text{Tr}(W_i) < \mu, \quad i = 1, \dots, N$$

$$\begin{bmatrix} W_i & C_i G & D_i \\ * & G + G^T - P_i & 0 \\ * & * & I \end{bmatrix} \succ 0, \quad i = 1, \dots, N \quad (5.25)$$

$$\begin{bmatrix} P_i & A_i G & B_i \\ * & G + G^T - P_i & 0 \\ * & * & I \end{bmatrix} \succ 0, \quad i = 1, \dots, N$$

Para o cálculo do custo garantido \mathcal{H}_∞ é utilizada a formulação baseada no Teorema 4 apresentado em de Oliveira et al. (2002): seja $G \in \mathbb{R}^{n \times n}$, $P_i = P_i^T \in \mathbb{R}^{n \times n}$, $i = 1, \dots, N$, então

$$\begin{aligned} \gamma_c^2 &= \min_{G, P_i} \mu \\ \text{sujeito a: } & P_i \succ 0, \quad i = 1, \dots, N \\ & \begin{bmatrix} P_i & A_i G & B_i & 0 \\ * & G + G^T - P_i & 0 & G^T C_i^T \\ * & * & I & D_i^T \\ * & * & * & \mu I \end{bmatrix} \succ 0, \quad i = 1, \dots, N \end{aligned} \quad (5.26)$$

5.3.3 Verificação do Posicionamento Regional de Pólos

No caso do projeto de controladores com posicionamento de pólos, é necessário verificar se as restrições de posicionamento regional de pólos são atendidas em todo o politopo e, em caso contrário, identificar um valor de $\alpha \in \Omega$ para o qual tais restrições sejam violadas. Também neste caso é utilizada a estratégia de partição do politopo combinada com formulações de análise baseadas em LMIs, conforme apresentado no Capítulo 3. Na implementação do procedimento geral de projeto proposto, as restrições de posicionamento regional de pólos são verificadas para cada região individualmente, adotando a formulação LMI considerada a mais adequada para cada caso. Para a região semi-plano esquerdo é aplicado o Lema 1 apresentado em Ramos e Peres (2002) (com a substituição de A por $(A - h_1 I_n)$): se existir $P_i = P_i^T \in \mathbb{R}^{n \times n}$ tal que as seguintes LMIs sejam atendidas

$$\begin{aligned} (A_i - h_1 I_n)^T P_i + P_i (A_i - h_1 I_n) &\prec -I_n, \quad P_i \succ 0, \quad i = 1, \dots, N \\ (A_i - h_1 I_n)^T P_j + P_j (A_i - h_1 I_n) + (A_j - h_1 I_n)^T P_i + P_i (A_j - h_1 I_n) & \\ &\prec -\frac{2}{N-1} I_n, \quad i = 1, \dots, N-1, \quad j = i+1, \dots, N \end{aligned} \quad (5.27)$$

então $\text{Real}(\lambda_i(\bar{A}(\alpha))) < h_1$ para todo i e $\alpha \in \Omega$. Para região disco e semi-plano direito é aplicada a formulação baseada no Teorema 4, apresentado em Peaucelle et al. (2000), e para a região setor é adotada a formulação baseada no Teorema 4 apresentado em Gao e Xue (2004), porém com a matriz Q fixa (mesmo grau de conservadorismo mas com menos variáveis escalares de otimização): se existir

$P_i = P_i^T \in \mathbb{R}^{n \times n}$ e $G_i \in \mathbb{R}^{n \times n}$, $i = 1, \dots, N$, tais que as seguintes LMIs sejam atendidas

$$\begin{aligned}
& \left[\begin{array}{c|c} R_{11} \otimes P_i + R_{12} \otimes (P_i A_i) + R_{12}^T \otimes (A_i^T P_i) & L \otimes (A_i^T G_i^T) \\ \hline L^T \otimes (G_i A_i) & I_m \otimes (P_i - G_i - G_i^T) \end{array} \right] \\
& \prec \left[\begin{array}{c|c} I_m \otimes I_n & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right], P_i \succ 0, i = 1, \dots, N \\
& \left[\begin{array}{c|c} R_{11} \otimes P_i + R_{12} \otimes (P_i A_j) + R_{12}^T \otimes (A_j^T P_i) & L \otimes (A_j^T G_i^T) \\ \hline L^T \otimes (G_i A_j) & I_m \otimes (P_i - G_i - G_i^T) \end{array} \right] \\
& + \left[\begin{array}{c|c} R_{11} \otimes P_j + R_{12} \otimes (P_j A_i) + R_{12}^T \otimes (A_i^T P_j) & L \otimes (A_i^T G_j^T) \\ \hline L^T \otimes (G_j A_i) & I_m \otimes (P_j - G_j - G_j^T) \end{array} \right] \\
& \prec \frac{2}{N-1} \left[\begin{array}{c|c} I_m \otimes I_n & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right], i = 1, \dots, N-1, j = i+1, \dots, N
\end{aligned} \tag{5.28}$$

então o modelo politópico é robustamente \mathcal{D} -estável. As matrizes que caracterizam a região \mathcal{D} , R_{11} , R_{12} e $LL^T = R_{22}$, são as mesmas apresentadas na Seção 3.2.

5.4 Conclusões

Foi apresentado um procedimento geral de projeto robusto $\mathcal{H}_2/\mathcal{H}_\infty$, com posicionamento regional de pólos, que pode ser utilizado em vários problemas de controle tais como: síntese de controladores robustos por realimentação de estado, realimentação estática de saída ou realimentação dinâmica de saída, com qualquer dimensão e estrutura, síntese de filtros robustos, com qualquer dimensão e estrutura e redução de modelos. Na implementação do procedimento proposto, as únicas diferenças para cada tipo de projeto são a forma de compor o sistema projetado (combinação do sistema original e o controlador/filtro/modelo reduzido) e as duas possibilidades de funções de análise (para sistemas a tempo contínuo ou a tempo discreto).

Apesar de não ser uma formulação convexa, o método de otimização proposto nesta tese tem levado a bons resultados em todos os exemplos em que foi testado, comparado com outras estratégias existentes na literatura, como será mostrado nos exemplos que serão apresentados nos próximos três capítulos. Tanto a etapa de síntese, como a etapa de análise do procedimento geral de projeto proposto, não precisam necessariamente serem implementadas conforme foi descrito neste capítulo, entretanto, tanto o algoritmo cone-elipsoidal como o algoritmo branch-and-bound têm demonstrado

serem bastante eficazes para este propósito. É claro que o algoritmo cone-elipsoidal obtém apenas o mínimo local da função objetivo não convexa, sendo interessante, de acordo com a situação, testar diferentes valores iniciais para os parâmetros de otimização na tentativa de obter melhores soluções.

Capítulo 6

Projeto de Controladores

6.1 Introdução

No artigo de Chilali e Gahinet (1996) foi apresentada uma importante metodologia de síntese de controladores robustos $\mathcal{H}_2/\mathcal{H}_\infty$ por realimentação de estado, com alocação regional de pólos, para uma classe geral de sub-regiões convexas do plano complexo, baseada em uma formulação LMI. Entretanto, existe uma desvantagem potencial da formulação apresentada, e de outras que empregam a mesma metodologia, que é fixar uma mesma matriz de Lyapunov para garantir o desempenho do sistema e as restrições de alocação regional de pólos. Isto significa que as únicas soluções possíveis para tais procedimentos de projeto são aquelas que admitem uma mesma matriz de Lyapunov para todo o conjunto de incertezas e para a formulação LMI de todos os objetivos de controle. Como estas soluções são apenas um subconjunto de todas as soluções possíveis relevantes, isto constitui uma fonte de conservadorismo em várias metodologias de síntese de controladores robustos para sistemas com incertezas politópicas.

Vários trabalhos recentes têm abordado este problema, visando caracterizações menos conservadoras, através do uso de multiplicadores ou matrizes de Lyapunov dependentes de parâmetros. Em Peaucelle et al. (2000) é apresentada uma nova condição suficiente para \mathcal{D} -estabilidade robusta baseada em LMIs, que pode ser empregada na síntese de controladores por realimentação de estado, com função de Lyapunov dependente de parâmetros que fornece graus de liberdade adicionais através da introdução de novas variáveis matriciais. Em Apkarian et al. (2001) é proposta uma caracterização LMI baseada em função de Lyapunov dependente de parâmetros para tratar do problema de posicionamento exato dos pólos em malha-fechada otimizando simultaneamente o critério de desempenho \mathcal{H}_2 no caso de sistemas contínuos no tempo. Em Shaked (2001) é apresentada uma formulação LMI com função de Lyapunov dependente de parâmetros, com busca unidimensional de um parâmetro escalar de ajuste, para tratar do problema \mathcal{H}_∞ de sistemas incertos a tempo contínuo. O artigo (Ebihara e

Hagiwara, 2002a) descreve uma caracterização LMI dilatada para síntese de controle robusto \mathcal{H}_2 com alocação regional de pólos para sistemas com incertezas politópicas baseada no uso de variáveis de Lyapunov dependentes de parâmetros. O artigo (Ebihara e Hagiwara, 2002b) estende os resultados de Ebihara e Hagiwara (2002a) através de uma formulação LMI dilatada aperfeiçoada que introduz um parâmetro arbitrário ajustável. Em de Oliveira et al. (2002), é apresentada uma formulação LMI, com funções de Lyapunov dependentes de parâmetros, para o caso de sistemas discretos no tempo. Em Wang e Wilson (2003), a formulação proposta por Apkarian et al. (2001) foi ampliada para incluir o desempenho \mathcal{H}_∞ . Em Coutinho et al. (2005) é apresentada uma formulação LMI para tratar do problema multiobjetivo de sistemas precisamente conhecidos considerando um método iterativo de projeto para determinação do valor ótimo de um parâmetro escalar que conduz a um resultado menos conservador.

A caracterização de controladores por realimentação estática ou dinâmica de saída resulta inicialmente em formulações BMI. Infelizmente, a solução de problemas BMI é difícil não sendo possível o desenvolvimento de um procedimento para obter a solução de tais problemas em tempo polinomial (Toker e Özbay, 1995). Vários trabalhos foram dedicados para obter formulações LMI, sendo alguns baseados em mudança de variáveis linearizantes (Gahinet et al., 1995; Scherer, 1995; Scherer et al., 1997; Masubuchi et al., 1998; Apkarian et al., 2001; Trofino, 2002; de Oliveira et al., 2002; Ebihara e Hagiwara, 2004a) e outros baseados em eliminação de variáveis com a transformação da BMI em duas LMI's acopladas por uma relação não convexa (Grigoriadis e Skelton, 1994; Grigoriadis e Skelton, 1996; Iwasaki, 1999; Shimomura e Fujii, 2000; de Oliveira et al., 2000). Nestas formulações, as matrizes do controlador são funções das matrizes do sistema, o que limita a utilização a sistemas precisamente conhecidos (ou sistema com incertezas limitadas em norma se o canal \mathcal{H}_∞ é empregado para acomodar as incertezas do modelo, usando o teorema do ganho pequeno). Um outro fator limitante é a ordem elevada dos controladores dinâmicos obtidos, principalmente quando são incluídas funções de ponderação para obtenção do desempenho desejado do sistema em malha-fechada, sendo também difícil a caracterização de controladores de ordem reduzida em termos de LMIs (Safonov et al., 1994). Até o presente momento, não existe uma caracterização por LMIs para tratar o caso de sistemas de controle por realimentação estática ou dinâmica da saída multiobjetivo com incertezas politópicas. Isto significa que não existe um algoritmo “globalmente convergente”, nem mesmo um conservador, para resolver esta classe de problemas. Em Kanev et al. (2004); Pereira e de Araújo (2004) e de Araújo e Neumann (2004) são apresentadas estratégias de projeto para controladores robustos por realimentação de saída. Em Kanev et al. (2004) é apresentado um algoritmo para solução de problemas de otimização BMI. Em Pereira e de Araújo (2004) e de Araújo e Neumann (2004) é apresentada uma estratégia de projeto baseada em algoritmo genético com a função objetivo sendo o custo garantido \mathcal{H}_2 calculado por um problema de otimização LMI satisfazendo uma restrição no

custo \mathcal{H}_∞ com as matrizes do controlador fixadas. Em Bernussou et al. (2005) é apresentada uma formulação LMI para projeto de controladores dinâmicos \mathcal{H}_2 para modelos politópicos.

Vários trabalhos são dedicados ao projeto de realimentação estática de saída (Geromel et al., 1996; de Oliveira e Geromel, 1997; El Ghaoui et al., 1997; Syrmos et al., 1997; Peaucelle et al., 2000; Peaucelle, Arzelier e Bertrand, 2002; Arzelier e Peaucelle, 2002; Arzelier et al., 2003; Shaked, 2003). A vantagem da realimentação estática é a sua simplicidade de implementação. O problema com o projeto de controladores por realimentação estática de saída é que as caracterizações são da forma de BMIs, não sendo possível aplicar diretamente os procedimentos de programação convexa padrões, mesmo quando todas as matrizes do sistema são precisamente conhecidas (Shaked, 2003).

Neste capítulo, o procedimento geral de projeto proposto no Capítulo 5 será empregado para o projeto de controladores robustos $\mathcal{H}_2/\mathcal{H}_\infty$, por realimentação de estado ou de saída, estático ou dinâmico, com posicionamento regional de pólos aplicado a sistemas com domínio politópico de incertezas com modelos politópicos ou com dependência afim de parâmetros. Apesar das formulações baseadas em LMIs serem problemas de otimização convexa, o que é uma característica bastante desejada uma vez que permite a utilização de algoritmos com garantia de convergência para o ótimo global, será verificado por intermédio de exemplos ilustrativos, que o procedimento geral de projeto, apesar de não ser um problema de otimização convexa, leva a resultados menos conservadores do que os obtidos pelas formulações LMIs existentes, mesmo aquelas baseadas em funções de Lyapunov dependentes de parâmetros. Resultados obtidos com uma versão preliminar do procedimento de projeto proposto são apresentados em Gonçalves, Palhares e Takahashi (2004a), Gonçalves, Palhares e Takahashi (2004b) e Gonçalves, Palhares e Takahashi (2004). Resultados com o procedimento de projeto geral são apresentados em Gonçalves, Palhares e Takahashi (2005a) e Gonçalves, Palhares e Takahashi (2005b).

6.2 Formulação do Problema

Considere um sistema linear invariante no tempo descrito por

$$\begin{aligned}
 \delta[x(\tau)] &= Ax(\tau) + B_u u(\tau) + B_w w(\tau) \\
 z_\infty(\tau) &= C_{z1}x(\tau) + D_{zu1}u(\tau) + D_{zw1}w(\tau) \\
 z_2(\tau) &= C_{z2}x(\tau) + D_{zu2}u(\tau) + D_{zw2}w(\tau) \\
 y(\tau) &= C_y x(\tau) + D_{yu}u(\tau) + D_{yw}w(\tau)
 \end{aligned} \tag{6.1}$$

sendo

$$\delta[x(\tau)] \triangleq \frac{dx(t)}{dt}, \quad \tau \triangleq t \in \mathbb{R} \quad \text{para sistemas contínuos no tempo}$$

$$\delta[x(\tau)] \triangleq x(k+1), \quad \tau \triangleq k \in \mathbb{N} \quad \text{para sistemas discretos no tempo}$$

Na eq. (6.1), $x \in \mathbb{R}^n$ é o vetor de variáveis de estado, $u \in \mathbb{R}^{n_u}$ é o vetor de entradas de controle, $w \in \mathbb{R}^{n_w}$ é o vetor de entradas exógenas (tais como sinais de distúrbios, ruído de medições ou sinais de referência), $z_\infty \in \mathbb{R}^{n_{z1}}$ é o vetor de variáveis controladas relacionadas com o desempenho \mathcal{H}_∞ , $z_2 \in \mathbb{R}^{n_{z2}}$ é o vetor de variáveis controladas relacionadas com o desempenho \mathcal{H}_2 e $y \in \mathbb{R}^{n_y}$ é o vetor de saídas medidas.

Defina a matriz do sistema:

$$S \triangleq \begin{bmatrix} A & B_u & B_w \\ C_{z1} & D_{zu1} & D_{zw1} \\ C_{z2} & D_{zu2} & D_{zw2} \\ C_y & D_{yu} & D_{yw} \end{bmatrix} \in \mathcal{P} \quad (6.2)$$

Considere que a matriz sistema S não é precisamente conhecida, mas pertence a um conjunto poliédrico convexo fechado, ou politopo: $S \in \mathcal{P}$.

No caso de modelos politópicos, o conjunto \mathcal{P} é um politopo no espaço de matrizes definido pelo conjunto de todas matrizes obtidas pela combinação convexa de seus N vértices:

$$\mathcal{P} \triangleq \left\{ S(\theta) : S(\theta) = \sum_{i=1}^N \theta_i S_i, \theta \in \Omega_M \right\} \quad (6.3)$$

sendo

$$S_i \triangleq \begin{bmatrix} A_i & B_{u,i} & B_{w,i} \\ C_{z1,i} & D_{zu1,i} & D_{zw1,i} \\ C_{z2,i} & D_{zu2,i} & D_{zw2,i} \\ C_{y,i} & D_{yu,i} & D_{yw,i} \end{bmatrix}, \quad i = 1, \dots, N \quad (6.4)$$

os vértices do politopo e

$$\Omega_M \triangleq \left\{ \theta \in \mathbb{R}^N : \theta_i \geq 0, i = 1, \dots, N, \sum_{i=1}^N \theta_i = 1 \right\} \quad (6.5)$$

No caso de sistemas dependentes de parâmetros, a matriz sistema é dependente afim do vetor de parâmetros incertos $p = [p_1, p_2, \dots, p_d]^T \in \mathbb{R}^d$:

$$\mathcal{P} \triangleq \{ S(p) : S(p) = S_0 + p_1 S_1 + \dots + p_d S_d, p \in \Omega_p \} \quad (6.6)$$

Se os parâmetros incertos variam entre valores limites, i.e. $p_i \in [\underline{p}_i, \bar{p}_i]$, sendo \underline{p}_i e \bar{p}_i os valores mínimo e máximo do i -ésimo parâmetro incerto, o vetor p pertence a um hiper-retângulo no espaço d -dimensional:

$$\Omega_p \triangleq \left\{ p \in \mathbb{R}^d : \underline{p}_i \leq p_i \leq \bar{p}_i, i = 1, \dots, d \right\} \quad (6.7)$$

O domínio Ω_p também pode ser um polítopo de formato qualquer se existirem restrições lineares adicionais sobre os parâmetros.

Sem perda de generalidade, será considerando $D_{yu} = 0$ de modo a simplificar as equações apresentadas a seguir. Considere o diagrama de blocos geral de um sistema de controle realimentado, apresentado na Figura 6.1, cujo controlador \mathcal{K} pode ser um controlador dinâmico representado por:

$$\begin{aligned} \delta[x_c(\tau)] &= A_c x_c(\tau) + B_c y(\tau) \\ u(t) &= C_c x_c(\tau) + D_c y(\tau) \end{aligned} \quad (6.8)$$

sendo $x_c \in \mathbb{R}^k$ o vetor de estado do controlador, ou um controlador estático com a seguinte ação de controle

$$u(\tau) = Ky(\tau) \quad (6.9)$$

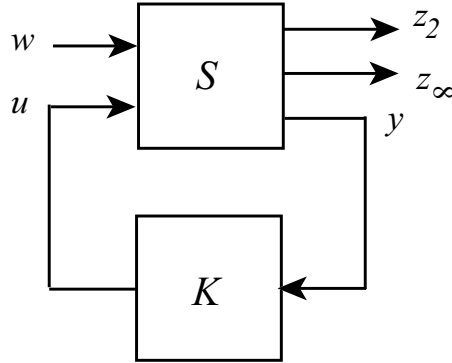


Figura 6.1: Diagrama de blocos geral do sistema de controle em malha-fechada.

Defina as equações dinâmicas do sistema em malha-fechada como

$$\begin{aligned} \delta[\bar{x}(\tau)] &= \bar{A}\bar{x}(\tau) + \bar{B}w(\tau) \\ z_\infty(\tau) &= \bar{C}_1\bar{x}(\tau) + \bar{D}_1w(\tau) \\ z_2(\tau) &= \bar{C}_2\bar{x}(\tau) + \bar{D}_2w(\tau) \end{aligned} \quad (6.10)$$

em que $\bar{x}(\tau) \triangleq [x^T(\tau) \ x_c^T(\tau)]^T$ no caso de realimentação dinâmica ou simplesmente $\bar{x}(\tau) \triangleq x(\tau)$ no caso de realimentação estática. Seja $z(t) \triangleq [z_\infty(\tau)^T \ z_2(\tau)^T]^T$, $C_z = [C_{z1}^T \ C_{z2}^T]^T$, $D_{zw} = [D_{zw1}^T \ D_{zw2}^T]^T$, $D_{zu} = [D_{zu1}^T \ D_{zu2}^T]^T$, $\bar{C} \triangleq [\bar{C}_1^T \ \bar{C}_2^T]^T$ e $\bar{D} \triangleq [\bar{D}_1^T \ \bar{D}_2^T]^T$. Considerando realimenta-

ção dinâmica de saída, as matrizes em malha-fechada são dadas por:

$$\left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array} \right] = \left[\begin{array}{cc|c} A + B_u D_c C_y & B_u C_c & B_w + B_u D_c D_{yw} \\ B_c C_y & A_c & B_c D_{yw} \\ \hline C_z + D_{zu} D_c C_y & D_{zu} C_c & D_{zw} + D_{zu} D_c D_{yw} \end{array} \right] \quad (6.11)$$

No caso do controle por realimentação estática de saída, $u(\tau) = \mathcal{K}y(\tau)$, ou realimentação de estado ($C_y = I_n$, tal que $y(\tau) = x(\tau)$), as matrizes em malha-fechada são dadas por:

$$\left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array} \right] = \left[\begin{array}{cc|c} A + B_u \mathcal{K} C_y & B_w + B_u \mathcal{K} D_{yw} \\ \hline C_z + D_{zu} \mathcal{K} C_y & D_{zw} + D_{zu} \mathcal{K} D_{yw} \end{array} \right] \quad (6.12)$$

As matrizes do sistema em malha-fechada também podem ser determinadas, de uma forma geral, a partir das matrizes ampliadas do sistema:

$$\left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array} \right] = \left[\begin{array}{cc} \hat{A} & \hat{B}_w \\ \hat{C}_z & D_{zw} \end{array} \right] + \left[\begin{array}{c} \hat{B}_u \\ \hat{D}_{zu} \end{array} \right] \mathcal{K} \left[\begin{array}{cc} \hat{C}_y & \hat{D}_{yw} \end{array} \right] \quad (6.13)$$

sendo as matrizes aumentadas definidas como

$$\begin{aligned} \hat{A} &\triangleq \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix}, & \hat{B}_u &\triangleq \begin{bmatrix} 0 & B_u \\ I_k & 0 \end{bmatrix}, & \hat{B}_w &\triangleq \begin{bmatrix} B_w \\ 0 \end{bmatrix}, \\ \hat{C}_z &\triangleq \begin{bmatrix} C_z & 0 \end{bmatrix}, & \hat{D}_{zu} &\triangleq \begin{bmatrix} 0 & D_{zu} \end{bmatrix}, \\ \hat{C}_y &\triangleq \begin{bmatrix} 0 & I_k \\ C_y & 0 \end{bmatrix}, & \hat{D}_{yw} &\triangleq \begin{bmatrix} 0 \\ D_{yw} \end{bmatrix} \end{aligned} \quad (6.14)$$

e

$$\mathcal{K} \triangleq \left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] \quad (6.15)$$

sendo que, no caso particular da realimentação estática de saída (controlador de ordem $k = 0$), as matrizes do controlador A_c , B_c e C_c são vazias (dimensão zero) e $D_c = K$.

Para um sistema com $D_{yu} \neq 0$, pode-se projetar o controlador dinâmico \mathcal{K} considerando como saída $\hat{y} \triangleq C_y x + D_{yw} w$ e depois substituir \hat{y} por $y - D_{yu} u$ para obter o controlador $\bar{\mathcal{K}}$ para o sistema

original desde que exista $(I + D_c D_{yu})^{-1}$ (Iwasaki e Skelton, 1994):

$$\bar{\mathcal{K}} = \begin{bmatrix} A_c - B_c D_{yw} (I + D_c D_{yu})^{-1} C_c & B_c (I - D_{yw} (I + D_c D_{yu})^{-1} D_c) \\ (I + D_c D_{yu})^{-1} C_c & (I + D_c D_{yu})^{-1} D_c \end{bmatrix}$$

Seja $T_\infty(\alpha, \mathcal{K})$ a matriz de transferência em malha-fechada de w para z_∞ e $T_2(\alpha, \mathcal{K})$ a matriz de transferência em malha-fechada de w para z_2 , ambas para um determinado sistema do politopo, definido pelo vetor $\alpha \in \Omega$, com α representando θ ou p e Ω representando Ω_M ou Ω_p , e um determinado controlador \mathcal{K} . O objetivo de controle é calcular o controlador \mathcal{K} que minimiza a norma \mathcal{H}_∞ de $T_\infty(\alpha, \mathcal{K})$, $\|T_\infty(\alpha, \mathcal{K})\|_\infty$, e a norma \mathcal{H}_2 de $T_2(\alpha, \mathcal{K})$, $\|T_2(\alpha, \mathcal{K})\|_2$, e aloque os pólos do sistema em malha-fechada (ou os autovalores da matriz $\bar{A}(\alpha, \mathcal{K})$) em regiões \mathcal{D} do plano complexo, para todo $\alpha \in \Omega$. No caso de controle por realimentação dinâmica de saída de sistemas contínuos no tempo, D_c deve ser tal que $\bar{D}_2 = D_{zw2} + D_{zu2} D_c D_{yw} = 0$, de modo a garantir norma $\|T_2(\alpha, \mathcal{K})\|_2$ finita.

6.3 Exemplos Ilustrativos

Os seguintes exemplos de projeto de controladores são baseados no procedimento de projeto proposto, apresentado no Capítulo 5, implementado no LMI Control Toolbox para MATLAB[®], sendo utilizado um computador baseado no Pentium IV 2.8GHz, com 1 Gb de memória RAM. A menos que seja informado diferente, os resultados são obtidos adotando o critério de parada do procedimento de projeto $\varepsilon_\delta = 0,1$; os critérios de parada do algoritmo de otimização cone-elipsoidal $\epsilon = \varphi$ e $N_\epsilon = 10$ e a especificação de precisão no algoritmo branch-and-bound $\varepsilon = \varphi$, sendo $\varphi = 0,001$ para realimentação estática e $\varphi = 0,01$ para realimentação dinâmica. Nos exemplos a seguir, se nada for mencionado, considere que o controlador é obtido em apenas uma iteração do procedimento de projeto (piores casos ocorrem nos próprios vértices do domínio de incerteza politópico), o que acontece na maioria dos casos analisados.

6.3.1 Exemplos Ilustrativos de Controle por Realimentação de Estado

Exemplo 6.1 Considere o modelo no espaço de estado de um satélite consistindo de dois corpos rígidos (módulo principal e módulo de sensores) conectados por eixo elástico (Gahinet et al., 1995):

$$\begin{aligned} \frac{dx(t)}{dt} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k}{J_1} & \frac{k}{J_1} & -\frac{f}{J_1} & \frac{f}{J_1} \\ \frac{k}{J_2} & -\frac{k}{J_2} & \frac{f}{J_2} & -\frac{f}{J_2} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_1} \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_1} \\ 0 \end{bmatrix} w(t) \\ z_\infty(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x(t) \\ z_2(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) \end{aligned} \quad (6.16)$$

com $x = [\theta_1 \ \theta_2 \ \dot{\theta}_1 \ \dot{\theta}_2]^T$, sendo θ_1 e θ_2 os ângulos de rotação do corpo principal e do módulo de sensores, respectivamente, u é o conjugado de controle e w é um conjugado de distúrbio sobre o corpo principal. São considerados $J_1 = 1$ e $J_2 = 1$. Os parâmetros incertos k e f podem variar nas faixas $0,09 \leq k \leq 0,4$ e $0,0038 \leq f \leq 0,04$ resultando em um domínio de incerteza na forma de retângulo cujos vértices são as combinações dos valores extremos de k e f : $\Omega = \{\alpha = [k \ f]^T \in \mathbb{R}^2 : 0,09 \leq k \leq 0,4, 0,0038 \leq f \leq 0,04\}$.

O objetivo de controle é projetar um controlador robusto por realimentação de estado que obtenha um compromisso entre as normas $\|T_2(\alpha, \mathcal{K})\|_2$ e $\|T_\infty(\alpha, \mathcal{K})\|_\infty$, com $\|T_\infty(\alpha, \mathcal{K})\|_\infty \leq 1$, e posicione os pólos em malha-fechada na intersecção das regiões semi-plano $\text{Real}(s) < 0,1$ e o setor cônico centrado na origem com ângulo interno $2\theta = \frac{3}{4}\pi$ para todo $\alpha \in \Omega$: $\lambda_i(\bar{A}(\alpha, \mathcal{K})) \in \mathcal{D}$, $\forall i, \forall \alpha \in \Omega$, $\mathcal{D} = \{s \in \mathbb{C} : \text{Real}(s) < -0,1, |\angle(s)| < \pi - \frac{3}{8}\pi\}$.

No procedimento proposto, os parâmetros de otimização são relacionados com a controlador da seguinte forma:

$$\mathcal{K} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}$$

As soluções obtidas pelo procedimento proposto com $\gamma = 0,1; 0,2; \dots; 1$ ($\delta = 0$, $\lambda_2 = 1$ e $\lambda_\infty = 0$) são comparadas com as soluções calculadas com a função `msf_syn` disponível no *LMI Control Toolbox* do MATLAB[®] (Gahinet et al., 1995). No algoritmo de otimização foram adotados os valores iniciais $x_0 = \begin{bmatrix} -0,4326 & -1,6656 & 0,1253 & 0,2877 \end{bmatrix}^T$ (gerado de forma aleatória) e $Q_0 = 10^3 I_4$. Os resultados de projeto do procedimento proposto são apresentados na Tabela 6.1 sendo que os custos ε -garantidos ou simplesmente custos \mathcal{H}_2 e \mathcal{H}_∞ são calculados pelo algoritmo branch-and-

bound (BnB) com precisão relativa $\varepsilon = 0,001$. É possível reduzir os tempos computacionais após o cálculo do primeiro controlador, obtendo resultados similares, reduzindo o raio do elipsóide inicial e utilizando o controlador obtido para o valor de γ anterior, como condição inicial para o seguinte, ao invés da condição inicial aleatória.

Tabela 6.1: Resultados de projeto do procedimento proposto (Ex. 6.1).

γ	Custo \mathcal{H}_∞	Custo \mathcal{H}_2	Tempo Proc. (s)	Ganhos \mathcal{K}
0,1	0,0996	1,9180	26,5	[-8,8399 -5,3100 -4,3728 -29,2253]
0,2	0,1993	1,7261	32,9	[-5,9889 -1,0892 -3,6235 -16,1648]
0,3	0,2998	1,6214	33,6	[-4,7609 0,0518 -3,1938 -11,2636]
0,4	0,3977	1,5504	38,1	[-4,0231 0,4722 -2,9210 -8,6864]
0,5	0,4974	1,5002	38,3	[-3,4946 0,6592 -2,7811 -7,0037]
0,6	0,5998	1,4614	36,6	[-3,1119 0,7900 -2,6497 -5,8526]
0,7	0,6996	1,4300	35,8	[-2,8221 0,8154 -2,5216 -4,9975]
0,8	0,7957	1,4055	36,6	[-2,6076 0,8352 -2,3822 -4,3916]
0,9	0,8956	1,3872	34,2	[-2,3894 0,8277 -2,3515 -3,8527]
1,0	0,9953	1,3749	31,2	[-2,2430 0,8825 -2,2926 -3,4853]

A Figura 6.2 apresenta a curva tipo Pareto obtida pelos dois procedimentos, sendo que os valores de custo obtidos pela função msf_{syn} também são calculados pelo algoritmo BnB. É verificado que o procedimento proposto apresenta duas vantagens em relação à formulação LMI baseada em estabilidade quadrática. A primeira vantagem é que a curva obtida pelo procedimento proposto cobre a faixa de valores especificada para o custo \mathcal{H}_∞ enquanto que a curva obtida pela função msf_{syn} cobre apenas a faixa de 0,037 a 0,323. A segunda vantagem do procedimento proposto é que o mesmo resulta em menores valores do custo \mathcal{H}_2 para mesmos valores de custo \mathcal{H}_∞ do que a formulação LMI.

Considere o melhor controlador obtido pela função msf_{syn} como sendo aquele para o qual a redução relativa no custo \mathcal{H}_2 é menor que o aumento relativo no custo \mathcal{H}_∞ , o que é verificado para $\gamma = 0,2$, com o correspondente controlador:

$$\mathcal{K}_m = \begin{bmatrix} -16,6769 & 2,9441 & -6,7480 & -59,6433 \end{bmatrix}$$

que resulta nos custos $\gamma_c = 0,074$ e $\delta_c = 2,245$. Para o mesmo γ , o controlador \mathcal{K}_p calculado com o procedimento de projeto proposto resulta nos custos $\gamma_c = 0,199$ e $\delta_c = 1,726$. O custo \mathcal{H}_2 é 23% menor do que o obtido pela formulação baseada em LMI. O tempo de processamento requerido neste

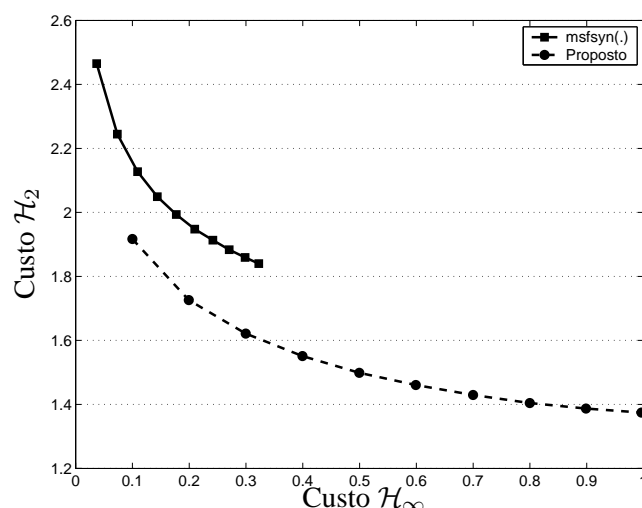


Figura 6.2: Curva tipo Pareto obtida com a função `msfsyn` e o procedimento proposto.

caso é de 22,92s no passo 3 (otimização), 4,94s no passo 4 (cálculo do custo \mathcal{H}_2), 4,24s no passo 5 (cálculo do custo \mathcal{H}_∞) e 0,74s no passo 6 (verificação do posicionamento regional de pólos).

Como exemplo da capacidade do procedimento proposto para obter soluções factíveis, considere que, além da especificação semi-plano e setor cônico, é necessário que os pólos se localizem dentro do disco de raio r com centro na origem: $\mathcal{D} = \{s \in \mathbb{C} : |s| < r, \text{Real}(s) < -0,1, |\angle(s)| < \pi - \frac{3}{8}\pi\}$. Neste caso será analisada a capacidade de cada método de obter solução factível com a redução do raio do disco com as outras restrições fixas. Para $\gamma = 0,2$, a formulação LMI obteve solução para $r \geq 3,70$ enquanto que o procedimento proposto obtém solução para $r \geq 2,41$, que é 35% menor que o primeiro. Neste caso, o controlador obtido com o procedimento proposto e $r = 2,41$ é obtido como

$$\mathcal{K}_{pd} = \begin{bmatrix} -5,9686 & -1,0743 & -3,7566 & -16,1295 \end{bmatrix}$$

que leva a custos similares em relação ao projeto sem a restrição de disco. A Figura 6.3 apresenta a distribuição dos pólos em malha-fechada para o controlador \mathcal{K}_{pd} com k e f variando entre os seus respectivos valores limites.

Para ilustrar o comportamento do algoritmo de otimização cone-elipsoidal, são apresentados nas Figuras 6.4 e 6.5 a evolução da função objetivo e dos parâmetros de otimização com o número de iterações no cálculo do controlador \mathcal{K}_p . O algoritmo busca inicialmente a região de factibilidade para depois iniciar a otimização da função objetivo. As oscilações apresentadas são característica deste algoritmo de otimização cujo vetor de parâmetros de otimização corresponde ao centro do elipsóide que contém a solução ótima local.

A aplicação do procedimento geral de projeto para síntese de controladores por realimentação

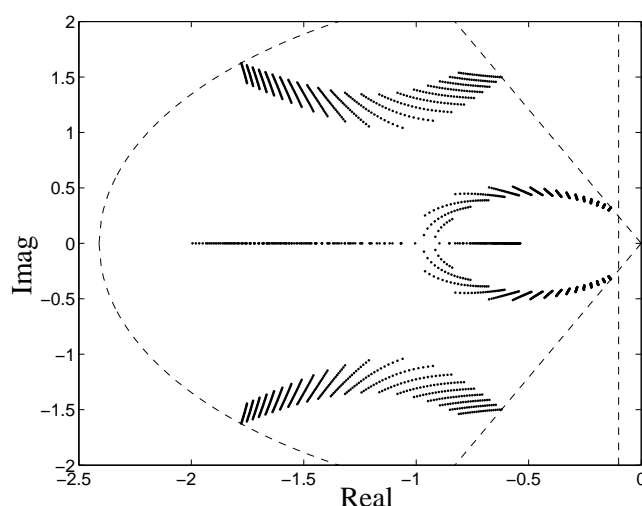


Figura 6.3: Posicionamento dos pólos em malha-fechada com \mathcal{K}_{pd} para k e f variando entre os valores limites.

dinâmica de saída, de ordem completa e reduzida, considerando este mesmo exemplo, é apresentada em Gonçalves, Palhares e Takahashi (2005b).

Exemplo 6.2 Considere o exemplo do sistema de controle a tempo discreto, analisado em de Oliveira et al. (2002) e em Ebihara et al. (2004), obtido do sistema contínuo no tempo apresentado em Veillette et al. (1992). O modelo no espaço de estado do sistema a tempo contínuo é dado por (Veillette et al., 1992):

$$\frac{dx(t)}{dt} = \begin{bmatrix} -2 & 1 & 1 & 1 \\ 3 & 0 & 0 & 2 \\ -1 & 0 & -2 & -3 \\ -2 & -1 & 2 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ b_1 & 0 \\ 0 & 0 \\ 0 & b_2 \end{bmatrix} u(t) + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} w(t)$$

$$z_2(t) = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u(t)$$

sendo que as matrizes do sistema a tempo discreto são obtidas considerando um amostrador de ordem zero com período de amostragem igual a 0,1s (de Oliveira et al., 2002).

Em Gonçalves, Palhares e Takahashi (2005a) é tratado o problema de projeto de um controlador por realimentação de estado descentralizado cujo objetivo é minimizar a norma \mathcal{H}_2 da função de transferência em malha-fechada discreta T_2 de w para z_2 considerando que o sistema é precisamente conhecido. Neste exemplo será tratado o problema de confiabilidade com objetivo de projetar

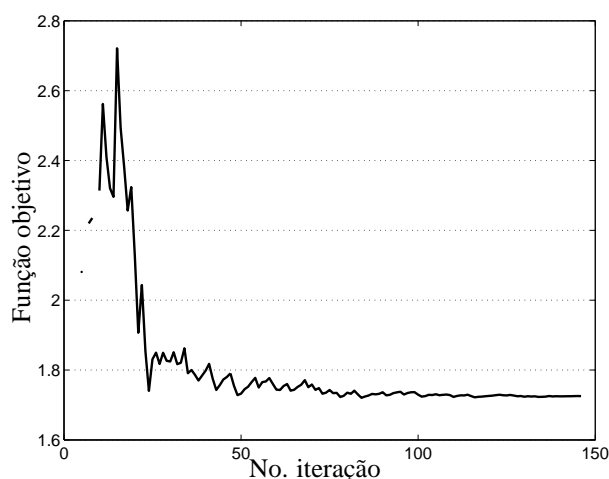


Figura 6.4: Evolução da função objetivo no processo de otimização para o cálculo do controlador \mathcal{K}_p .

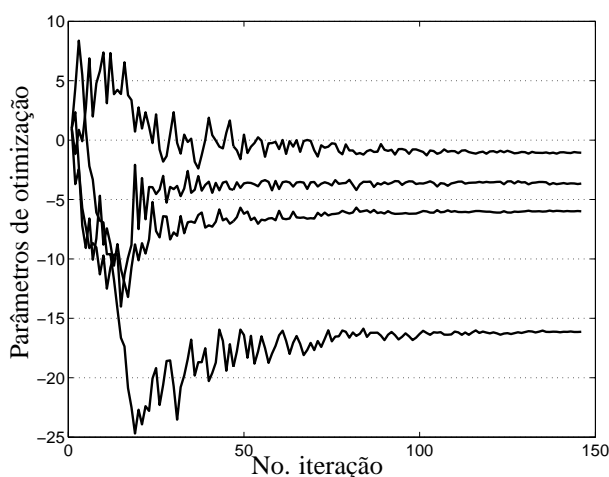


Figura 6.5: Evolução dos parâmetros de otimização no cálculo do controlador \mathcal{K}_p .

um controlador por realimentação de estado ($y(t) = x(t)$) centralizado para lidar com três cenários distintos (de Oliveira et al., 2002): a planta nominal ($b_1 = 1$ e $b_2 = 1$), falha do primeiro atuador ($b_1 = 0$ e $b_2 = 1$) e falha do segundo atuador ($b_1 = 1$ e $b_2 = 0$). O domínio de incerteza é considerado como sendo um triângulo com os vértices correspondendo aos três cenários listados: $\Omega = \{\alpha = [b_1 \ b_2]^T : 0 \leq b_1 \leq 1, 0 \leq b_2 \leq 1, b_1 + b_2 \geq 1\}$. Os custos \mathcal{H}_2 apresentados a seguir são calculados pelo método de análise proposto com precisão $\epsilon = 0,01$, considerando as quatro primeiras casas decimais dos ganhos dos controladores ao invés do valor exato.

Utilizando uma formulação LMI baseada em estabilidade quadrática, é obtido o controlador apre-

sentado em de Oliveira et al. (2002) e o custo \mathcal{H}_2 correspondente:

$$\mathcal{K} = \begin{bmatrix} -0,5473 & -0,7060 & -0,5289 & -0,6190 \\ -0,6590 & -0,9840 & -0,6073 & -0,3613 \end{bmatrix} \Rightarrow \text{custo } \mathcal{H}_2 : \delta_c^2 = 0,566$$

O controlador obtido com a caracterização estendida é dado por (de Oliveira et al., 2002):

$$\mathcal{K} = \begin{bmatrix} -0,6521 & -0,6783 & -0,3365 & -0,5451 \\ -0,6294 & -0,9690 & -0,6510 & -0,4170 \end{bmatrix} \Rightarrow \text{custo } \mathcal{H}_2 : \delta_c^2 = 0,536$$

Considere no procedimento de projeto proposto a seguinte estrutura do controlador:

$$\mathcal{K} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \end{bmatrix}$$

São adotados condições iniciais $x = \mathbf{0}$ e $Q(0) = I_8$, e critérios de parada $\epsilon = 0,01$ e $N_\epsilon = 10$. Se forem considerados apenas os três cenários, após 3,48s de processamento (apenas a etapa de síntese), é obtido o seguinte controlador:

$$\mathcal{K} = \begin{bmatrix} -0,1030 & -0,8536 & 0,1873 & -0,3135 \\ -0,8584 & -0,0499 & -0,5785 & 0,1355 \end{bmatrix}$$

com o pior caso da norma correspondendo à falha no primeiro atuador:

$$\|T_2(\alpha, \mathcal{K})\|_2^2 = 0,434, \quad \alpha = [0 \ 1]^T$$

Considerando todo o politopo, como o pior caso da norma não ocorre nos vértices do politopo, são necessárias iterações do procedimento de projeto proposto com a inclusão de pontos adicionais no conjunto $\tilde{\Omega}$, definido na Seção 5.3, inicializado com os vértices do politopo. Com $\epsilon_\delta = 0,1$, são necessárias três iterações, com a inclusão de dois novos pontos em $\tilde{\Omega}$ (ver Figura 6.6) obtidos pelo algoritmo BnB, resultando na seguinte seqüência de custos \mathcal{H}_2 para cada iteração: $\delta_c^2 = \{0,618; 0,500; 0,488\}$. Após três iterações e 170,41s de processamento é obtido o seguinte controlador e o custo \mathcal{H}_2 correspondente:

$$\mathcal{K} = \begin{bmatrix} -0,3071 & -0,4967 & -0,0695 & -0,1033 \\ -0,8649 & -0,2167 & -0,5481 & 0,0593 \end{bmatrix}, \Rightarrow \text{custo } \mathcal{H}_2 : \delta_c^2 = 0,488$$

que é 9% menor do que o melhor resultado obtido com as formulações LMI. A Figura 6.7 apresenta o perfil da norma \mathcal{H}_2 sobre a aresta $b_1 + b_2 = 1$ (onde ocorrem os piores casos de norma \mathcal{H}_2) para

as três iterações, através da qual pode ser observado como a inclusão de novos pontos no conjunto $\tilde{\Omega}$ atua sobre a minimização do custo \mathcal{H}_2 . Na terceira iteração o conjunto de pontos finitos utilizados é $\tilde{\Omega} = \{[1 \ 1]^T; [0 \ 1]^T; [1 \ 0]^T; [0,4688 \ 0,5313]^T; [0,25 \ 0,75]^T\}$.

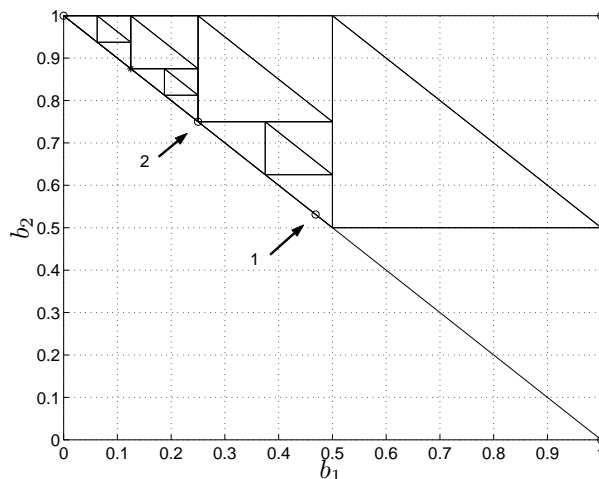


Figura 6.6: Subdivisão do espaço de incerteza após a terceira iteração (Ex. 6.2).

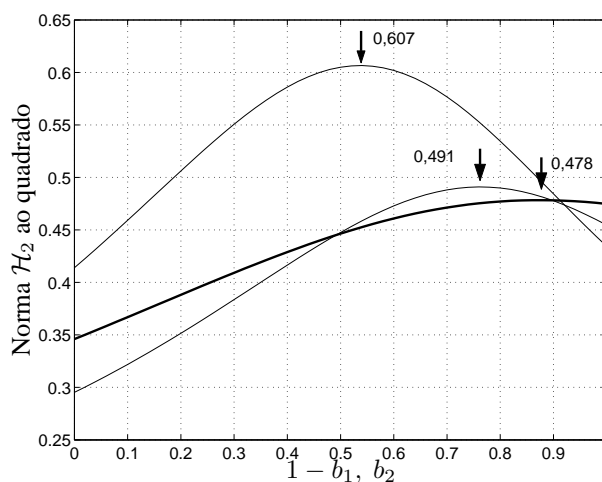


Figura 6.7: Evolução da norma \mathcal{H}_2 sobre a aresta $b_1 + b_2 = 1$ para cada iteração (Ex. 6.2).

6.3.2 Exemplos Ilustrativos de Controle por Realimentação de Saída

Exemplo 6.3 Considere o modelo linearizado de um helicóptero VTOL, obtido para condições de voo e carga típicos na velocidade de 135 nós, apresentado em Keel et al. (1988) e também considerado em Geromel et al. (1996), El Ghaoui et al. (1997), Pereira e de Araújo (2004) e de Araújo e Neumann

(2004), com as seguintes equações dinâmicas:

$$\dot{x}(t) = \begin{bmatrix} -0,0366 & 0,0271 & 0,0188 & -0,4555 \\ 0,0482 & -1,0100 & 0,0024 & -4,0208 \\ 0,1002 & a_{32} & -0,7070 & a_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} w(t) + \begin{bmatrix} -0,4422 & -0,1761 \\ b_{21} & 7,5922 \\ 5,5200 & -4,4900 \\ 0 & 0 \end{bmatrix} u(t)$$

$$z_{\infty}(t) = I_4 x(t) + 0_4 w(t) + 0_{4 \times 2} u(t)$$

$$z_2(t) = \begin{bmatrix} I_4 \\ 0_{2 \times 4} \end{bmatrix} x(t) + \begin{bmatrix} 0_4 \\ 0_{2 \times 4} \end{bmatrix} w(t) + \begin{bmatrix} 0_{4 \times 2} \\ I_2 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} w(t) + \begin{bmatrix} 0 & 0 \end{bmatrix} u(t)$$

O sistema possui três parâmetros incertos que variam nas faixas $a_{32} \in [-0,6319; 1,3681]$, $a_{34} \in [1,22; 1,62]$ e $b_{21} \in [2,7446; 4,3446]$ (Geromel et al., 1996).

O objetivo de controle é obter um compromisso entre os desempenhos \mathcal{H}_2 e \mathcal{H}_{∞} através de controladores por realimentação da saída, dinâmicos ou estáticos.

Como relatado em Pereira e de Araújo (2004), utilizando uma estratégia que utiliza um algoritmo híbrido que combina algoritmos genéticos de otimização com formulações baseadas em LMIs, é obtido um controlador dinâmico de 2ª ordem que resulta no custo garantido \mathcal{H}_2 igual a 4,3576 e no custo garantido \mathcal{H}_{∞} igual a 18,9714. Analisando o controlador relatado em Pereira e de Araújo (2004) com o cálculo dos custos pelo algoritmo BnB, com precisão $\varepsilon = 0,01$, resultou nos custos $\delta_c = 4,81$ e $\gamma_c = 18,97$. Em de Araújo e Neumann (2004), utilizando a mesma técnica, mas alterando os parâmetros do algoritmo genético, é obtido um controlador dinâmico de 2ª ordem que resulta no custo garantido \mathcal{H}_2 igual a 3,6992 e no custo garantido \mathcal{H}_{∞} igual a 13,6338. Analisando o controlador relatado em de Araújo e Neumann (2004) com o cálculo dos custos pelo algoritmo BnB, com precisão $\varepsilon = 0,01$, resultou nos custos $\delta_c = 3,56$ e $\gamma_c = 13,63$. Para comparar os resultados obtidos com o procedimento proposto com os resultados apresentados em Pereira e de Araújo (2004) e em de Araújo e Neumann (2004), foram projetados vários controladores estáticos e dinâmicos de 2ª ordem, cujos resultados são apresentados na Tabela 6.2 com a correspondente curva tipo Pareto

apresentada na Figura 6.8. Para a síntese dos controladores dinâmicos de 2ª ordem são adotadas as condições iniciais equivalentes à $A_c = -I$, B_c e C_c com todos elementos iguais a 1 e $D_c = 0$ e a elipse inicial definida por $Q_o = 100I_{12}$. Para a síntese dos controladores estáticos são adotadas as condições iniciais equivalente à $K = [1 \ 1]^T$ e a elipse inicial definida por $Q_o = 100I_2$. Pode ser observado que qualquer uma das soluções obtidas pelo procedimento de projeto proposto dominam a solução apresentada em Pereira e de Araújo (2004), o que significa que os dois objetivos são minimizados simultaneamente, independentemente de se considerar o valor de custo garantido relatado ou o valor de custo calculado. A solução obtida em de Araújo e Neumann (2004) é dominada pelas soluções obtidas com o procedimento proposto para $12,6 \leq \gamma \leq 13,8$. As marcas '*' na Figura 6.8 representam os resultados obtidos para síntese de controladores dinâmicos de 2ª ordem considerando a minimização do custo \mathcal{H}_2 ou do custo \mathcal{H}_∞ sem restrições.

Tabela 6.2: Resultados de projeto para os controladores por realimentação da saída (Ex. 6.3).

γ	Controladores estáticos				Controladores de 2ª ordem		
	γ_c	δ_c	Tempo	\mathcal{K}^T	γ_c	δ_c	Tempo
11,4	11,35	5,17	1min34s	[-1,2931 -14,9188]	11,35	4,08	8min10s
11,8	11,79	4,46	2min36s	[-1,1053 -22,9237]	11,78	3,73	9min4s
12,2	12,20	4,17	3min40s	[-0,3337 -19,4220]	12,19	3,58	14min11s
12,6	12,60	4,00	3min36s	[0,1816 -16,2335]	12,56	3,51	12min14s
13,0	12,99	3,91	4min26s	[0,5227 -14,0367]	12,98	3,48	12min52s
13,4	13,38	3,84	5min43s	[0,7441 -11,9000]	13,26	3,48	18min12s
13,8	13,74	3,79	6min58s	[0,8438 -9,8096]	13,58	3,46	18min34s

As matrizes do controlador de 2ª ordem são relacionadas com os parâmetros do vetor de otimização x como:

$$A_c = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}, B_c = \begin{bmatrix} x_5 \\ x_6 \end{bmatrix}, C_c = \begin{bmatrix} x_7 & x_8 \\ x_9 & x_{10} \end{bmatrix}, D_c = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$$

As matrizes do controlador que podem ser consideradas como as que proporcionam melhor compromisso entre os desempenhos \mathcal{H}_2 e \mathcal{H}_∞ são aquelas obtidas com $\gamma = 12,6$:

$$\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{cc|c} -1,3646 & 1,7720 & -1,7714 \\ 1,6965 & -1,9245 & -1,3347 \\ \hline -0,2454 & 0,0703 & 1,8588 \\ 3,1202 & 2,8109 & -2,8281 \end{array} \right]$$

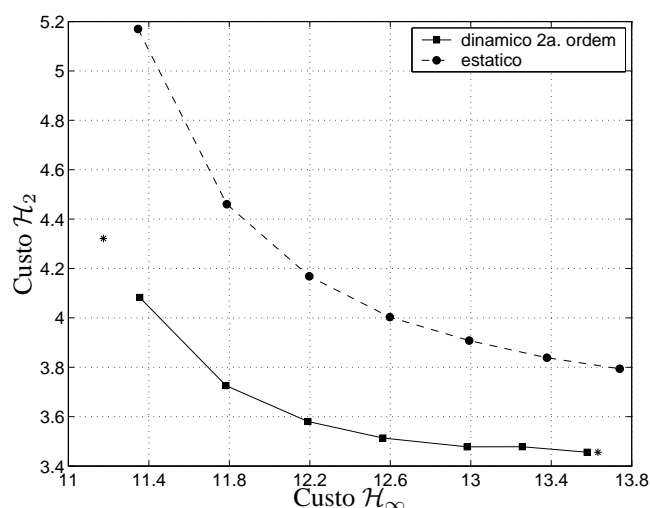


Figura 6.8: Curvas tipo Pareto para os controladores por realimentação de saída (Ex. 6.3).

Para $\gamma = 12,6$, adotando $Q_0 = 10^3 I_{30}$ e x_0 obtido de forma aleatória (distribuição normal com média zero e variância unitária), o procedimento proposto fornece o seguinte controlador dinâmico de ordem completa, $k = 4$:

$$\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{cccc|c} -0,6976 & 0,4934 & -0,6406 & -0,0136 & -1,5299 \\ 0,2446 & -0,4692 & -1,0192 & -0,8485 & -1,9637 \\ 0,5204 & 0,3141 & -1,0033 & 0,6900 & -0,2011 \\ -1,1613 & 0,7820 & -1,0482 & -0,0377 & 1,2562 \\ \hline 0,5954 & -0,1363 & -1,0434 & -0,1042 & -1,7465 \\ -2,4890 & -2,2604 & -1,5834 & 0,3871 & 2,5546 \end{array} \right]$$

Para o projeto deste controlador, são necessários 11min53s de processamento pelo algoritmo de otimização, 24min5s para o cálculo do custo ε -garantido \mathcal{H}_2 , após 11 iterações do algoritmo BnB, e 42s para o cálculo do custo ε -garantido \mathcal{H}_∞ , resultando nos custos $\delta_c = 3,51$ e $\gamma_c = 12,59$.

O tempo de processamento na etapa de síntese pode ser reduzido aproveitando a flexibilidade do procedimento de projeto proposto que permite a síntese de controladores com qualquer estrutura fixa desejada, propriedade que dificilmente pode ser obtida por caracterizações baseadas em LMIs. Impondo a relação entre as variáveis escalares de otimização e as matrizes do controlador de ordem

completa ($k = 4$) de modo a obter controladores descentralizados com a seguinte estrutura

$$\left[\begin{array}{cc|cc} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ x_1 & x_2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_3 & x_4 & 1 \\ \hline x_5 & x_6 & 0 & 0 & x_9 \\ 0 & 0 & x_7 & x_8 & x_{10} \end{array} \right]$$

o número de variáveis de otimização é reduzido de 30 para 10.

Para $\gamma = 12,6$, $Q_0 = 100I_{10}$ e x_0 obtido de forma aleatória, o procedimento proposto fornece o seguinte controlador dinâmico com a estrutura determinada:

$$\left[\begin{array}{cc|cc} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ -2,2406 & -2,9203 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -0,0738 & -1,4505 & 1 \\ \hline 2,1237 & 1,5338 & 0 & 0 & -1,8366 \\ 0 & 0 & 4,6930 & 5,5229 & 2,9472 \end{array} \right]$$

após apenas 1min40s de processamento do algoritmo de otimização cone-elipsoidal, resultando nos custos $\delta_c = 3,52$ e $\gamma_c = 12,57$.

Exemplo 6.4 Considere o sistema a tempo contínuo apresentado em Ebihara e Hagiwara (2004a):

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -0,32 & 0,04 & 0,01 \\ 0,45 & 0,99 & 0,64 \\ -0,76 & -0,37 & -0,40 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} w(t) + \begin{bmatrix} 0,42 \\ 0,74 \\ 0,31 \end{bmatrix} u(t) \\ z_2(t) &= \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0,72 & 0,85 & 0,88 \end{bmatrix} x(t) + 2w(t) \end{aligned} \tag{6.17}$$

cujo objetivo é projetar um controlador dinâmico por realimentação da saída que minimize a norma $\|T_2\|_2$ e que posicione os pólos em malha-fechada no semi-plano $\text{Real}(\lambda_i(\bar{A})) < -0,3$, $\forall i$. Neste caso, para obter norma \mathcal{H}_2 finita é necessário que o controlador seja estritamente próprio. Por se tratar

de um sistema precisamente conhecido, apenas a etapa de síntese do procedimento de projeto proposto será necessária. Este exemplo é interessante para comparar a eficácia do algoritmo de otimização cone-elipsoidal para resolver o problema não convexo em comparação com formulações LMIs, mais simples, porém mais conservadoras.

Partindo de condições iniciais aleatórias (distribuição normal com média zero e variância unitária), com $\epsilon = 0,01$ é obtido $\|T_2\|_2 = 61,46$ após 223 iterações do algoritmo cone-elipsoidal e tempo de processamento de 8,64s. Pode ser obtido um melhor resultado, com $\|T_2\|_2 = 50,46$, adotando $\epsilon = 0,001$, sendo necessárias 2075 iterações e tempo de processamento de 73,50s. Neste exemplo é observado que uma mudança no passo utilizado para o cálculo do gradiente (ver algoritmo na página 101) pode resultar em um melhor comportamento do algoritmo para $\epsilon = 0,001$. Mudando de $\Delta = 10^{-5}$ para $\Delta = 10^{-10}$, após 1733 iterações, 64,70s de tempo de processamento, é obtido o seguinte controlador dinâmico com $\|T_2\|_2 = 50,22$:

$$\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{ccc|c} -1,3778 & -1,0489 & 1,9774 & -2,0671 \\ -0,5750 & -0,1463 & 1,9559 & -0,1829 \\ 1,6772 & 1,1927 & -2,7185 & 2,8847 \\ \hline 0,5435 & -0,3552 & -2,0032 & 0 \end{array} \right]$$

com a correspondente função de transferência:

$$\mathcal{K}(s) = \frac{-6,8371(s + 0,3717)(s + 0,1547)}{(s + 4,653)(s - 0,407)(s - 0,003175)}$$

Através da Tabela 6.3 pode-se comparar o desempenho deste controlador com os controladores projetados com a formulação baseada em estabilidade quadrática (Scherer et al., 1997):

$$\mathcal{K}_S(s) = \frac{-15,93(s + 0,53)(s + 0,41)}{(s + 6,21)(s - 1,56)(s + 0,55)}$$

e com a formulação baseada em função de Lyapunov dependente de parâmetros com um parâmetro de sintonia escalar (requer busca unidimensional do melhor valor) apresentada em Ebihara e Hagiwara (2004a):

$$\mathcal{K}_E(s) = \frac{-10,60(s^2 + 0,84s + 0,18)}{(s + 4,99)(s - 1,19)(s + 0,44)}$$

O resultado obtido com o procedimento de projeto proposto é cerca de 26,7% menor do que a formulação apresentada por Ebihara e Hagiwara (2004a).

Tabela 6.3: Resultados de projeto (Ex. 6.4).

Método	(Scherer et al., 1997)		(Ebihara e Hagiwara, 2004a)		Proposto
	Síntese	Real	Síntese	Real	Real
Norma H_2 , $\ T_2\ _2$	115,62	79,09	73,60	68,49	50,22

Exemplo 6.5 Considere o sistema analisado em Arzelier e Peaucelle (2002):

$$\frac{dx(t)}{dt} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w_2(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w_\infty(t)$$

$$z_2(t) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$z_\infty(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t)$$

Este exemplo é interessante pelo fato de haver uma solução analítica para o problema de controle misto $\mathcal{H}_2/\mathcal{H}_\infty$, por realimentação estática de saída, que é apresentada em Arzelier e Peaucelle (2002). Seja $T_2(s)$ a função de transferência em malha-fechada de w_2 para z_2 e $T_\infty(s)$ a função de transferência em malha-fechada de w_∞ para z_∞ . A solução ótima para o problema \mathcal{H}_2 por realimentação estática de saída pode ser calculada analiticamente como $\mathcal{K} = -\sqrt{2/3} \simeq -0,8165$ resultando em $\|T_2\|_2 = 6^{1/4} \simeq 1,5651$ e $\|T_\infty\|_\infty = 3/\sqrt{5} \simeq 1,3416$. Para o problema misto $\mathcal{H}_2/\mathcal{H}_\infty$, com $\gamma = 1,2$, a solução analítica é obtida como $\mathcal{K} = -0,9458$ resultando em $\|T_2\|_2 = 1,5735$.

Com o procedimento proposto, adotando condições iniciais $Q_0 = 2$ e $x_0 = -1$, a solução do problema \mathcal{H}_2 resulta no mesmo controlador obtido analiticamente após 0,172s de processamento, sendo necessário somente o passo de otimização (sistema precisamente conhecido). Para o problema misto também é obtido o mesmo controlador obtido analiticamente após 0,625s de processamento.

O procedimento apresentado em Arzelier e Peaucelle (2002) não resulta nos mesmos controladores obtidos analiticamente e os autores reconhecem a dificuldade de síntese de controladores por realimentação dinâmica de saída, mesmo para um sistema simples e precisamente conhecido.

Em Kim et al. (2005), é projetado um controlador para o problema misto com $\gamma = 1,2$, utilizando um método de síntese iterativo baseado em LMI com uma restrição de posto não convexa substituída por uma função de penalidade, de modo que, a cada iteração é necessário resolver um problema de otimização convexo. É obtido o controlador $\mathcal{K} = -0,9735$ e $\|T_2\|_2 = 1,5772$. Os resultados para o

problema misto são apresentados na Tabela 6.4.

Tabela 6.4: Resultados de projeto para o problema misto (Ex. 6.5).

Método	\mathcal{K}	$\ T_2\ _2$	$\ T_\infty\ _\infty$
Ótimo	-0,9458	1,5735	1,2000
(Arzelier e Peaucelle, 2002)	-0,9782	1,6825	1,1706
(Kim et al., 2005)	-0,9735	1,5838	1,1746
Proposto	-0,9458	1,5735	1,2000

Em Geromel et al. (1996), é projetado um controlador somente para o problema \mathcal{H}_2 , utilizando um algoritmo tipo planos de corte, resultando em um controlador praticamente igual ao controlador ótimo: $\mathcal{K} = -0,8198$ e $\|T_2\|_2 = 1,5651$.

Em Pereira e de Araújo (2004), é projetado um controlador para o problema misto com $\gamma = 1$, utilizando um algoritmo de otimização genético, resultando no controlador $\mathcal{K} = -3,4176$ e $\|T_2\|_2 = 2,3279$. Pelo procedimento proposto é obtido $\mathcal{K} = -1,5303$ e $\|T_2\|_2 = 1,7173$.

Exemplo 6.6 Considere o problema de estabilização e atenuação de distúrbios do modo de período curto longitudinal do avião de combate F4E analisado em Shaked (2003):

$$\frac{dx(t)}{dt} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & -30 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ 0 \\ 30 \end{bmatrix} u(t) + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} w(t)$$

$$z(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t)$$

cujos parâmetros incertos, determinados para quatro pontos de operação, são apresentados na Tabela 6.5. O sistema não é estável em malha-aberta em todos os pontos de operação. Será considerado o problema de projeto do controlador robusto \mathcal{H}_2 considerando o politopo definido pelos quatro vértices, associados a cada ponto de operação.

Em Shaked (2003), ao invés de considerar o controlador como um ganho fixo, é utilizado um

Tabela 6.5: Valores dos parâmetros incertos nos quatro pontos de operação distintos (Ex. 6.6).

Pt. Op.	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	b_1
1	-0,9896	17,41	96,15	0,2648	-0,8512	-11,39	-97,78
2	-0,6607	18,11	84,34	0,08201	-0,6587	-10,81	-272,2
3	-1,702	50,72	263,5	0,2201	-1,418	-31,99	-85,09
4	-0,5162	29,96	178,9	-0,6896	-1,225	-30,38	-175,6

controlador estritamente próprio:

$$\dot{\eta}(t) = -\rho\eta(t) + \rho y(t), \quad u(t) = K\eta(t) \quad (6.18)$$

sendo $\eta \in \mathbb{R}^m$ e $\rho \gg 1$ um escalar, a ser fixado, formando um filtro passa-baixa com frequência de corte bastante elevada. Segundo (Shaked, 2003), este controlador, além de facilitar a formulação de projeto, representa de forma mais realista os amplificadores que possuem faixa de passagem finita. Cada função de Lyapunov dependente de parâmetro envolve um parâmetro escalar de sintonia β_i , $i = 1, \dots, N$, que devem ser fixados, junto com ρ , para o problema ser representado na forma de LMI. É necessário adotar um algoritmo de otimização para encontrar a combinação de parâmetros que conduz à solução do problema com o atendimento das especificações.

A Tabela 6.6 apresenta o ganho, o pior caso de norma no politopo (o valor entre parênteses corresponde ao pior caso nos vértices) e o custo ε -garantido \mathcal{H}_2 , com precisão 0,1%, considerando os resultados relatados em Shaked (2003), obtidos para $\rho = 10^{10}$, e os obtidos com o procedimento de projeto proposto. Os dois primeiros resultados são obtidos pelos métodos baseados no Teorema 1, com função de Lyapunov comum para os quatro vértices, e no Teorema 3, com funções de Lyapunov dependentes de parâmetros, ambos apresentados em Shaked (2003). Nos resultados obtidos com o procedimento proposto, é adotado o critério de parada na síntese (algoritmo cone-elipsoidal) $\epsilon = 0,001$ e a precisão no cálculo do custo de $\varepsilon = 0,001$. Com $\varepsilon_\delta = 0,1$, o controlador é obtido em uma iteração do procedimento de projeto, partindo da solução inicial aleatória $x_0 = [-1,1465 \ 1,1909]^T$ e $Q_0 = 100I_2$, após 1,84s de tempo de processamento na síntese e 50,06s na análise de desempenho \mathcal{H}_2 , sendo que o pior caso de norma ocorre na coordenada $\alpha_{(2)} = [0 \ 0,7813 \ 0 \ 0,2187]^T$, fora dos vértices do politopo. Para $\varepsilon_\delta = 0,01$, o controlador é obtido em duas iterações do procedimento de projeto, com a inclusão no conjunto $\tilde{\Omega}$ do ponto de pior caso obtido na primeira iteração, sendo necessários 100,58s de processamento total. Neste caso, como nos projetos obtidos em Shaked (2003), o pior caso de norma corresponde ao vértice associado ao ponto de operação 3. Em ambos os projetos com o procedimento proposto são obtidos melhores resultados que os obtidos em Shaked (2003).

No caso de um controle robusto \mathcal{H}_∞ , para as mesmas condições do caso anterior, foram obtidos

Tabela 6.6: Resultados do projeto \mathcal{H}_2 (Ex. 6.6).

Método	\mathcal{K}	$\delta_{p.c.}$	Custo \mathcal{H}_2, δ_c
Teorema 1 (Shaked, 2003)	[0,0634 0,6444]	4,2885	4,2921
Teorema 3 (Shaked, 2003)	[0,0495 0,3870]	5,3547	5,3596
Proposto, $\varepsilon_\delta = 0,1$: 1 iteração	[1,1909 10,2325]	3,6886(3,5020)	3,6917
Proposto, $\varepsilon_\delta = 0,01$: 2 iterações	[1,3081 10,7745]	3,5730(3,5730)	3,5750

os resultados apresentados na Tabela 6.7. Para $\varepsilon_\delta = 0,1$, o tempo de processamento necessário é de 9,64s para síntese e 10,13s para análise do desempenho \mathcal{H}_∞ , com o pior caso de norma ocorrendo na coordenada $\alpha_{(\infty)} = [0 \ 0,875 \ 0 \ 0,125]^T$. Para $\varepsilon_\delta = 0,01$, são necessárias duas iterações e 32,53s de tempo computacional para convergência do procedimento de projeto com a inclusão do ponto $\alpha_{(\infty)}$ no conjunto $\tilde{\Omega}$. Os valores entre parênteses correspondem aos piores casos nos vértices. Também neste caso, o procedimento de projeto proposto apresenta melhores resultados do que os obtidos em Shaked (2003), sendo que o Teorema 2 (Shaked, 2003) utiliza função de Lyapunov comum para os quatro vértices, e o Teorema 4 (Shaked, 2003) adota funções de Lyapunov dependentes de parâmetros.

Tabela 6.7: Resultados do projeto \mathcal{H}_∞ (Ex. 6.6).

Método	\mathcal{K}	$\gamma_{p.c.}$	Custo $\mathcal{H}_\infty, \gamma_c$
Teorema 2 (Shaked, 2003)	[0,0706 0,7663]	4,3015(3,7478)	4,3028
Teorema 4 (Shaked, 2003)	[0,0530 0,4596]	3,1278(3,0622)	3,1280
Proposto, $\varepsilon_\delta = 0,1$: 1 iteração	[2,7708 17,0632]	2,9830(2,8770)	2,9842
Proposto, $\varepsilon_\delta = 0,01$: 2 iterações	[3,6401 22,2378]	2,9663(2,8945)	2,9672

Considere agora que, além de minimizar o pior caso de norma \mathcal{H}_2 no politopo formado pelos quatro pontos de operações, deseja-se posicionar os pólos na intersecção das regiões semi-plano $\text{Real}(s) < -1,25$, disco centrado na origem com raio $r = 25$ e setor cônico com ângulo interno $2\theta = \frac{\pi}{2}$: $\lambda_i(\bar{A}(\alpha, \mathcal{K})) \in \mathcal{D}, \forall i, \forall \alpha \in \Omega, \mathcal{D} = \{s \in \mathbb{C} : \text{Real}(s) < -1,25, |s| < 25, |\angle(s)| < \pi - \frac{\pi}{4}\}$. Neste caso são necessárias três iterações para posicionar os pólos na região especificada. Após a primeira iteração são acrescentados três novos pontos no conjunto $\tilde{\Omega}$: $[0 \ 0,7969 \ 0 \ 0,2031]^T$, correspondente ao pior caso de norma \mathcal{H}_2 ; $[0 \ 0,75 \ 0 \ 0,25]^T$ e $[0 \ 0,5 \ 0 \ 0,5]^T$ associados com sistemas que não atendem às restrições de posicionamento regional de pólos no semi-plano e no disco, respectivamente. Na segunda iteração é acrescentado o ponto com coordenada $[0 \ 0,6875 \ 0 \ 0,3125]^T$ associado a um sistema que não atende à restrição de posicionamento de pólos no disco. Após a terceira iteração, com tempo de processamento total de 108,44s, é obtido o controlador: $K = [0,0641 \ 0,8014]$. A Figura 6.9(a) apresenta o posicionamento de pólos na iteração 1 do procedimento de projeto, em que

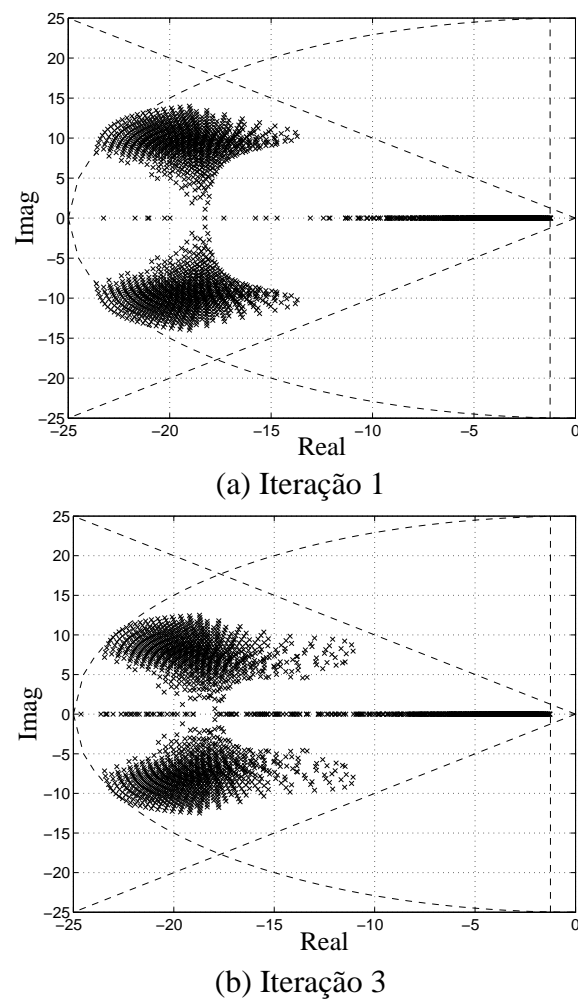


Figura 6.9: Posicionamento regional de pólos (Ex. 6.6).

é verificado que existem casos de sistemas fora dos vértices cujos pólos se localizam fora da região semi-plano e disco. Como pode ser verificado pela Figura 6.9(b), após 3 iterações, o sistema em malha fechada é robustamente \mathcal{D} -estável.

Exemplo 6.7 Considere o sistema incerto discreto no tempo analisado em Yang et al. (2005):

$$x(k+1) = \begin{bmatrix} 0,01 & 0 & 0 \\ 0 & p_1 & 1 \\ 0 & 0,45 & 0,3 + p_2 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0,1 & 0 & 0 \end{bmatrix} w(k) + \begin{bmatrix} 0,1 \\ 0,3 \\ 0,02 \end{bmatrix} u(k)$$

$$z(k) = \begin{bmatrix} 0,1 & 0,3 & 0,4 + p_3 \end{bmatrix} x(k) + \begin{bmatrix} 0,01 & 0,02 & 0,45 \end{bmatrix} w(k) + u(k)$$

$$y(k) = \begin{bmatrix} 0 & 0,1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0 & 0 & 0,414 \end{bmatrix} w(k)$$

cujos parâmetros incertos variam nas seguintes faixas: $p_1 \in [0; 0,02]$, $p_2 \in [0; 0,01]$ e $p_3 \in [0; 0,02]$. Em Yang et al. (2005) é estudado inicialmente o projeto de controladores que minimizam o custo \mathcal{H}_2 utilizando uma formulação não convexa que é transformada em uma formulação LMI através da fixação de um parâmetro de sintonia ϵ , determinado através de busca linear. É apresentado um segundo procedimento de projeto mais complexo, para reduzir o conservadorismo da primeira formulação, na qual é necessário minimizar o traço de um produto de variáveis matriciais utilizando um método de programação linear seqüencial desenvolvido em Leibfritz (2001). Com o procedimento mais simples é obtido o controlador que garante o custo de 0,5139 após 6min40s de processamento em um computador com processador de 3GHz e 1Gb de RAM (como relatado em Yang et al. (2005)). Com procedimento mais complexo é obtido o controlador que garante o custo de 0,4975 após 53min21s de processamento (Yang et al., 2005). Com o procedimento de projeto proposto, adotando $Q_0 = 10I_{16}$, $x_0 \in \mathbb{R}^{16}$ com elementos aleatórios com distribuição normal (média zero e variância unitária), é obtido o seguinte controlador

$$\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{ccc|c} 0,1972 & -0,0013 & -0,0255 & 0,0478 \\ -2,2597 & 0,3273 & -0,1233 & 0,6864 \\ 0,9628 & 0,9975 & -0,6041 & -1,2034 \\ \hline -0,7895 & 0,0970 & 0,0096 & -0,3453 \end{array} \right]$$

que garante um custo de 0,4970, similar ao obtido no melhor projeto apresentado em Yang et al. (2005) mas com tempo de processamento de 1min34s, consideravelmente inferior aos tempo mencionados em Yang et al. (2005). Todos os custos garantidos \mathcal{H}_2 foram calculados com o procedimento proposto de análise com precisão de 1%.

Em Yang et al. (2005) também é considerado o projeto de controladores que minimizam o custo \mathcal{H}_∞ deste mesmo sistema com formulação de projeto semelhante. Com o procedimento mais simples é obtido o controlador que resulta no custo garantido de 1,7941 após 3min5s de processamento. Com procedimento mais complexo é obtido o controlador que garante o custo de 1,6849 após 15min36s de processamento. Com o procedimento de projeto proposto e mesmas especificações informadas no caso \mathcal{H}_2 , é obtido o seguinte controlador

$$\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{ccc|c} 0,1883 & 0,1263 & -0,0562 & 0,3557 \\ -2,2281 & 0,3301 & -0,1866 & 1,1302 \\ 0,9900 & 1,0870 & -1,0267 & -1,4483 \\ \hline -1,1182 & -0,0258 & 0,0375 & 0,1969 \end{array} \right]$$

que garante um custo de 1,6843, similar ao obtido no melhor projeto apresentado em Yang et al. (2005), mas com tempo de processamento de apenas 47s. Todos os custos garantidos \mathcal{H}_∞ foram

calculados com o procedimento proposto de análise com precisão de 1%.

6.4 Síntese de Controladores PID

6.4.1 Introdução

Ao se tratar de controladores dinâmicos por realimentação de saída é importante considerar o controlador proporcional-integral-derivativo (PID). O controlador PID continua nos dias de hoje a configuração de controlador mais aplicada na indústria. Cerca de 90% de todas as malhas de controle na prática utilizam controladores PID em uma extensa lista de aplicações: controle de processos, acionamentos elétricos, automóveis, aviação, instrumentação etc. (Åström e Hägglund, 2001). Esta larga aplicação é devida a sua simplicidade (apenas três parâmetros de ajuste), eficiência na maioria das aplicações industriais e a existência de um método de sintonia simples desenvolvido por John G. Ziegler e Nathaniel B. Nichols em 1942. Desde 1942, vários pesquisadores têm buscado desenvolver procedimentos de síntese para atender as necessidades específicas de cada aplicação (auto-sintonia, controle preditivo etc.) considerando objetivos e complexidade do algoritmo de sintonia completamente diferentes (ver referências em Åström e Hägglund (2001)). Uma revisão a respeito de controladores PID e métodos de sintonia é apresentada em Cominos e Munro (2002). O método da resposta ao degrau de Ziegler-Nichols é revisto e novas regras de sintonia são desenvolvidas em Hang et al. (1991) e Åström e Hägglund (2004).

Apesar das várias técnicas de sintonia PID disponíveis para as diferentes aplicações e configurações de controladores PID, ainda é interessante pesquisar novos métodos que possam resultar em melhor desempenho e que possam ser aplicados a uma classe mais abrangente de problemas, em especial, síntese robusta multiobjetivo para tratar de sistemas incertos representadas por modelos politópicos ou com dependência afim de parâmetros. O desenvolvimento de caracterizações de custo garantido $\mathcal{H}_2/\mathcal{H}_\infty$ em termos de desigualdades matriciais lineares (LMIs) é uma das alternativas para o desenvolvimento de procedimentos de sintonia PID para serem aplicados a sistemas incertos. Entretanto, o controlador PID é um controlador por realimentação dinâmica de saída, de ordem reduzida, com restrição de estrutura, cujo problema de sintonia é difícil de ser representado por formulações LMI. Formular o problema como controle por realimentação de saída estático resulta em restrições na forma desigualdade matriciais bilineares (BMIs). Em Takahashi et al. (1996), a sintonia PID é caracterizada como um problema de otimização multiobjetivo de realimentação estática robusto, com a equação de estados ampliada, sendo que formulações LMI são aplicadas para determinar os custos garantidos resultantes. Em Huang e Wang (2000), o teorema de Kharitonov para intervalos de plantas é explorado com o propósito de caracterizar todos os controladores PID que estabilizam o sistema

incerto. Em Ge et al. (2002), um método de projeto de controladores PID robustos para sistemas com modelos múltiplos é desenvolvido pelo método LQR-LMI para controle por realimentação de saída estática. Nesta seção, será desenvolvida uma estratégia para aplicar o procedimento geral de projeto, apresentado no Capítulo 5, para a sintonia PID. Será mostrado, através de exemplos, que o procedimento geral de projeto proposto pode ser aplicado eficientemente para a sintonia robusta de controladores PID, podendo apresentar melhores resultados que outros métodos já publicados para lidar com o mesmo tipo de problema. Os resultados que serão apresentados na seqüência também podem ser encontrados em Gonçalves, Palhares e Takahashi (2006b) e Gonçalves, Palhares e Takahashi (2006c).

6.4.2 Formulação do Problema

Considere a configuração PID-ISA com dois-graus-de-liberdade apresentada na Figura 6.10 no qual $R(s)$ é o sinal de referência, $D(s)$ é o distúrbio de carga, $U(s)$ é o sinal de controle, $C(s)$ é a saída do sistema e $\eta(s)$ é o ruído de medição. A lei de controle PID-ISA é dada por

$$U(s) = -k_p \left(\frac{T_i s + 1}{T_i s} + \frac{T_d s}{\rho T_d s + 1} \right) [C(s) + \eta(s)] + k_p \frac{T_i s + 1}{T_i s} R(s) \quad (6.19)$$

em que k_p é o ganho proporcional, T_i é o tempo da ação integral (“reset time”), T_d é o tempo da ação derivativa (“rate time”) e $\mathcal{N} \triangleq 1/\rho$ é uma constante do filtro de ruído que geralmente varia entre 3 e 10 (Hang et al., 1991) com $\mathcal{N} = 10$ sendo o valor típico.

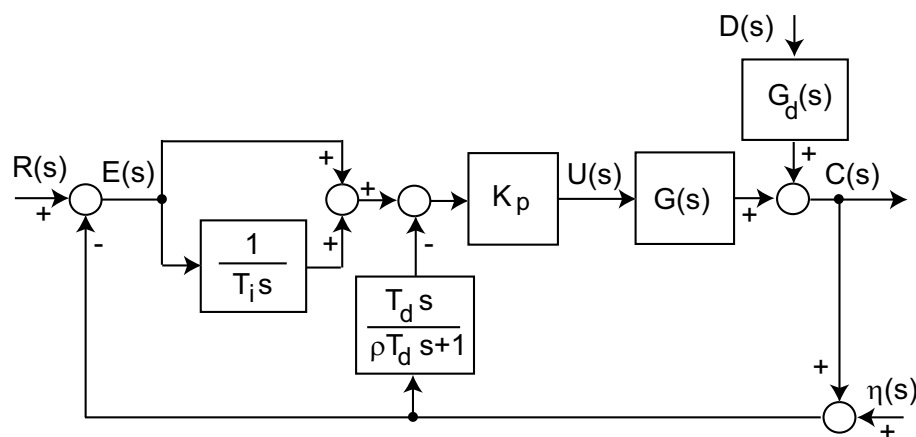


Figura 6.10: Diagrama de blocos da configuração PID-ISA.

A sintonia do controlador PID deve ter como objetivo obter sistemas com as seguintes características: boa resposta de rastreamento, rejeição ao distúrbio de carga, pequena influência do ruído de medição sobre os demais sinais do sistema, sinal de controle com amplitude e taxa de variação

limitadas e robustez em relação às incertezas do modelo. Existem diferentes formas de tratar estes objetivos nas estratégias de sintonia PID. Por exemplo, a rejeição ao distúrbio de carga pode ser obtida pela minimização da norma da função de transferência em malha-fechada relacionando a saída e o distúrbio. As especificações em termos de resposta transitória podem ser atendidas através de posicionamento regional de pólos. Para obter um controlador PID robusto, o método de sintonia PID deve ter capacidade para lidar com sistemas incertos representados por modelos politópicos ou com dependência afim de parâmetros.

Considere o vetor de entrada exógenas definido como $w(t) = [r(t), d(t), \eta(t)]^T$ e o vetor de saídas medidas definido como $y(t) = [c(t) + \eta(t), r(t)]^T$. O controlador PID-ISA pode ser descrito pela seguinte realização: considerando as variáveis de estado $x_{c1}(t)$ e $x_{c2}(t)$ do diagrama de blocos modificado apresentado na Figura 6.11:

$$\mathcal{K}(s) = \left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{cc|cc} 0 & 0 & -\frac{1}{T_i} & \frac{1}{T_i} \\ 0 & -\frac{1}{\rho T_d} & \frac{1}{\rho^2 T_d} & 0 \\ \hline k_p & k_p & -k_p \left(1 + \frac{1}{\rho}\right) & k_p \end{array} \right] \quad (6.20)$$

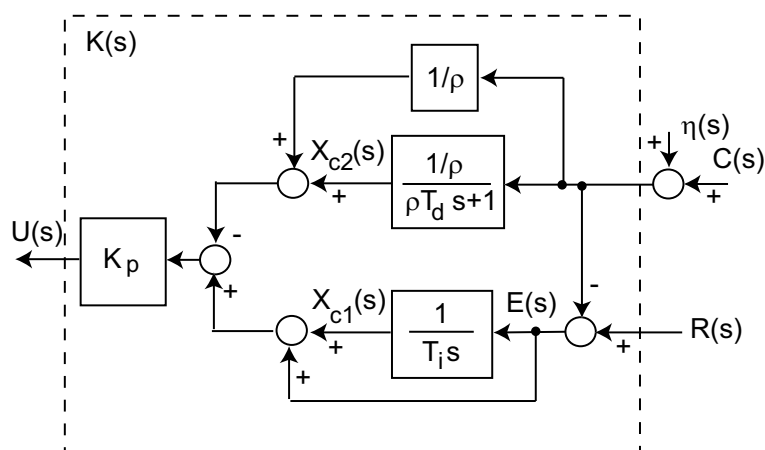


Figura 6.11: Diagrama de blocos modificado do PID-ISA.

O problema de sintonia PID robusta pode ser estabelecido como: encontre os valores de k_p , T_i e T_d que minimizam os valores máximos das normas, $\gamma_{p.c.}$ e $\delta_{p.c.}$, sujeitos a $k_p > 0$, $T_i > 0$, $T_d \geq 0$ e $\lambda_i(\bar{A}(\alpha, \mathcal{K})) \in \mathcal{D}, \forall i, \forall \alpha \in \Omega$.

A restrição de posicionamento regional de pólos é útil porque os controladores ótimos \mathcal{H}_∞ podem resultar em resposta de rastreamento lentas (Takahashi et al., 1996).

6.4.3 Exemplos Ilustrativos

Exemplo 6.8 Considere o sistema incerto a tempo contínuo, apresentado em Huang e Wang (2000), modelado pela função de transferência:

$$G(s) = \frac{5,2(s+2)}{s(s^3 + b_2s^2 + b_1s + b_0)}$$

Os coeficientes do denominador da função de transferência não são precisamente conhecidos podendo variar de acordo com o seguinte domínio de incerteza: $\Omega = \{\alpha = [b_0 \ b_1 \ b_2]^T \in \mathbb{R}^3 : 9,5 \leq b_0 \leq 11,5, 12 \leq b_1 \leq 15, 3,5 \leq b_2 \leq 4,8\}$. Em Huang e Wang (2000), um controlador PID clássico robusto é projetado para garantir as margens de ganho e fase. A seguinte realização é considerada aqui com a definição das variáveis controladas e os acréscimos do distúrbio de entrada, $G_d(s) = G(s)$, e do ruído de medição (ver Figura 6.10):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -b_0 \\ 0 & 1 & 0 & -b_1 \\ 0 & 0 & 1 & -b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 10,4 & 0 \\ 0 & 5,2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ d \\ \eta \end{bmatrix} + \begin{bmatrix} 10,4 \\ 5,2 \\ 0 \\ 0 \end{bmatrix} u$$

$$z_1 = \begin{bmatrix} 0 & 0 & 0 & w_1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ w_2 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ d \\ \eta \end{bmatrix}$$

O procedimento proposto de sintonia PID é aplicado para obter o controlador PID-ISA que minimiza o custo garantido \mathcal{H}_∞ e posiciona os pólos na região semi-plano $\text{Real}(\lambda_i(\bar{A}(\alpha, \mathcal{K}))) < -0,5$, $\forall i, \forall \alpha \in \Omega$. O procedimento proposto obtém $k_p = 1,6910$, $T_i = 3,2082$ e $T_d = 0,5316$ considerando $w_1 = 1$ e $w_2 = 0,01$. Os parâmetros PID $k_p = 3,1950$, $T_i = 1,3975$ e $T_d = 0,2236$, calculados pelo método de resposta em frequência de Ziegler-Nichols são considerados como condições iniciais no algoritmo de otimização. O custo garantido \mathcal{H}_∞ é reduzido de 9,2467 para 2,0082 após 64s de processamento. A Figura 6.12 apresenta as respostas transitórias do sinal de saída para o controlador PID obtido em (Huang e Wang, 2000) e o obtido pelo procedimento proposto, para um degrau unitário

em $r(t)$, um degrau unitário negativo em $d(t)$ iniciando em $t = 15s$ e um ruído de medição aleatório variando na faixa $|\eta(t)| \leq 0,01$. O controlador PID obtido pelo procedimento proposto apresenta melhor rejeição ao distúrbio e é menos influenciado pela variação dos parâmetros incertos.

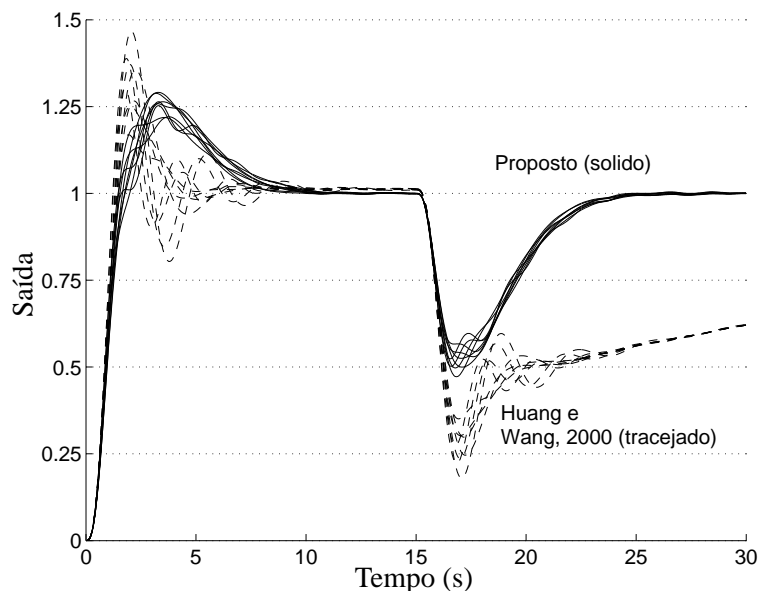


Figura 6.12: Resposta transitória de $y(t)$ nos oito vértices do politopo (Ex. 6.8).

Exemplo 6.9 Considere o sistema com modelos múltiplos apresentado em Ge et al. (2002) e também considerado em Toscano (2005). Para uma faixa de operação estável são obtidos três modelos para diferentes pontos de operação:

$$G_1(s) = \frac{0,04612}{s^2 + 9,251s + 22,19}$$

$$G_2(s) = \frac{0,04107}{s^2 + 2,674s + 10,97}$$

$$G_3(s) = \frac{0,03707}{s^2 + 0,01248s + 5,862}$$

Em Ge et al. (2002), o controlador PID clássico é projetado para minimizar o custo LQR e posicionar os pólos na intersecção das regiões semi-plano e setor cônico com ângulo interno igual a $\frac{3}{4}\pi$: $\mathcal{D} = \{s \in \mathbb{C} : \text{Real}(s) < -1, \pi - |\angle(s)| < \frac{3}{8}\pi\}$. Em Ge et al. (2002), os parâmetros PID são calculados como sendo $k_p = 516,6$, $T_d = 0,2784$ e $T_i = 0,6749$. Em Toscano (2005), os parâmetros PID são calculados como sendo $k_p = 698,1$, $T_d = 0,5259$ e $T_i = 0,6197$. No procedimento pro-

posto, considerando um distúrbio com $G_d(s, \alpha) = 500G(s, \alpha)$ e ruído de medição, os três pontos de operação são tratados como os três vértices do simplexo $\Omega \subset \mathbb{R}^2$ representados por

$$A_1 = \begin{bmatrix} 0 & -22,19 \\ 1 & -9,251 \end{bmatrix}, \quad B_{w,1} = \begin{bmatrix} 0 & 23,06 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_{u,1} = \begin{bmatrix} 0,04612 \\ 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & -10,97 \\ 1 & -2,674 \end{bmatrix}, \quad B_{w,2} = \begin{bmatrix} 0 & 20,5350 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_{u,2} = \begin{bmatrix} 0,04107 \\ 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & -5,862 \\ 1 & -0,01248 \end{bmatrix}, \quad B_{w,3} = \begin{bmatrix} 0 & 18,5350 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_{u,3} = \begin{bmatrix} 0,03707 \\ 0 \end{bmatrix},$$

$$C_{z1,1} = C_{z1,2} = C_{z1,3} = \begin{bmatrix} 0 & w_1 \\ 0 & 0 \end{bmatrix}, \quad D_{zu1,1} = D_{zu1,2} = D_{zu1,3} = \begin{bmatrix} 0 \\ w_2 \end{bmatrix},$$

$$C_{y,1} = C_{y,2} = C_{y,3} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad D_{yw,1} = D_{yw,2} = D_{yw,3} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

O procedimento proposto de sintonia PID é aplicado para minimizar o custo garantido \mathcal{H}_∞ e posicionar os pólos na intersecção do semi-plano $\text{Real}(s) < -2,5$ e do setor cônico centrado na origem com ângulo interno igual a $\frac{\pi}{2}$: $\lambda_i(\bar{A}(\alpha, \mathcal{K})) \in \mathcal{D}, \forall i, \forall \alpha \in \Omega, \mathcal{D} = \{s \in \mathbb{C} : \text{Real}(s) < -2,5, |\angle(s)| < \pi - \frac{\pi}{4}\}, \forall i, \forall \alpha \in \Omega$. O método da resposta ao degrau de Ziegler-Nichols resulta nos parâmetros $k_p = 2,5209 \times 10^3$, $T_i = 0,25$ e $T_d = 0,04$. Partindo destes valores e considerando $w_1 = 1$ e $w_2 = 6 \times 10^{-5}$, o procedimento proposto obtém $k_p = 2.518,4$, $T_i = 0,5703$ e $T_d = 0,1706$. A otimização reduz o custo garantido \mathcal{H}_∞ de 408,37 para 1,6706 após 6,33s de processamento. As resposta transitórias dos controladores PID obtidos com a formulação LMI em Ge et al. (2002), o método em Toscano (2005) e o procedimento proposto são apresentados na Figura 6.13 para um degrau unitário em $r(t)$, um degrau unitário negativo em $d(t)$ iniciando em $t = 5s$ e um ruído de medição aleatório na faixa $|\eta(t)| \leq 0,01$.

Em Ge et al. (2002), também são considerados três pontos de operação em uma região instável:

$$G_1(s) = \frac{0,036}{s^2 - 0,405s + 4,996}$$

$$G_2(s) = \frac{0,026}{s^2 - 2,647s - 0,879}$$

$$G_3(s) = \frac{0,016}{s^2 - 2,251s - 2,252}$$

Os parâmetros do controlador PID, $k_p = 804,6$, $T_i = 1,3249$ e $T_d = 0,3930$ são calculados em Ge et al. (2002). O procedimento geral de projeto proposto, considerando o posicionamento regional de pólos na região $\mathcal{D} = \{s \in \mathbb{C} : \text{Real}(s) < -2, |\angle(s)| < \pi - \frac{3}{8}\pi\}$, obtém os seguintes parâmetros: $k_p = 2.518,1$, $T_i = 0,8748$ e $T_d = 0,2762$. A otimização inicia a partir de um sistema em malha-fechada instável para obter ao final o custo \mathcal{H}_∞ garantido igual a 1,9794 após 16,45s de processamento. As respostas transitórias dos controladores PID obtidos em Ge et al. (2002) e pelo procedimento proposto são apresentadas na Figura 6.13 para as mesmas condições anteriores.

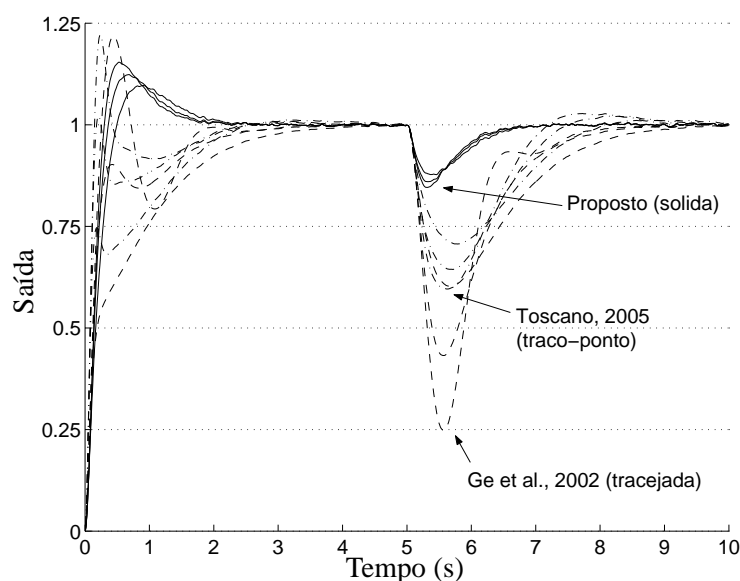


Figura 6.13: Resposta transitória de $y(t)$ para os três vértices (Ex. 6.9).

O controlador PID obtido pelo procedimento proposto apresenta melhor resposta de rastreamento e melhor rejeição do distúrbio do que os controladores apresentados em Ge et al. (2002) e Toscano (2005).

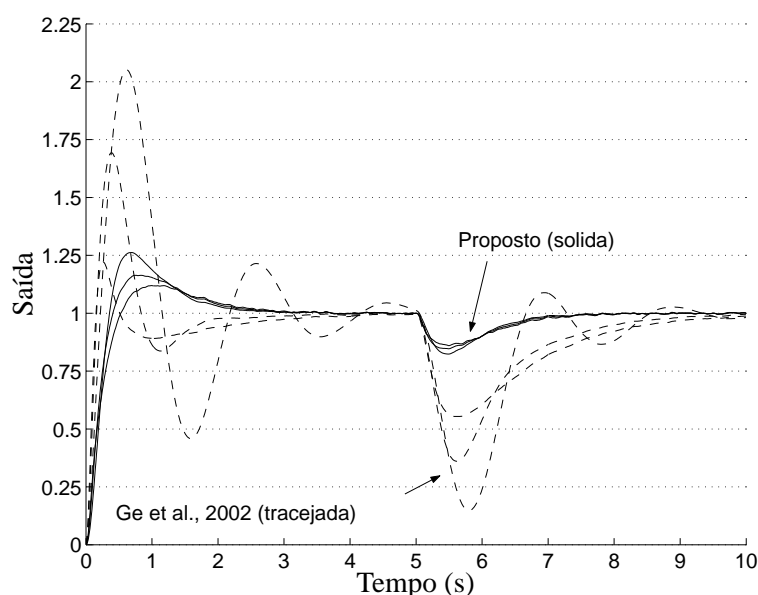


Figura 6.14: Respostas transitórias de $y(t)$ para os três vértices do politopo para os sistemas instáveis em malha-aberta (Ex. 6.9).

6.5 Conclusões

Foi demonstrado através de vários exemplos que o procedimento geral de projeto proposto é um método eficaz para a síntese de controladores robustos $\mathcal{H}_2 / \mathcal{H}_\infty$, com posicionamento regional de pólos, tanto por realimentação de estado como por realimentação estática ou dinâmica da saída, aplicado a sistemas lineares invariantes no tempo, com domínio politópico de incerteza, representados por modelos politópicos ou por dependência afim de parâmetros. O procedimento proposto pode tratar de síntese de controladores dinâmicos com qualquer dimensão e estrutura especificadas como, por exemplo, controle descentralizado, controlador de ordem reduzida e controladores PID. A combinação do procedimento de otimização, considerando diretamente os parâmetros do controlador e um número finito de pontos do domínio infinito de incerteza, com a validação por meio de formulações de análise baseadas em LMIs resultam em controladores robustos menos conservadores que os obtidos baseados puramente em formulações LMIs.

No caso do controle por realimentação de estado, considerando os exemplos analisados, o procedimento geral de projeto proposto apresenta melhores resultados que as formulações LMI. Uma segunda vantagem é a possibilidade de fornecer soluções factíveis para problemas em que as formulações baseadas em LMIs não são factíveis.

O desenvolvimento do procedimento de projeto de controladores robustos por realimentação estática ou dinâmica da saída é uma contribuição bastante relevante uma vez que a maioria das formula-

ções existentes na literatura também consideram problemas de otimização não convexos sem garantia de convergência para o ótimo global.

O procedimento de projeto proposto também pode ser utilizado para aprimorar resultados de projetos obtidos com estratégias mais simples, como ocorreu para os casos de projetos de controladores PID. Para isso pode ser considerado um elipsóide inicial de menor volume no algoritmo elipsoidal que resulta em convergência muito mais rápida.

Capítulo 7

Projeto de Filtros

7.1 Introdução

O problema de projeto de filtros robustos $\mathcal{H}_2/\mathcal{H}_\infty$ para sistemas com incertezas politópicas tem sido tratado em vários artigos nos últimos anos. A maioria das estratégias de projeto apresentadas é baseada em problemas de otimização com função objetivo linear e restrições na forma de LMIs. No caso de sistemas contínuos no tempo, formulações para o projeto de filtro \mathcal{H}_2 e \mathcal{H}_∞ , baseadas no conceito de estabilidade quadrática, podem ser encontradas em Geromel (1999), Geromel e de Oliveira (2001) e Jin e Park (2001); uma formulação generalizada de projeto de filtro \mathcal{H}_2 é apresentada em Palhares e Peres (2000a); uma formulação de projeto de filtro \mathcal{H}_∞ com posicionamento regional de pólos é tratada em Palhares e Peres (1999) e formulações baseadas em funções de Lyapunov dependentes de parâmetros para o projeto de filtros \mathcal{H}_2 são desenvolvidas em Tuan et al. (2001) e Barbosa et al. (2005), sendo que o último considera um parâmetro escalar adicional que deve ser pesquisado para resultar em um menor conservadorismo da formulação. No caso de sistemas discretos no tempo, formulações baseadas em estabilidade quadrática para o problema de projeto de filtro \mathcal{H}_2 e \mathcal{H}_∞ são apresentadas em Geromel et al. (1998); uma formulação baseada em estabilidade quadrática para o problema de projeto de filtro $\mathcal{H}_2/\mathcal{H}_\infty$ misto pode ser obtida em Palhares e Peres (2001); uma formulação de projeto de filtro \mathcal{H}_∞ com posicionamento regional de pólos pode ser encontrada em Palhares e Peres (2000b); formulações baseadas em funções de Lyapunov dependentes de parâmetros para o projeto de filtros \mathcal{H}_2 podem ser vistas em Shaked et al. (2001) e Geromel et al. (2002) e formulações baseadas em funções de Lyapunov dependentes de parâmetros para o projeto de filtros \mathcal{H}_2 e \mathcal{H}_∞ , com parâmetros livres adicionais (que precisam ser pesquisados para reduzir o conservadorismo), são apresentadas em Xie et al. (2004). Projetos de filtro \mathcal{H}_2 com confiabilidade contra falhas de sensores, para sistemas contínuos ou discretos no tempo, podem ser vistos em Liu et al. (2003).

Mesmo considerando a facilidade de solução dos problemas de projeto com formulações baseadas

em LMIs, é interessante desenvolver métodos alternativos que possibilitam: (i) verificar o grau de conservadorismo das formulações de projeto baseadas em LMI; (ii) tratar de situações para as quais os problemas de otimização com restrições tipo LMI não são factíveis e (iii) evitar situações em que os resultados obtidos pelos projetos baseados em LMI não apresentam desempenho satisfatório.

Este capítulo trata da aplicação do procedimento geral de projeto apresentado no Capítulo 5 ao projeto de filtros robustos $\mathcal{H}_2/\mathcal{H}_\infty$ com posicionamento regional dos pólos do filtro. Uma das vantagens do procedimento proposto é a possibilidade de definir a estrutura desejada e a ordem para o filtro a ser projetado.

Várias conseqüências do procedimento proposto são exploradas nos exemplos apresentados na Seção 7.3, que revelam a flexibilidade e o desempenho superior quando comparado com métodos existentes.

Parte dos resultados de projetos de filtros com o procedimento de projeto geral proposto, apresentados a seguir, também podem ser encontrados em Gonçalves, Palhares e Takahashi (2006a).

7.2 Formulação do Problema

Considere o sistema linear invariante no tempo descrito por

$$\begin{aligned}\delta[x(\tau)] &= Ax(\tau) + Bv(\tau) + Ew(\tau) \\ y(\tau) &= Cx(\tau) + Dv(\tau) + Fw(\tau) \\ z(\tau) &= Lx(\tau)\end{aligned}\tag{7.1}$$

sendo $x(\tau) \in \mathbb{R}^n$ o vetor de estado, com $x(0) = 0$, $y(\tau) \in \mathbb{R}^m$ o vetor de saídas medidas, $z(\tau) \in \mathbb{R}^p$ o vetor do sinal de saída a ser estimado, $v(\tau) \in \mathbb{R}^l$ um vetor de entradas tipo ruído branco gaussiano com média zero e covariância e densidade espectral de potência conhecidos e $w(\tau) \in \mathbb{R}^q$ um vetor de distúrbios não nulo com energia limitada, ou seja, $w \in \mathcal{L}_2[0, \infty)$ para sinais contínuos ou $w \in l_2[0, \infty)$ para sinais discretos. O operador $\delta[\cdot]$ representa o operador derivada para sistemas contínuos no tempo ou o operador deslocamento no tempo para sistemas discretos e τ pode representar o tempo contínuo t ou discreto kT , sendo T o período de amostragem e $k \in \mathbb{N}$.

As matrizes do sistema em (7.1) podem ser agrupadas na matriz

$$S \triangleq \begin{bmatrix} A & B & E \\ C & D & F \\ L & 0 & 0 \end{bmatrix}\tag{7.2}$$

Considere que a matriz S não é precisamente conhecida, mas pertence a um conjunto poliédrico

convexo fechado, ou politopo: $S \in \mathcal{P}$. Como no caso de projeto de controladores, para modelos politópicos, o conjunto \mathcal{P} é um politopo no espaço de matrizes definido pelo conjunto de todas as matrizes obtidas pela combinação convexa de seus N vértices:

$$\mathcal{P} \triangleq \left\{ S(\theta) : S(\theta) = \sum_{i=1}^N \theta_i S_i, \theta \in \Omega_M \right\} \quad (7.3)$$

sendo

$$S_i \triangleq \begin{bmatrix} A_i & B_i & E_i \\ C_i & D_i & F_i \\ L_i & 0 & 0 \end{bmatrix}, \quad i = 1, \dots, N \quad (7.4)$$

os vértices do politopo e

$$\Omega_M \triangleq \left\{ \theta \in \mathbb{R}^N : \theta_i \geq 0, i = 1, \dots, N, \sum_{i=1}^N \theta_i = 1 \right\} \quad (7.5)$$

No caso de sistemas dependentes de parâmetros, a matriz S é dependente afim do vetor de parâmetros incertos $p = [p_1, p_2, \dots, p_d]^T \in \mathbb{R}^d$:

$$\mathcal{P} \triangleq \{S(p) : S(p) = S_0 + p_1 S_1 + \dots + p_d S_d, p \in \Omega_p\} \quad (7.6)$$

Se os parâmetros incertos variam entre valores limites, i.e. $p_i \in [\underline{p}_i, \bar{p}_i]$, sendo \underline{p}_i e \bar{p}_i os valores mínimo e máximo do i -ésimo parâmetro incerto, o vetor p pertence a um hiper-retângulo no espaço d -dimensional:

$$\Omega_p \triangleq \left\{ p \in \mathbb{R}^d : \underline{p}_i \leq p_i \leq \bar{p}_i, i = 1, \dots, d \right\} \quad (7.7)$$

O domínio Ω_p também pode ser um politopo de formato qualquer se existirem restrições adicionais sobre os parâmetros. A partir deste ponto, α será utilizado para representar θ ou p e Ω para representar Ω_M ou Ω_p .

O problema de filtragem consiste em gerar as estimativas $\hat{z}(\tau)$ do sinal $z(\tau)$ baseadas no vetor $y(\tau)$ de saídas medidas, conforme apresentado na Figura 7.1. Para realizar esta tarefa é necessário projetar um filtro \mathcal{F} com a seguinte descrição:

$$\begin{aligned} \delta[\hat{x}(\tau)] &= A_f \hat{x}(\tau) + B_f y(\tau), \hat{x}(0) = 0 \\ \hat{z}(\tau) &= C_f \hat{x}(\tau) + D_f y(\tau) \end{aligned} \quad (7.8)$$

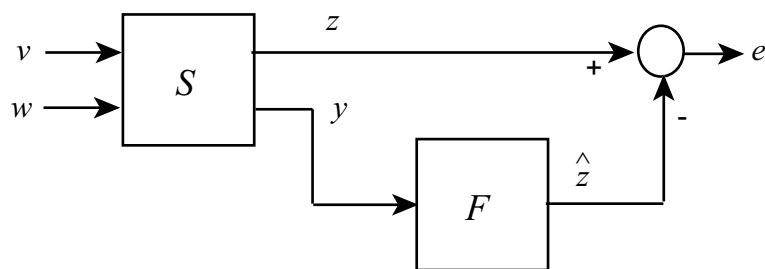


Figura 7.1: Diagrama de blocos do problema de filtragem.

O sistema ampliado \mathcal{E} composto pelo modelo do sistema \mathcal{S} e do filtro \mathcal{F} pode ser descrito como:

$$\begin{aligned}\delta[\bar{x}(\tau)] &= \bar{A}\bar{x}(\tau) + \bar{B}v(\tau) + \bar{E}w(\tau) \\ e(\tau) &= \bar{C}\bar{x}(\tau) + \bar{D}v(\tau) + \bar{F}w(\tau)\end{aligned}\quad (7.9)$$

sendo $\bar{x}(\tau) \triangleq [x^T(\tau) \hat{x}^T(\tau)]^T$, $e(\tau) \triangleq z(\tau) - \hat{z}(\tau)$ e

$$\begin{aligned}\bar{A} &= \begin{bmatrix} A & 0 \\ B_f C & A_f \end{bmatrix}, & \bar{B} &= \begin{bmatrix} B \\ B_f D \end{bmatrix}, & \bar{E} &= \begin{bmatrix} E \\ B_f F \end{bmatrix}, \\ \bar{C} &= \begin{bmatrix} L - D_f C & -C_f \end{bmatrix}, & \bar{D} &= [-D_f D], & \bar{F} &= [-D_f F],\end{aligned}\quad (7.10)$$

com a imposição de $\bar{D} = -D_f D = 0$ no caso de ser considerado desempenho \mathcal{H}_2 em sistemas contínuos no tempo.

As funções de transferência para os dois canais do sistema ampliado \mathcal{E} , para um dado modelo, especificado por α , e um dado filtro \mathcal{F} , são dadas por:

$$T_{ev}(\alpha, \mathcal{F}) = T_2(\alpha, \mathcal{F}) \triangleq \bar{D} + \bar{C}[\lambda I - \bar{A}]^{-1} \bar{B} \quad (7.11)$$

$$T_{ew}(\alpha, \mathcal{F}) = T_\infty(\alpha, \mathcal{F}) \triangleq \bar{F} + \bar{C}[\lambda I - \bar{A}]^{-1} \bar{E} \quad (7.12)$$

sendo que λ representa s ou z no caso de sistemas a tempo contínuo ou discreto, respectivamente (novamente, o significado da variável z fica claro com o contexto).

O problema de projeto de filtro robusto $\mathcal{H}_2/\mathcal{H}_\infty$ consiste em obter a estimativa $\hat{z}(t)$ do sinal $z(t)$ tal que o seguinte vetor de objetivos seja minimizado:

$$J(\mathcal{F}) = \begin{bmatrix} \max_{\alpha \in \Omega} \|T_{ev}(\alpha, \mathcal{F})\|_2 \\ \max_{\alpha \in \Omega} \|T_{ew}(\alpha, \mathcal{F})\|_\infty \end{bmatrix} \quad (7.13)$$

Neste caso, o conjunto Γ , definido na Seção 5.3, é o conjunto dos filtros que satisfazem as restrições de posicionamento de pólos:

$$\Gamma \triangleq \{\mathcal{F} : \lambda(A_f) \subset \mathcal{D}\} \quad (7.14)$$

sendo \mathcal{D} a região no plano complexo desejada para a localização dos pólos do filtro.

7.3 Exemplos Ilustrativos

Os resultados obtidos com o procedimento proposto nos exemplos ilustrativos apresentados a seguir adotam como critério de parada no algoritmo de otimização cone-elipsoidal $\epsilon = 0,001$ e $N_\epsilon = 10$, a precisão nos cálculos dos custos ϵ -garantido de $\epsilon = 0,001$ e como critério de parada do procedimento de projeto $\epsilon_\delta = 0,1$. Nestes exemplos, os custos \mathcal{H}_2 são representados pelo custo ϵ -garantido ao quadrado, δ_c^2 , conforme outros trabalhos publicados nesta área.

Exemplo 7.1 Considere o seguinte sistema incerto a tempo contínuo analisado em Geromel (1999), Tuan et al. (2001), e Barbosa et al. (2005):

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0 & -1 + 0,3\sigma \\ 1 & -0,5 \end{bmatrix} x(t) + \begin{bmatrix} -2 & 0 \\ 1 & 0 \end{bmatrix} v(t) \\ y(t) &= \begin{bmatrix} -100 + 10\beta & 100 \end{bmatrix} x(t) + [0 \ 1] v(t) \\ z(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{aligned}$$

cujo domínio de incerteza é especificado como: $\Omega = \{\alpha = [\sigma \ \beta]^T \in \mathbb{R}^2 : |\sigma| \leq 1, |\beta| \leq 1\}$. Deseja-se projetar filtros estritamente próprios de ordem completa e reduzida que minimizem $\|T_{ev}(\alpha, \mathcal{F})\|_2$, $\forall \alpha \in \Omega$.

Em Geromel (1999) é apresentada uma formulação de projeto de filtro robusto \mathcal{H}_2 de ordem completa para sistemas contínuos no tempo baseada em LMIs com função de Lyapunov fixa e em Tuan et al. (2001) e Barbosa et al. (2005) são apresentadas estratégias de projeto de filtro robusto \mathcal{H}_2 com funções de Lyapunov dependentes de parâmetros. Em Tuan et al. (2001) é também apresentada uma formulação para projeto de filtros de ordem reduzida.

Neste exemplo, a relação entre os elementos do vetor de otimização e as matrizes do filtro de 2ª ordem é definida como

$$A_f = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}, B_f = \begin{bmatrix} x_5 \\ x_6 \end{bmatrix}, C_f = \begin{bmatrix} x_7 & x_8 \end{bmatrix}, D_f = 0$$

Neste exemplo são adotadas, no procedimento proposto, as condições iniciais equivalentes a $A_f = -I_k$, sendo k a ordem do filtro, B_f e C_f com todos elementos iguais a 1, e $Q_0 = 100I_{k^2+2k}$. As funções de transferência dos filtros de 2ª ordem calculados com o método apresentado em Geromel (1999), $\mathcal{F}_G(s)$, com o método apresentado em Tuan et al. (2001), $\mathcal{F}_T(s)$, com o método apresentado em Barbosa et al. (2005), $\mathcal{F}_B(s)$, e com o método proposto, $\mathcal{F}_p(s)$, são

$$\mathcal{F}_G(s) = \frac{-0,010742(s + 1,453)}{(s + 1,937)(s + 0,7119)}$$

$$\mathcal{F}_T(s) = \frac{-0,053057(s + 1,146)}{(s + 11,29)(s + 0,1689)}$$

$$\mathcal{F}_B(s) = \frac{-0,094768(s + 1,464)}{(s + 21,91)(s + 0,1682)}$$

$$\mathcal{F}_p(s) = \frac{-0,35284(s + 1,49)}{(s + 82,17)(s + 0,1648)}$$

Para o cômputo do filtro de 2ª ordem pelo procedimento proposto foi necessário 43,781s de tempo de processamento.

As funções de transferência dos filtros de 1ª ordem calculadas com o método proposto, $\mathcal{F}_{pr}(s)$, após 7,031s de tempo computacional, e com o método apresentado em Tuan et al. (2001), $\mathcal{F}_{Tr}(s)$, são

$$\mathcal{F}_{pr}(s) = \frac{-0,0089029}{s + 0,5362}, \quad \mathcal{F}_{Tr}(s) = \frac{-0,0091873}{s + 0,7297}$$

Na formulação apresentada em Barbosa et al. (2005) é necessário a busca de um parâmetro escalar ε que resulta no melhor desempenho. Neste exemplo, foi utilizado o método da seção áurea com $0,01 \leq \varepsilon \leq 30$ e critério de parada quando a seção de busca for menor que 0,01. O filtro $\mathcal{F}_B(s)$ é obtido para $\varepsilon = 0,8794$ após 47,2810s de tempo de processamento, superior ao requerido pelo procedimento de projeto proposto.

A Tabela 7.1 compara os custos $\mathcal{H}_2, \delta_c^2$ obtidos com os quatro métodos e $|\sigma| \leq \sigma_L, \sigma_L \in \{1; 3,3\}$. O método apresentado em Geromel (1999) obtém solução factível apenas para $\sigma_L < 1,6132$, o método apresentado em Tuan et al. (2001) obtém solução factível para $\sigma_L < 3,05$, e para $\sigma_L = 10/3$, o sistema é instável (Barbosa et al., 2005). Os resultados de minimização da função objetivo na síntese são também apresentados para Geromel (1999), Tuan et al. (2001) e Barbosa et al. (2005), e representam o custo garantido \mathcal{H}_2 obtido.

As funções de transferência dos filtros de 1ª e 2ª ordem, para $\sigma_L = 3,3$, calculadas com o método proposto, são dadas por

$$\mathcal{F}_{pr}(s) = \frac{0,22975}{s + 28,26}$$

Tabela 7.1: Custo \mathcal{H}_2 , δ_c^2 (Ex. 7.1).

Método	Filtro	σ_L	δ_c^2 (síntese)	δ_c^2 (análise)
Geromel (1999)	2 ^a	1	5,278	3,013
Tuan et al. (2001)	2 ^a	1	2,381	2,245
Barbosa et al. (2005)	2 ^a	1	2,188	2,141
Proposto	2 ^a	1	2,120	2,120
Tuan et al. (2001)	1 ^a	1	3,001	2,910
Proposto	1 ^a	1	2,799	2,799
Geromel (1999)	2 ^a	3,3	não factível	
Tuan et al. (2001)	2 ^a	3,3	não factível	
Barbosa et al. (2005)	2 ^a	3,3	31.876	28.826
Proposto	2 ^a	3,3	26,725	26,725
Proposto	1 ^a	3,3	29,413	29,413

$$\mathcal{F}_p(s) = \frac{0,30175(s + 0,4291)}{(s + 49,7)(s + 0,2656)}$$

Neste caso o tempo de processamento, de 311,813s, foi bem superior ao caso em que $\sigma_L = 1$. O tempo de otimização é praticamente o mesmo, o aumento é devido ao tempo necessário para a validação do custo \mathcal{H}_2 . Para $\sigma_L = 1$, o próprio custo garantido já possui a precisão requerida, ao passo que, para $\sigma_L = 3,3$, são necessárias 144 iterações do algoritmo branch-and-bound para o cálculo do custo \mathcal{H}_2 .

Exemplo 7.2 Considere o sistema linear invariante no tempo, a tempo contínuo, com dois parâmetros incertos, adaptado de de Oliveira et al. (2004a):

$$\dot{x}(t) = \begin{bmatrix} -2,02 & 0,16 + \sigma & -0,60 + \beta \\ -0,45 - 2\sigma & -1,85 + \beta & 0,59 - 3\sigma \\ -0,19 - 3\beta & 0,97 + 4\sigma & -0,57 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(t)$$

$$z(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t)$$

sendo o domínio de incerteza dado por: $\Omega = \{\alpha = [\beta \ \sigma]^T \in \mathbb{R}^2 : |\beta| \leq 1, |\sigma| \leq 1\}$. O objetivo de projeto é calcular um filtro estritamente próprio que minimiza $\|T_{ev}(\alpha, \mathcal{F})\|_2, \forall \alpha \in \Omega$.

Neste caso, as formulações apresentadas em Geromel (1999) e em Tuan et al. (2001) falham em obter soluções factíveis. Com a formulação apresentada em Barbosa et al. (2005), é necessário realizar uma pesquisa unidimensional para obter o melhor valor do parâmetro escalar ε . O problema de otimização é factível apenas para $0 < \varepsilon < 0,1$. O melhor desempenho é obtido com $\varepsilon = 0,0212$ que resulta no limite superior de custo na síntese de 8,3692 e no valor de custo ε -garantido \mathcal{H}_2 de $\delta_c^2 = 1,2218$. As funções de transferência do filtro são obtidas como sendo:

$$\mathcal{F}_{B1}(s) = \frac{35477,9909(s - 4,322)(s + 1,329)}{(s + 2,68 \times 10^5)(s^2 + 6,699s + 20,64)}$$

$$\mathcal{F}_{B2}(s) = \frac{-19551,3248(s + 19,54)(s - 1,621)}{(s + 2,68 \times 10^5)(s^2 + 6,699s + 20,64)}$$

Com o procedimento de projeto proposto, considerando a versão dual da formulação LMI para cálculo do custo garantido \mathcal{H}_2 , e adotando condições iniciais x_0 com valores aleatórios com distribuição normal (média zero e variância unitária) e $Q(0) = 100I_{18}$, após 1.431,8s ou 23 minutos e 52 segundos de processamento, são obtidas as seguintes funções de transferência para o filtro:

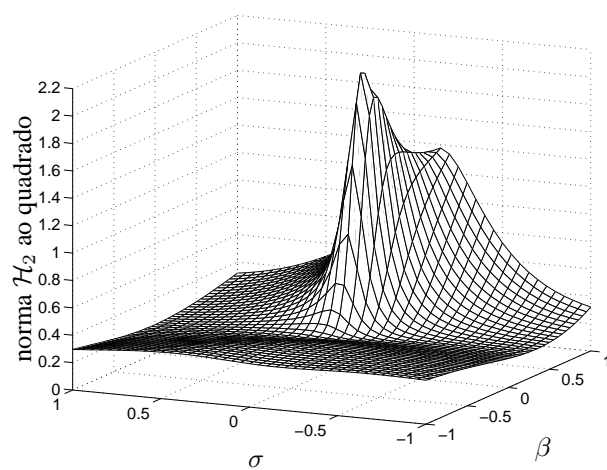
$$\mathcal{F}_{p1}(s) = \frac{-199,8773(s + 2,321)(s + 0,459)}{(s + 1,265)(s^2 + 26,64s + 1166)}$$

$$\mathcal{F}_{p2}(s) = \frac{100,6101(s^2 - 0,3812s + 9,53)}{(s + 1,265)(s^2 + 26,64s + 1166)}$$

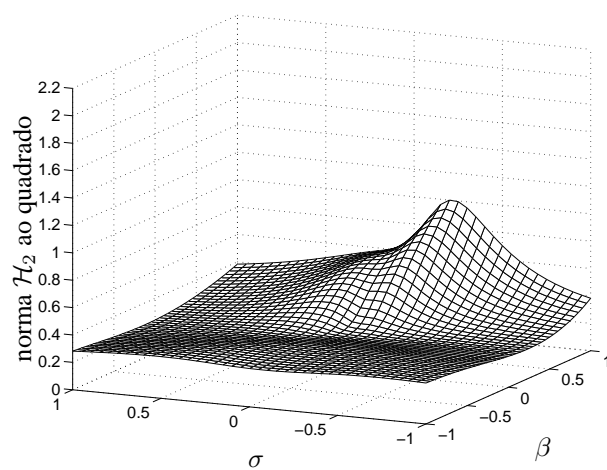
Este projeto garante o custo $\delta_c^2 = 0,5168$, que é cerca de 57,6% menor que o obtido com a formulação apresentada em Barbosa et al. (2005).

Neste caso, como o pior caso da norma \mathcal{H}_2 ocorre no interior do politopo, são necessárias quatro iterações do procedimento, com a adição de três pontos de pior caso no conjunto finito considerado na etapa de síntese, inicializado como o conjunto dos vértices do politopo Ω e finalizado como sendo $\tilde{\Omega} = \{[-1 \ -1]^T, [1 \ -1]^T, [1 \ 1]^T, [-1 \ 1]^T, [0,3750 \ -0,0156]^T, [1 \ -0,2188]^T, [0,3281 \ 0,0469]^T\}$.

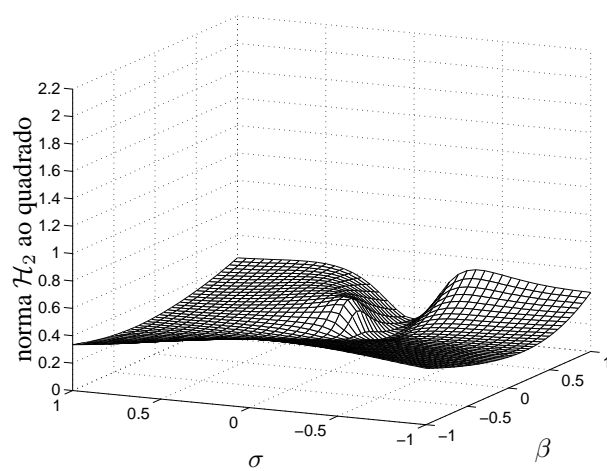
A Figura 7.2 apresenta as superfícies da norma \mathcal{H}_2 ao quadrado nas primeira, segunda e quarta iterações do procedimento de projeto. Através desta figura é possível verificar o funcionamento do procedimento de projeto que, a cada iteração, busca reduzir o pico da superfície nas coordenadas correspondentes aos vértices e aos pontos de pior caso obtidos nas iterações anteriores. O efeito é o de “suavizar” a superfície de forma gradativa. A Figura 7.3 mostra a evolução do custo \mathcal{H}_2 ao quadrado com o número de iterações, na qual é verificada a convergência do procedimento proposto.



(a) 1a. iteração.



(b) 2a. iteração.



(c) 4a. iteração.

Figura 7.2: Evolução da superfície da norma \mathcal{H}_2 ao quadrado (Ex. 7.2).

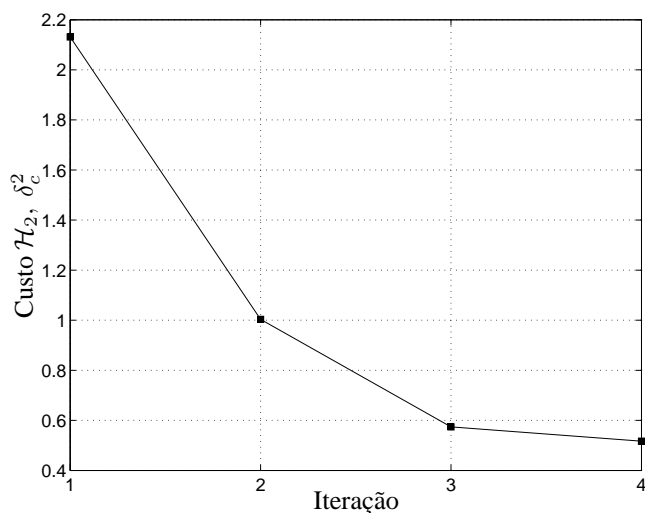


Figura 7.3: Evolução do custo \mathcal{H}_2 (Ex. 7.2).

A Figura 7.4(a) apresenta a partição do espaço de incerteza e a localização dos três pontos adicionais no cálculo do custo \mathcal{H}_2 na última iteração.

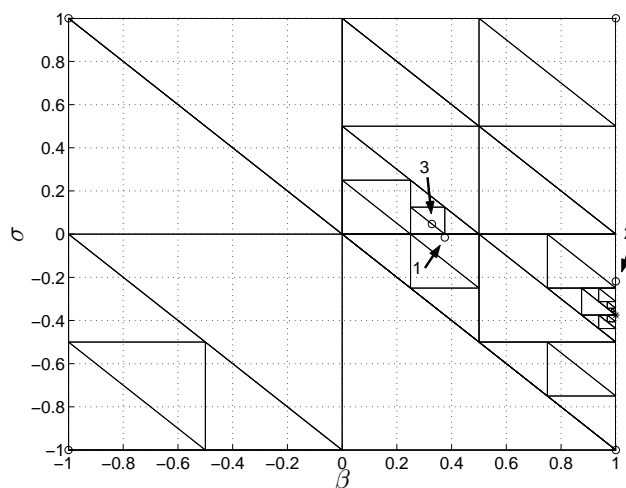


Figura 7.4: Partição do domínio de incerteza e localização das coordenadas de pior caso de norma \mathcal{H}_2 (Ex. 7.2).

Observe que tanto o filtro obtido com a formulação apresentada em Barbosa et al. (2005) como os filtros obtidos com o procedimento de projeto apresentam funções de transferência de fase não-mínima. Através do procedimento proposto é fácil impor que as funções de transferência sejam de fase mínima por intermédio da introdução de restrições adicionais sobre os zeros no problema de

otimização auxiliar. As funções de transferência de fase mínima são obtidas como:

$$\mathcal{F}_1(s) = \frac{-158,3148(s + 2,236)(s + 0,4624)}{(s + 1,265)(s^2 + 25,98s + 941,1)}$$

$$\mathcal{F}_2(s) = \frac{75,1569(s^2 + 0,01986s + 10,29)}{(s + 1,265)(s^2 + 25,98s + 941,1)}$$

correspondendo a $\delta_c^2 = 0,5195$, valor similar ao obtido sem a restrição sobre os zeros do filtro.

Exemplo 7.3 Considere o sistema incerto a tempo discreto, analisado em Xie et al. (2004), descrito pelas seguintes matrizes:

$$A = \begin{bmatrix} 0,6 + 0,05a & 0,1 & 0,2 + 0,05b & -0,3 & -0,2 + 0,05c & 0 \\ 0,05a & 0,4 & -0,3 + 0,05b & 0,2 & 0,1 + 0,05c & 0,1 \\ 0,3 + 0,05a & -0,2 & 0,1 + 0,05b & -0,1 & 0,05c & -0,2 \\ -0,1 + 0,05a & 0,3 & 0,1 + 0,05b & -0,3 & 0,1 + 0,05c & 0,05 \\ 0,1 + 0,05a & 0,2 & -0,1 + 0,05b & 0,1 & 0,3 + 0,05c & 0,1 \\ 0,3 + 0,05a & 0,1 & -0,2 + 0,05b & 0,3 & 0,2 + 0,05c & -0,3 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T, \quad C = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$D = 0,5, \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

cujo domínio de incerteza é um hiper-retângulo com 8 vértices: $\Omega = \{\alpha = [a \ b \ c]^T \in \mathbb{R}^3 : |a| \leq 5, |b| \leq 3, |c| \leq 2\}$. O objetivo de projeto é obter filtros estritamente próprios de ordem completa e reduzida que minimizem o pior caso da norma $\|T_{ev}(\alpha, \mathcal{F})\|_2, \forall \alpha \in \Omega$. Como condições iniciais do algoritmo de otimização, x_0 é formado por números aleatórios com distribuição normal (média zero e variância unitária) e $Q_0 = 10I_{k(k+2)}$, para o projeto do filtro de ordem k . Neste exemplo, para reduzir o custo computacional, é considerado o cálculo dos custos com $\varepsilon = 0,01$.

A Tabela 7.3 apresenta os custos ε -garantidos \mathcal{H}_2 ao quadrado, δ_c^2 , calculados com precisão relativa de 1%, dos filtros obtidos com as formulações apresentadas em Shaked et al. (2001), em Geromel et al. (2002) e em Xie et al. (2004), todas com funções de Lyapunov dependentes de parâmetros, sendo que a última considera a busca de dois parâmetros escalares para redução de conservadorismo, e dos filtros de ordem completa e reduzidos calculados com o procedimento de projeto proposto. Observe que o procedimento de análise apresentado no Capítulo 4 proporciona uma comparação muito mais justa do que a realizada em Xie et al. (2004) para este mesmo exemplo.

Tabela 7.2: Custos \mathcal{H}_2 , δ_c^2 (Ex. 7.3).

Método	Ordem do Filtro	Custo \mathcal{H}_2 , δ_c^2
Shaked et al. (2001)	6 ^a	2,06
Geromel et al. (2002)	6 ^a	1,80
Xie et al. (2004)	6 ^a	1,31
Proposto	6 ^a	0,94
Proposto	5 ^a	0,94
Proposto	4 ^a	0,94
Proposto	3 ^a	0,94
Proposto	2 ^a	1,01
Proposto	1 ^a	1,04

Os filtros calculados com o procedimento de projeto proposto de ordem $k \geq 3$ resultam no mesmo custo \mathcal{H}_2 , sendo 54%, 48% e 28% menor do que os custos obtidos pelas formulações apresentadas em Shaked et al. (2001), em Geromel et al. (2002) e em Xie et al. (2004), respectivamente. A título de ilustração, as funções de transferência dos filtros de ordem completa e de terceira ordem calculadas pelo procedimento de projeto proposto são dadas por:

$$\mathcal{F}_6(z) = \frac{1,0263(z - 0,7741)(z + 0,7283)(z - 0,2053)(z^2 + 0,4884z + 0,1063)}{(z - 0,8423)(z + 0,6364)(z^2 - 0,1047z + 0,0215)(z^2 + 0,6921z + 0,2428)}$$

$$\mathcal{F}_3(z) = \frac{1,0227(z - 0,8156)(z + 0,5411)}{(z - 0,8677)(z^2 + 0,7243z + 0,2075)}$$

A Figura 7.5 apresenta a partição final do domínio de incerteza no cálculo do custo \mathcal{H}_2 para o filtro de 1^a ordem após 13 iterações do procedimento de análise de desempenho apresentado no Capítulo 4.

Como exemplo de posicionamento regional de pólos, considere o projeto do filtro de terceira ordem com a restrição adicional de que os pólos do filtro estejam localizados no semi-plano direito dentro do disco de raio 0,8 e centrado na origem: $\mathcal{D} = \{z \in \mathbb{C} : \text{Real}(z) > 0, |z| < 0,8\}$. A função de transferência do filtro calculada neste caso é dada por:

$$\mathcal{F}(z) = \frac{1,0472(z - 0,6881)(z - 0,253)}{(z - 0,7962)(z^2 - 0,007812z + 0,0908)}$$

que resulta em $\delta_c^2 = 0,94$, igual ao obtido sem a restrição de posicionamento regional de pólos.

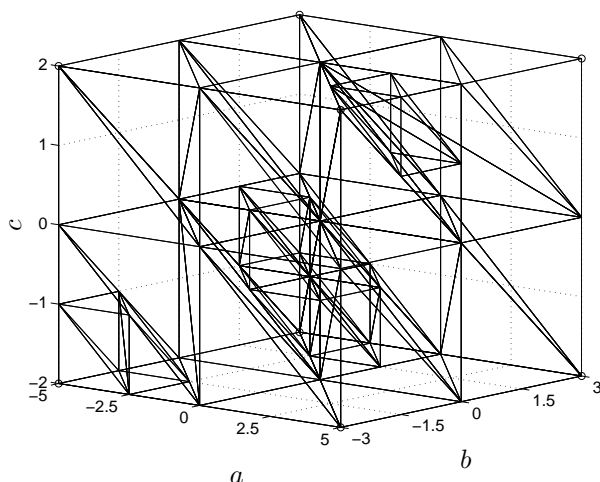


Figura 7.5: Partição final do domínio de incerteza no cálculo do custo \mathcal{H}_2 (Ex. 7.3).

Exemplo 7.4 Considere o seguinte sistema incerto discreto no tempo tratado em Palhares e Peres (2001):

$$x(k+1) = \begin{bmatrix} 0 & -0,8 \\ 1,2 - \sigma & -0,5 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v(k) + \begin{bmatrix} -0,45 \\ 0,35 \end{bmatrix} w(k)$$

$$y(k) = \begin{bmatrix} 0,35 + \beta & -0,65 \end{bmatrix} x(k) + 1,3v(k) + 0,4w(k)$$

$$z(k) = \begin{bmatrix} 0,2 & 0 \end{bmatrix} x(k)$$

sendo o domínio de incerteza dado por: $\Omega = \{\alpha = [\sigma \ \beta]^T \in \mathbb{R}^2 : 0 \leq \sigma \leq 0,5, 0 \leq \beta \leq 1\}$.

O objetivo de projeto é obter um filtro estritamente próprio de ordem completa que minimiza o pior caso da norma $\|T_{ev}(\alpha, \mathcal{F})\|_2$ e que garanta que o pior caso da norma $\|T_{ew}(\alpha, \mathcal{F})\|_\infty$ seja menor que $\gamma = 2, \forall \alpha \in \Omega$.

A Tabela 7.3 apresenta os resultados de projeto para a formulação LMI baseada em estabilidade quadrática, como apresentado em Palhares e Peres (2001), e para o procedimento proposto. Os resultados do procedimento proposto foram obtidos com as condições iniciais equivalentes a $A_f = 0,9I_2$, B_f e C_f com todos os elementos iguais a 1 e $Q_0 = 100I_8$. Neste caso, o procedimento proposto obtém um resultado 24% menor do que a formulação LMI.

Tabela 7.3: Custos H_2 , δ_c^2 (Ex. 7.4).

Método	Palhares e Peres (2001)	Proposto
Custo H_2 , δ_c^2	0,0871	0,0666
Custo H_∞ , γ_c	1,5157	1,4169

As matrizes do filtro calculadas com o procedimento proposto para $\gamma = 2$ são

$$A_f = \begin{bmatrix} 0,3080 & 0,5976 \\ -0,9214 & -0,2979 \end{bmatrix}, \quad B_f = \begin{bmatrix} 0,2548 \\ 0,1170 \end{bmatrix}$$

$$C_f = \begin{bmatrix} -0,1196 & 0,3225 \end{bmatrix}$$

Para este exemplo, o procedimento proposto obtém soluções factíveis para $\gamma \geq 1,39$ enquanto a formulação LMI é factível apenas para $\gamma \geq 1,90$.

7.4 Conclusões

Neste capítulo, o procedimento geral de projeto proposto foi utilizado para síntese de filtros robustos $\mathcal{H}_2/\mathcal{H}_\infty$, com posicionamento regional de pólos, para sistemas lineares invariantes no tempo, a tempo contínuo ou discreto, com domínios politópicos de incerteza. A síntese por intermédio de otimização realizada diretamente no espaço dos parâmetros do filtro, reduz o conservadorismo que aparece nas formulações baseadas em LMIs. Além disso, o procedimento proposto permite incluir restrições adicionais sobre a estrutura do filtro. Um exemplo interessante desta flexibilidade de projeto é considerar restrições sobre os zeros das funções de transferência do filtro de modo a garantir filtros de fase mínima. Tais restrições são difíceis de serem tratadas por formulações LMI.

Considerando os exemplos analisados, o procedimento de projeto obtém resultados melhores ou iguais aos dos procedimentos baseados em LMIs além de ter a capacidade de obter soluções para problemas em que as formulações LMI não são factíveis.

Capítulo 8

Redução de Modelos

8.1 Introdução

Aproximação de sistemas complexos de alta ordem por modelos de baixa ordem mais simples é um dos problemas fundamentais da teoria de sistemas lineares merecendo ainda a atenção de vários pesquisadores. Seja $G(\lambda)$ um sistema linear invariante no tempo estável de ordem n , com λ representando s ou z em caso de sistemas a tempo contínuo ou discreto, respectivamente. Nesta tese, o problema de redução de modelos é obter uma aproximação $G_r(\lambda)$ de ordem r , com $r < n$, de modo que uma norma p do erro de redução, $\|E(\lambda)\|_p$, com $E(\lambda) \triangleq G(\lambda) - G_r(\lambda)$, seja minimizada. O termo ordem é empregado como sendo a dimensão do vetor de estado da realização mínima (realização controlável e observável) de $G(\lambda)$, também conhecido como grau de McMillan. A escolha da norma depende da aplicação, podendo ser consideradas, por exemplo, as normas \mathcal{H}_∞ e/ou \mathcal{H}_2 .

No caso de sistemas precisamente conhecidos, os métodos mais conhecidos de redução de modelos são os métodos de redução balanceada (Moore, 1981) e de redução pela aproximação da norma de Hankel ótima (Glover, 1984), ambas aplicadas sobre modelos balanceados. Outra possibilidade de redução de modelos é a minimização da norma do erro de redução considerando problemas de otimização baseados em desigualdades matriciais lineares (LMIs) estendendo resultados já conhecidos na área de síntese de controladores (Helmersson, 1994). O problema de redução de modelos é similar ao problema de síntese de controladores dinâmicos de ordem reduzida que resulta em formulações não convexas em termos de desigualdades matriciais bilineares (BMIs). Em Grigoriadis (1995) é apresentado um algoritmo iterativo para a minimização da norma \mathcal{H}_∞ do erro de redução de sistemas lineares a tempo contínuo e discreto. Em Geromel et al. (2004) o problema de redução de modelo \mathcal{H}_∞ e \mathcal{H}_2 de sistemas discretos no tempo em termos de BMIs é transformado em um problema sub-ótimo, baseado em LMIs, através da fixação de parte das variáveis evitando com isso a necessidade de um procedimento iterativo. De forma semelhante, em Ebihara e Hagiwara (2004b) é apresentada uma

formulação convexa baseada em LMIs para o problema de redução \mathcal{H}_∞ de sistemas contínuos no tempo, no qual uma matriz é fixada para tornar o problema convexo, obtendo resultados relacionados com os obtidos pelo método de aproximação da norma de Hankel ótima. O mesmo problema é resolvido por estratégia semelhante em Geromel et al. (2005). Em Kanno (2005) o problema de redução de modelos, com critério \mathcal{H}_2 , para sistemas a tempo contínuo e discreto, é resolvido pelo método de projeções alternadas, apresentando uma discussão interessante sobre a escolha da condição inicial no algoritmo de otimização. Mais informações sobre redução de modelos conhecidos, baseada em formulações balanceadas, podem ser encontradas em Gugercin e Antoulas (2000), Gugercin e Antoulas (2004) e em suas referências.

No caso de sistemas incertos, alguns trabalhos recentes apresentam formulações para obtenção de modelos reduzidos fixos. Uma das estratégias utilizadas é a generalização do método de redução balanceada para tratar de sistemas incertos como, por exemplo, em Beck et al. (1996) que apresenta um método baseado na solução de um par de LMIs acopladas (problema não convexo) para lidar com modelos por transformação fracional linear (LFT). Em Assunção e Peres (1999) também é apresentado um método para obter um modelo reduzido fixo em que o problema de otimização é novamente formulado em termos de restrições LMIs acopladas, exigindo um algoritmo iterativo, que pode ser aplicado a sistemas discretos no tempo com incertezas politópicas. Em Trofino e Coutinho (2004) é tratado o problema de redução de modelos com critérios de otimalidade \mathcal{H}_2 e \mathcal{H}_∞ , considerando sistemas contínuos lineares invariantes no tempo, com modelo politópico, sendo apresentadas formulações baseadas em LMIs nas quais o modelo aproximado de ordem completa é formatado de tal modo que possa ser isolado o modelo reduzido com a ordem desejada, para isso sendo necessária a escolha de uma matriz não singular arbitrária. Em Halevi e Shaked (2004) é considerada uma abordagem similar para o critério \mathcal{H}_2 , porém incluindo dois parâmetros escalares de sintonia que devem ser pesquisados para gerar o resultado menos conservador. Outro problema importante é a obtenção de modelos reduzidos que reproduzem o modelo de incerteza. Em Halevi et al. (1997) é considerado o problema de obter modelos reduzidos dependentes de parâmetros, porém, ao invés de dependência afim de parâmetros é considerada uma dependência não linear dos parâmetros. Em Dolgin e Zeheb (2004) é apresentado um método de aproximação por modelos incertos porém limitado a sistemas incertos discretos no tempo monovariáveis (SISO) nos quais a incerteza aparece sobre os coeficientes do modelo por resposta ao impulso finita (FIR). Em Wu (1996) é apresentado um procedimento para cálculo de modelos reduzidos politópicos de sistemas politópicos lineares a tempo contínuo, considerando como critério de otimalidade o custo garantido \mathcal{H}_∞ do erro de redução, baseado no método de projeções alternadas. Em Wu e Jaramillo (2002) é abordado o mesmo problema só que considerando um procedimento baseado no algoritmo de complementaridade do cone.

Neste capítulo, o procedimento geral de projeto será aplicado ao problema de redução de modelos

com aproximação por modelos reduzidos fixos, politópicos ou com dependência afim de parâmetros.

8.2 Redução de Modelos Balanceada

As formas mais clássicas de redução de modelos são baseadas em modelos balanceados. Considere o sistema linear invariante no tempo, estável, de grau de McMillan n , representado por

$$G(\lambda) \triangleq \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \Leftrightarrow \begin{cases} \delta[x(t)] = Ax(t) + Bw(t) \\ y(t) = Cx(t) + Dw(t) \end{cases} \quad (8.1)$$

sendo $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{q \times n}$, $D \in \mathbb{R}^{q \times p}$, λ representando s ou z e $\delta[\cdot]$ denotando o operador derivada, $\delta[x(t)] = dx(t)/dt$, ou o operador deslocamento, $\delta[x(t)] = x(t+1)$, para sistemas a tempo contínuo ou discreto, respectivamente.

Os gramianos de controlabilidade e observabilidade, $P = P^T \succ 0$ e $Q = Q^T \succ 0$, respectivamente, podem ser calculados como sendo as soluções das seguintes equações de Lyapunov:

$$AP + PA^T + BB^T = 0 \quad (8.2)$$

$$A^T Q + QA + C^T C = 0 \quad (8.3)$$

no caso de sistemas a tempo contínuo e

$$APA^T - P + BB^T = 0 \quad (8.4)$$

$$A^T QA - Q + C^T C = 0 \quad (8.5)$$

no caso de sistemas a tempo discreto.

Se $G(\lambda)$ é controlável e observável, uma realização balanceada de $G(\lambda)$ é uma realização em que os gramianos de controlabilidade e observabilidade são iguais a uma matriz diagonal:

$$P = Q = \Sigma \triangleq \begin{bmatrix} \sigma_1 I_{m_1} & 0 & \dots & 0 \\ 0 & \sigma_2 I_{m_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_k I_{m_k} \end{bmatrix} \quad (8.6)$$

sendo $\sigma_1 > \sigma_2 > \dots > \sigma_k > 0$ os valores singulares de Hankel de $G(\lambda)$ com multiplicidades m_i tais que $\sum_{i=1}^k m_i = n$. Os valores singulares de Hankel são definidos como $\sigma_i \triangleq \sqrt{\lambda_i(PQ)}$, $i = 1, \dots, k$, não sendo alterados por uma transformação de similaridade. Estes valores informam o peso

da influência de cada variável de estado da realização balanceada sobre a relação entrada-saída de $G(\lambda)$. Deste modo, uma forma de redução de modelo é descartar as variáveis de estado associadas aos menores valores singulares de Hankel (Moore, 1981).

Considere a seguinte partição da realização balanceada de $G(\lambda)$, $x = [x_1^T \ x_2^T]^T$, tal que:

$$G(\lambda) = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array} \right] \quad (8.7)$$

e a respectiva partição do gramiano: $\Sigma = \text{diag}(\Sigma_1, \Sigma_2)$, com $\Sigma_1 \triangleq \text{diag}(\sigma_1 I_{m_1}, \dots, \sigma_s I_{m_s})$ e $\Sigma_2 \triangleq \text{diag}(\sigma_{s+1} I_{m_{s+1}}, \dots, \sigma_k I_{m_k})$, sendo $\sigma_s \ll \sigma_{s+1}$. A eliminação das variáveis de estados $x_2(t)$, menos significativas na relação entrada-saída de $G(s)$, pode ser feita por dois modos diferentes: truncamento ou residualização (Skogestad e Postlethwaite, 1996). No truncamento, o modelo reduzido de ordem $r = \sum_{i=1}^s m_i$, $G_r(\lambda)$, é obtido eliminando diretamente as variáveis x_2 do modelo balanceado particionado dado por (8.7):

$$G_r(\lambda) \triangleq \left[\begin{array}{c|c} A_r & B_r \\ \hline C_r & D_r \end{array} \right] = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array} \right] \quad (8.8)$$

com $A_r \in \mathbb{R}^{r \times r}$, $B_r \in \mathbb{R}^{r \times p}$, $C_r \in \mathbb{R}^{q \times r}$, $D_r \in \mathbb{R}^{q \times p}$.

Na residualização, considera-se nula a variação de $x_2(t)$, ou seja $\delta[x_2(t)] = 0$, e a equação resultante é resolvida para $x_1(t)$ de modo a obter as matrizes do modelo reduzido assumindo que A_{22} é não singular:

$$A_r \triangleq A_{11} - A_{12}A_{22}^{-1}A_{21} \quad (8.9)$$

$$B_r \triangleq B_1 - A_{12}A_{22}^{-1}B_2 \quad (8.10)$$

$$C_r \triangleq C_1 - C_2A_{22}^{-1}A_{21} \quad (8.11)$$

$$D_r \triangleq D - C_2A_{22}^{-1}B_2 \quad (8.12)$$

Em ambos os casos de redução balanceada pode ser provado que

$$\|G(\lambda) - G_r(\lambda)\|_\infty \leq 2(\sigma_{r+1} + \sigma_{r+2} + \dots + \sigma_k) \quad (8.13)$$

As diferenças entre as duas formas de redução balanceada é que o truncamento garante o mesmo ganho em altas-freqüências, $G(\infty) = G_r(\infty) = D$, ao passo que a residualização garante o mesmo ganho em regime estacionário, $G(0) = D - CA^{-1}B = G_r(0) = D_r - C_rA_r^{-1}B_r$.

8.3 Redução de Modelos Incertos

Neste trabalho, o objetivo é apresentar uma estratégia de redução de modelos incertos em que as matrizes do sistema dado por (8.1) não são precisamente conhecidas. No caso de modelos politópicos, as matrizes do sistema pertencem a um domínio convexo fechado, ou politopo, que é obtido pela combinação convexa de seus vértices:

$$G(\lambda, \theta) = \sum_{i=1}^N \theta_i G_i \quad (8.14)$$

sendo

$$G_i \triangleq \left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & D_i \end{array} \right] \quad (8.15)$$

o conjunto de N vértices conhecidos e $\theta = [\theta_1 \dots \theta_N]^T \in \Omega_M$ o vetor de coordenadas do politopo, sendo Ω_M definido como

$$\Omega_M \triangleq \left\{ \theta : \theta_i > 0, i = 1, \dots, N, \sum_{i=1}^N \theta_i = 1 \right\} \quad (8.16)$$

No caso de modelos com dependência afim de um vetor de parâmetros incertos $p = [p_1 \dots p_d]^T$, sendo que p_i varia entre valores mínimo, \underline{p}_i , e máximo, \bar{p}_i , conhecidos, as matrizes do sistema podem ser calculadas como

$$G(\lambda, p) = G_0 + \sum_{i=1}^d p_i G_i \quad (8.17)$$

sendo $G_i, i = 0, \dots, d$, como em (8.15), as matrizes que definem a dependência afim dos parâmetros incertos.

Neste caso, $p \in \Omega_p$, sendo Ω_p um politopo na forma de um hiper-retângulo, definido pela combinação dos valores extremos dos parâmetros incertos, ou um politopo de formato qualquer se existirem restrições lineares adicionais sobre os parâmetros incertos.

O modelo reduzido pode ser um modelo fixo ou um modelo incerto que reproduz a estrutura do modelo incerto original. No caso de modelos politópicos, o modelo reduzido incerto é dado por

$$G_r(\lambda, \theta) \triangleq \sum_{i=1}^N \theta_i \left[\begin{array}{c|c} A_{r_i} & B_{r_i} \\ \hline C_{r_i} & D_{r_i} \end{array} \right] \quad (8.18)$$

sendo $(A_{r_i}, B_{r_i}, C_{r_i}, D_{r_i}), i = 1, \dots, N$, os vértices do politopo a ser determinado pelo procedimento de projeto e $\theta \in \Omega_M$.

No caso de modelos com dependência afim de parâmetros, o modelo reduzido incerto é dado por

$$G_r(\lambda, p) \triangleq \left[\begin{array}{c|c} A_{r_0} & B_{r_0} \\ \hline C_{r_0} & D_{r_0} \end{array} \right] + \sum_{i=1}^d p_i \left[\begin{array}{c|c} A_{r_i} & B_{r_i} \\ \hline C_{r_i} & D_{r_i} \end{array} \right] \quad (8.19)$$

sendo $(A_{r_i}, B_{r_i}, C_{r_i}, D_{r_i})$, $i = 0, \dots, d$, as matrizes que caracterizam a dependência afim dos parâmetros incertos a serem determinadas pelo procedimento de projeto.

8.4 Procedimento de Projeto

Nesta seção é descrito como o procedimento geral de projeto pode ser aplicado ao problema de redução de modelo. Considere que α representa θ ou p e Ω representa Ω_M ou Ω_p . Defina $\tilde{\Omega}$ como o conjunto finito de pontos do domínio de incerteza politópico inicializado como o conjunto dos vértices de Ω :

$$\tilde{\Omega} \triangleq \text{Vert}(\Omega) \quad (8.20)$$

Seja $E(\lambda) \triangleq G(\lambda) - G_r(\lambda)$ o erro de redução de modelo dado por:

$$E(\lambda) = \left[\begin{array}{cc|c} A & 0 & B \\ 0 & A_r & B_r \\ \hline C & -C_r & D - D_r \end{array} \right] \quad (8.21)$$

A partir deste ponto, os argumentos λ e/ou α poderão ser omitidos por conveniência. Considere os pontos de “pior caso” no conjunto finito de pontos $\tilde{\Omega}$, dado um modelo reduzido G_r :

$$\begin{aligned} \tilde{\gamma}_{p.c.} &= \max_{\alpha \in \tilde{\Omega}} \|E(\alpha)\|_{\infty} \\ \tilde{\delta}_{p.c.} &= \max_{\alpha \in \tilde{\Omega}} \|E(\alpha)\|_2 \end{aligned} \quad (8.22)$$

e os pontos de “pior caso” em todo o politopo:

$$\begin{aligned} \gamma_{p.c.} &= \max_{\alpha \in \Omega} \|E(\alpha)\|_{\infty} \\ \delta_{p.c.} &= \max_{\alpha \in \Omega} \|E(\alpha)\|_2 \end{aligned} \quad (8.23)$$

com suas correspondente coordenadas

$$\begin{aligned} \alpha_{(\infty)} &= \arg \max_{\alpha \in \Omega} \|E(\alpha)\|_{\infty} \\ \alpha_{(2)} &= \arg \max_{\alpha \in \Omega} \|E(\alpha)\|_2 \end{aligned} \quad (8.24)$$

Defina agora o conjunto de modelos reduzidos robustamente estáveis:

$$\tilde{\Gamma} \triangleq \left\{ G_r(\alpha) : \lambda_i(A_r(\alpha)) \subset \mathcal{D}, \quad i = 1, \dots, r, \quad \forall \alpha \in \tilde{\Omega} \right\} \quad (8.25)$$

sendo $\mathcal{D} \subset \mathbb{C}$ a região semi-plano esquerdo ou o disco centrado na origem de raio unitário em caso de sistemas a tempo contínuo ou discreto, respectivamente.

Será utilizado o seguinte problema auxiliar definido por:

Problema Auxiliar: *Dado os escalares $\lambda_1 > 0$, $\lambda_2 > 0$, $\lambda_1 + \lambda_2 = 1$, $\epsilon_1 > 0$ e $\epsilon_2 > 0$, encontre o modelo reduzido G_r^* , tal que:*

$$G_r^* = \arg \min_{G_r} \max_{\alpha} (\lambda_1 \|E(\alpha)\|_{\infty} + \lambda_2 \|E(\alpha)\|_2)$$

$$\text{sujeito a: } \begin{cases} \alpha \in \tilde{\Omega} \\ G_r \in \tilde{\Gamma} \\ \|E(\alpha)\|_{\infty} \leq \epsilon_1 \\ \|E(\alpha)\|_2 \leq \epsilon_2 \end{cases}$$

O procedimento de projeto apresentado a seguir é adaptado do procedimento geral de projeto apresentado no Capítulo 5. O procedimento finaliza quando todas as restrições são atendidas e não existe possibilidade de reduzir a função objetivo pelo acréscimo de novos pontos, o que é verificado por um índice de precisão relativa ϵ_{δ} . Como no procedimento geral de projeto, γ_c e δ_c representam os custos ϵ -garantidos \mathcal{H}_{∞} e \mathcal{H}_2 , respectivamente.

Procedimento de Projeto

Passo 1. Inicialize $i \leftarrow 0$,

$$\tilde{\Omega}_0 \leftarrow \text{Vert}(\Omega).$$

Passo 2. $i \leftarrow i + 1$, $\tilde{\Omega}_i \leftarrow \tilde{\Omega}_{i-1}$.

Passo 3. Resolva o problema auxiliar para encontrar G_r^* e $\tilde{\gamma}_{p.c.}$ e/ou $\tilde{\delta}_{p.c.}$.

Passo 4. Se $(\lambda_1 > 0$ ou $\epsilon_1 < \infty)$ então calcule γ_c , $\gamma_{p.c.}$ e $\alpha_{(\infty)}$ para G_r^* .

Se $\alpha_{(\infty)} \notin \tilde{\Omega}_i$ então se $\{\lambda_1 > 0$ e $(\gamma_{p.c.} - \tilde{\gamma}_{p.c.})/\tilde{\gamma}_{p.c.} > \epsilon_{\delta}\}$ ou $\gamma_c > \epsilon_1$, então $\tilde{\Omega}_i \leftarrow \tilde{\Omega}_i \cup \alpha_{(\infty)}$.

Passo 5. Se $(\lambda_2 > 0$ ou $\epsilon_2 < \infty)$ então calcule δ_c , $\delta_{p.c.}$ e $\alpha_{(2)}$ para G_r^* .

Se $\alpha_{(2)} \notin \tilde{\Omega}_i$ então se $\{\lambda_2 > 0 \text{ e } (\delta_{p.c.} - \tilde{\delta}_{p.c.})/\tilde{\delta}_{p.c.} > \varepsilon_\delta\}$ ou $\delta_c > \varepsilon_2$, então $\tilde{\Omega}_i \leftarrow \tilde{\Omega}_i \cup \alpha_{(2)}$.

Passo 6. Se $\tilde{\Omega}_i \neq \tilde{\Omega}_{i-1}$ vá para o passo 2. Caso contrário, fim.

8.5 Exemplos Ilustrativos

Os resultados apresentados nos exemplos a seguir foram computados através do LMI Control Toolbox para MATLAB[®] em um computador com processador de 2.8Ghz e 1Gb de memória RAM.

Exemplo 8.1 Considere o sistema incerto, linear invariante no tempo, a tempo contínuo (Wu, 1996; Wu e Jaramillo, 2002; Trofino e Coutinho, 2004):

$$G(s, p) = \left[\begin{array}{cccc|ccc} -2 & 3 & -1 & 1 & -2,5 & b_{12} & -1,2 \\ 0 & -1 & 1 & 0 & 1,3 & -1 & 1 \\ 0 & 0 & a_{33} & 12 & 1,6 & 2 & 0 \\ 0 & 0 & 0 & -4 & -3,4 & 0,1 & 2 \\ \hline -2,5 & 1,3 & 1,6 & -3,4 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0,1 & 0 & 0 & 0 \\ -1,2 & 1 & 0 & 2 & 0 & 0 & 0 \end{array} \right]$$

em que $p = [a_{33} \ b_{12}]^T \in \Omega_p$, sendo $\Omega_p = \{p \in \mathbb{R}^2 : -3,5 \leq a_{33} \leq -2,5, -0,5 \leq b_{12} \leq 0,5\}$. O problema considerado é calcular um modelo reduzido de 2^a ordem que minimize o pior caso de norma \mathcal{H}_∞ do erro de redução no domínio de incerteza. Em Trofino e Coutinho (2004) é relatado que foi obtido o limitante $\gamma_{c.g.} = 5,54$ para um modelo reduzido de 2^a ordem fixo. Os procedimentos de aproximação baseados no método de projeções alternadas (Wu, 1996) e no algoritmo de complementaridade do cone (Wu e Jaramillo, 2002) calculam modelos reduzidos politópicos que resultam em $\gamma_{c.g.} = 3,79$ e $\gamma_{c.g.} = 3,67$, respectivamente.

O procedimento proposto é aplicado para calcular os modelos reduzidos de 2^a ordem fixo e com dependência afim de parâmetros. São utilizados como condições iniciais o modelo reduzido obtido pelo truncamento do modelo balanceado e $Q_0 = 10I_{25}$. São adotados os critérios de parada do algoritmo cone-elipsoidal, $\epsilon = 0,01$ e $N_\epsilon = 10$ e do algoritmo BnB, $\varepsilon = 0,01$. No caso do modelo

reduzido fixo, após 8 minutos e 46 segundos de processamento é obtido:

$$G_r(s) = \left[\begin{array}{cc|ccc} -1,1062 & -1,3692 & -2,4644 & 0,9103 & 2,5686 \\ 0,6334 & -0,4245 & 0,1724 & -0,5937 & 0,8357 \\ \hline 2,4290 & 0,0377 & 3,1647 & 0,0607 & -0,4967 \\ 2,7042 & 1,0050 & 0,1718 & 0,4603 & -0,6666 \\ 1,2101 & 0,0083 & 0,3425 & -0,2217 & 0,4432 \end{array} \right]$$

que resulta em $\gamma_c = 3,24$, inferior aos resultados obtidos com as formulações baseadas em LMIs, mesmo comparado com os modelos reduzidos politópicos. A Figura 8.1 apresenta o diagrama de valores singulares máximos de $E(s, p)$ para os quatro vértices do politopo.

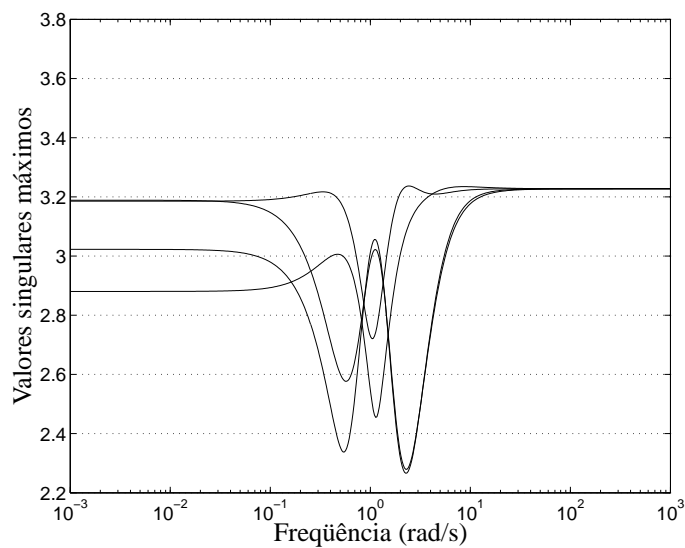


Figura 8.1: Diagramas de valores singulares máximos de $E(s, p)$ do modelo reduzido fixo para os vértices do politopo (Ex. 8.1).

No caso do modelo reduzido dependente de parâmetros, para $Q_0 = 50I_{75}$, após 24 minutos e 39 segundos de processamento é obtido:

$$G_r(s, p) = \left[\begin{array}{cc|ccc} -0,7961 & -1,0070 & -3,3290 & 1,2579 & 2,8928 \\ 0,6251 & -0,1640 & 0,5037 & -0,5529 & 0,7351 \\ \hline 3,3810 & 0,4994 & 4,8763 & -0,0235 & -0,7463 \\ 2,7028 & 1,0227 & -0,3917 & 0,5720 & -0,7653 \\ 1,4950 & 0,2076 & 0,9797 & -0,3661 & 0,6494 \end{array} \right]$$

$$\begin{aligned}
 & + a_{33} \left[\begin{array}{cc|ccc} 0,0540 & 0,0728 & -0,4728 & 0,0390 & 0,2890 \\ 0,0116 & 0,1255 & 0,2360 & -0,0718 & -0,0836 \\ \hline 0,3006 & 0,1822 & 0,7158 & -0,0568 & -0,0849 \\ -0,1343 & -0,0159 & -0,2320 & 0,1559 & -0,1008 \\ 0,0681 & 0,0599 & 0,2629 & -0,0485 & 0,0364 \end{array} \right] \\
 & + b_{12} \left[\begin{array}{cc|ccc} -0,0377 & 0,0452 & -0,0150 & -0,1038 & 0,0063 \\ -0,0058 & 0,0479 & -0,0037 & -0,0140 & -0,0008 \\ \hline -0,0947 & -0,0075 & -0,0068 & -0,0649 & -0,0315 \\ -0,0016 & 0,0388 & -0,0327 & -0,0333 & -0,0094 \\ -0,0343 & 0,0078 & 0,0209 & -0,0180 & -0,0051 \end{array} \right]
 \end{aligned}$$

que resulta em $\gamma_c = 3,19$, que é apenas 1,6% maior do que o limite inferior estabelecido pelo Teorema 2 apresentado em (Wu, 1996) e (Wu e Jaramillo, 2002), e 13,0% menor do que o melhor resultado relatado em (Wu e Jaramillo, 2002). A Figura 8.2 apresenta o diagrama de valores singulares de $E(s, p)$ para os quatro vértices do politopo que pode ser comparado com as Figuras 1 e 2 em Wu e Jaramillo (2002).

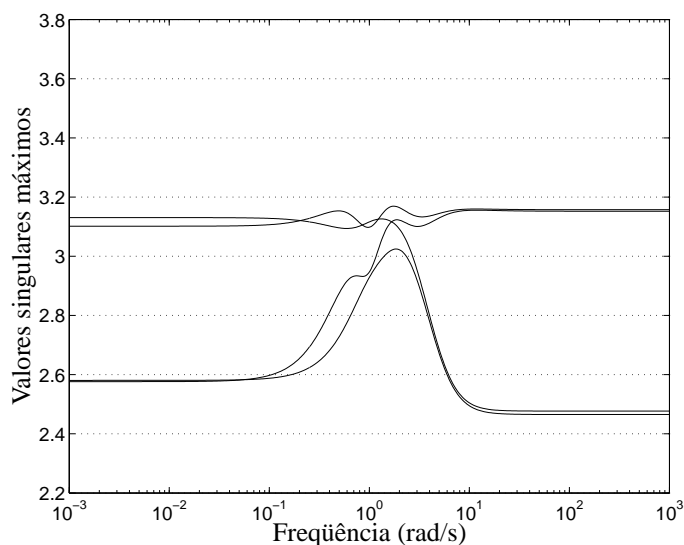


Figura 8.2: Diagramas de valores singulares máximos de $E(s, p)$ do modelo reduzido dependente de parâmetros para os vértices do politopo (Ex. 8.1).

Exemplo 8.2 Considere o seguinte sistema incerto, linear invariante no tempo, a tempo discreto,

apresentado em Assunção e Peres (1999):

$$G(z, p) = \left[\begin{array}{ccc|c} 0,150 & 0,340 + p_1 & 0,0315 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 1 & -0,2 + p_2 & 0 \\ 0,50 & 0,80 & 0 & 0 \end{array} \right]$$

com $p = [p_1 \ p_2]^T \in \Omega_p$, sendo $\Omega_p = \{p \in \mathbb{R}^2 : |p_1| \leq 0,2, |p_2| \leq 0,1\}$.

Os valores singulares de Hankel do sistema nominal ($p_1 = p_2 = 0$) são $\sigma_1 = 2,0241$, $\sigma_2 = 1,2414$ e $\sigma_3 = 0,1576$. Considere o problema de reduzir o modelo incerto de ordem 3 para um modelo de ordem 2 minimizando a norma \mathcal{H}_∞ para $p \in \Omega_p$. Com o algoritmo B apresentado em Assunção e Peres (1999), o valor de $\gamma_{c.g.}$ é reduzido de 2,4944 (redução balanceada) para 1,7616 (4,9dBs). Com o procedimento de projeto proposto, após 18s de processamento, é obtido o seguinte modelo reduzido fixo

$$G_r(z) = \left[\begin{array}{cc|c} 0,7249 & -0,2342 & -1,0576 \\ 0,3352 & -0,7032 & 0,6546 \\ \hline -0,5161 & -0,6504 & 0,0832 \\ -0,8833 & -0,3449 & 0,0239 \end{array} \right]$$

que resulta em $\gamma_c = 1,76$, que é similar ao anterior. Projetando um modelo reduzido com dependência afim de parâmetros, é obtido o seguinte modelo, após 9 minutos e 46 segundos de processamento:

$$G_r(z, p) = \left[\begin{array}{cc|c} 0,7651 & -0,2780 & -0,9028 \\ 0,4187 & -0,7148 & 0,6217 \\ \hline -0,4284 & -0,7022 & -0,0065 \\ -0,7558 & -0,2389 & 0,0044 \end{array} \right]$$

$$+ \Delta \left[\begin{array}{cc|c} 0,5747 & 0,2868 & 0,0952 \\ -0,2406 & -0,3133 & -0,0185 \\ \hline 0,2025 & 0,0244 & 0,0142 \\ -0,0248 & 0,0516 & -0,0269 \end{array} \right]$$

$$+ \delta \left[\begin{array}{cc|c} 0,0649 & 0,0085 & -0,0261 \\ 0,0091 & 0,1376 & -0,0666 \\ \hline -0,1344 & 0,0456 & 0,0142 \\ 0,0467 & 0,0252 & -0,0134 \end{array} \right]$$

que resulta em $\gamma_c = 0,26$ ($-11,8\text{dBs}$), que é 85% menor do que o valor obtido pelos modelos reduzidos fixos. A Figura 8.3 compara os diagramas de valores singulares de $E(z, p)$ obtidos com o modelo reduzido fixo e o modelo reduzido dependente de parâmetros, para os quatro vértices do politopo. É verificado que o modelo reduzido fixo pode amplificar o efeito da entrada sobre o erro para determinada faixa de frequências ao passo que o modelo reduzido dependente de parâmetros sempre atenua o efeito da entrada em todas as faixas de frequência e para todo $p \in \Omega_p$.

A Figura 8.4 mostra as respostas ao impulso do modelo original, modelo fixo de segunda ordem (Assunção e Peres, 1999), e o modelo incerto de segunda ordem, para $p = [0,2 \ 0,1]^T$, que mostra a melhor aproximação do modelo reduzido incerto comparado ao fixo.

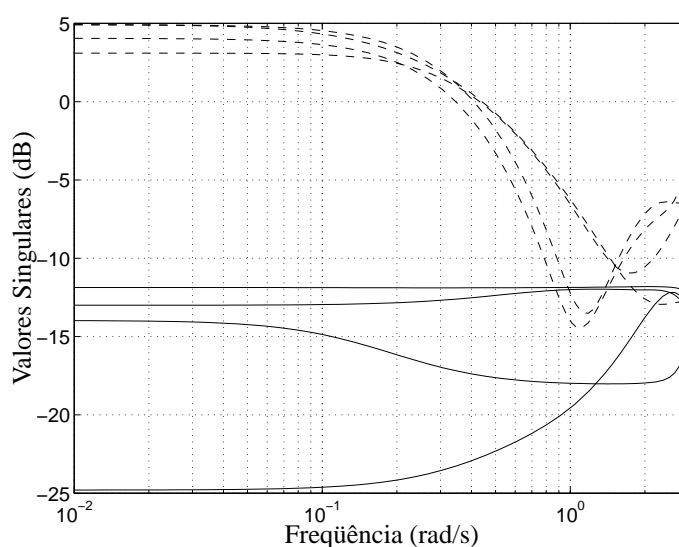


Figura 8.3: Diagramas de valores singulares de $E(z, p)$ com o modelo reduzido fixo (tracejado) e modelo reduzido incerto (linha sólida) para os vértices do politopo (Ex. 8.2).

Os resultados obtidos com uma única iteração do procedimento proposto utilizam como condições iniciais o modelo reduzido obtido pelo truncamento do modelo balanceado e $Q_0 = I_{12}$. São adotados os critérios de parada do algoritmo cone-elipsoidal, $\epsilon = 0,01$ e $N_\epsilon = 10$ e do algoritmo BnB, $\epsilon = 0,01$.

Exemplo 8.3 Considere o sistema discreto no tempo de 8ª ordem, com incerteza politópica, $G(z) = \alpha_1 G_1(z) + \alpha_2 G_2(z)$, $\alpha = [\alpha_1 \ \alpha_2]^T \in \Omega$, sendo $A_1 = \text{diag}(0,5, 0,55, 0,6, \dots, 0,85)$, A_2 igual a uma

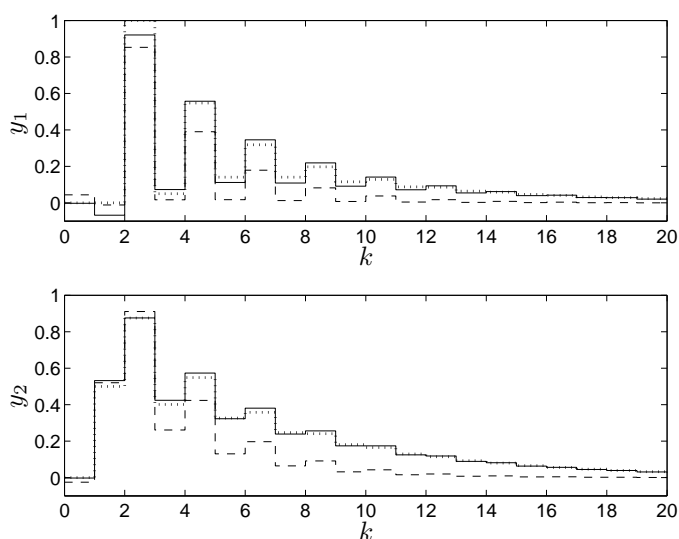


Figura 8.4: Respostas ao impulso do modelo original (linha pontilhada), modelo fixo de ordem reduzida (linha tracejada), e modelo incerto de ordem reduzida (linha sólida) para $p = [0,2 \ 0,1]^T$ (Ex. 8.2).

matriz bloco diagonal:

$$A_2 = A_1 + \begin{bmatrix} 0 & 0,2 & 0 & \dots & 0 \\ 0,2 & 0 & & \ddots & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & \ddots & & 0 & 0,2 \\ 0 & \dots & 0 & 0,2 & 0 \end{bmatrix}$$

e as demais matrizes são fixas:

$$B_1 = B_2 = \begin{bmatrix} 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & \dots & 0 & 1 \end{bmatrix}^T,$$

$$C_1 = C_2 = [1 \ \dots \ 1] \text{ e } D_1 = D_2 = [0 \ 0].$$

O modelo “nominal”, dado por $\alpha = [0,5 \ 0,5]^T$, apresenta os seguintes quatro maiores valores singulares de Hankel: 11,5543; 2,9482; 0,2377 e 0,0709. Estes valores indicam que modelos reduzidos de 1^a ou 2^a ordem produzem uma boa aproximação. Neste exemplo são consideradas aproximações de 2^a ordem e minimização da norma \mathcal{H}_∞ do erro de redução para $\alpha \in \Omega$.

O modelo reduzido balanceado considerando o modelo “nominal” gera um valor baixo de norma, $\|E(z)\|_\infty = 0,4441$, porém, considerando todo o politopo é obtido $\gamma_{p.c.} = 7,0637$. Para o procedimento proposto, são adotados como condições iniciais o modelo reduzido obtido pelo truncamento do modelo balanceado e $Q_0 = I_{12}$ e como critérios de parada do algoritmo cone-elipsoidal, $\epsilon = 0,01$

e $N_\epsilon = 5$, e do algoritmo BnB, $\epsilon = 0,01$. Após 22s de processamento, é obtido o seguinte modelo reduzido fixo

$$G_r(z) = \left[\begin{array}{cc|cc} 0,6985 & 0,2777 & -1,2923 & -1,9760 \\ -0,0698 & 0,5281 & 0,9764 & -0,4889 \\ -2,3635 & 0,8454 & 0,1782 & 0,0374 \end{array} \right]$$

que resulta em $\gamma_{p.c.} = 5,79$ que é similar ao valor 5,82 obtido pelo algoritmo B em Assunção e Peres (1999). Projetando um modelo reduzido politópico, é obtido, após 14m58s de processamento, o modelo reduzido $G_r(z, \alpha) = \alpha_1 G_{r1} + \alpha_2 G_{r2}$, sendo

$$G_{r1} = \left[\begin{array}{cc|cc} 0,7466 & 0,0746 & -1,3406 & -1,8109 \\ 0,0796 & 0,7758 & 1,0135 & -0,3597 \\ -2,2779 & 0,7934 & 0,1189 & 0,0584 \end{array} \right]$$

$$G_{r2} = \left[\begin{array}{cc|cc} 0,7153 & 0,3622 & -1,2202 & -1,9234 \\ -0,0876 & 0,7616 & 1,0298 & -0,6619 \\ -2,3653 & 0,7753 & 0,1270 & -0,0154 \end{array} \right]$$

que resulta em $\gamma_{p.c.} = 0,47$, que é 92% menor que o resultado obtido com o modelo reduzido fixo calculado com o algoritmo B em Assunção e Peres (1999).

Este exemplo é interessante porque ele requer duas iterações do procedimento de projeto. Na 1ª iteração, o cálculo do modelo reduzido pelo procedimento de otimização, considerando apenas os dois vértices, $\tilde{\Omega} = \{[1 \ 0]^T; [0 \ 1]^T\}$, resulta em uma variação da norma \mathcal{H}_∞ em função de α_1 (com $\alpha_2 = 1 - \alpha_1$) tal que o valor máximo ocorre para $\alpha = [0,5625 \ 0,4375]^T$, fora dos vértices, como indicado na Figura 8.5 (curva tracejada). Acrescentado este ponto ao conjunto finito $\tilde{\Omega}$ e repetindo a otimização, a curva da norma \mathcal{H}_∞ é suavizada, como mostrado na Figura 8.5 (curva sólida), tal que a diferença entre os valores máximos para $\alpha \in \Omega$ e para $\alpha \in \tilde{\Omega}$ não é significativa (de acordo com o critério de parada, $\epsilon_\delta = 0,1$) não sendo necessária nova iteração.

8.6 Conclusões

O procedimento geral de projeto foi aplicado ao problema de redução de modelos de sistemas incertos lineares invariantes no tempo estáveis. Uma vantagem importante do procedimento proposto é sua capacidade de obter modelos reduzidos fixos ou que reproduzem a estrutura do modelo incerto original, seja ele modelo politópico ou com dependência afim de parâmetros. Como outras formulações não convexas, não existe garantia de convergência para o mínimo global da norma do erro de redução. Entretanto, os vários testes já realizados mostram que o procedimento proposto obtém na

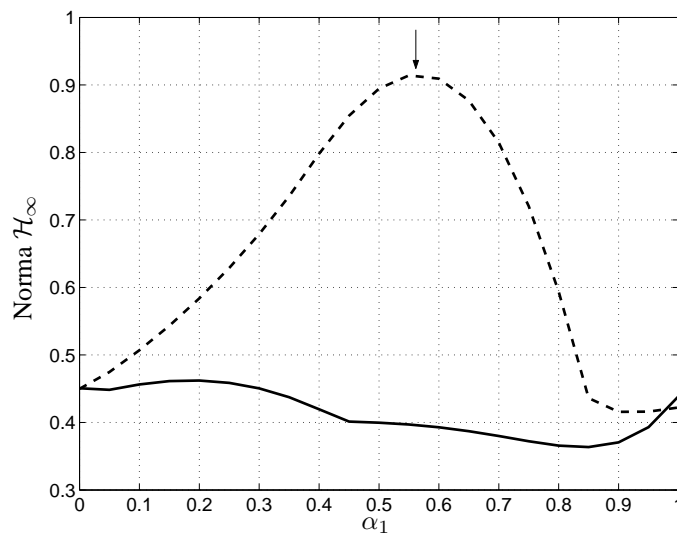


Figura 8.5: Norma \mathcal{H}_∞ do erro de redução de $G_r(z, \alpha)$ na primeira (tracejada) e segunda (sólida) iterações do procedimento proposto (Ex. 8.3).

prática resultados bastante satisfatórios que podem compensar em determinadas situações o maior tempo requerido. O problema de cálculo de modelos reduzidos é um problema mais simples que os tratados nos capítulos anteriores e que, a princípio, pode ser tratado de forma eficiente com procedimentos baseados em formulações LMI, iterativos ou não, que podem ser facilmente estendidos para o caso de obtenção de modelos reduzidos politópicos. Apesar disto, o procedimento de aproximação por modelos reduzidos proposto ainda pode ser interessante nos casos em que as formulações LMI não são factíveis ou não resultem no desempenho desejado.

Capítulo 9

Conclusões Finais

9.1 Sumário das contribuições da tese

Esta tese apresenta as seguintes contribuições na área de Geometria Computacional e Controle Robusto:

- **procedimento de divisão de simplexo** orientada pelas arestas em k^d , $k \geq 2$, para espaços de dimensão d qualquer, de fácil implementação, baseado em um modelo de simplexo existente na literatura, que reproduz os resultados já conhecidos limitados aos espaços de duas e três dimensões (Gonçalves, Palhares, Takahashi e Mesquita, 2006a);
- **procedimento de análise de \mathcal{D} -estabilidade robusta** para sistemas incertos lineares, invariantes no tempo, com domínio politópico de incerteza, que informa se o sistema incerto é robustamente \mathcal{D} -estável ou, em caso contrário, localiza um sistema no domínio de incerteza que não é \mathcal{D} -estável (Gonçalves, Palhares, Takahashi e Mesquita, 2006d; Gonçalves, Palhares, Takahashi e Mesquita, 2006b);
- **procedimento de análise de desempenho robusto** baseado no algoritmo branch-and-bound, considerando formulações LMI para cálculo de custos garantidos como funções limitantes superiores, que calcula os custos \mathcal{H}_2 ou \mathcal{H}_∞ de sistemas incertos lineares invariantes no tempo, com domínio politópico de incerteza, com qualquer precisão requerida (Gonçalves, Palhares, Takahashi e Mesquita, 2004; Gonçalves, Campos, Ekel, Palhares, Takahashi e Mesquita, 2004; Gonçalves, Bastos, Palhares, Takahashi, Mesquita, Campos e Ekel, 2005; Gonçalves, Palhares, Takahashi e Mesquita, 2006c);
- **procedimento geral de projeto de controladores, filtros ou modelos reduzidos**, considerando especificações de custo \mathcal{H}_2 , \mathcal{H}_∞ e posicionamento regional de pólos, para sistemas line-

ares invariantes no tempo com domínios politópicos de incerteza, considerando diretamente os elementos das matrizes do controlador, filtro ou modelo reduzido como parâmetros de otimização (Gonçalves, Palhares e Takahashi, 2004a; Gonçalves, Palhares e Takahashi, 2004b; Gonçalves, Palhares e Takahashi, 2005a; Gonçalves, Palhares e Takahashi, 2004; Gonçalves, Palhares e Takahashi, 2005b; Gonçalves, Palhares e Takahashi, 2006b; Gonçalves, Palhares e Takahashi, 2006c; Gonçalves, Palhares e Takahashi, 2006a; Gonçalves, Chasin, Palhares, Takahashi e Mesquita, 2006).

O método simples e eficiente de partição de politopos de qualquer dimensão, apresentado no Capítulo 2, combina a triangularização de Delaunay, para a decomposição do politopo em uma malha de simplexos, e a técnica proposta de divisão de simplexo orientada pelas arestas. Este procedimento é bastante útil para vários problemas na área de otimização, em especial, para os algoritmos de otimização global tipo branch-and-bound que são considerados nas estratégias de análise propostas. Além da simplicidade de implementação, o método proposto também possui a vantagem de trabalhar com malha de simplexos, que são os politopos com menor número de vértices em uma determinada dimensão. Esta característica possui influência direta sobre a eficiência dos procedimentos de análise propostos nesta tese. Outra vantagem de extrema importância do método de partição de politopos é a aplicação da técnica de divisão de simplexo orientada pelas arestas que produz o mínimo número de classes congruentes, evitando a geração de simplexos degenerados que podem comprometer a convergência do algoritmo branch-and-bound.

O procedimento de análise de \mathcal{D} -estabilidade robusta, apresentado em detalhes no Capítulo 3, combina a redução do conservadorismo de condições suficientes LMI pela partição do domínio de incerteza com um método de grade (discretização do domínio de incerteza). O método de grade é eficiente devido a eliminação de regiões do domínio de incerteza a partir dos resultados obtidos pelas condições suficientes LMI. O procedimento proposto funciona, na prática, como uma condição necessária e suficiente se não for considerado o tempo computacional requerido, limitado pelo número máximo de iterações (ou partições) permitido. Foram mostradas, através de exemplos, a eficácia e a eficiência do procedimento proposto, que identificou a estabilidade de 100% dos casos analisados e requereu, em média, menores tempos computacionais do que as formulações LMI mais eficazes, porém mais complexas.

A principal vantagem do procedimento proposto para o cômputo dos custos garantidos \mathcal{H}_2 ou \mathcal{H}_∞ , apresentado no Capítulo 4, é a capacidade de cálculo com a precisão requerida, o que não é possível com formulações LMI isoladas. A princípio, são esperados altos tempos computacionais com a aplicação do algoritmo branch-and-bound, sendo necessário, nos casos mais complexos, estabelecer um compromisso entre precisão e custo computacional. Apesar disto, é observado em vários exemplos que a combinação do algoritmo branch-and-bound com formulações LMI mais simples pode requerer

menores tempos computacionais do que formulações LMI mais complexas isoladas. Similar ao caso de análise de \mathcal{D} estabilidade, tal eficiência é em parte justificada pela adoção da malha simplicial na implementação do algoritmo branch-and-bound. Uma segunda vantagem do procedimento proposto é a sua capacidade de localizar as coordenadas do sistema com pior caso de norma, informação que é útil para o procedimento geral de projeto proposto. O procedimento de análise de desempenho robusto proposto é uma ferramenta útil para aferir o conservadorismo das formulações LMI e para comparar sistemas projetados por diferentes estratégias de síntese.

O procedimento geral de projeto, apresentado no Capítulo 5, combina uma etapa de síntese (otimização no espaço de parâmetros do controlador, filtro ou modelo reduzido) e uma de análise (análise robusta de estabilidade e desempenho com localização dos pontos de pior caso), transformando o problema de otimização do pior caso de um número infinito de sistemas em um domínio de incerteza em um problema com um conjunto finito de pontos. O conjunto finito de pontos considerados na etapa de síntese é aumentado gradativamente de acordo com a necessidade avaliada na etapa de análise. Apesar do procedimento de projeto ser baseado em um problema de otimização não convexo, sem garantia de convergência para o ótimo global, através de vários exemplos, apresentados nos capítulos 6 e 7, foi verificado que é possível projetar controladores robustos por realimentação de estado, realimentação de saída estática e realimentação de saída dinâmica de ordem fixa, admitindo restrições de estrutura, inclusive controladores PID, e filtros robustos de ordem fixa com desempenhos iguais ou superiores em relação aos projetos baseados em formulações LMIs ou BMIs. No Capítulo 8, o procedimento geral de projeto também é aplicado à aproximação de sistemas incertos por modelos reduzidos fixos ou com a mesma estrutura do modelo incerto.

9.2 Comentários finais e perspectivas futuras

Apesar do procedimento geral de projeto apresentar desempenho satisfatório em todos os exemplos analisados, tendo evoluído desde a primeira versão publicada, ainda existe bastante campo de pesquisa para o aprimoramento do passo de projeto do controlador ou filtro, pelo algoritmo de otimização cone-elipsoidal, e do passo de validação pela combinação do algoritmo branch-and-bound com as formulações de análise baseadas em LMIs.

Com relação ao algoritmo de otimização cone-elipsoidal, como foi verificado, o mesmo tem apresentado bastante eficiência, tanto na obtenção de soluções factíveis como na obtenção dos ótimos locais, mas ainda pode ser aperfeiçoado no quesito custo computacional, que é relevante quando o número de parâmetros de otimização é elevado, como é comum em projetos de controladores por realimentação dinâmica de saída de ordem completa ou filtros de ordem completa.

O aspecto mais importante, ao se aplicar o procedimento geral de projeto em um sistema em

que o espaço de incertezas é de ordem maior ou igual a três, com número de vértices elevado, é o passo de validação do controlador, filtro ou modelo reduzido pelo algoritmo branch-and-bound. A aplicação deste algoritmo no problema em análise demanda conhecimentos nas áreas de otimização, geometria computacional e controle robusto. O mais importante obstáculo enfrentado para utilização deste algoritmo na análise de sistemas robustos com incertezas politópicas foi encontrar uma técnica simples e eficiente de partição do espaço. Com o desenvolvimento do algoritmo de divisão orientado pelas arestas, esta etapa do trabalho pode ser considerada finalizada. Entretanto, o conservadorismo das formulações de cálculo do custo garantido, utilizado como função limitante superior, e das formulações de análise de posicionamento regional de pólos, ainda é um problema a ser trabalhado. As formulações baseadas em estabilidade quadrática, com função de Lyapunov fixa, exigem pequeno tempo de processamento por iteração mas demandam número elevado de iterações para convergência. Por outro lado, as formulações de análise baseadas em funções de Lyapunov dependentes de parâmetros produzem resultados menos conservadores, necessitando menos iterações, mas com tempo de processamento por iteração elevado em função do número maior de parâmetros de otimização. Infelizmente, não foi observada, nos vários exemplos analisados, uma formulação que possa ser considerada a mais adequada em qualquer situação. No caso do custo \mathcal{H}_2 , mesmo para uma mesma formulação, as versões primal e dual apresentam comportamentos diferentes de acordo com o sistema analisado.

Apesar do tempo de processamento elevado em algumas situações, o procedimento de análise proposto baseado no algoritmo branch-and-bound pode ser considerado, independente da aplicação no procedimento de projeto proposto, uma importante ferramenta de análise para validação e comparação de desempenho das formulações de análise e projeto baseadas em LMIs. Além disso, a técnica desenvolvida de partição de espaço pode ser aplicada em conjunto com o algoritmo branch-and-bound em várias outras situações, não limitadas à área de controle.

Do ponto de vista da aplicação do procedimento geral de projeto proposto, outras possibilidades a serem pesquisadas incluem:

- Projeto de controladores e filtros robustos aplicados a sistemas incertos com retardo no tempo (Palhares et al., 2001; Palhares et al., 2005).
- Projeto de controladores e filtros dependentes de parâmetros (Leite e Peres, 2002; Gao, Lam, Shi e Wang, 2005; Gao, Lam, Xie e Wang, 2005).
- Inclusão de novos tipos de restrições, considerando, por exemplo, a questão da saturação do controlador (Gomes da Silva Jr. et al., 2003).
- Controle por realimentação da derivada do estado (Abdelaziz e Valášek, 2004).

- Projeto aplicado a sistemas com incertezas não-estruturadas.
- Projeto de filtros robustos para detecção de falhas (Nobrega et al., 2000; Zhong et al., 2003; Casavola et al., 2005).
- Projeto de controladores baseados em modelos de referência considerando critérios de otimalidade \mathcal{H}_2 e \mathcal{H}_∞ .

Com objetivo de aperfeiçoar o desempenho do procedimento geral de projeto, é importante pesquisar a aplicação de:

- outras técnicas disponíveis de cálculo dos elipsóides que podem proporcionar uma convergência mais rápida do método de otimização baseado no algoritmo elipsoidal;
- outras possibilidades de critério de parada no método de otimização para evitar iterações desnecessárias do algoritmo, sem redução relevante da função objetivo;
- outros métodos de otimização para determinação dos parâmetros do controlador, filtro ou modelo reduzido, considerando inclusive algoritmos de otimização multiobjetivos;
- outras formulações de análise de custo garantido ou posicionamento regional de pólos, já publicadas ou sendo desenvolvidas, que poderão proporcionar um desempenho melhor do procedimento de análise proposto.

Referências Bibliográficas

- Abdelaziz, T. H. S. e Valášek, M. (2004). Pole-placement for SISO linear systems by state-derivative feedback, *IEE Proceedings Control Theory & Applications* **151**(4): 377–385.
- Apkarian, P., Tuan, H. D. e Bernussou, J. (2001). Continuous-time analysis, eigenstructure assignment and \mathcal{H}_2 synthesis with enhanced LMI characterizations, *IEEE Transactions on Automatic Control* **46**(12): 1941–1946.
- Arzelier, D. e Peaucelle, D. (2002). An iterative method for mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis via static output-feedback, *Proceedings of the 41st IEEE Conference on Decision and Control*, IEEE, Las Vegas, Nevada, USA, pp. 3464–3469.
- Arzelier, D., Peaucelle, D. e Salhi, S. (2003). Robust static output feedback stabilization for polytopic uncertain systems: improving the guaranteed performance bound, *Proceedings of the 4th IFAC Symposium on Robust Control Design*, IFAC, Milan, Italy.
- Assunção, E. e Peres, P. L. D. (1999). \mathcal{H}_2 and/or \mathcal{H}_∞ -norm model reduction of uncertain discrete-time systems, *Proceedings of the American Control Conference*, AACC, San Diego, California, pp. 4466–4470.
- Åström e Hägglund, T. (2001). The future of PID control, *Control Engineering Practice* **9**: 1163–1175.
- Åström e Hägglund, T. (2004). Revisiting the Ziegler-Nichols step response method for PID control, *Journal of Process Control* **14**: 635–650.
- Balakrishnan, V. e Boyd, S. (1991). Computation of the worst-case covariance for linear systems with uncertain parameters, *Proceedings of the IEEE Conference on Decision and Control*, IEEE, Brighton, U.K, pp. 1941–1942.
- Balakrishnan, V., Boyd, S. e Balemi, S. (1991). Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems, *International Journal of Robust and Nonlinear Control* **1**(4): 295–317.

- Balemi, S. e Balakrishnan, V. (1992). Branch and bound algorithm for global optimization in control theory, *In Proceedings of IFAC Workshop on Mutual Impact of Computing Power and Control Theory*, IFAC, Plenum Publishing Corporation, Prague, Czech Republic, pp. 169–179.
- Balemi, S., Boyd, S. e Balakrishnan, V. (1991). Computation of the maximum \mathcal{H}_∞ -norm of parameter-dependent linear systems by a branch and bound algorithm, *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems*, Kobe, Japan, pp. 305–310.
- Barber, C. B. (1996). The quickhull algorithm for convex hulls, *ACM Transactions on Mathematical Software* **22**(4): 469–483.
- Barbosa, K. A., de Souza, C. E. e Trofino, A. (2005). Robust \mathcal{H}_2 filtering for uncertain linear systems: LMI based methods with parametric Lyapunov functions, *System & Control Letters* **54**: 251–262.
- Beck, C. L., Doyle, J. e Glover, K. (1996). Model reduction of multidimensional and uncertain systems, *IEEE Transactions on Automatic Control* **41**(10): 1466–1477.
- Beran, E. (1997). *Methods for Optimization-based Fixed-Order Control Design*, PhD Thesis, Technical University of Denmark.
- Beran, E., Vandenberghe, L. e Boyd, S. (1997). A global BMI algorithm based on the generalized Benders decomposition, *Proceedings of the European Control Conference*, Brussels, Belgium.
- Bernussou, J., Geromel, J. C. e Korogui, R. H. (2005). On robust output feedback control for polytopic systems, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, IEEE, Seville, Spain, pp. 5018–5023.
- Bernussou, J., Peres, P. L. D. e Geromel, J. C. (1989). A linear programming oriented procedure for quadratic stabilization of uncertain systems, *Systems & Control Letters* **13**: 65–72.
- Bey, J. (2000). Simplicial grid refinement: on Freudenthal’s algorithm and the optimal number of congruence classes, *Numerische Mathematik* **85**: 1–29.
- Boyd, S., Balakrishnan, K. e Kabamba, P. (1989). A bisection method for computing the \mathcal{H}_∞ norm of a transfer matrix and related problems, *Mathematics of Control, Signals, and Systems* **2**(3): 207–219.
- Boyd, S., El Ghaoui, L., Feron, E. e Balakrishnan, V. (1994). *Linear Matrix Inequalities in Systems and Control Theory*, SIAM, Philadelphia, PA.

- Brewer, J. W. (1978). Kronecker products and matrix calculus in system theory, *IEEE Transactions on Circuits and Systems* **CAS-25**(9): 772–781.
- Casavola, A., Famularo, D. e Franzè, G. (2005). A robust deconvolution scheme for fault detection and isolation of uncertain linear systems: an LMI approach, *Automatica* **41**: 1463–1472.
- Chankong, V. e Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology*, Vol. 8 of *System Science and Engineering*, Elsevier North-Holland Publishing Co, New York.
- Chesi, G. (2005). Establishing stability and instability of matrix hypercubes, *Systems & Control Letters* **54**: 381–388.
- Chesi, G., Garulli, A., Tesi, A. e Vicino, A. (2005a). Polynomially parameter-dependent Lyapunov functions for robust \mathcal{H}_∞ performance analysis, in P. Horacek, M. Simandl e P. Zitek (eds), *Preprints of the 16th IFAC World Congress*, IFAC, Prague, Czech Republic.
- Chesi, G., Garulli, A., Tesi, A. e Vicino, A. (2005b). Polynomially parameter-dependent lyapunov functions for robust stability of polytopic systems: An LMI approach, *IEEE Transactions on Automatic Control* **50**(3): 365–370.
- Chilali, M. e Gahinet, P. (1996). \mathcal{H}_∞ design with pole placement constraints: An LMI approach, *IEEE Transactions on Automatic Control* **41**(3): 358–367.
- Chilali, M. e Gahinet, P. (1999). Robust pole placement in LMI regions, *IEEE Transactions on Automatic Control* **44**(12): 2257–2270.
- Clausen, J. e Zilinskas, A. (2002). Subdivision, sampling, and initialization strategies for simplicial branch and bound in global optimization, *Computers and Mathematics with Applications* **44**: 943–955.
- Cominos, P. e Munro, N. (2002). PID controllers: recent tuning methods and design to specification, *IEE Proceedings Control Theory & Applications* **149**(1): 46–53.
- Coutinho, M., Coutinho, D. F., Trofino, A. e Barbosa, K. A. (2005). A new strategy to the multi-objective control of linear systems, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, IEEE, Seville, Spain, pp. 3742–3746.
- de Araújo, H. X. e Neumann, D. (2004). Controle robusto $\mathcal{H}_2/\mathcal{H}_\infty$: Síntese via algoritmos genéticos, *Anais do XV Congresso Brasileiro de Automática*, SBA, Gramado, RS, Brasil.

- de Berg, M., Kreveld, M. V., Overmars, M. e Schwarzkopt, O. (2000). *Computational Geometry: Algorithms and Applications*, 2 edn, Springer-Verlag, Berlin.
- de Oliveira, M. C. (2004). Investigating duality on stability conditions, *Systems & Control Letters* **52**: 1–6.
- de Oliveira, M. C., Bernussou, J. e Geromel, J. C. (1999). A new discrete-time robust stability condition, *System & Control Letters* **37**(4): 261–265.
- de Oliveira, M. C. e Geromel, J. C. (1997). Numerical comparison of output feedback design methods, *Proceedings of the American Control Conference, AACC*, Albuquerque, New Mexico, USA, pp. 72–76.
- de Oliveira, M. C., Geromel, J. C. e Bernussou, J. (2000). Design of dynamic output feedback decentralized controllers via a separation procedure, *International Journal of Control* **73**(5): 371–381.
- de Oliveira, M. C., Geromel, J. C. e Bernussou, J. (2002). Extended \mathcal{H}_2 and \mathcal{H}_∞ norm characterizations and controller parametrizations for discrete-time systems, *International Journal of Control* **75**(9): 666–679.
- de Oliveira, M. C., Geromel, J. C. e Hsu, L. (1999). LMI characterization of structural and robust stability: the discrete-time case, *Linear Algebra and its Applications* **296**: 27–38.
- de Oliveira, M. C. e Skelton, R. E. (2002). On stability tests for linear systems, *Proceedings of the 15th Triennial World Congress, IFAC*, Barcelona, Spain.
- de Oliveira, P. J., Oliveira, R. C. L. F., Leite, V. J. S., Montagner, V. F. e Peres, P. L. D. (2004a). \mathcal{H}_2 guaranteed cost computation by means of parameter-dependent Lyapunov functions, *International Journal of Systems Science* **35**(5): 305–315.
- de Oliveira, P. J., Oliveira, R. C. L. F., Leite, V. J. S., Montagner, V. F. e Peres, P. L. D. (2004b). \mathcal{H}_∞ guaranteed cost computation by means of parameter-dependent Lyapunov functions, *Automatica* **40**: 1053–1061.
- DeMarco, C. L., Balakrishnan, V. e Boyd, S. (1990). A branch and bound methodology for matrix polytope stability problems arising in power systems, *Proceedings of the 29th IEEE Conference on Decision and Control*, IEEE, Honolulu, Hawaii, pp. 3022–3027.
- Dolgin, Y. e Zeheb, E. (2004). Model reduction of uncertain FIR discrete-time systems, *IEEE Transactions on Circuits and Systems–II* **51**(8): 406–411.

- Doyle, J. C. (1978). Guaranteed margins for LQG regulators, *IEEE Transactions on Automatic Control* **AC-23**(4): 756–757.
- Doyle, J. C., Glover, K., Khargonekar, P. P. e Francis, B. A. (1989). State-space solutions to standard \mathcal{H}_2 and \mathcal{H}_∞ control problems, *IEEE Transactions in Automatic Control* **34**(8): 831–847.
- Doyle, J., Packard, A. e Zhou, K. (1991). Review of LFTs, LMIs, and μ , *Proceedings of the 30th Conference on Decision and Control*, IEEE, Brighton, England, pp. 1227–1232.
- Ebihara, Y. e Hagiwara, T. (2002a). New dilated LMI characterizations for continuous-time control design and robust multiobjective control, *Proceedings of the American Control Conference*, AACC, Anchorage, AK, USA, pp. 47–52.
- Ebihara, Y. e Hagiwara, T. (2002b). Robust controller synthesis with parameter-dependent Lyapunov variables: A dilated LMI approach, *Proceedings of the 41st IEEE Conference on Decision and Control*, IEEE, Las Vegas, Nevada, USA, pp. 4179–4184.
- Ebihara, Y. e Hagiwara, T. (2004a). New dilated LMI characterizations for continuous-time multiobjective controller synthesis, *Automatica* **40**: 2003–2009.
- Ebihara, Y. e Hagiwara, T. (2004b). On \mathcal{H}_∞ model reduction using LMIs, *IEEE Transactions on Automatic Control* **49**(7): 1187–1191.
- Ebihara, Y., Peaucelle, D., Arzelier, D. e Hagiwara, T. (2005). Robust performance analysis of linear time-invariant uncertain systems by taking higher-order time-derivatives of the state, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, IEEE, Seville, Spain, pp. 5030–5035.
- Ebihara, Y., Tokuyama, K. e Hagiwara, T. (2004). Structured controller synthesis using LMI and alternating projection method, *International Journal of Control* **77**(12): 1137–1147.
- Edelsbrunner, H. e Grayson, D. R. (2000). Edgewise subdivision of a simplex, *Discrete & Computational Geometry* **24**: 707–719.
- El Ghaoui, L. e Balakrishnan, V. (1994). Synthesis of fixed-structure controllers via numerical optimization, *Proceedings of the 33rd Conference on Decision and Control*, IEEE, Lake Buena Vista, FL, pp. 2678–2683.
- El Ghaoui, L., Oustry, F. e AitRami, M. (1997). A cone complementarity linearization algorithm for static output-feedback and related problems, *IEEE Transactions on Automatic Control* **42**(8): 1171–1176.

- Gahinet, P., Nemirovski, A., Laub, A. J. e Chilali, M. (1995). *LMI Control Toolbox: For Use with MATLAB[®]*, The MATH WORKS Inc., Natick.
- Gao, H., Lam, J., Shi, P. e Wang, C. (2005). Parameter-dependent filter design with guaranteed \mathcal{H}_∞ performance, *IEE Proceedings Control Theory & Applications* **152**(5): 531–537.
- Gao, H., Lam, J., Xie, L. e Wang, C. (2005). New approach to mixed $\mathcal{H}_2/\mathcal{H}_\infty$ filtering for polytopic discrete-time systems, *IEEE Transactions on Signal Processing* **53**(8): 3183–3192.
- Gao, L. e Xue, A. (2004). On LMI robust \mathcal{D} -stability condition for real convex polytopic uncertainty, *Proceedings of the 5th World Congress on Intelligent Control and Automation*, IEEE, P. R. China, pp. 151–154.
- Ge, M., Chiu, M.-S. e Wang, Q.-G. (2002). Robust PID controller design via LMI approach, *Journal of Process Control* **12**: 3–13.
- Geromel, J. C. (1999). Optimal linear filtering under parameter uncertainty, *IEEE Transactions on Signal Processing* **47**(1): 168–175.
- Geromel, J. C., Bernussou, J., Garcia, G. e de Oliveira, M. C. (1998). \mathcal{H}_2 and \mathcal{H}_∞ robust filtering for discrete-time linear systems, *Proceedings of the 37th IEEE Conference on Decision & Control*, IEEE, Tampa, Florida USA, pp. 632–637.
- Geromel, J. C. e de Oliveira, M. C. (2001). \mathcal{H}_2 and \mathcal{H}_∞ robust filtering for convex bounded uncertain systems, *IEEE Transactions on Automatic Control* **46**(1): 100–107.
- Geromel, J. C., de Oliveira, M. C. e Bernussou, J. (2002). Robust filtering of discrete-time linear systems with parameter dependent Lyapunov functions, *SIAM Journal of Control Optimization* **41**(3): 700–711.
- Geromel, J. C., Egas, R. G. e Kawaoka, F. R. R. (2005). \mathcal{H}_∞ model reduction with application to flexible systems, *IEEE Transactions on Automatic Control* **50**(3): 402–406.
- Geromel, J. C., Kawaoka, F. R. R. e Egas, R. G. (2004). Model reduction of discrete time systems through linear matrix inequalities, *International Journal of Control* **77**(10): 978–984.
- Geromel, J. C., Peres, P. L. D. e Souza, S. R. (1996). Convex analysis of output feedback control problems: Robust stability and performance, *IEEE Transactions on Automatic Control* **41**(7): 997–1003.

- Glover, K. (1984). All optimal Hankel-norm approximations of linear multivariable systems and their error bounds, *International Journal of Control* **39**: 1115–1193.
- Goh, K.-C. (1995). *Robust Control Synthesis via Bilinear Matrix Inequalities*, PhD Thesis, Faculty of the Graduate School, University of Southern California.
- Goh, K.-C., Safonov, M. G. e Papavassilopoulos, G. P. (1995). Global optimization for the biaffine matrix inequality problem, *Journal of Global Optimization* **7**: 365–380.
- Goldfarb, D. e Todd, M. J. (1982). Modifications and implementation of the ellipsoid algorithm for linear programming, *Mathematical Programming* **23**: 1–19.
- Gomes da Silva Jr., J. M., Tarbouriech, S. e Garcia, G. (2003). Local stabilization of linear systems under amplitude and rate saturating actuators, *IEEE Transactions on Automatic Control* **48**(5): 842–847.
- Gonçalves, E. N., Campos, C. D., Ekel, P. Y., Palhares, R. M., Takahashi, R. H. C. e Mesquita, R. C. (2004). Cálculo menos conservador do nível de atenuação de distúrbios \mathcal{H}_∞ de sistemas incertos sujeitos a retardo no tempo, *Anais do XV Congresso Brasileiro de Automática*, SBA, Gramado, RS, Brasil.
- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2004). Síntese de controladores robustos $\mathcal{H}_2/\mathcal{H}_\infty$ por realimentação dinâmica de saída para sistemas com incertezas politópicas, *Anais do XV Congresso Brasileiro de Automática*, SBA, Gramado, RS, Brasil.
- Gonçalves, E. N., Bastos, S. B., Palhares, R. M., Takahashi, R. H. C., Mesquita, R. C., Campos, C. D. e Ekel, P. Y. (2005). \mathcal{H}_∞ guaranteed cost computation for uncertain time-delay systems, *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference 2005*, IEEE, Seville, Spain, pp. 7169–7174.
- Gonçalves, E. N., Chasin, A. N. V., Palhares, R. M., Takahashi, R. H. C. e Mesquita, R. C. (2006). Redução $\mathcal{H}_2/\mathcal{H}_\infty$ de modelos de sistemas incertos: Aproximação por modelos incertos, *Anais do XVI Congresso Brasileiro de Automática*, SBA, Salvador, Brasil. To appear.
- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2004a). Multiobjective optimization applied to robust $\mathcal{H}_2/\mathcal{H}_\infty$ state-feedback control synthesis, *Proceedings of the 2004 American Control Conference*, AACC, Boston, MA, USA, pp. 4619–4624.
- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2004b). Otimização multi-objetivo aplicada a síntese de controladores $\mathcal{H}_2/\mathcal{H}_\infty$ por realimentação de estado, *Anais do XV Congresso Brasileiro de Automática*, SBA, Gramado, RS, Brasil.

- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2005a). Improved optimisation approach to robust $\mathcal{H}_2/\mathcal{H}_\infty$ control problem for linear systems, *IEE Proceedings Control Theory & Applications* **152**(2): 171–176.
- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2005b). Robust $\mathcal{H}_2/\mathcal{H}_\infty$ dynamic output-feedback control synthesis for systems with polytope-bounded uncertainty, in P. Horacek, M. Simandl e P. Zitek (eds), *Preprints of the 16th IFAC World Congress*, IFAC, Prague, Czech Republic.
- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2006a). $\mathcal{H}_2/\mathcal{H}_\infty$ filter design for systems with polytope-bounded uncertainty, *IEEE Transactions on Signal Processing* **54**(9): 3620–3626.
- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2006b). $\mathcal{H}_2/\mathcal{H}_\infty$ robust PID synthesis for uncertain systems, *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, San Diego, CA. To appear.
- Gonçalves, E. N., Palhares, R. M. e Takahashi, R. H. C. (2006c). Síntese de controladores PID $\mathcal{H}_2/\mathcal{H}_\infty$ robustos para sistemas incertos, *Anais do XVI Congresso Brasileiro de Automática*, SBA, Salvador, Brasil. To appear.
- Gonçalves, E. N., Palhares, R. M., Takahashi, R. H. C. e Mesquita, R. C. (2004). Cálculo exato dos custos \mathcal{H}_2 e \mathcal{H}_∞ para sistemas com incertezas politópicas, *Anais do XV Congresso Brasileiro de Automática*, SBA, Gramado, RS, Brasil.
- Gonçalves, E. N., Palhares, R. M., Takahashi, R. H. C. e Mesquita, R. C. (2006a). Algorithm 8xx: SimpleS: An extension of Freudenthal's simplex subdivision, *ACM Transactions on Mathematical Software* . To appear.
- Gonçalves, E. N., Palhares, R. M., Takahashi, R. H. C. e Mesquita, R. C. (2006b). Análise de estabilidade robusta de sistemas discretos no tempo, *Anais do XVI Congresso Brasileiro de Automática*, SBA, Salvador, Brasil. To appear.
- Gonçalves, E. N., Palhares, R. M., Takahashi, R. H. C. e Mesquita, R. C. (2006c). \mathcal{H}_2 and \mathcal{H}_∞ ϵ -guaranteed cost computation of uncertain linear systems, *IEE Proceedings Control Theory & Applications* . To appear.
- Gonçalves, E. N., Palhares, R. M., Takahashi, R. H. C. e Mesquita, R. C. (2006d). New approach to robust \mathcal{D} -stability analysis of linear time-invariant systems with polytope-bounded uncertainty, *IEEE Transactions on Automatic Control* . To appear.

- Goodman, T. N. T. e Peters, J. (1995). Bézier nets, convexity and subdivision on higher dimensional simplices, *Computer Aided Geometric Design* **12**: 53–65.
- Grigoriadis, K. M. (1995). Optimal \mathcal{H}_∞ model reduction via linear matrix inequalities: Continuous- and discrete-time cases, *Systems & Control Letters* **26**: 321–333.
- Grigoriadis, K. M. e Skelton, R. S. (1994). Fixed-order control design for LMI control problems using alternating projection methods, *Proceedings of the 33rd Conference on Decision and Control*, IEEE, Lake Buena Vista, FL, USA, pp. 2003–2008.
- Grigoriadis, K. M. e Skelton, R. S. (1996). Low-order control design for LMI problems using alternating projection methods, *Automatica* **32**(8): 1117–1125.
- Gugercin, S. e Antoulas, A. C. (2000). A comparative study of 7 algorithms for model reduction, *Proceedings of the 39th IEEE Conference on Decision and Control*, IEEE, Sydney, Australia, pp. 2367–2372.
- Gugercin, S. e Antoulas, A. C. (2004). A survey of model reduction by balanced truncation and some new results, *International Journal of Control* **77**(8): 748–766.
- Halevi, Y. e Shaked, U. (2004). Robust robust model reduction, *Proceeding of the 2004 American Control Conference*, AACC, Boston, Massachusetts, pp. 96–101.
- Halevi, Y., Zlochevsky, A. e Gilat, T. (1997). Parameter-dependent model order reduction, *International Journal of Control* **66**(3): 463–485.
- Hang, C. C., Åström, K. J. e Ho, W. K. (1991). Refinements of the Ziegler-Nichols tuning formula, *IEE Proceedings D Control Theory & Applications* **138**(2): 111–118.
- He, Y., Wu, M. e She, J.-H. (2005). Improved bounded-real-lemma representation and \mathcal{H}_∞ control of systems with polytopic uncertainties, *IEEE Transactions on Circuits and Systems-II: Express Briefs* **52**(7): 380–383.
- Helmerson, A. (1994). Model reduction using LMIs, *Proceedings of the 33rd Conference on Decision and Control*, IEEE, Lake Buena Vista, FL, pp. 3217–3222.
- Henrion, D., Arzelier, D., Peaucelle, D. e Lasserre, J.-B. (2004). On parameter-dependent Lyapunov functions for robust stability of linear systems, *Proceedings of the 43rd IEEE Conference on Decision and Control*, IEEE, Atlantis, Paradise Island, Bahamas, pp. 887–892.

- Huang, Y. J. e Wang, Y.-J. (2000). Robust PID tuning strategy for uncertain plants based on the Kharitonov theorem, *ISA Transactions* **39**: 419–431.
- Iwasaki, T. (1999). The dual iteration for fixed-order control, *IEEE Transactions on Automatic Control* **44**(4): 783–788.
- Iwasaki, T. e Skelton, R. E. (1994). All controllers for the general \mathcal{H}_∞ control problem: LMI existence conditions and state space formulas, *Automatica* **30**(8): 1307–1317.
- Iwasaki, T. e Skelton, R. R. (1995). The XY-centering algorithm for the dual LMI problem: A new approach to fixed order control design, *International Journal of Control* **62**(6): 1257–1272.
- Jin, S. H. e Park, J. B. (2001). Robust \mathcal{H}_∞ filtering for polytopic uncertain systems via convex optimisation, *IEE Proceedings Control Theory & Applications* **148**(1): 55–59.
- Kanev, S., Scherer, C., Verhaegen, M. e de Schutter, B. (2004). Robust output-feedback controller design via local BMI optimization, *Automatica* **40**: 1115–1127.
- Kanno, M. (2005). \mathcal{H}_2 model reduction using LMIs, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, IEEE, Seville, Spain, pp. 5887–5892.
- Kau, S.-W., Liu, Y.-S., Hong, L., Lee, C.-H., Fang, C.-H. e Lee, L. (2005). A new LMI condition for robust stability of discrete-time uncertain systems, *Systems & Control Letters* **54**: 1195–1203.
- Keel, L. H., Bhattacharyya, S. P. e Howze, J. W. (1988). Robust control with structured perturbations, *IEEE Transactions on Automatic Control* **33**(1): 68–78.
- Kim, S.-J., Moon, Y.-H., Kwon, S. e Kim, K. (2005). Rank-constrained LMI approach to mixed $\mathcal{H}_2/\mathcal{H}_\infty$ static output feedback controllers, in P. Horacek, M. Simandl e P. Zitek (eds), *Preprints of the 16th IFAC World Congress*, IFAC, Prague, Czech Republic.
- Leibfritz, F. (2001). An LMI-based algorithm for designing suboptimal static $\mathcal{H}_2/\mathcal{H}_\infty$ output feedback controllers, *SIAM Journal of Control and Optimization* **39**: 1711–1735.
- Leibfritz, F. e Lipinski, W. (2003). Description of the benchmark examples in COMPl_eib 1.0, *Tech. report*, University of Trier. URL: <http://www.compleib.de>.
- Leite, V. J. S. e Peres, P. L. D. (2002). Robust pole location for an active suspension quarter-car model through parameter dependent control, *Proceedings of the 2002 IEEE International Conference on Control Applications*, IEEE, Glasgow, Scotland, U.K., pp. 447–452.

- Leite, V. J. S. e Peres, P. L. D. (2003). An improved LMI condition for robust \mathcal{D} -stability of uncertain polytopic systems, *IEEE Transactions on Automatic Control* **48**(3): 500–504.
- Liu, J., Wang, J. L. e Yang, G. (2003). Reliable guaranteed variance filtering against sensor failures, *IEEE Transactions on Signal Processing* **51**(5): 1403–1411.
- Masubuchi, I., Ohara, A. e Suda, N. (1998). LMI-based controller synthesis: a unified formulation and solution, *International Journal of Robust and Nonlinear Control* **8**: 669–686.
- M'Closkey, R. T., Packard, A. K. e Sipila, J. D. (2000). Branch and bound computation of the minimum norm of a linear fractional transformation over a structured set, *IEEE Transactions on Automatic Control* **45**(2): 369–375.
- Moore, B. C. (1981). Principal component analysis in linear systems: Controllability, observability, and model reduction, *IEEE Transactions on Automatic Control* **AC-26**(1): 17–32.
- Moore, D. W. (1992). *Simplicial Mesh Generation with Applications*, PhD thesis, Cornell University, Department of Computer Science, Ithaca, New York.
- Nobrega, E. G., Abdalla, M. O. e Grigoriadis, K. M. (2000). LMI-based filter design for fault detection and isolation, *Proceedings of the 39th IEEE Conference on Decision and Control*, IEEE, Sydney, Australia, pp. 4329–4334.
- Oliveira, R. C. L. F. e Peres, P. L. D. (2005a). LMI conditions for the existence of polynomially parameter-dependent Lyapunov functions assuring robust stability, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, IEEE, Seville, Spain, pp. 1660–1665.
- Oliveira, R. C. L. F. e Peres, P. L. D. (2005b). \mathcal{H}_∞ guaranteed cost computation via polynomially parameter-dependent Lyapunov functions, in P. Horacek, M. Simandl e P. Zitek (eds), *Preprints of the 16th IFAC World Congress*, IFAC, Prague, Czech Republic.
- Oliveira, R. C. L. F. e Peres, P. L. D. (2005c). Stability of polytopes of matrices via affine parameter-dependent Lyapunov functions: Asymptotically exact LMI conditions, *Linear Algebra and its Applications* **405**: 209–228.
- Oliveira, R. C. L. F. e Peres, P. L. D. (2006). LMI conditions for robust stability analysis based on polynomially parameter-dependent lyapunov functions, *Systems & Control Letters* **55**(1): 52–61.

- Palhares, R. M., Campos, C. D., Ekel, P. Y., Leles, M. C. R. e D'Angelo, M. F. S. V. (2005). Delay-dependent robust \mathcal{H}_∞ control of uncertain linear systems with lumped delays, *IEE Proceedings Control Theory & Applications* **152**(1): 27–33.
- Palhares, R. M., de Souza, C. E. e Peres, P. L. D. (2001). Robust \mathcal{H}_∞ filtering for uncertain discrete-time state-delayed systems, *IEEE Transactions on Signal Processing* **49**(8): 1696–1703.
- Palhares, R. M. e Peres, P. L. D. (1999). Robust \mathcal{H}_∞ -filtering design with pole placement constraints via Linear Matrix Inequalities, *Journal of Optimization Theory and Applications* **102**(2): 239–261.
- Palhares, R. M. e Peres, P. L. D. (2000a). Robust filtering with guaranteed energy-to-peak performance - an LMI approach, *Automatica* **36**: 851–858.
- Palhares, R. M. e Peres, P. L. D. (2000b). Robust \mathcal{H}_∞ filter design with pole constraints for discrete-time systems, *Journal of the Franklin Institute* **337**: 713–723.
- Palhares, R. M. e Peres, P. L. D. (2001). LMI approach to the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ filtering design for discrete-time uncertain systems, *IEEE Transactions on Aerospace and Electronic Systems* **37**(1): 292–296.
- Palhares, R. M., Takahashi, R. H. C. e Peres, P. L. D. (1997). \mathcal{H}_∞ and \mathcal{H}_2 guaranteed costs computation for uncertain linear systems, *International Journal of Systems Science* **28**(2): 183–188.
- Peaucelle, D., Arzelier, D., Bachelier, O. e Bernussou, J. (2000). A new robust \mathcal{D} -stability condition for real convex polytopic uncertainty, *Systems & Control Letters* **40**: 21–30.
- Peaucelle, D., Arzelier, D. e Bertrand, R. (2002). Ellipsoidal sets for static output feedback, *Proceedings of the 15th Triennial World Congress, IFAC, Barcelona, Spain*.
- Peaucelle, D., Henrion, D., Labit, Y. e Taitz, K. (2002). User's guide for sedumi interface 1.04. Version 1.04 available from <http://www.laas.fr/henrion/papers/sdmguide.pdf>.
- Pereira, G. J. e de Araújo, H. X. (2004). Robust output feedback controller design via genetic algorithms and LMIs: The mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problem, *Proceeding of the 2004 American Control Conference, AACC, Boston, MA, USA*, pp. 3309–3314.
- Ramos, D. C. W. e Peres, P. L. D. (2001). A less conservative LMI condition for the robust stability of discrete-time uncertain systems, *Systems & Control Letters* **43**: 371–378.

- Ramos, D. C. W. e Peres, P. L. D. (2002). An LMI condition for the robust stability of uncertain continuous-time linear systems, *IEEE Transactions on Automatic Control* **47**(4): 675–678.
- Safonov, M. G., Goh, K. C. e Ly, J. H. (1994). Control synthesis via Bilinear Matrix Inequalities, *Proceedings of the American control Conference*, AACC, Baltimore, Maryland, pp. 45–49.
- Scherer, C., Gahinet, P. e Chilali, M. (1997). Multiobjective output-feedback control via LMI optimization, *IEEE Transactions on Automatic Control* **42**(7): 896–911.
- Scherer, C. W. (1995). From single-channel LMI analysis to multi-channel mixed LMI synthesis: a general procedure, *Selected Topics on Identification, Modelling and Control* **8**: 1–8.
- Shaked, U. (2001). Improved LMI representations for the analysis and the design of continuous-time systems with polytopic type uncertainty, *IEEE Transactions on Automatic Control* **46**(4): 652–656.
- Shaked, U. (2003). An LPD approach to robust \mathcal{H}_2 and \mathcal{H}_∞ static output-feedback design, *IEEE Transactions on Automatic Control* **48**(5): 866–872.
- Shaked, U., Xie, L. e Soh, Y. C. (2001). New approaches to robust minimum variance filter design, *IEEE Transactions on Signal Processing* **49**(11): 2620–2629.
- Shimomura, T. e Fujii, T. (2000). Multiobjective control design via successive over-bounding of quadratic terms, *Proceedings of the 39th Conference on Decision and Control*, IEEE, Sydney, Australia, pp. 2763–2768.
- Sipila, J. D., M'Closkey, R. e Packard, A. (1999). Optimal structure design using branch and bound, *Proceedings of the American Control Conference*, AACC, San Diego, California, pp. 1861–1865.
- Skelton, R. E., Iwasaki, T. e Grigoriadis, K. (1998). *A Unified Algebraic Approach to Linear Control Design*, System and Control Book Series, Taylor & Francis, London.
- Skogestad, S. e Postlethwaite, I. (1996). *Multivariable Feedback Control: Analysis and Design*, John Wiley & Sons, West Sussex, England.
- Sturm, J. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optimization Methods and Software* **11–12**: 625–653. Version 1.05 available from <http://fewcal.kub.nl/sturm>.

- Syrmos, V. L., Abdallah, C. T., Dorato, P. e Grigoriadis, K. (1997). Static output feedback - A survey, *Automatica* **33**(2): 125–137.
- Takahashi, R. H. C., Peres, P. L. D. e Ferreira, P. A. V. (1996). $\mathcal{H}_2/\mathcal{H}_\infty$ guaranteed cost PID design for uncertain systems: A multiobjective approach, *Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design*, IEEE, Dearborn, MI, USA, pp. 315–320.
- Takahashi, R. H. C., Saldanha, R. R., Dias-Filho, W. e Ramírez, J. A. (2003). A new constrained ellipsoidal algorithm for nonlinear optimization with equality constraints, *IEEE Transactions on Magnetics* **39**(3): 1289–1292.
- Toker, O. e Özbay, H. (1995). On the NP-hardness of solving Bilinear Matrix Inequalities and simultaneously stabilization via static output feedback, *Proceedings of the American Control Conference*, AACC, Seattle, Washington, pp. 2525–2526.
- Toscano, R. (2005). A simple robust PI/PID controller design via numerical optimization approach, *Journal of Process Control* **15**: 81–88.
- Trofino, A. (2002). Robust, stable and reduced order dynamic output feedback controllers with guaranteed \mathcal{H}_2 performance, *Proceedings of the 41st IEEE Conference on Decision and Control*, IEEE, Las Vegas, Nevada, USA, pp. 3470–3475.
- Trofino, A. e Coutinho, D. F. (2004). Robust order reduction, *Proceeding of the 2004 American Control Conference*, AACC, Boston, Massachusetts, pp. 3363–3368.
- Trofino, A., Coutinho, D. F. e Barbosa, K. A. (2005). Improved \mathcal{H}_2 and \mathcal{H}_∞ conditions for robust analysis and control synthesis of linear systems, *Revista Controle & Automação* **16**(4): 427–434.
- Tuan, H. D., Apkarian, P. e Nakashima, Y. (2000). A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities, *International Journal of Robust and Nonlinear Control* **10**: 561–578.
- Tuan, H. D., Apkarian, P. e Nguyen, T. Q. (2001). Robust and reduced-order filtering: New LMI-based characterizations and methods, *IEEE Transactions on Signal Processing* **49**(12): 2975–2984.
- Vandenberghe, L. e Balakrishnan, V. (1997). Algorithms and software for LMI problems in control, *IEEE Control System Magazine* **17**(5): 89–95.

- Veillette, R. J., Medamic, S. V. e Perkins, W. R. (1992). Design of reliable control systems, *IEEE Transactions on Automatic Control* **37**(3): 280–304.
- Vendenbergue, L. e Boyd, S. (1996). Semidefinite programming, *SIAM Review* **38**(1): 49–95.
- Wang, J. e Wilson, D. A. (2003). Continuous-time $\mathcal{H}_2/\mathcal{H}_\infty$ control with non-common Lyapunov variables via convergent iterations, *Proceedings of the European Control Conference*, IEE, Cambridge, UK.
- Willems, J. C. (1971). Least squares stationary optimal control and the algebraic Riccati equation, *IEEE Transaction on Automatic Control* **AC-16**(6): 621–634.
- Wu, F. (1996). Induced \mathcal{L}_2 norm model reduction of polytopic uncertain linear systems, *Automatica* **32**(10): 1417–1426.
- Wu, F. e Jaramillo, J. (2002). Numerical algorithm for model reduction of polytopic uncertain linear systems, *Proceedings of the American Control Conference*, AACC, Anchorage, AK, pp. 1748–1752.
- Xie, L., Lu, L., Zhang, D. e Zhang, H. (2004). Improved robust \mathcal{H}_2 and \mathcal{H}_∞ filtering for uncertain discrete-time systems, *Automatica* **40**: 873–880.
- Yang, R., Lu, L. e Xie, L. (2005). Robust \mathcal{H}_2 and \mathcal{H}_∞ control of discrete-time systems with polytopic uncertainties via dynamic output feedback, *International Journal of Control* **78**(16): 1285–1294.
- Zames, G. (1981). Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses, *IEEE Transaction on Automatic Control* **AC-26**(2): 301–320.
- Zhong, M., Ding, S. X., Lam, J. e Wang, H. (2003). An LMI approach to design robust fault detection filter for uncertain LTI systems, *Automatica* **39**: 543–550.
- Zhou, K. e Doyle, J. C. (1998). *Essentials of Robust Control*, Prentice-Hall. Inc., New Jersey.