

DONALD NEUMANN

SÍNTESE DE CONTROLADORES ROBUSTOS $\mathcal{H}_2/\mathcal{H}_\infty$
 \mathcal{D} -ESTÁVEIS VIA ALGORITMOS EVOLUTIVOS: UMA
ABORDAGEM LMI

CURITIBA

2006

DONALD NEUMANN

**SÍNTESE DE CONTROLADORES ROBUSTOS $\mathcal{H}_2/\mathcal{H}_\infty$
 \mathcal{D} –ESTÁVEIS VIA ALGORITMOS EVOLUTIVOS: UMA
ABORDAGEM LMI**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Orientador:
Humberto Xavier de Araújo

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ - PUCPR

CURITIBA

2006

”Como podemos reconciliar nossa capacidade de refletir sobre o mundo e sobre nós mesmos com o fato de que nossas vidas são tão curtas, de que por mais que amemos e aprendamos, teremos sempre muito mais o que amar e aprender?”

(Marcelo Gleiser)

Aos meus pais
que possibilitaram este trabalho.

Agradecimentos

Agradecer aos nossos pais não deixa de ser um clichê, mas nem por isso se faz menos importante. Agradeço a eles a possibilidade e o apoio concedidos por poucos pais a poucos filhos para a realização de um trabalho como este. Considero-me privilegiado. À minha Pátria amada e idolatrada Brasil, sou eternamente grato por, através da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), apoiar financeiramente este projeto, acreditar no meu potencial e me proporcionar alguns dos melhores anos da minha vida, indiscutivelmente. Ao meu orientador, professor Dr. Humberto Xavier Araújo agradeço a força, a confiança e a paciência com os estudos. Ao professor Dr. Leandro dos Santos Coelho, agradeço os valiosíssimos conselhos, idéias e ajuda. Aos professores Dr. Robert Carlisle Burnett, Sheila dos Santos Rheiner, Dr. Paulo Henrique Müller do Prado e Dr. Darli Rodrigues Vieira, agradeço o empenho em suas aulas, que despertou em mim o anseio pela pesquisa científica e o amor pela academia. A todos os meus amigos, agradeço por entenderem minha ausência por conta dos estudos. E, finalmente, à Sarah agradeço pelo seu simples sorriso, uma inspiração para viver.

Um sincero Muito Obrigado!

Sumário

Lista de Publicações	1
Lista de Figuras	2
Lista de Tabelas	7
Lista de Símbolos	9
Lista de Abreviaturas	11
Resumo	12
Abstract	13
1 Introdução	14
2 Considerações Preliminares	25
2.1 Conjunto Convexo	25
2.2 Funcional Convexo	25
2.3 Otimização	26
2.3.1 Problema de Otimização	26
2.3.2 Factibilidade	26
2.3.3 Vizinhança	27
2.3.4 Ótimo Local	27
2.3.5 Ótimo Global	27
2.4 Função Analítica	27

2.5	Espaços	27
2.5.1	Espaço de Hilbert	27
2.5.2	Espaço \mathcal{L}_2	28
2.5.3	Espaço de Hardy \mathcal{H}_2	28
2.5.4	Espaço \mathcal{L}_∞	28
2.5.5	Espaço de Hardy \mathcal{H}_∞	29
2.6	Desigualdades Matriciais Lineares	29
2.6.1	Introdução	29
2.6.2	Definição	30
2.6.3	Complemento de Schur	31
2.6.4	<i>Bounded Real Lemma</i>	32
2.6.5	Regiões LMI de Alocação de Pólos e \mathcal{D} -estabilidade	33
2.7	Conclusão	37
3	Sistemas Incertos e Controle Robusto $\mathcal{H}_2/\mathcal{H}_\infty$	38
3.1	Introdução	38
3.2	Sistemas Incertos	39
3.3	Estabilidade Quadrática	41
3.4	Problema \mathcal{H}_2	42
3.5	Problema \mathcal{H}_∞	52
3.6	Problema de Controle Robusto Misto $\mathcal{H}_2/\mathcal{H}_\infty$ Sujeito à \mathcal{D} -estabilidade	59
3.7	Conclusão	63
4	Algoritmos Evolutivos	65
4.1	Introdução	65
4.2	Algoritmos Genéticos	68
4.2.1	Representação	71

4.2.2	Função Objetivo e Função Aptidão	71
4.2.3	Seleção	72
4.2.4	Probabilidades de Cruzamento e Mutação	73
4.2.5	Reinserção e Elitismo	74
4.3	Algoritmo de Michalewicz	74
4.3.1	Mutação Uniforme	76
4.3.2	Mutação de Contorno	76
4.3.3	Mutação Não-Uniforme	76
4.3.4	Cruzamento Aritmético	77
4.3.5	Cruzamento Simples	77
4.3.6	Cruzamento Heurístico	78
4.4	Evolução Diferencial	79
4.5	Algoritmo de Salomon	80
4.6	Conclusão	85
5	Algoritmo Híbrido	87
5.1	Estrutura Geral	87
5.2	Função Objetivo e Função Aptidão	89
5.3	Seleção	89
5.4	Estratégia Elitista	89
5.5	População Inicial	90
5.6	Operadores de Recombinação	92
5.6.1	Grupo de Operadores de Recombinação I	92
5.6.1.1	Cruzamento Aritmético	93
5.6.1.2	Cruzamento Simples	93
5.6.1.3	Cruzamento Heurístico	94
5.6.2	Grupo de Operadores de Recombinação II	94

5.6.2.1	Operador de Recombinação 1	94
5.6.2.2	Operador de Recombinação 2	94
5.6.2.3	Operador de Recombinação 3	95
5.6.2.4	Operador de Recombinação 4	95
5.6.3	Operador de Evolução Diferencial	95
5.7	Operadores de Mutação	97
5.7.1	Grupo de Operadores de Mutação I	98
5.7.1.1	Mutação Uniforme	99
5.7.1.2	Mutação de Contorno	99
5.7.1.3	Mutação Não-Uniforme	99
5.7.2	Grupo de Operadores de Mutação II	100
5.7.2.1	Operador de Mutação 1	100
5.7.2.2	Operador de Mutação 2	100
5.7.2.3	Operador de Mutação 3	101
5.7.2.4	Operador de Mutação 4	101
5.7.3	Operador de Salomon	101
5.8	Conclusão	102
6	Resultados	103
6.1	Exemplo 1	105
6.1.1	Caso 1: Realimentação Estática de Saída	106
6.1.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida . .	109
6.1.3	Caso 3: Realimentação Dinâmica de Saída de Ordem Completa .	112
6.1.4	Análise do Exemplo 1	115
6.2	Exemplo 2	117
6.2.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida . .	119
6.2.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Completa .	122

6.2.3	Análise do Exemplo 2	124
6.3	Exemplo 3	124
6.3.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida . .	125
6.3.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Completa .	128
6.3.3	Análise do Exemplo 3	131
6.4	Exemplo 4	132
6.4.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida . .	133
6.4.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida com Alocação de Pólos	135
6.4.3	Caso 3: Realimentação Dinâmica de Saída de Ordem Completa .	137
6.4.4	Análise do Exemplo 4	140
6.5	Exemplo 5	141
6.5.1	Caso 1: Realimentação Estática de Saída	141
6.5.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida . .	144
6.5.3	Caso 3: Realimentação Dinâmica de Saída de Ordem Completa .	145
6.5.4	Análise do Exemplo 5	147
6.6	Exemplo 6	148
6.6.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida . .	149
6.7	Exemplo 7	152
6.7.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida . .	152
6.7.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida com Alocação de Pólos	155
6.7.3	Caso 3: Realimentação Dinâmica de Saída de Ordem Reduzida com Alocação de Pólos e População de 40 Elementos	158
6.7.4	Análise do Exemplo 7	161
6.8	Exemplo 8	162

6.8.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Completa e $\gamma \leq 100$	162
6.8.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Completa e $\gamma \leq 12,5$	166
6.8.3	Análise do Exemplo 8	169
6.9	Exemplo 9	170
6.9.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Completa	170
6.9.2	Caso 2: Realimentação Dinâmica de Saída de Ordem Completa com População de 40 Elementos	173
6.9.3	Análise do Exemplo 9	175
6.10	Exemplo 10	176
6.10.1	Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida	177
6.11	Conclusão	180
7	Conclusão Geral	185
	Referências	189

Lista de Publicações

1. NEUMANN, Donald; DE ARAÚJO, Humberto X. *Hybrid differential evolution method for the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ robust control problem under pole assignment*. 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 - Sevilha, Espanha.
2. NEUMANN, Donald; DE ARAÚJO, Humberto X. *Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for uncertain systems under pole placement constraints using genetic algorithms and LMIs*. ISIC04 - IEEE International Conference on Intelligent Control 2004 - Taipei, Taiwan.
3. DE ARAÚJO, Humberto X.; NEUMANN, Donald. *Controle robusto $\mathcal{H}_2/\mathcal{H}_\infty$: síntese via algoritmos genéticos*. Congresso Brasileiro de Automática - CBA04, Gramado, RS - Brasil.

Lista de Figuras

1	Região S de alocação de pólos.	34
2	Sistema em malha fechada para otimização LQG.	43
3	Sistema genérico realimentado - caso \mathcal{H}_2	44
4	Sistema genérico realimentado - caso \mathcal{H}_∞	53
5	Sistema genérico realimentado - caso $\mathcal{H}_2/\mathcal{H}_\infty$	60
6	Fluxograma de um algoritmo genético básico.	69
7	Diagrama do cruzamento.	70
8	Diagrama da mutação.	70
9	Algoritmo evolutivo híbrido	88
10	Operadores de recombinação.	92
11	Operadores de mutação.	97
12	Evolução dos algoritmos no exemplo 1 - caso 1.	107
13	Comparação dos melhores resultados do exemplo 1 - caso 1.	108
14	Número médio de elementos inactíveis gerados no exemplo 1 - caso 1.	108
15	Pólos em malha fechada do exemplo 1 - caso 1.	109
16	Evolução dos algoritmos no exemplo 1 - caso 2.	110
17	Comparação dos melhores resultados do exemplo 1 - caso 2.	111
18	Número médio de elementos inactíveis gerados no exemplo 1 - caso 2.	111
19	Pólos em malha fechada do exemplo 1 - caso 2.	112
20	Evolução dos algoritmos no exemplo 1 - caso 3.	113
21	Comparação dos melhores resultados do exemplo 1 - caso 3.	114
22	Número médio de elementos inactíveis gerados no exemplo 1 - caso 3.	114

23	Pólos em malha fechada do exemplo 1 - caso 3.	115
24	Número médio de elementos inactíveis gerados no exemplo 1.	116
25	Sistema de pêndulo invertido.	117
26	Evolução dos algoritmos no exemplo 2 - caso 1.	120
27	Comparação dos melhores resultados do exemplo 2 - caso 1.	120
28	Pólos em malha fechada do exemplo 2 - caso 1.	121
29	Detalhe dos pólos em malha fechada do exemplo 2 - caso 1.	121
30	Evolução dos algoritmos no exemplo 2 - caso 2.	122
31	Comparação dos melhores resultados do exemplo 2 - caso 2.	123
32	Pólos em malha fechada do exemplo 2 - caso 2.	123
33	Detalhe dos pólos em malha fechada do exemplo 2 - caso 2.	124
34	Evolução média dos algoritmos no exemplo 3 - caso 1.	126
35	Comparação dos melhores resultados do exemplo 3 - caso 1.	127
36	Número médio de elementos inactíveis gerados no exemplo 3 - caso 1.	127
37	Pólos em malha fechada do exemplo 3 - caso 1.	128
38	Evolução média dos algoritmos no exemplo 3 - caso 2.	129
39	Comparação dos melhores resultados do exemplo 3 - caso 2.	130
40	Número de elementos inactíveis gerados no exemplo 3 - caso 2.	130
41	Pólos em malha fechada do exemplo 3 - caso 2.	131
42	Número médio de elementos inactíveis do exemplo 3.	132
43	Evolução média dos algoritmos no exemplo 4 - caso 1.	133
44	Comparação dos melhores resultados do exemplo 4 - caso 1.	134
45	Número de elementos inactíveis gerados no exemplo 4 - caso 1.	134
46	Pólos em malha fechada do exemplo 4 - caso 1.	135
47	Evolução média dos algoritmos no exemplo 4 - caso 2.	136
48	Comparação dos melhores resultados do exemplo 4 - caso 2.	137

49	Número de elementos inactivos gerados no exemplo 4 - caso 2.	137
50	Pólos em malha fechada do exemplo 4 - caso 2.	138
51	Evolução média dos algoritmos no exemplo 4 - caso 3.	139
52	Comparação dos melhores resultados do exemplo 4 - caso 3.	140
53	Número de elementos inactivos gerados no exemplo 4 - caso 3.	140
54	Pólos em malha fechada do exemplo 4 - caso 3.	141
55	Número médio de elementos inactivos gerados no exemplo 4.	142
56	Evolução dos algoritmos no exemplo 5 - caso 1.	142
57	Comparação dos melhores resultados do exemplo 5 - caso 1.	143
58	Pólos em malha fechada do exemplo 5 - caso 1.	143
59	Evolução média dos algoritmos no exemplo 5 - caso 2.	145
60	Comparação dos melhores resultados do exemplo 5 - caso 2.	146
61	Pólos em malha fechada do exemplo 5 - caso 2.	146
62	Evolução média dos algoritmos no exemplo 5 - caso 3.	147
63	Comparação dos melhores resultados do exemplo 5 - caso 3.	148
64	Pólos em malha fechada do exemplo 5 - caso 3.	148
65	Evolução média dos algoritmos do exemplo 6 - caso 1.	150
66	Comparação dos melhores resultados do exemplo 6 - caso 1.	151
67	Número de elementos inactivos gerado no exemplo 6 - caso 1.	151
68	Pólos em malha fechada do exemplo 6 - caso 1.	151
69	Evolução média dos algoritmos no exemplo 7 - caso 1.	153
70	Comparação dos melhores resultados do exemplo 7 - caso 1.	153
71	Número de elementos inactivos gerados no exemplo 7 - caso 1.	154
72	Pólos em malha fechada do exemplo 7 - caso 1.	154
73	Evolução média dos algoritmos no exemplo 7 - caso 2.	155
74	Comparação dos melhores resultados do exemplo 7 - caso 2.	156

75	Número médio de elementos inactivos gerados no exemplo 7 - caso 2.	156
76	Pólos em malha fechada do exemplo 7 - caso 2.	157
77	Detalhe dos pólos em malha fechada do exemplo 7 - caso 2.	157
78	Evolução média dos algoritmos no exemplo 7 - caso 3.	158
79	Comparação dos melhores resultados do exemplo 7 - caso 3.	159
80	Número médio de elementos inactivos gerados no exemplo 7 - caso 3.	159
81	Pólos em malha fechada do exemplo 7 - caso 3.	160
82	Detalhe dos pólos em malha fechada do exemplo 7 - caso 3.	160
83	Número de elementos inactivos gerado no exemplo 7.	161
84	Evolução média dos algoritmos no exemplo 8 - caso 1.	163
85	Comparação dos melhores resultados do exemplo 8 - caso 1.	164
86	Número médio de elementos inactivos gerados no exemplo 8 - caso 1.	164
87	Pólos em malha fechada do exemplo 8 - caso 1.	165
88	Evolução média dos algoritmos no exemplo 8 - caso 2.	166
89	Comparação dos melhores resultados do exemplo 8 - caso 2.	167
90	Número médio de elementos inactivos gerados no exemplo 8 - caso 2.	168
91	Pólos em malha fechada do exemplo 8 - caso 2.	168
92	Número médio de elementos inactivos gerados no exemplo 8.	169
93	Evolução média dos algoritmos no exemplo 9 - caso 1.	171
94	Comparação dos melhores resultados do exemplo 9 - caso 1.	171
95	Número médio de elementos inactivos gerados no exemplo 9 - caso 1.	172
96	Pólos em malha fechada do exemplo 9 - caso 1.	172
97	Evolução média dos algoritmos no exemplo 9 - caso 2.	174
98	Comparação dos melhores resultados do exemplo 9 - caso 2.	175
99	Número médio de elementos inactivos gerados no exemplo 9 - caso 2.	175
100	Pólos em malha fechada do exemplo 9 - caso 2.	176

101	Número médio de elementos inactivos gerados no exemplo 9.	177
102	Evolução média dos algoritmos no exemplo 10 - caso 1.	178
103	Comparação dos melhores resultados do exemplo 10 - caso 1.	179
104	Número médio de elementos inactivos gerados no exemplo 10 - caso 1. .	179
105	Pólos em malha fechada do exemplo 10 - caso 1.	180
106	Percentual de melhores resultados x grupos de operadores.	181
107	Percentual de piores resultados x grupos de operadores.	182

Lista de Tabelas

1	Exemplo de algoritmos de seleção.	73
2	Probabilidades do grupo I de operadores.	104
3	Probabilidades do grupo II de operadores.	105
4	Probabilidades do grupo III de operadores.	105
5	Comparação dos resultados encontrados no exemplo 1 - caso 1.	109
6	Resultados encontrados na literatura para o exemplo 1 - caso 1.	109
7	Comparação dos resultados encontrados no exemplo 1 - caso 2.	111
8	Comparação dos resultados do exemplo 1 - caso 3.	115
9	Controladores ótimos para o sistema nominal do exemplo 1 - caso 3.	116
10	Constantes do pêndulo invertido.	118
11	Comparação dos melhores resultados do exemplo 2 caso 1.	121
12	Comparação dos melhores resultados do exemplo 2 - caso 2.	124
13	Resultados da literatura e ótimos para o sistema nominal do exemplo 2 - caso 2.	125
14	Comparação dos resultados do exemplo 3 - caso 1.	128
15	Comparação dos resultados do exemplo 3 - caso 2.	130
16	Controladores ótimos para o sistema nominal do exemplo 3 - caso 2.	131
17	Comparação dos melhores resultados do exemplo 4 - caso 1.	135
18	Comparação dos melhores resultados do exemplo 4 - caso 2.	138
19	Comparação dos melhores resultados do exemplo 4 - caso 3.	139
20	Controladores ótimos para o sistema nominal do exemplo 4 - caso 3.	141
21	Comparação dos melhores resultados do exemplo 5 - caso 1.	144

22	Resultados encontrados na literatura para o exemplo 5 - caso 1.	144
23	Comparação dos melhores resultados do exemplo 5 - caso 2.	145
24	Comparação dos melhores resultados do exemplo 5 - caso 3.	147
25	Controladores ótimos para o exemplo 5 - caso 3.	149
26	Comparação dos melhores resultados do exemplo 6 - caso 1.	150
27	Comparação dos melhores resultados do exemplo 7 - caso 1.	155
28	Comparação dos melhores resultados do exemplo 7 - caso 2.	157
29	Comparação dos melhores resultados do exemplo 7 - caso 3.	160
30	Comparação dos melhores resultados do exemplo 8 - caso 1.	166
31	Controladores ótimos para o sistema nominal do exemplo 8 - caso 1. . . .	166
32	Comparação dos melhores resultados do exemplo 8 - caso 2.	168
33	Comparação dos resultados ótimos para o sistema nominal do exemplo 8 - caso 2.	169
34	Comparação dos melhores resultados do exemplo 9 - caso 1.	173
35	Resultados ótimo obtidos para o sistema nominal do exemplo 9 - caso 1.	173
36	Comparação dos melhores resultados do exemplo 9 - caso 2.	174
37	Resultados ótimo obtidos para o sistema nominal do exemplo 9 - caso 2.	176
38	Comparação dos melhores resultados do exemplo 10 - caso 1.	180
39	Estatísticas gerais do capítulo.	180

Lista de Símbolos

$f(\bullet)$ - Funcional

$\nabla f(\bullet)$ - Gradiente de $f(\bullet)$

$\mathbb{R}^{n \times m}$ - Conjunto dos vetores de números reais com dimensão $n \times m$

$\mathbb{C}^{n \times m}$ - Conjunto dos vetores de números complexos com dimensão $n \times m$

$\text{Re}(s)$ - Parte real do número complexo s

$\text{Imag}(s)$ - Parte imaginária do número complexo s

x_{opt} - Vetor minimizador x_{opt}

$\text{apt}(x)$ - Aptidão do cromossomo x

$\text{máximo}(\Omega)$ - Máximo do conjunto Ω

$\text{mínimo}(\Omega)$ - Mínimo do conjunto Ω

$E(\bullet)$ - Esperança matemática

L_C - Gramiano de controlabilidade

L_O - Gramiano de observabilidade

$\mathcal{L}\{h(t)\}$ - Transformada de Laplace de $h(t)$

$\mathcal{L}^{-1}\{H(s)\}$ - Transformada de Laplace inversa de $H(s)$

$|H(s)|$ - Módulo da função $H(s)$

\mathcal{L}_2 - Espaço \mathcal{L}_2

\mathcal{L}_∞ - Espaço \mathcal{L}_∞

\mathcal{H}_2 - Espaço de Hardy \mathcal{H}_2

\mathcal{H}_∞ - Espaço de Hardy \mathcal{H}_∞

$\text{ess}(\bullet)$ - Essencial

sup_σ - Supremo de σ

G^* - Matriz transposta conjugada de G

G^T - Matriz transposta de G

G^{-1} - Matriz inversa de G

$\text{tr}[G]$ - Traço da matriz G

$\det(G)$ - Determinante da matriz G

$\|G\|_n$ - Norma n da matriz G

$\langle F, G \rangle$ - Produto interno das matrizes F e G

$\text{diag}(G_1, \dots, G_p)$ - Matriz diagonal formada pelas matrizes G_1, \dots, G_p

$G(\geq) > 0$ - Matriz (semi)definida positiva, *i.e.* $\{G \in \mathbb{R}^{n \times n}, | u^T G u(\geq) > 0, \forall u \in \mathbb{R}^n, u \neq 0\}$

$G(\leq) < 0$ - Matriz (semi)definida negativa, *i.e.* $\{G \in \mathbb{R}^{n \times n}, | u^T G u(\leq) < 0, \forall u \in \mathbb{R}^n, u \neq 0\}$

Lista de Abreviaturas

AE - Algoritmo Evolutivo

AG - Algoritmo Genético

AMI - Desigualdade Matricial Afim (*Affine Matrix Inequality*)

CE - Computação Evolutiva

DNA - Ácido Desoxirribonucleico (*Deoxyribonucleic Acid*)

ED - Evolução Diferencial

EGS - Algoritmo de Salomon (*Evolutionary Gradient Search Procedure*)

LMI - Desigualdade Linear Matricial (*Linear Matrix Inequality*)

LQ - Linear Quadrático

LQG - Linear Quadrático Gaussiano

MIMO - Múltiplas Entradas, Múltiplas Saídas (*Multiple Inputs, Multiple Outputs*)

SDP - Programação Semidefinida (*Semidefinite Programming*)

SISO - Uma Entrada, Uma Saída (*Single Input, Single Output*)

Resumo

Este trabalho aborda o problema de controle misto $\mathcal{H}_2/\mathcal{H}_\infty$ de sistemas lineares incertos por realimentação de saída estática ou dinâmica, sujeitos à \mathcal{D} -estabilidade. As incertezas do sistema são do tipo paramétricas e os elementos das matrizes dinâmicas do sistema podem estar contidos em um politopo no espaço de parâmetros. Para a \mathcal{D} -estabilidade, utiliza-se o critério de estabilidade quadrática e regiões de alocação de pólos descritas por desigualdades matriciais lineares (*Linear Matrix Inequalities* - LMIs). O problema misto é formulado com diferentes funções de Lyapunov, cada uma associada a um tipo de restrição, permitindo soluções menos conservadoras do ponto de vista matemático. Para a síntese de controladores de estrutura fixa, propõe-se um algoritmo híbrido, baseado nos Algoritmos Evolutivos e nas LMIs. No caso de realimentação de saída dinâmica, esta abordagem pode ser usada na síntese de controladores de ordem reduzida ou completa. Para validar a aplicação do algoritmo desenvolvido, diversos problemas da literatura são estudados.

Palavras-chave: 1. Controle Robusto; 2. Controle Ótimo $\mathcal{H}_2/\mathcal{H}_\infty$; 3. Algoritmos Evolutivos; 4. \mathcal{D} -estabilidade; 5. LMI.

Abstract

This work is concerned with the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ robust control problem combined with robust \mathcal{D} -stability in linear matrix inequality (LMI) regions. Continuous-time linear systems subject to polytopic parametric uncertainties are studied. Based on Evolutionary Algorithms and LMIs, an hybrid algorithm is presented for numerical computation of a robust fixed-order (static or dynamic) output feedback controller. This approach does not require that all specifications are enforced by a single closed-loop Lyapunov function. This fact allows to reduce the conservatism of the usual existing methods. In addition, this approach can be used for synthesis of reduced or full order controllers. Examples borrowed from the literature are discussed to validate this approach.

Key-words: 1. Robust Control; 2. $\mathcal{H}_2/\mathcal{H}_\infty$ Control; 3. Evolutionary Algorithms; 4. \mathcal{D} -stability; 5. LMI.

1 Introdução

O primeiro passo no projeto de um controlador em um sistema de controle é a modelagem matemática do processo em questão. Esta tarefa nem sempre é simples. Quando o processo pode ser precisamente descrito por um modelo matemático, em geral, as técnicas conhecidas do controle clássico, desenvolvidas no domínio da frequência, podem ser aplicadas. Entretanto, muitas vezes o processo não admite a modelagem matemática precisa, seja por influência dos instrumentos de medição, seja por erros de modelagem ou aproximações. Este fato resulta em modelos matemáticos, até certo ponto, inexatos. Para se aumentar a acurácia destes modelos, pode-se abordar o sistema físico como um sistema incerto, no qual seus parâmetros podem assumir infinitos valores dentro de um conjunto com limites conhecidos. Para os sistemas incertos, exige-se do controlador um certo grau de robustez, garantindo-se estabilidade e desempenho para todo o conjunto de incertezas. O campo da teoria de controle de processos, que aborda estes problemas, denomina-se controle robusto. Embora alguns conceitos clássicos, como as margens de ganho e de fase, possibilitem quantificar a robustez de um sistema em malha fechada, a síntese de um controlador robusto é uma tarefa complexa e, em geral, as técnicas de controle clássico não são eficientes.

As técnicas de controle moderno, desenvolvidas sobretudo no domínio do tempo, obtiveram crescente atenção da comunidade científica a partir da década de 60. Dentre as características destes métodos estão possibilidade de síntese de controladores robustos e também o tratamento direto de processos com múltiplas entradas e múltiplas saídas (MIMO). Desde então, técnicas capazes de agregar requisitos de desempenho ao controlador projetado foram desenvolvidas, permitindo que o projetista garanta não só a estabilidade do sistema, mas também um comportamento eficiente. Em geral, os requisitos de desempenho são expressos na forma de índices e o campo da teoria de controle moderno que busca a otimização destes índices denomina-se controle ótimo. Estes índices podem representar, por exemplo, a influência de sinais de distúrbio ou ruído na saída do sistema e o que se busca projetar é o controlador que minimize este efeito.

Os primeiros índices de desempenho utilizados foram os custos Linear Quadrático (LQ) e Linear Quadrático Gaussiano (LQG), este último baseado no conhecimento estocástico dos sinais de ruído. Uma grande euforia na comunidade científica foi observada com a teoria LQG, porém, logo se percebeu que estas técnicas não forneciam soluções robustas o suficiente para diversas aplicações [94]. Além disso, outras formulações mais simples e diretas foram desenvolvidas, como é o caso do controle ótimo \mathcal{H}_2 , uma abordagem alternativa e determinística para o problema LQG.

Um índice de desempenho frequentemente utilizado é a norma \mathcal{H}_∞ . Em 1981, George Zames publicou um artigo [135] no qual descreve os fundamentos do que viria a ser denominado controle \mathcal{H}_∞ . Inicialmente proposto no domínio da frequência, o problema de controle \mathcal{H}_∞ aborda a síntese de controladores que garantem desempenho face ao custo chamado norma \mathcal{H}_∞ e tem como objetivo a redução da sensibilidade de processos lineares. Técnicas posteriores (final da década de 80) possibilitaram a abordagem da teoria \mathcal{H}_∞ no domínio do tempo, simplificando significativamente a abordagem e permitindo um tratamento mais direto dos processos incertos. A idéia, introduzida inicialmente em [102] e posteriormente demonstrada em [43], apresenta o problema de controle \mathcal{H}_∞ por realimentação de saída com solução através de duas equações algébricas de Riccati.

Outra abordagem para o problema de controle ótimo robusto teve início com os estudos de Aleksandr Lyapunov e é baseado nas funções de Lyapunov para a garantia da estabilidade do sistema em um ponto de equilíbrio. O problema reside, em geral, em encontrar uma função de Lyapunov adequada.

O uso das funções de Lyapunov na síntese de controladores pode ser observado nos trabalhos pioneiros de Kalman e Bertram [75]. O desenvolvimento inicial da teoria teve como foco a busca de funções de Lyapunov adequadas para os sistemas incertos e os principais trabalhos desenvolvidos foram [65, 83]. Recentemente, porém, grande esforço da comunidade científica foi dedicado ao estudo das funções quadráticas de Lyapunov, resultando na denominada teoria de estabilidade quadrática de sistemas lineares [8].

Informalmente, o conceito de estabilidade quadrática pressupõe funções de Lyapunov do tipo quadrático, tanto para o domínio de incerteza do problema, quanto para as diferentes restrições envolvidas [35, 94]. Assim, um sistema é denominado quadraticamente estabilizável se existe um controlador para o qual o sistema em malha fechada é quadraticamente estável. Diversas soluções foram propostas para este problema. Uma das soluções, obtida através das equações de Riccati [104, 102, 77], permite a presença de incertezas limitadas em norma e é descrita no espaço de estados. A literatura desenvolvida

utilizando o conceito de estabilidade quadrática é vasta e aqui citam-se apenas alguns dos principais trabalhos [105, 106, 107, 108, 132, 40, 121, 99, 112, 133, 12, 13, 39, 54, 134, 103].

Uma abordagem de particular interesse é a que trata do controle a custo garantido, proposta por Chang e Peng [26]. Nessa abordagem uma função de custo quadrático é utilizada para se calcular o limite superior do custo para o sistema incerto em malha fechada. Essa abordagem permite tratar não só a estabilidade robusta do sistema, mas também garante um certo grau de desempenho.

Apesar de poderosas, as soluções propostas através das equações algébricas de Riccati são matematicamente complexas. Além disso, não permitem uma aplicação ampla e genérica para diferentes tipos de problemas e funções de custo estabelecidas [113, 80]. Com o avanço da computação, nos anos seguintes à segunda grande guerra, diversos problemas, que não possuíam soluções formais e bem definidas matematicamente, receberam um enfoque numérico, através de algoritmos de otimização. Assim, os problemas de controle ótimo começaram a ser reescritos na forma de problemas de otimização numérica. Um dos métodos que ganhou significativo impulso devido à evolução da computação foi o método baseado em desigualdades matriciais lineares - LMI.

O estudo das LMIs data, provavelmente, dos primeiros trabalhos feitos por Lyapunov e só não se fortaleceram antes devido à falta de algoritmos eficientes de solução. O impulso no estudo dos algoritmos de programação semi-definida, como o algoritmo de pontos interiores, em meados dos anos 90 [60], bem como o grande desenvolvimento tecnológico possibilitaram a solução de problemas LMI em tempo finito, motivando a comunidade científica. Simplificadamente, um problema LMI é escrito na forma de um problema de otimização com objetivo linear e restrições semi-definidas positivas, envolvendo matrizes simétricas que são afins nas variáveis de decisão [60]. A principal vantagem da abordagem LMI é a convexidade inerente ao método, que permite a aplicação direta e simples em alguns problemas de controle robusto. Como a abordagem LMI é descrita na forma de um problema de otimização, diferentes índices de desempenho, possivelmente conflitantes, podem ser associados ao problema em uma estrutura única e geral de síntese.

Diversos problemas de controle foram escritos na forma de LMIs [17], como por exemplo os problemas \mathcal{H}_2 [43, 57, 58] e \mathcal{H}_∞ [29, 43]. A otimização conjunta dos critérios \mathcal{H}_2 e \mathcal{H}_∞ também é interessante do ponto de vista de rejeição de ruídos. Este problema denomina-se problema de controle ótimo misto $\mathcal{H}_2/\mathcal{H}_\infty$ e foi tratado no contexto de LMIs em [67, 44]. Apesar de eficiente, a maioria destes problemas só possui solução geral LMI para o caso de realimentação de estado. Scherer desenvolveu uma solução para o caso

de realimentação de saída em [116]. Esta solução não permite, porém, o tratamento de sistemas incertos e se aplica somente nos casos onde a ordem do controlador é igual ou superior à ordem do processo.

Em [7, 29, 30, 98] o problema de controle misto $\mathcal{H}_2/\mathcal{H}_\infty$ é formulado em termos de LMIs e consideram-se também restrições de alocação de pólos. Nestas abordagens, buscase um controlador que atenda não somente os critérios de otimalidade \mathcal{H}_2 ou \mathcal{H}_∞ , mas que os pólos do sistema em malha fechada estejam localizados em uma região pré-estabelecida do plano complexo. Isso permite a escolha *a priori* de limites para o comportamento transitório do sistema incerto. Apesar de permitir ao projetista grande flexibilidade na síntese, estas abordagens, desenvolvidas com o conceito de estabilidade quadrática, são intrinsecamente conservadoras ao exigir uma mesma função de Lyapunov para as diferentes restrições do problema e para todo o domínio de incerteza.

Na busca de se diminuir o grau de conservadorismo das soluções que utilizam o conceito de estabilidade quadrática, novas abordagens que consideram funções de Lyapunov dependentes de parâmetros foram desenvolvidas [45, 6, 37]. A utilização de funções de Lyapunov dependentes de parâmetros tem se solidificado na teoria de controle moderno e seus resultados são inspiradores. Entretanto, uma técnica geral e não restritiva de síntese de controladores robustos mistos $\mathcal{H}_2/\mathcal{H}_\infty$ com restrições de alocação de pólos através de funções de Lyapunov dependentes de parâmetros ainda não foi desenvolvida.

Na busca por soluções menos restritivas para o problema de síntese de controladores robustos, diversos autores utilizaram métodos menos ortodoxos, como é o caso da inteligência computacional. Em geral, estas técnicas geram soluções satisfatórias calcadas no menor número de hipóteses matemáticas assumidas para o problema. A principal desvantagem destas técnicas reside na falta de uma descrição matemática completa, como, por exemplo, a prova de convergência.

Na maioria dos problemas do “mundo real” é necessário se considerar diversos critérios de desempenho. Estes problemas são, por isso, denominados problemas com múltiplos objetivos ou multi-objetivo. Em muitos casos, os critérios de desempenho não são mensuráveis e freqüentemente são conflitantes. Assim, existe uma relação de compromisso entre estes critérios e a melhoria de um objetivo implica necessariamente a deterioração de outro. Exemplos de critérios conflitantes são sobre-sinal e tempo de subida de um sistema com uma entrada e uma saída (SISO) e as normas \mathcal{H}_2 e \mathcal{H}_∞ . Para os problemas multi-objetivo, não existe uma única solução ótima, porém um conjunto de soluções ótimas de Pareto. Informalmente, as soluções ótimas não-dominantes (como também são

conhecidas as soluções de Pareto) são aquelas com as quais não se consegue otimizar um critério sem se prejudicar outro. Como não se pode comparar objetivamente duas soluções ótimas de Pareto, é função do projetista escolher a que melhor se adapta ao problema.

No passado os problemas multi-objetivo eram tratados como problemas de otimização com um objetivo único. Assim, uma única função custo era utilizada, responsável por agregar os diversos objetivos conflitantes, geralmente através de uma soma ponderada. Para se definir a função custo como a combinação de diversos objetivos, necessita-se de grande conhecimento da relação entre os objetivos e a mensuração de cada critério individualmente. Essas informações nem sempre estão disponíveis facilmente e constitui-se um desafio para o projetista não só solucionar o problema, mas também escrevê-lo da forma mais adequada. Os métodos clássicos de otimização eram então aplicados resultando em uma única solução para o problema. Para que um conjunto de soluções de Pareto fosse construído, permitindo ao projetista escolher uma dentre várias soluções ótimas, o método de otimização necessitava ser aplicado diversas vezes, variando-se as ponderações na função custo. No intuito de driblar estas desvantagens da abordagem clássica de otimização através de um único objetivo, a comunidade científica voltou-se para técnicas permitem considerar múltiplos objetivos em conjunto. Uma das teorias que podem ser aplicadas em tais tipos de problemas é a Computação Evolutiva (CE) e seus Algoritmos Evolutivos (AEs).

Os algoritmos evolutivos são métodos de busca e otimização globais e paralelos, fundados sobre os princípios da seleção natural de Darwin [34] e genética de Mendel. Suas principais vantagens são robustez, flexibilidade e aplicabilidade a um grande escopo de problemas, principalmente aqueles mal condicionados ou onde o espaço de busca associado é complexo e desconhecido. A principal desvantagem dos AEs é, em muitos casos, a ineficiência computacional, que limita sua aplicabilidade em problemas de segurança e em tempo-real [48].

Uma característica representativa das técnicas evolutivas é a capacidade de operar sobre um conjunto de soluções candidatas do problema, denominado população, e não somente sobre uma única solução, como no caso dos métodos clássicos. Em geral, qualquer método iterativo, baseado em populações de elementos, em critérios de seleção e operadores estocásticos para se gerar novas soluções, pode ser denominado um AE [48].

De forma bastante simples, um AE evolui uma população inicial através de operadores de seleção, recombinação, mutação e reinserção até que uma população de soluções satisfatórias seja encontrada. A seleção é responsável por, de maneira estocástica e a

exemplo da seleção natural, garantir a sobrevivência e propagar as melhores soluções (ou, no jargão da área, as soluções mais aptas). A aptidão de cada cromossomo, como também são denominadas as soluções candidatas, é medida através de uma função aptidão que é relacionada à uma função custo especificada. A exemplo do código genético na natureza, cada solução (fenótipo) é codificada e representada por um código respectivo (seu genótipo). É sobre o código ou genótipo que os operadores de recombinação e mutação são aplicados. A recombinação (também denominada cruzamento em alguns dos AEs) é responsável pela troca de informação genética entre soluções, permitindo a exploração de determinadas regiões no espaço de busca. A mutação, por sua vez, introduz uma variação aleatória no código genético explorando outras regiões do espaço de busca. Por fim, a reinserção é o mecanismo responsável por decidir quais das soluções geradas são incluídas na população subsequente. De maneira iterativa, estas etapas são repetidas e as soluções evoluem até que um critério de parada seja atingido. Dentre os critérios de parada mais comuns estão o número máximo de iterações (também chamadas gerações) e um limite máximo entre a melhor e a pior aptidão dos elementos da população.

Os diferentes algoritmos pertencentes ao ramo da computação evolutiva tiveram sua origem em quatro abordagens aparentemente distintas: programação evolutiva [50], estratégias evolutivas [110], algoritmos genéticos, propostos inicialmente por Holland [70], e programação genética [78]. O algoritmo genético (AG) proposto por Goldberg [61] obteve imensa popularidade por sua simplicidade, robustez e amplo escopo de aplicação. É, provavelmente, o algoritmo que mais foi aplicado no campo da teoria de controle até os dias de hoje. É importante notar, porém, que com os avanços na computação evolutiva não se reconhecem mais fronteiras rígidas e bem definidas entre as quatro principais técnicas citadas [48]. Recentemente, a tendência de se unificar o desenvolvimento das quatro abordagens iniciais deu origem à teoria moderna e mais ampla dos algoritmos evolutivos [92].

Diversos autores aplicaram as técnicas de computação evolutiva nos problemas de controle ótimo e robusto, impulsionados principalmente pela aplicabilidade dos AEs em problemas multi-objetivo e pela sua capacidade de operar com populações. Estas características permitem que, a cada simulação do algoritmo, um conjunto de soluções ótimas de Pareto seja gerado.

Algumas aplicações dos algoritmos evolutivos na síntese de controladores no domínio da frequência são apresentadas em [27, 32, 74, 79, 71]. Em [74] aborda-se a síntese de controladores discretos para sistemas SISO sujeitos à incertezas e utilizando-se critérios de

desempenho temporais. A abordagem descrita em [27] considera o critério de desempenho misto $\mathcal{H}_2/\mathcal{H}_\infty$ na síntese de PIDs para sistemas incertos, utilizando um algoritmo genético canônico. Krohling *et al.* [79] descrevem um algoritmo baseado em estratégias evolutivas para a síntese de controladores $\mathcal{H}_2/\mathcal{H}_\infty$ de estrutura fixa. Os sistemas considerados são incertos e possuem apenas uma entrada e uma saída. Esta abordagem não se limita a uma estrutura definida, como PID, desde que a estrutura do controlador seja dada. Em [71], considera-se um algoritmo genético para a solução do problema de controle ótimo misto $\mathcal{H}_2/\mathcal{H}_\infty$ para sistemas discretos precisamente conhecidos, no domínio da frequência. O algoritmo proposto busca a minimização de distúrbios externos utilizando um AG de codificação binária. A vantagem das abordagens no domínio da frequência é a síntese direta das estruturas de controle simples e bastante difundidas, como os PIDs. Contudo, estas abordagens possuem limitada aplicação em processos complexos, com múltiplas entradas e múltiplas saídas (MIMO).

Descrito no domínio do tempo em uma estrutura LMI, o método proposto em [87, 88] busca a minimização da norma \mathcal{H}_∞ . Em uma estrutura análoga à discutida neste trabalho, apresenta-se um algoritmo genético que transforma o problema matricial bi-linear em um problema convexo, para o qual minimiza-se a norma \mathcal{H}_∞ do sistema em malha fechada. A lei de controle é dada por realimentação de saída dinâmica e não se consideram sistemas incertos nem restrições de alocação de pólos. Ressaltam os autores destes trabalhos que, tanto diferentes estruturas de controle (como por exemplo PIDs), quanto critérios de desempenho podem ser utilizados, desde que se conheça, *a priori*, uma população inicial. Esta característica não é, porém, inerente aos trabalhos [87, 88], mas uma propriedade dos métodos baseados em algoritmo evolutivos. A partir de uma população inicial definida, a flexibilidade destes algoritmos e a relativa simplicidade de análise de factibilidade dos cromossomos, através de LMIs, permite que, em geral, qualquer estrutura de controle no espaço de estados possa ser utilizada. O desafio reside, porém, em se construir a população inicial factível, dados os critérios de desempenho desejados e as restrições na estrutura de controle.

A síntese de controladores de ordem reduzida \mathcal{H}_∞ por realimentação dinâmica de saída através de algoritmos genéticos, utilizando-se a formulação LMI, é tratada em [73]. Nesta abordagem, as LMIs específicas de síntese de controladores \mathcal{H}_∞ de ordem reduzida são utilizadas e o objetivo do algoritmo genético é encontrar uma matriz com restrições de posto. Uma característica interessante do algoritmo proposto em [73] é permitir a busca do controlador de menor ordem possível, tal que o sistema abordado seja estável. Isto permite, não somente a otimização da norma \mathcal{H}_∞ , mas também a simplificação da

estrutura do controlador. Por outro lado, este método não admite o tratamento direto de sistemas incertos, de restrições de alocação de pólos e de outros índices de desempenho, como por exemplo a norma \mathcal{H}_2 .

A solução proposta em [81] permite a alocação de pólos, utilizando-se controladores por realimentação dinâmica de saída através de um algoritmo genético canônico. Esta alternativa não utiliza a formulação LMI e a alocação de pólos é aproximada através de estratégias de penalização para os cromossomos não factíveis. A formulação descrita em [81] limita a flexibilidade do algoritmo, pois a adição de novas restrições, como, por exemplo, restrições de síntese \mathcal{H}_∞ , não é direta.

O algoritmo genético proposto em [68] permite a síntese de controladores multi-objetivo, incluindo o critério misto $\mathcal{H}_2/\mathcal{H}_\infty$. A formulação desta proposta não utiliza a descrição LMI do problema e codifica as possíveis soluções através do código binário de Gray. Com isso, cada restrição é tratada como um novo objetivo para o algoritmo, não permitindo a fácil incorporação de restrições de alocação de pólos, por exemplo. Além disso, esta abordagem não admite processos incertos, reduzindo o escopo de aplicação do algoritmo. As características das abordagens descritas em [81, 68] evidenciam claramente as desvantagens de não se utilizar a estrutura LMI na descrição do problema.

Um algoritmo de otimização local BMI para a síntese de algoritmos robustos mistos $\mathcal{H}_2/\mathcal{H}_\infty$ por realimentação de saída dinâmica é proposto em [76]. Esta abordagem, apesar de não utilizar a teoria evolutiva, é bastante interessante, pois além de permitir a otimização não convexa, utilizando diferentes matrizes de Lyapunov para as restrições do problema, garante que a solução encontrada é um ótimo local do problema. Partindo de uma solução inicial, que pode ser restritiva, o algoritmo encontra um ótimo local para o problema. Esta abordagem apresenta melhores desempenho e resultados, quando comparada às abordagens tradicionais, nas quais o problema BMI, não convexo, é transformado em problemas convexos, como por exemplo, a iteração $D - K$, discutida em [42]. Uma vantagem do algoritmo proposto em [76] é a garantia de convergência para um ótimo local. Partindo-se de diversas soluções no espaço de busca, pode-se encontrar uma solução satisfatória para o respectivo problema. Isto supõe, porém, que o algoritmo seja executado diversas vezes na busca por soluções ótimas de Pareto, característica intrínseca dos algoritmos de busca tradicionais. Além disso, o algoritmo proposto em [76] trata apenas da síntese de controladores de ordem completa, reduzindo-se a flexibilidade na busca por uma solução, em comparação com a abordagem deste trabalho.

Em muitos casos práticos, é desejável encontrar controladores que não só satisfaçam

os critérios de desempenho \mathcal{H}_2 e \mathcal{H}_∞ , mas que também sejam, eles próprios, estáveis. Controladores instáveis tendem a ser sensíveis às incertezas e às não linearidades do sensor e atuador [25]. Uma aplicação dos algoritmos genéticos na síntese de controladores mistos $\mathcal{H}_2/\mathcal{H}_\infty$ é descrita no trabalho de Campos-Delgado e Zhou, que se desenvolve sobre a hipótese de estabilidade do controlador. Esta abordagem trata apenas de processos precisamente conhecidos e não considera restrições de alocação de pólos. A extensão da teoria apresentada em [25] para o caso incerto não é, porém, trivial.

Uma aplicação particularmente interessante dos algoritmos genéticos na síntese de controladores robustos é apresentada por Takahashi *et al.* em [125]. Através de um AG, esta abordagem permite a busca por controladores robustos factíveis. Diferentes estruturas de controle podem ser utilizadas, entre elas controladores por realimentação dinâmica de saída, controladores de ordem reduzida e controladores LPV. Utilizando-se a formulação LMI, a proposta de Takahashi *et al.* pressupõe funções de Lyapunov dependentes de parâmetros para o domínio de incerteza do problema. Com isso, melhores soluções podem ser encontradas. Apesar de não considerar a otimização de índices de desempenho, como o controle \mathcal{H}_2 ou \mathcal{H}_∞ , o trabalho de Takahashi *et al.* permite a síntese de controladores para os casos onde o critério de estabilidade quadrática é muito restritivo.

Uma das abordagens mais próximas à desenvolvida neste trabalho é descrita em [100]. Pedersen *et al.* oferecem um algoritmo evolutivo para a síntese de controladores multi-objetivo $\mathcal{H}_2/\mathcal{H}_\infty$, utilizando-se a representação em desigualdades matriciais. Da mesma forma que o presente trabalho, o algoritmo proposto em [100] transforma o problema bilinear matricial em um problema convexo, permitindo sua solução. Não é necessária a hipótese de uma matriz comum de Lyapunov para todas as restrições do problema. A principal diferença do algoritmo proposto em [100], em relação à abordagem proposta neste trabalho, é a consideração de sistemas discretos e precisamente conhecidos. Além disso, não são consideradas restrições de alocação de pólos.

Abordagens bastante similares à proposta neste trabalho são descritas por Gonçalves *et al.* em [62, 63]. Estas buscam solucionar o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$ por realimentação de estado e de saída dinâmica, respectivamente. Os sistemas estudados em [62, 63] são contínuos, lineares, invariantes no tempo e sujeitos a incertezas do tipo politopo. Consideram-se ainda restrições de alocação de pólos. Nenhuma hipótese é admitida com relação à ordem do controlador a ser sintetizado. Nesta solução, os autores utilizam uma matriz de Lyapunov comum para o domínio de incerteza, porém diferentes matrizes de Lyapunov para as diferentes restrições do problema. Para se solucionar o

problema, Gonçalves *et al.* [62, 63] utilizam a formulação LMI e o algoritmo genético com operador de cruzamento polarizado-real, descrito em [126]. A diferença entre algoritmos propostos em [62, 63] e os desenvolvidos neste trabalho está no método de geração da população inicial. Com esta finalidade, propõe-se em [62, 63] um algoritmo de cone-elipsoidal, baseado em informações de gradiente. A idéia é, a partir dos vértices do politopo, encontrar controladores factíveis para o problema incerto, analisando-se então a factibilidade robusta dos controladores encontrados. Esta proposta não garante, porém, que os controladores sintetizados para os vértices do politopo de incertezas respeitam a condição de robustez do problema abordado. Este fato pode inviabilizar a geração do conjunto de soluções iniciais. O método de inicialização utilizado neste trabalho é descrito em [35]. Esta técnica, por outro lado, apesar de restritiva ao impor algumas estruturas matriciais fixas, permite considerar todo o domínio de incerteza na geração da população inicial. Assim, garante-se que os controladores encontrados são factíveis do ponto de vista robusto.

Apesar de inúmeras as aplicações dos AEs na teoria de controle, ainda são poucas as referências que utilizam os algoritmos evolutivos em conjunto com a formulação LMI na solução do problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$. A grande vantagem desta abordagem é a não utilização de funções comuns de Lyapunov para as diferentes restrições associadas ao problema. A desconsideração desta hipótese resulta em uma abordagem menos restritiva, pois pode-se explorar um espaço de busca mais amplo. Apesar dos AEs não possuírem provas formais de convergência e de não garantirem que a solução encontrada representa o controlador ótimo para o problema, a prática evidencia que a abordagem é eficaz e fornece bons resultados, como se pode observar em [101, 97, 36, 63, 62, 100].

Embora a hipótese de uma função comum de Lyapunov para as diferentes restrições não seja considerada, a abordagem proposta neste trabalho assume o conceito de estabilidade quadrática, com funções de Lyapunov comuns para todo o domínio de incerteza, em cada tipo de restrição. Uma possível extensão é relaxar também esta hipótese, agregando-se, por exemplo, o conceito de funções de Lyapunov dependentes de parâmetros. Assim, é provável que melhores soluções possam ser encontradas.

A abordagem apresentada, utilizando LMIs e AEs, forma uma estrutura unificada de projeto de controladores robustos com custos $\mathcal{H}_2/\mathcal{H}_\infty$, com ou sem critérios de alocação de pólos, para realimentação de estado, de saída estática ou de saída dinâmica. Nenhuma restrição é assumida quanto à ordem do controlador projetado, podendo este ser de ordem

completa ou reduzida, no caso de realimentação de saída dinâmica. Isso permite ao projetista um amplo conjunto de opções de síntese em uma estrutura única e simples, aplicável a um conjunto diverso de problemas de controle.

Este trabalho está estruturado da seguinte forma: no capítulo 2, algumas definições e conceitos básicos da matemática são expostos, como por exemplo convexidade. Além disso, os conceitos de estabilidade quadrática e de sistemas incertos, bem como as LMIs e suas regiões de alocação de pólos, também são apresentadas no capítulo 2. No capítulo 3, os problemas de controle \mathcal{H}_2 , \mathcal{H}_∞ e o problema misto $\mathcal{H}_2/\mathcal{H}_\infty$ são abordados em conjunto com suas soluções LMI para o caso de realimentação de estado. A teoria dos Algoritmos Evolutivos é abordada no capítulo 4. Especial atenção é dedicada aos Algoritmos Genéticos, à Evolução Diferencial e ao algoritmo de Salomon, por serem estes o fundamento da solução proposta para o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$, no capítulo 5. Aplicações da abordagem proposta em diferentes exemplos encontrados na literatura e seus respectivos resultados são apresentados no capítulo 6. Finalmente, no capítulo 7, são discutidas algumas considerações finais, desafios enfrentados e algumas possíveis continuções deste trabalho.

2 Considerações Preliminares

Neste capítulo são descritos alguns conceitos estabelecidos na literatura, necessários ao desenvolvimento deste trabalho. Algumas propriedades dos conjuntos convexos e a definição de funcional convexo são apresentadas nas seções 2.1 e 2.2, respectivamente. Definições da teoria de otimização são expostas na seção 2.3. O conceito de função analítica é descrito na seção 2.4 e os espaços utilizados neste trabalho são descritos na seção 2.4. A definição formal, as características e as propriedades das LMIs são exploradas na seção 2.6. Por fim, algumas conclusões são apresentadas na seção 2.7.

2.1 Conjunto Convexo

Um conjunto S é dito convexo se um segmento de linha reta ligando dois pontos quaisquer de S também pertence a S .

Em outras palavras, se $x, y \in S$ então $\alpha x + (1 - \alpha)y \in S, \forall \alpha \in [0, 1], \forall x, y \in S$. O elemento $z = \alpha x + (1 - \alpha)y$ é chamado de combinação convexa dos elementos x e y .

Lema: Sejam S_1 e S_2 conjuntos convexos, então as seguintes propriedades são válidas:

1. $S_1 \cap S_2$ é um conjunto convexo;
2. $\alpha S_1 = \{x : x = \alpha y, y \in S_1\}$ é convexo;
3. $S_1 + S_2 = \{x_1 + x_2 : x_1 \in S_1, x_2 \in S_2\}$ é convexo;
4. $S_1 - S_2 = \{x_1 - x_2 : x_1 \in S_1, x_2 \in S_2\}$ é convexo.

2.2 Funcional Convexo

Um funcional $f(\bullet) : \mathcal{S} \subset \mathbb{R}^n \rightarrow \mathbb{R}$ é convexo sobre um subconjunto \mathcal{S} do \mathbb{R}^n se

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad (2.1)$$

para quaisquer $x, y \in \mathcal{S}$ e $\alpha \in [0, 1]$.

Se a desigualdade for estrita, diz-se que $f(\bullet)$ é estritamente convexo. Um funcional $f(\bullet)$ é côncavo se $-f(\bullet)$ for convexo. Qualquer norma define também um funcional convexo.

2.3 Otimização

2.3.1 Problema de Otimização

Diversos problemas na engenharia podem ser descritos como problemas de otimização. O objetivo deste tipo de problema é encontrar o vetor de variáveis de decisão $x \in \mathbb{R}^m$, que minimize (ou maximize) um índice de desempenho específico, considerando-se um conjunto restrições. Matematicamente, descrevem-se tais problemas como

$$\begin{aligned} & \underset{x \in \mathcal{X}}{\text{minimizar}} && f(x), \end{aligned} \tag{2.2}$$

onde $f(x)$ é uma função escalar no vetor $x \in \mathbb{R}^m$, denominada função objetivo. O conjunto \mathcal{X} é o conjunto de restrições formado por desigualdades na forma

$$f_i(x) \leq 0, \quad i = 1, \dots, p, \tag{2.3}$$

e representa um subconjunto do espaço \mathbb{R}^m . Sem perda de generalidade, pode-se considerar como problema de otimização apenas o caso de minimização da função $f(x)$. Os casos de maximização podem ser descritos como problemas de minimização de $-f(x)$.

2.3.2 Factibilidade

Uma solução x do problema de otimização restrito da função $f(x)$ dado por

$$\begin{aligned} & \underset{x \in \Omega}{\text{minimizar}} && f(x) \end{aligned} \tag{2.4}$$

é definida factível se $x \in \Omega$, caso contrário, a solução x é definida infactível.

2.3.3 Vizinhaça

Define-se a vizinhaça de tamanho $\varepsilon > 0$ do elemento x o conjunto de todo elemento y tal que, para uma determinada norma,

$$\|y - x\| \leq \varepsilon. \quad (2.5)$$

2.3.4 Ótimo Local

Define-se x_{opt} um ótimo local do problema de minimização descrito pela equação 2.2, se existe uma vizinhaça de x_{opt} de tamanho $\varepsilon > 0$, tal que $f(x_{opt}) \leq f(x)$ nesta vizinhaça, para todo $x \in \mathcal{X}$.

2.3.5 Ótimo Global

Define-se x_{opt} o ótimo global do problema de minimização descrito pela equação 2.2, se $f(x_{opt}) \leq f(x)$ para todo $x \in \mathcal{X}$. Neste caso, $f(x_{opt})$ é chamado de mínimo de f em \mathcal{X} .

2.4 Função Analítica

Seja um conjunto $S \subset \mathbb{C}$ aberto e seja a função $f(s)$ complexa e definida em S , tais que

$$f(s) : S \rightarrow \mathbb{C}. \quad (2.6)$$

A função $f(s)$ é dita analítica em um ponto s_0 em S se é diferenciável em s_0 e em qualquer ponto da vizinhaça de s_0 . A função $f(s)$ é dita analítica em S se é analítica em todos os pontos de S .

2.5 Espaços

2.5.1 Espaço de Hilbert

Um espaço de Hilbert é um espaço linear sobre \mathbb{C} , completo e com um produto interno definido.

2.5.2 Espaço \mathcal{L}_2

O espaço \mathcal{L}_2 é o espaço de Hilbert de todas as funções complexas $F(j\omega)$, $F : (-\infty < \omega < \infty) \rightarrow \mathbb{C}^{n \times m}$, quadraticamente integráveis, ou seja:

$$\int_{-\infty}^{\infty} \text{tr} [F(j\omega)^* F(j\omega)] d\omega < \infty \quad (2.7)$$

com o produto interno definido por

$$\langle F, G \rangle \triangleq \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr} [F(j\omega)^* G(j\omega)] d\omega, \quad (2.8)$$

para $F, G \in \mathcal{L}_2$. A norma induzida pelo produto interno em \mathcal{L}_2 é dada por

$$\|F\|_2 \triangleq \sqrt{\langle F, F \rangle}. \quad (2.9)$$

2.5.3 Espaço de Hardy \mathcal{H}_2

O espaço \mathcal{H}_2 é o subespaço fechado de \mathcal{L}_2 das funções $F(s)$, $F : \mathbb{C} \rightarrow \mathbb{C}^{n \times m}$, analíticas em $\text{Re}(s) > 0$ (semi-plano complexo direito) que satisfazem a condição de integrabilidade quadrática uniforme

$$\|F\|_2^2 \triangleq \sup_{\sigma > 0} \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr} [F(\sigma + j\omega)^* F(\sigma + j\omega)] d\omega \right\} < \infty. \quad (2.10)$$

Para as funções racionais com coeficientes reais em \mathcal{H}_2 , a norma induzida \mathcal{H}_2 é definida por

$$\|F\|_2^2 \triangleq \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr} [F(j\omega)^* F(j\omega)] d\omega. \quad (2.11)$$

2.5.4 Espaço \mathcal{L}_∞

O espaço \mathcal{L}_∞ é o espaço normado das funções $F(j\omega)$, $F : (-\infty < \omega < \infty) \rightarrow \mathbb{C}^{n \times m}$, que são essencialmente limitadas ($\sigma_{\max} [F(j\omega)] < \infty, \forall \omega$) com norma dada por

$$\|F\|_\infty \triangleq \text{ess sup}_{\omega \in \mathbb{R}} \sigma_{\max} [F(j\omega)]. \quad (2.12)$$

2.5.5 Espaço de Hardy \mathcal{H}_∞

O espaço \mathcal{H}_∞ é o subespaço fechado de \mathcal{L}_∞ das funções $F(s)$, $F : \mathbb{C} \rightarrow \mathbb{C}^{n \times m}$, analíticas e limitadas em $\text{Re}(s) > 0$ ($\sigma_{\max}[F(s)] < \infty, \text{Re}(s) > 0$). A norma \mathcal{H}_∞ é definida como

$$\|F\|_\infty \triangleq \sup_{\text{Re}(s) > 0} \sigma_{\max}[F(s)] \quad (2.13)$$

e para as funções racionais com coeficientes reais em \mathcal{H}_∞ , a norma \mathcal{H}_∞ é dada por

$$\|F\|_\infty = \sup_{\omega \in \mathbb{R}} \sigma_{\max}[F(j\omega)]. \quad (2.14)$$

2.6 Desigualdades Matriciais Lineares

2.6.1 Introdução

Diversos problemas na engenharia de controle são descritos na forma de problemas de otimização. O objetivo destes problemas é ajustar os ganhos do controlador para que o mesmo satisfaça determinados índices de desempenho. Muitas vezes, estes índices são conflitantes e, assim, necessita-se estabelecer uma relação de compromisso entre os mesmos, na busca pela solução mais adequada. Uma abordagem bastante difundida na formulação dos problemas de otimização na engenharia de controle são as Desigualdades Lineares Matriciais (LMIs). Dentre as principais características desta abordagem ressaltam-se a robustez na descrição de problemas com incertezas e facilidade na formulação de problemas multi-objetivo.

Ao contrário do que outros autores propõem, o estudo das LMIs provavelmente começou com o trabalho fundamental de Aleksandr Mikhailovich Lyapunov e seus ensaios sobre a estabilidade dos movimentos [117]. Nos anos 60, as LMIs obtiveram especial atenção pela sua aplicação na teoria de controle, mas foi somente recentemente que surgiram métodos de pontos interiores eficientes na solução destes problemas. Como resultado, um impulso no estudo da teoria de controle robusto foi observado.

O conceito básico associado aos métodos LMIs é a transformação do problema abordado em um problema de programação semi-definida (SDP), ou seja, em um problema de otimização com objetivo linear e restrições semi-definidas positivas, envolvendo matrizes simétricas que são afins nas variáveis de decisão [60].

2.6.2 Definição

A idéia básica da formulação via desigualdades matriciais lineares (LMIs) é descrever um problema com objetivo linear e restrições formuladas por LMIs. Uma restrição LMI para um vetor $x \in \mathbb{R}^m$ é dada por

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i \geq 0, \quad (2.15)$$

onde $x = \begin{bmatrix} x_1 & x_2 & \dots & x_l \end{bmatrix}^T \in \mathbb{R}^m$ é o vetor de variáveis de decisão e $F_i = F_i^T \in \mathbb{R}^{n \times n}$, $i = 0, \dots, m$, são matrizes conhecidas.

A matriz $F(x)$ é afim no vetor de variáveis de decisão $x_i = \begin{bmatrix} x_1 & x_2 & \dots & x_l \end{bmatrix}^T$. Estritamente, a equação (2.15) é denominada Desigualdade Matricial Afim - AMI. O termo LMI, originalmente utilizado por J. C. Willems, é, porém, universalmente aceito.

O problema de minimização

$$\min_{F(x) \geq 0} c^T x \quad (2.16)$$

onde $c \in \mathbb{R}^m$ e $F \geq 0$ é chamado de programação semi-definida.

A LMI (2.15) é uma restrição convexa em x , *i.e.*, o conjunto $\{x : F(x) \geq 0\}$ é convexo. Múltiplas LMIs $F_1(x) > 0, \dots, F_p(x) > 0$ podem ser expressas como uma única LMI dada por

$$\text{diag}(F_1(x) \geq 0, \dots, F_p(x) \geq 0). \quad (2.17)$$

Assim, nenhuma distinção é feita entre uma única LMI e um conjunto destas. Quando as matrizes F_i são diagonais, a LMI $F(x) \geq 0$ representa um conjunto de inequações lineares.

Desigualdades matriciais formadas pela combinação de um conjunto de variáveis matriciais X_1, X_2, \dots, X_K , são também referidas como LMIs, uma vez que podem ser expressas na forma (2.15), em termos das componentes de X_1, X_2, \dots, X_K .

A formulação LMI é interessante pelas seguintes razões [60]:

- **convexidade e solução numérica eficiente:** como os problemas na forma LMI se tornam convexos, estes podem ser solucionados utilizando-se métodos de pontos interiores para otimização convexa SDP [60]. Estes métodos possuem taxa de convergência polinomial, fornecendo uma solução numérica eficiente para problemas que não possuem solução analítica, ou que possuem solução analítica restritiva ou complicada;

- **robustez sobre as incertezas:** esta abordagem é adequada para problemas com parâmetros incertos. Baseando-se em uma descrição determinista das incertezas com estrutura e limites conhecidos, um procedimento sistemático permite a descrição dos problemas na forma SDP, resultando em uma solução robusta. Esta característica tem implicação especial nos problemas de engenharia onde constantemente erros de medida, modelagem etc. estão presentes;
- **problemas multi-objetivo:** esta abordagem permite a imposição de diversos e possivelmente conflitantes objetivos e restrições. Com esta característica pode-se explorar relações de compromisso entre os critérios de desempenho, bem como analisar seus limites de factibilidade. Esta é uma vantagem significativa quando se compara a abordagem LMI aos métodos clássicos de otimização, que empregam apenas um critério para refletir um conjunto deles. A tarefa de se escolher o critério de otimização mais relevante nem sempre é trivial;
- **amplo escopo de aplicação:** as técnicas utilizadas na abordagem LMI vão muito além de problemas de controle e estimação. Esta característica permite um horizonte amplo na pesquisa fornecendo a aplicação de um mesmo método em diversos tipos de problemas.

Por estas características, a abordagem LMI foi escolhida como objeto de estudo neste trabalho. Além do problema abordado possuir restrições conflitantes, as normas \mathcal{H}_2 e \mathcal{H}_∞ , seu objetivo é linear e incertezas do tipo politopo são consideradas. Portanto, é bastante adequada a formulação do problema em termos de desigualdades matriciais.

2.6.3 Complemento de Schur

Sejam $Q(x) = Q(x)^T$, $R(x) = R(x)^T$ e $S(x)$ dependentes de x de forma afim. A LMI

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} > 0 \quad (2.18)$$

é equivalente às desigualdades matriciais

$$R(x) > 0, \quad Q(x) - S(x)R(x)^{-1}S(x)^T > 0 \quad (2.19)$$

ou, de forma equivalente,

$$Q(x) > 0, \quad R(x) - S(x)^T Q(x)^{-1} S(x) > 0. \quad (2.20)$$

Estes resultados podem ser gerados através da aplicação da transformação de congruência

$$\begin{bmatrix} I & -S(x)R(x)^{-1} \\ 0 & I \end{bmatrix} \quad (2.21)$$

em (2.15), para se obter (2.16), e aplicando-se a transformação de congruência

$$\begin{bmatrix} I & 0 \\ -S(x)^T Q(x)^{-1} & I \end{bmatrix} \quad (2.22)$$

em (2.15), para se obter (2.17).

O complemento de Schur permite transformar as desigualdades não lineares (2.16) e (2.17) na desigualdade linear (2.15).

2.6.4 Bounded Real Lemma

Seja um sistema S controlável com função de transferência $H(s) = C(s\mathbf{I} - A)^{-1}B + D$ e descrito pelas equações de estado

$$S : \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du, \end{cases} \quad (2.23)$$

Seja a função quadrática $s(u, y) = \gamma^2 u^T u - y^T y$, com $\gamma \geq 0$. Então as seguintes proposições são equivalentes [117]:

1. (S, s) é dissipativo;
2. o sistema de LMIs

$$\begin{aligned} X = X^T &\geq 0, \\ \begin{bmatrix} A^T X + XA + C^T C & XB + C^T D \\ B^T X + D^T C & D^T D - \gamma^2 \mathbf{I} \end{bmatrix} &\leq 0 \end{aligned} \quad (2.24)$$

é factível;

3. para todo $\omega \in \mathbb{R}$ com $\det(j\omega\mathbf{I} - A) \neq 0$, $H(j\omega)^* H(j\omega) \leq \gamma^2 \mathbf{I}$.

Além disso, $V(x) = x^T X x$ define uma função quadrática de Lyapunov se e somente se X satisfaz o sistema de LMIs (2.24). Considerando-se que $x(0) = 0$, A possui todos os autovalores no semi-plano esquerdo aberto e $u \in \mathcal{L}_2$, então, para X satisfazendo o sistema

de LMIs (2.24),

$$\sup_{u \in \mathcal{L}_2} \frac{\|y\|_2}{\|u\|_2} \leq \gamma. \quad (2.25)$$

O lado esquerdo da inequação (2.25) é o chamado ganho \mathcal{L}_2 ou norma induzida \mathcal{L}_2 do sistema (2.23). O ganho \mathcal{L}_2 é o menor $\gamma \geq 0$ tal que (2.25) seja satisfeita.

De outra forma, e no contexto de LMIs, pode-se obter o limite superior de γ que satisfaz (2.25) se e somente se existe X que satisfaz o sistema (2.24). Este lema fornece um teste de factibilidade, parametrizado em $\gamma > 0$, para determinar o ganho \mathcal{L}_2 , do sistema. Para maiores detalhes, incluindo a demonstração deste lema, referir-se a [117].

2.6.5 Regiões LMI de Alocação de Pólos e \mathcal{D} -estabilidade

Um importante problema na teoria e prática de controle é a síntese de controladores que, além de garantir estabilidade e desempenho para o sistema em malha fechada, posicionem seus pólos em uma região pré-determinada pelo projetista. A motivação para a determinação *a priori* da localização dos pólos reside no comportamento transitório do sistema. É consenso que um comportamento transitório satisfatório pode ser atingido com a adequada alocação dos pólos em malha fechada [29].

Quando o modelo do processo é precisamente conhecido, é possível determinar, sob a hipótese de controlabilidade, de forma exata, a localização dos pólos em malha fechada. Porém, quando o modelo do processo não é precisamente conhecido, isto não é possível. Busca-se então alocar os pólos em uma região definida no plano complexo. Este problema chamou a atenção de diversos autores e é vasta a literatura no assunto, como se pode conferir em [66, 29, 94, 14, 3] e nas referências lá citadas. As regiões de maior interesse são setores, faixas verticais e discos. Uma região de especial atenção, representada na figura 1, é a região $S(\alpha, r, \theta)$ dada pelo conjunto de números complexos $x + jy$ tais que

$$x < -\alpha < 0; \quad |x + jy| < r; \quad \text{tg}(\theta) x < -|y|. \quad (2.26)$$

Alocar os pólos na região S garante um decaimento mínimo α , um mínimo coeficiente de amortecimento $\zeta = \cos \theta$ e uma frequência natural não amortecida máxima $\omega_d = r \sin \theta$. Estes limites para os pólos do sistema forçam uma resposta transitória satisfatória.

A vantagem do uso das LMIs no problema de alocação de pólos é o tratamento das regiões de alocação como um conjunto de restrições e não como um conjunto de funções

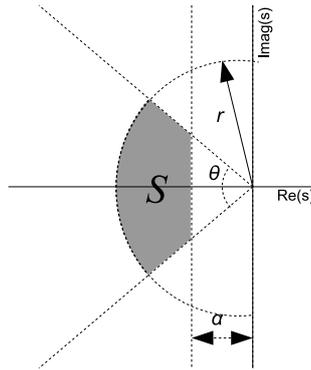


Figura 1: Região S de alocação de pólos.

objetivo do problema, proporcionando-se grande flexibilidade à abordagem. Como as restrições podem ser facilmente adicionadas ao problema, diversas LMIs de alocação de pólos em regiões simples podem ser usadas conjuntamente, construindo-se regiões complexas no semi-plano $Re(s) < 0$.

Os conceitos apresentados nesta seção foram extraídos principalmente dos trabalhos de Chilali e Gahinet [29, 55].

Seja \mathcal{D} uma região contida no semi-plano esquerdo do plano complexo. Um sistema dinâmico $\dot{x} = Ax$ é definido \mathcal{D} -estável se todos os seus pólos pertencem à região \mathcal{D} . Isto é, se todos os autovalores da matriz A pertencem a \mathcal{D} . Quando \mathcal{D} corresponde à totalidade do semi-plano complexo esquerdo, este conceito se reduz ao conceito de estabilidade assintótica e é descrito na forma de LMIs pelo teorema de Lyapunov. Assim sendo, A é assintoticamente estável se e somente se existe uma matriz simétrica $X_D > 0$, tal que

$$A^T X_D + X_D A < 0. \quad (2.27)$$

Gutman [64] expandiu o teorema de Lyapunov para uma variedade de regiões distintas no plano complexo esquerdo. As regiões consideradas por Gutman são regiões polinomiais da forma

$$\mathcal{D} = \left\{ z \in \mathbb{C} : \sum_{0 \leq k, l \leq m} c_{kl} z^k \bar{z}^l < 0 \right\}, \quad (2.28)$$

onde os coeficientes c_{kl} são reais e satisfazem $c_{kl} = c_{lk}$. Observa-se que as regiões polinomiais não são gerais e não descrevem qualquer região no plano complexo, como o caso da região S , na figura 1.

O principal resultado de Gutman [64, 29] demonstra que a matriz A é \mathcal{D} -estável se e

somente se existir uma matriz simétrica definida positiva X_D tal que

$$\sum_{k,l} c_{kl} A^k X_D (A^T)^l < 0. \quad (2.29)$$

Infelizmente, a síntese de controladores utilizando os resultados propostos por Gutman é complicada devido à sua caracterização polinomial. Para o tratamento da síntese de controladores através de LMIs é necessário manter a dependência afim das matrizes em malha fechada e do controlador. A função polinomial de Gutman não é necessariamente convexa nem facilmente transformada em LMI. Essa foi a motivação para uma outra abordagem relativa à representação de regiões de \mathcal{D} -estabilidade.

Uma região \mathcal{D} do plano complexo é chamada de região LMI se existe uma matriz simétrica $t = [t]_{rs} \in \mathbb{R}^{q \times q}$ e uma matriz $\tau = [\tau_{rs}] \in \mathbb{R}^{q \times q}$ tais que

$$\mathcal{D} = \{z \in \mathbb{C} : f_{\mathcal{D}}(z) < 0\}, \quad (2.30)$$

onde

$$f_{\mathcal{D}}(z) \triangleq t + z\tau + \bar{z}\tau^T \quad (2.31)$$

$$= [t_{rs} + \tau_{rs}z + \tau_{sr}\bar{z}]_{1 \leq r,s \leq q}. \quad (2.32)$$

Assim, uma região LMI é um sub-conjunto do plano complexo e é representada por uma LMI em z e \bar{z} . Como resultado, tem-se que regiões LMIs são convexas e simétricas em relação ao eixo real, uma vez que para qualquer $z \in \mathcal{D}$, $f_{\mathcal{D}}(\bar{z}) = \overline{f_{\mathcal{D}}(z)} < 0$.

De maneira interessante, existe um teorema paralelo ao teorema de Gutman para regiões LMI. A alocação de pólos em uma região LMI é caracterizada pela matriz

$$\begin{aligned} M_{\mathcal{D}}(A, X_D) &\triangleq t \otimes X_D + \tau \otimes (AX_D) + \tau^T \otimes (AX_D)^T \\ M_{\mathcal{D}}(A, X_D) &= [t_{rs}X_D + \tau_{rs}AX_D + \tau_{sr}X_D A^T]_{1 \leq r,s \leq q} < 0 \end{aligned} \quad (2.33)$$

onde o símbolo \otimes denota o operador de Kronecker. Uma matriz A é \mathcal{D} -estável se e somente se existe uma matriz X tal que

$$M_{\mathcal{D}}(A, X_D) < 0 \text{ e } X_D > 0. \quad (2.34)$$

A relação entre $M_{\mathcal{D}}(A, X_D)$ na equação (2.33) e $f_{\mathcal{D}}$ na equação (2.31) é dada pela substituição $(X_D, AX_D, X_D A^T) \leftrightarrow (1, z, \bar{z})$ [29]. Diversas regiões simétricas com relação ao eixo real podem ser descritas como regiões LMI. As principais - o disco, a faixa vertical

e o setor cônico, formadoras da região S na figura 1 - são dadas por:

1. disco com centro em $(q, 0)$ e raio r :

$$f_{\mathcal{D}}(z) = \begin{bmatrix} -r & -q - z \\ -q - \bar{z} & -r \end{bmatrix}; \quad (2.35)$$

2. faixa vertical com limites $h_1 \leq \text{Re}(z) \leq h_2$:

$$f_{\mathcal{D}}(z) = \begin{bmatrix} 2h_1 - (z + \bar{z}) & 0 \\ 0 & (z + \bar{z}) - 2h_2 \end{bmatrix}; \quad (2.36)$$

3. setor cônico com vértice na origem e ângulo interno 2θ :

$$f_{\mathcal{D}}(z) = \begin{bmatrix} \sin\theta(z + \bar{z}) & \cos\theta(z - \bar{z}) \\ -\cos\theta(z - \bar{z}) & \sin\theta(z + \bar{z}) \end{bmatrix}. \quad (2.37)$$

Utilizando-se a transformação $f_{\mathcal{D}} \leftrightarrow M_{\mathcal{D}}$, tem-se que as regiões LMI descritas assumem a seguinte forma:

1. disco com centro em $(q, 0)$ e raio r :

$$M_{\mathcal{D}}(A, X_D) = \begin{bmatrix} -rX_D & qX_D + AX_D \\ qX_D + X_DA^T & -rX_D \end{bmatrix} < 0; \quad (2.38)$$

2. faixa vertical com limites $h_1 \leq \text{Re}(z) \leq h_2$:

$$M_{\mathcal{D}}(A, X_D) = \begin{cases} AX_D + X_DA^T - 2h_1X_D > 0 \\ AX_D + X_DA^T - 2h_2X_D < 0; \end{cases} \quad (2.39)$$

3. setor cônico com vértice na origem e ângulo interno 2θ :

$$M_{\mathcal{D}}(A, X_D) = \begin{bmatrix} \sin\theta(AX_D + X_DA^T) & \cos\theta(AX_D - X_DA^T) \\ \cos\theta(X_DA^T - AX_D) & \sin\theta(AX_D + X_DA^T) \end{bmatrix} < 0. \quad (2.40)$$

Nota-se que a equação de estabilidade de Lyapunov, um caso especial onde \mathcal{D} é definido como sendo o semi-plano complexo esquerdo, possui função característica $f_{\mathcal{D}}(z) = z + \bar{z}$.

A abordagem LMI é bastante flexível. Para a construção da região S na figura 1 basta-se utilizar as equações (2.38), (2.39) e (2.40) simultaneamente. Diversas regiões mais

complexas podem ser construídas apenas com a utilização das LMIs básicas apresentadas, conferindo grande flexibilidade para a solução de diversos problemas de controle distintos. Aos controladores que satisfazem as restrições de \mathcal{D} -estabilidade de um dado problema, denominam-se controladores \mathcal{D} -estáveis.

2.7 Conclusão

Neste capítulo as principais definições matemáticas, fundamentos para este trabalho, foram abordadas. Além disso, as LMIs e suas propriedades mais significativas foram descritas. Salientou-se suas características, no intuito de justificar o seu uso na solução do problema proposto no capítulo 3. Uma importante vantagem na utilização de LMIs para a formulação dos problemas é a propriedade de convexidade. Esta permite uma solução eficiente do ponto de vista computacional e, além disso, garante que o ótimo global do problema convexo seja encontrado.

3 Sistemas Incertos e Controle Robusto $\mathcal{H}_2/\mathcal{H}_\infty$

Neste capítulo abordam-se os conceitos de sistemas incertos e estabilidade quadrática, bem como os problemas de controle robusto utilizados neste trabalho. Uma introdução sobre os problemas deste capítulo é realizada na seção 3.1. Na seção 3.2, descrevem-se o conceito de sistemas incertos e a definição de incertezas paramétricas do tipo politopo. A definição de estabilidade quadrática é apresentada na seção 3.3. Apesar de restritiva, a abordagem quadrática permite, para o caso de realimentação de estado, o tratamento direto dos sistemas incertos. Os problemas de controle robusto \mathcal{H}_2 e \mathcal{H}_∞ são discutidos nas seções 3.4 e 3.5, respectivamente. O problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$, sujeito à \mathcal{D} -estabilidade, foco deste trabalho, é apresentado na seção 3.6. Finalmente, uma breve conclusão do capítulo é realizada na seção 3.7.

3.1 Introdução

Um problema fundamental, de interesse tanto teórico quanto prático, que reside no centro da teoria de controle, é a síntese de controladores que proporcionam um desempenho aceitável não somente para um único processo e sob entradas conhecidas, mas para toda uma família de processos sob diversos tipos de entradas e distúrbios.

A importância deste problema tem sido amplamente reconhecida e, com o passar dos anos, diversos métodos de solução foram desenvolvidos [41]. A solução proposta inicialmente foi uma descrição matemática formal utilizando a otimização de um índice de desempenho. Dois destes métodos, o método da sensibilidade e o problema linear quadrático Gaussiano (LQG), dominaram a teoria de controle ótimo nos anos 60 e 70. Estas teorias permitem, respectivamente, pequenos distúrbios sobre o modelo nominal [33, 51] e uma descrição estatística dos distúrbios desconhecidos [5].

Com a evolução científica, percebeu-se que o problema LQG poderia ser reescrito

de forma determinística e mais geral, dando origem ao problema de controle \mathcal{H}_2 . Este problema recebeu grande atenção da comunidade e diversas soluções foram apresentadas [80]. As principais razões para o desenvolvimento desta teoria são: histórica, pois foi um dos primeiros métodos de síntese de controladores ótimos com solução; e prática, pois este problema pode ser resolvido utilizando-se métodos computacionais confiáveis e eficientes [60].

Em paralelo com a teoria \mathcal{H}_2 , surgiu a teoria de controle \mathcal{H}_∞ no trabalho pioneiro de G. Zames na década de 80 [135]. Introduzido na forma de um problema de otimização da norma \mathcal{H}_∞ , Zames considerou a minimização do efeito de distúrbios em sistemas realimentados. Mais tarde, percebeu-se que a norma \mathcal{H}_∞ poderia ser utilizada não somente na quantificação de distúrbios, mas também na medida da robustez do modelo face a incertezas. Reconheceu-se assim a norma \mathcal{H}_∞ como uma importante medida de desempenho do controlador.

Desejando-se uma abordagem mais completa, surgiu a possibilidade de se considerar os problemas \mathcal{H}_2 e \mathcal{H}_∞ em conjunto, originando o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$. O critério de desempenho misto $\mathcal{H}_2/\mathcal{H}_\infty$ se popularizou rapidamente, pois fornece uma medida bastante representativa da robustez do modelo com relação a incertezas, bem como o nível de influência de distúrbios nos sinais de saída.

Os conceitos de sistemas incertos e estabilidade quadrática, fundamentais para a formulação dos problemas de controle robusto, são discutidos a seguir. Na seqüência, apresentam-se os problemas de controle \mathcal{H}_2 , \mathcal{H}_∞ e, por fim, o problema central deste trabalho: o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$, em uma abordagem LMI.

3.2 Sistemas Incertos

Em diversos problemas de controle, os dados do problema definido por (2.2) não são totalmente conhecidos, mas incertos. A formulação necessita então ser adaptada a estes problemas sujeitos a incertezas.

A principal motivação para se considerar as incertezas é que os fenômenos do mundo real são, em geral, bastante complexos (não lineares, variantes no tempo etc.) e por isso são necessárias aproximações. Uma maneira elegante de se considerar estes problemas complexos é formular as aproximações como incertezas.

Nos problemas de otimização em que os dados do sistema não são totalmente conheci-

dos, a função objetivo $f(x)$ depende não somente do vetor x mas também de um vetor de incertezas δ , que pertence a um conjunto de perturbações admissíveis Δ . Este conjunto denomina-se conjunto de factibilidade robusta do problema e é definido como

$$\mathcal{X}(\Delta) = \{x \in X \mid f(x, \delta) \leq 0, \forall \delta \in \Delta\}. \quad (3.1)$$

O problema de análise de uma solução x consiste em identificar se x pertence ao conjunto de factibilidade robusta do problema. O problema abordado neste trabalho é um problema de estabilidade robusta, que busca uma solução x pertencente ao conjunto $\mathcal{X}(\Delta)$. A seguir, apresenta-se o conceito de sistemas incertos.

Seja um sistema dinâmico contínuo linear e invariante no tempo, sujeito a incertezas paramétricas, definido pela equação de estado

$$\dot{x}(t) = A(r(t))x(t) + B_2(s(t))u(t), \quad (3.2)$$

onde $x(t) \in \mathbb{R}^n$ é o vetor de estado, $u(t) \in \mathbb{R}^m$ é o vetor de variáveis de controle, $A(r(t)) \in \mathbb{R}^{n \times n}$ é a matriz dinâmica do sistema e $r(t) \in \mathcal{R} \subset \mathbb{R}^{n_r}$ e $s(t) \in \mathcal{S} \subset \mathbb{R}^{n_s}$ são os vetores de incertezas paramétricas das matrizes dinâmica e de controle, respectivamente. Os conjuntos \mathcal{S} e \mathcal{R} são compactos e as funções $r(t) : [0, \infty) \rightarrow \mathcal{R}$ e $s(t) : [0, \infty) \rightarrow \mathcal{S}$ são consideradas mensuráveis no sentido de Lebesgue [35].

As matrizes $A(r(t))$ e $B_2(s(t))$ podem ser decompostas utilizando-se as matrizes do sistema nominal (A, B_2) , quando este é definido, como segue

$$\begin{cases} A(r(t)) = A + \Delta A(r(t)) \\ B_2(s(t)) = B_2 + \Delta B_2(s(t)), \end{cases} \quad (3.3)$$

onde $\Delta A(r(t))$ e $\Delta B_2(s(t))$ correspondem às matrizes de incertezas. Por hipótese, o par (A, B_2) é considerado constante e estabilizável [9].

A maneira como as funções $r(t)$ e $s(t)$ são definidas caracteriza os diferentes tipos de incerteza. As hipóteses sobre $r(t)$ e $s(t)$ e sua relação com as matrizes $A(r)$ e $B(s)$, para o caso de incertezas do tipo politopo, são discutidas na seqüência.

A partir do modelo geral (3.2), considera-se a hipótese de que as componentes dos vetores de incertezas $r(t) \in \mathcal{R}$ e $s(t) \in \mathcal{S}$ são limitadas superior e inferiormente, como segue:

$$\begin{aligned} \underline{r}_i &\leq r_i \leq \bar{r}_i, \quad i = 1, 2, \dots, n_r \\ \underline{s}_j &\leq s_j \leq \bar{s}_j, \quad j = 1, 2, \dots, n_s. \end{aligned} \quad (3.4)$$

Os domínios \mathcal{R} e \mathcal{S} , definidos por (3.4), são poliedros convexos com $N = 2^{n_r}$ e $M = 2^{n_s}$ vértices, respectivamente.

Com a hipótese adicional que os elementos das matrizes $A(r)$ e $B_2(s)$ são funções lineares de $r(t)$ e $s(t)$, (3.4) define também as matrizes “vértices” A_i e B_{2j} , $i = 1, 2, \dots, N$, e $j = 1, 2, \dots, M$, associadas aos vértices dos poliedros \mathcal{R} e \mathcal{S} . A este tipo de incerteza denomina-se incerteza paramétrica do tipo politopo, ou simplesmente, incerteza politópica.

Assim, as matrizes A e B_2 pertencem aos domínios convexos fechados definidos por:

$$\mathcal{D}_A = \left\{ A \in \mathbb{R}^{n \times n}; A = \sum_{i=1}^N \alpha_i A_i, \sum_{i=1}^N \alpha_i = 1, \alpha_i \geq 0 \right\}, \quad (3.5)$$

$$\mathcal{D}_B = \left\{ B_2 \in \mathbb{R}^{n \times m}; B_2 = \sum_{j=1}^M \beta_j B_{2j}, \sum_{j=1}^M \beta_j = 1, \beta_j \geq 0 \right\}. \quad (3.6)$$

Com base nas definições (3.5) e (3.6), o sistema geral (3.2) sujeito a incertezas politópicas pode ser descrito como

$$\begin{cases} x(t) = Ax(t) + B_2u(t) \\ A \in \mathcal{D}_A, B_2 \in \mathcal{D}_B. \end{cases} \quad (3.7)$$

Os sistemas considerados neste trabalho são incertos, invariantes no tempo e descritos por (3.7), onde as incertezas são do tipo politopo.

3.3 Estabilidade Quadrática

O sistema (3.2) é quadraticamente estabilizável se existe uma lei de controle contínua, função do estado, $p(\bullet) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ com $p(0) = 0$, uma matriz P simétrica e definida positiva e uma constante real $\alpha > 0$ que satisfazem

$$x^T [A(r(t))^T P + PA(r(t))] x + 2x^T P B_2(s(t)) p(x) \leq -\alpha \|x\|^2, \quad (3.8)$$

$$\forall x \neq 0 \in \mathbb{R}^n, \forall r(t) \in \mathcal{R}, \forall s(t) \in \mathcal{S}, \forall t.$$

Pode-se demonstrar que se a inequação (3.8) é verificada então $v(x) = x^T P x$ é uma função de Lyapunov para o sistema em malha fechada (3.2) que garante a estabilidade assintótica do ponto de equilíbrio $x = 0$, quaisquer que sejam as incertezas admissíveis. Assim, o conceito de estabilidade quadrática estabelece que uma mesma função quadrática

de Lyapunov seja válida para todo o domínio de incerteza admissível. A estabilidade quadrática é uma condição suficiente para a estabilidade assintótica do sistema incerto (3.2) [35, 9].

Para o sistema (3.7), o conceito de estabilidade quadrática se reduz à existência de uma matriz $P \in \mathbb{R}^{n \times n}$, simétrica e definida positiva, tal que

$$\begin{aligned} A^T P + P A &< 0 \\ \forall A \in \mathcal{D}_A, \forall B_2 \in \mathcal{D}_B. \end{aligned} \quad (3.9)$$

Observando-se a convexidade da inequação de Lyapunov (3.9) em relação à matriz P , pode-se considerar apenas os vértices do conjunto \mathcal{D}_A para a análise de estabilidade quadrática do sistema (3.7). Assim, um sistema sujeito a incertezas do tipo politopo (3.7) é quadraticamente estável se e somente se existe uma matriz $P \in \mathbb{R}^{n \times n}$, simétrica e definida positiva, tal que

$$A_i^T P + P A_i < 0, \quad i = 1, \dots, N, \quad (3.10)$$

onde A_i são as matrizes “vértices” do domínio convexo \mathcal{D}_A . Nota-se que (3.10) representa um conjunto de LMIs facilmente solucionado em P .

3.4 Problema \mathcal{H}_2

Durante as últimas décadas, diversas técnicas de análise e síntese de controladores para sistemas multi-variáveis têm sido estudadas. A interpretação estocástica da teoria de controle ótimo \mathcal{H}_2 , conhecida como teoria de controle linear quadrático Gaussiano (LQG), é uma das técnicas mais recentes e poderosas que apareceram. Como o controle LQG possibilitou a síntese de controladores dinâmicos multi-variáveis de forma metódica, a teoria LQG veio a ser denominada teoria de controle ótimo moderno [113].

No problema de controle LQG, dado um processo descrito por equações de estado, assume-se que os distúrbios e ruídos de medida são processos estocásticos Gaussianos com potência e densidade espectral conhecidas. As especificações de síntese são transformadas em um critério de desempenho quadrático e consistem de algumas variáveis de estado e sinais de entrada. O objetivo da síntese é minimizar o critério de desempenho utilizando um controlador por realimentação de estado ou de saída, garantindo ao mesmo tempo a estabilidade em malha fechada. A teoria de controle LQG é apresentada a seguir.

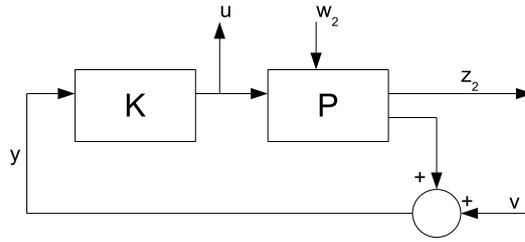


Figura 2: Sistema em malha fechada para otimização LQG.

Seja o processo P do sistema representado pela figura 2 e descrita pela equação de estado

$$\dot{x}(t) = Ax(t) + B_1 w_2(t) + B_2 u(t), \quad (3.11)$$

onde o sinal de distúrbios externos $w_2(t)$ corresponde a um ruído branco¹. A saída controlada $z_2(t)$ e a saída medida $y(t)$ são dadas por

$$\begin{aligned} z_2(t) &= D_{22}x(t) \\ y(t) &= C_y x(t) + v(t), \end{aligned} \quad (3.12)$$

onde o ruído branco $v(t)$ representa o ruído do sensor. Assume-se que as matrizes do sistema possuem dimensões adequadas e que a lei de controle é dada por $u(t) = Ky(t)$. O problema de otimização quadrática linear Gaussiana é dado por:

Encontrar um controlador K que estabiliza o sistema em malha fechada e minimiza

$$\lim_{t \rightarrow \infty} E \left(z_2^T(t) z_2(t) + u^T(t) u(t) \right), \quad (3.13)$$

onde $E(\bullet)$ indica a esperança matemática.

Este famoso problema foi amplamente estudado e sua solução é bem conhecida [5]. Sob as hipóteses de controlabilidade e observabilidade, o controlador ótimo K é a interconexão de um filtro de Kalman [80] e a lei de controle de realimentação de estado. O filtro de Kalman e a respectiva lei de controle podem ser obtidos através da solução de duas equações algébricas de Riccati.

Apesar da formulação do problema de controle LQG e do problema \mathcal{H}_2 serem matematicamente equivalentes, a formulação determinística da teoria \mathcal{H}_2 é preferível por dois motivos: a solução do problema LQG requer certa intimidade com a teoria de processos estocásticos, sendo que o controle \mathcal{H}_2 é menos complexo que a teoria LQG; pode-se

¹Ruído branco é um sinal aleatório com distribuição Gaussiana, média zero, potência e densidade espectral conhecidas.

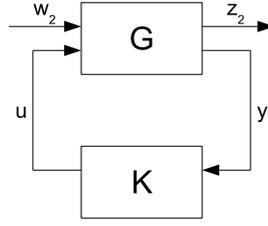


Figura 3: Sistema genérico realimentado - caso \mathcal{H}_2 .

considerar que o problema LQG não é inerentemente estocástico, uma vez que pode ser descrito de forma equivalente pela teoria determinística [113].

O caráter estocástico do problema LQG é eliminado quando o mesmo é transformado no problema “padrão” de controle mostrado na figura 3 e abordado na forma de um problema de minimização da norma \mathcal{H}_2 do sistema em malha fechada.

Observa-se que a configuração apresentada na figura 2 é um caso especial daquela na figura 3, onde G representa o processo generalizado. A entrada $w_2(t)$ é composta do sinal de distúrbio externo $w_2(t)$ e do sinal de ruído do sensor $v(t)$. O sinal $z_2(t)$ é formado pelos sinais de saída controlada $z_2(t)$ e de controle $u(t)$. A saída medida é dada por $y(t)$ e $u(t)$ é o sinal de controle. Ao se redefinir os sinais $z_2(t)$ e $w_2(t)$, tais que

$$z_2(t) \triangleq \begin{bmatrix} z_2(t) \\ u(t) \end{bmatrix}, \quad w_2(t) \triangleq \begin{bmatrix} w_2(t) \\ v(t) \end{bmatrix}, \quad (3.14)$$

o sistema da figura 2 se reduz ao sistema da figura 3. O processo generalizado G é descrito pela matriz de transferência

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}, \quad (3.15)$$

onde

$$\begin{aligned} G_{11} &= \begin{bmatrix} D_{22}(s\mathbf{I} - A)^{-1}B_1 & 0 \\ 0 & 0 \end{bmatrix}, & G_{12} &= \begin{bmatrix} D_{22}(s\mathbf{I} - A)^{-1}B_2 \\ \mathbf{I} \end{bmatrix}, \\ G_{21} &= \begin{bmatrix} C_y(s\mathbf{I} - A)^{-1}B_1 & \mathbf{I} \end{bmatrix}, & G_{22} &= C_y(s\mathbf{I} - A)^{-1}B_2. \end{aligned} \quad (3.16)$$

Para o sistema realimentado genérico da figura 3, o problema de controle LQG é dado por:

Encontrar o controlador K que estabilize o sistema em malha fechada e minimize

$$\lim_{t \rightarrow \infty} E(z_2^T(t)z_2(t)). \quad (3.17)$$

A saída z_2 , na figura 3, pode ser expressa via transformada de Laplace como

$$z_2 = T_{z_2 w_2}(s)w_2, \quad (3.18)$$

onde $T_{z_2 w_2}(s)$ é a matriz de transferência em malha fechada dada por

$$T_{z_2 w_2}(s) = G_{11}(s) + G_{12}(s) (\mathbf{I} - K(s)G_{22}(s))^{-1} K(s)G_{21}(s). \quad (3.19)$$

Se w_2 corresponde aos distúrbios externos modelados como ruído branco, com matriz de intensidade \mathbf{I} , e a função de transferência $T_{z_2 w_2}(s)$ é estável, então

$$\lim_{t \rightarrow \infty} E(z_2^T(t)z_2(t)) = \frac{1}{2\pi} \text{tr} \int_{-\infty}^{\infty} T_{z_2 w_2}^T(-j\omega)T_{z_2 w_2}(j\omega)d\omega. \quad (3.20)$$

O lado direito dessa expressão corresponde ao quadrado da norma \mathcal{H}_2 da matriz de transferência estável $T_{z_2 w_2}(s)$, dada por

$$\|T_{z_2 w_2}\|_2 = \left(\frac{1}{2\pi} \text{tr} \int_{-\infty}^{\infty} T_{z_2 w_2}^T(-j\omega)T_{z_2 w_2}(j\omega)d\omega \right)^{\frac{1}{2}}. \quad (3.21)$$

Assim, o problema de otimização LQG é tratado como a minimização \mathcal{H}_2 do sistema em malha fechada descrito por $T_{z_2 w_2}(s)$. Pode-se então redefinir o problema de controle ótimo \mathcal{H}_2 como:

Encontrar um controlador K tal que a matriz de transferência $T_{z_2 w_2}(s)$ do sistema em malha fechada seja estável e possua norma \mathcal{H}_2 , definida por (3.21), mínima.

Diversas soluções estão disponíveis para o problema de controle \mathcal{H}_2 . A solução inicial do problema \mathcal{H}_2 padrão foi proposta por Doyle, em [43]. O ponto de partida da solução apresentada por Doyle [43, 136] é a própria representação no espaço de estados e resulta na solução de duas equações algébricas de Riccati.

Desde a solução inicial, diversas outras soluções foram propostas por diferentes autores. Hunt, Sebek e Kucera [72] propuseram uma solução polinomial matricial baseada em equações Diofantinas. Meinsma [90] descreve uma solução no domínio da frequência que utiliza fatorações de matrizes polinomiais e matrizes estáveis. Takaba e Katayama [124] apresentaram uma solução por descritores na qual o problema \mathcal{H}_2 é solucionado para sistemas descritores com métodos polinomiais. A solução de Takaba e Katayama foi estendida por Kwakernaak em [80] e mistura os métodos de descritores e os métodos polinomiais. Na formulação LMI, o problema possui solução geral apenas para o caso por realimentação de estado [43]. Scherer [115] apresentou uma solução na forma de LMIs

para o problema com realimentação de saída. Sua solução, porém, não é geral, uma vez que considera apenas controladores de ordem igual ou superior à ordem do processo. Além disso, todas as soluções citadas são válidas somente para o caso onde o processo considerado é precisamente conhecido. No caso específico da solução proposta por Scherer, apesar de ser descrita na forma LMI, esta não permite uma fácil adaptação para o problema com incertezas, uma vez que as matrizes dinâmicas do sistema A e B_2 estão envolvidas na transformação utilizada.

Para processos sujeitos a incertezas, porém, o problema \mathcal{H}_2 possui resposta apenas para casos específicos apresentados na literatura, geralmente calcados em hipóteses restritivas do ponto de vista matemático. A solução proposta por realimentação de estado apresentada em [43], por exemplo, é facilmente estendida para o caso incerto utilizando-se o conceito de estabilidade quadrática. Entretanto, uma solução geral para o problema de controle robusto \mathcal{H}_2 por realimentação de saída ainda não foi desenvolvida. Assim, justifica-se o uso de métodos alternativos na solução deste problema. A utilização de métodos menos ortodoxos, como os algoritmos evolutivos, permite que algumas hipóteses sejam desconsideradas, possibilitando-se encontrar melhores soluções. Esta é uma das mais importantes motivações para o desenvolvimento deste trabalho.

O problema de controle \mathcal{H}_2 por realimentação de estado, para sistemas precisamente conhecidos, e sua formulação LMI é apresentado na seqüência. No capítulo 6, esta teoria é utilizada para a geração da população inicial do algoritmo evolutivo desenvolvido.

Considera-se o processo genérico G , demonstrado na figura 3 e descrito pelas seguintes equações de estado:

$$G : \begin{cases} \dot{x}(t) = Ax(t) + B_1w_2(t) + B_2u(t) \\ z_2(t) = C_2x(t) + D_{22}u(t) \\ y(t) = C_yx(t), \end{cases} \quad (3.22)$$

onde $x(t) \in \mathbb{R}^n$ é o vetor de estados, $w_2(t) \in \mathbb{R}^l$ é o vetor de distúrbios externos, $y(t) \in \mathbb{R}^r$ é o vetor de saída do sensor, $z_2(t) \in \mathbb{R}^{p_2}$ é a saída controlada e $u(t) \in \mathbb{R}^m$ é o vetor de controle. Todas as matrizes são reais com dimensões apropriadas.

Para uma lei de controle por realimentação de estado linear $u(t) = Kx(t)$, $K \in \mathbb{R}^{m \times n}$, a matriz de transferência entre z_2 e w_2 é dada por:

$$T_{z_2w_2}(s) = C_{2f}[s\mathbf{I} - A_f]^{-1}B_1. \quad (3.23)$$

Assim, o sistema em malha fechada resultante é descrito pelas equações de estado

$$\begin{cases} \dot{x}(t) = A_f x(t) + B_1 w_2(t) \\ z_2(t) = C_{2f} x(t) \\ y(t) = C_y x(t), \end{cases} \quad (3.24)$$

onde as A_f e C_{2f} são dadas por:

$$\begin{cases} A_f = A + B_2 K \\ C_{2f} = C_2 + D_{22} K. \end{cases} \quad (3.25)$$

Assumindo-se que A_f seja assintoticamente estável, os Gramianos de controlabilidade L_C e observabilidade L_O são definidos, respectivamente, por [28]:

$$L_C \triangleq \int_0^\infty \mathbf{e}^{A_f t} B_1 B_1^T \mathbf{e}^{A_f^T t} dt, \quad (3.26)$$

$$L_O \triangleq \int_0^\infty \mathbf{e}^{A_f^T t} C_{2f}^T C_{2f} \mathbf{e}^{A_f t} dt. \quad (3.27)$$

Os Gramianos L_C e L_O satisfazem as equações de Lyapunov

$$A_f L_C + L_C A_f^T + B_1 B_1^T = 0, \quad (3.28)$$

$$A_f^T L_O + L_O A_f + C_{2f}^T C_{2f} = 0 \quad (3.29)$$

e possuem as seguintes propriedades:

1. $L_C > 0$ se e somente se o par (A_f, B_1) é controlável;
2. $L_O > 0$ se e somente se o par (A_f, C_{2f}) é observável.

Assumindo-se, por hipótese, que $T_{z_2 w_2}(s)$ seja assintoticamente estável, tem-se que

$$h(t) = \mathcal{L}^{-1}\{T_{z_2 w_2}(s)\} = \begin{cases} C_{2f} e^{A_f t} B_1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (3.30)$$

e

$$\begin{aligned} \|T_{z_2 w_2}(s)\|_2^2 &= \int_0^\infty \mathbf{tr}(t_{z_2 w_2}^T(t) t_{z_2 w_2}(t)) dt \\ &= \int_0^\infty \mathbf{tr}(t_{z_2 w_2}(t) t_{z_2 w_2}(t)^T) dt \end{aligned} \quad (3.31)$$

$$\|T_{z_2 w_2}(s)\|_2^2 = \int_0^\infty \mathbf{tr} \left(B_1^T e^{A_f^T t} C_{2f}^T C_{2f} e^{A_f t} B_1 \right) dt \quad (3.32)$$

$$= \int_0^\infty \mathbf{tr} \left(C_{2f} e^{A_f t} B_1 B_1^T e^{A_f^T t} C_{2f}^T \right) dt. \quad (3.33)$$

Juntando-se as equações (3.32) e (3.33) com as definições (3.26) e (3.27) dos Gramianos de observabilidade e controlabilidade tem-se que o cálculo da norma \mathcal{H}_2 é dado por

$$\|T_{z_2 w_2}(s)\|_2^2 = \mathbf{tr} (B_1^T L_O B_1) \quad (3.34)$$

$$= \mathbf{tr} (C_{2f} L_C C_{2f}^T), \quad (3.35)$$

onde L_C e L_O são extraídos das equações de Lyapunov (3.28) e (3.29).

O problema de controle ótimo \mathcal{H}_2 pode ser reformulado como segue:

Encontrar um controlador K , admissível e próprio, tal que a matriz de transferência $T_{z_2 w_2}(s)$ do sistema em malha fechada seja estável e possua norma \mathcal{H}_2 , definida por (3.34) ou (3.35), mínima.

Seja $X_2 = X_2^T$ uma matriz tal que $X_2 \geq L_C$. Com isso, equação (3.28) torna-se uma inequação dada por

$$A_f X_2 + X_2 A_f^T + B_1 B_1^T \leq 0 \quad (3.36)$$

e a equação (3.34) torna-se

$$\|T_{z_2 w_2}(s)\|_2^2 \leq \mathbf{tr} (B_1^T X_2 B_1). \quad (3.37)$$

Da mesma forma, para a equação (3.29) e para uma matriz $X_2 = X_2^T$ tal que $X_2 \geq L_O$, a equação (3.29) torna-se

$$A_f^T X_2 + X_2 A_f + C_{2f}^T C_{2f} \leq 0 \quad (3.38)$$

e a equação (3.35) resulta em

$$\|T_{z_2 w_2}(s)\|_2^2 \leq \mathbf{tr} (C_{2f} X_2 C_{2f}^T). \quad (3.39)$$

Introduzindo-se a mudança de variáveis $KW_1 = W_2$, onde K é o controlador por realimentação de estado, tal que $u(t) = Kx(t)$ e $X_2 = W_1$, a inequação (3.36) torna-se uma desigualdade matricial linear (LMI) em W_1 e W_2 dada por

$$AW_1 + W_1 A^T + B_2 W_2 + W_2^T B_2^T + B_1 B_1^T \leq 0. \quad (3.40)$$

Da mesma forma, a inequação (3.37) resulta na LMI

$$\|T_{z_2 w_2}(s)\|_2^2 \leq \mathbf{tr} [(C_2 + D_{22}K)W_1(C_2 + D_{22}K)^T], \quad (3.41)$$

que equivale a

$$\|T_{z_2 w_2}(s)\|_2^2 \leq \mathbf{tr} [(C_2W_1 + D_{22}W_2)W_1^{-1}(C_2W_1 + D_{22}W_2)^T]. \quad (3.42)$$

Seja uma matriz W_3 tal que

$$W_3 \geq (C_2W_1 + D_{22}W_2)W_1^{-1}(C_2W_1 + D_{22}W_2)^T, \quad (3.43)$$

então

$$\|T_{z_2 w_2}(s)\|_2^2 \leq \mathbf{tr}(W_3). \quad (3.44)$$

Aplicando-se o complemento de Schur à inequação (3.43), obtém-se a seguinte desigualdade matricial linear (LMI):

$$\begin{bmatrix} W_1 & W_1C_2^T + W_2^TD_{22}^T \\ C_2W_1 + D_{22}W_2 & W_3 \end{bmatrix} \geq 0. \quad (3.45)$$

Da mesma maneira, a dedução das LMIs pode ser feita para o Gramiano de observabilidade (equações (3.38) e (3.39)) através da troca de variáveis $KW_1 = W_2$, $X_2 = W_1^{-1}$ e $X_2 \geq L_O$. As LMIs resultantes, para este caso, são dadas por:

$$\begin{bmatrix} AW_1 + W_1A^T + B_2W_2 + W_2^TB_2^T & (C_2W_1 + D_{22}W_2)^T \\ C_2W_1 + D_{22}W_2 & -\mathbf{I} \end{bmatrix} \leq 0 \quad (3.46)$$

e

$$\begin{bmatrix} W_3 & B_1^T \\ B_1 & W_1 \end{bmatrix} \geq 0. \quad (3.47)$$

O problema \mathcal{H}_2 pode ser interpretado como um problema de otimização em W_1 , W_2 e W_3 , onde a solução ótima corresponde à igualdade na equação (3.44). Na forma de LMIs, pode-se definir o problema como:

Encontrar as matrizes W_1 , W_2 e W_3 tal que $\mathbf{tr}(W_3)$ seja mínimo e as inequações (3.40) e (3.45) ou (3.46) e (3.47) sejam satisfeitas.

Na forma de um problema de otimização LMI, o problema de controle ótimo \mathcal{H}_2

corresponde a

$$\min_{W_1 > 0, W_2, W_3} \mathbf{tr}(W_3) \quad (3.48)$$

sujeito a (3.40) e (3.45) ou (3.46) e (3.47) onde $W_1 = W_1^T \in \mathbb{R}^{n \times n}$, $W_1 > 0$, $W_2 \in \mathbb{R}^{m \times n}$, $W_3 = W_3^T \in \mathbb{R}^{q \times q}$ e $W_3 > 0$. Se a solução ótima $[W_1, W_2, W_3]$ para o problema é encontrada, então o controlador que minimiza a norma \mathcal{H}_2 para o sistema na figura 3 é dado por $K = W_2 W_1^{-1}$ e $\|H(s)\|_2^2 = \mathbf{tr}(W_3)$.

Nesta forma, o problema é convexo, porém não estritamente, e é facilmente solucionado com a utilização de algoritmos de otimização específicos, como o método de pontos interiores [111, 96, 131]. As vantagens de se trabalhar com a formulação LMI são agregadas ao problema que se torna de fácil solução, flexível e convexo.

Utilizando-se o conceito de estabilidade quadrática, pode-se estender a solução do problema de controle \mathcal{H}_2 , descrito por (3.48), para a síntese de controladores para processos incertos. Com isso, garante-se um limitante superior para a norma \mathcal{H}_2 do sistema incerto em malha fechada. A solução para este caso é descrita a seguir.

Seja o o processo G , linear, invariante no tempo e sujeito a incertezas paramétricas do tipo politopo, descrito pelas equações de estado:

$$G : \begin{cases} \dot{x}(t) = Ax(t) + B_1 w_2(t) + B_2 u(t) \\ z_2(t) = C_2 x(t) + D_{22} u(t) \\ y(t) = C_y x(t), \end{cases} \quad (3.49)$$

onde $x(t) \in \mathbb{R}^n$ é o vetor de estados, $w_2(t) \in \mathbb{R}^l$ é o vetor de distúrbios externos, $y(t) \in \mathbb{R}^r$ é o vetor de saída do sensor, $z_2(t) \in \mathbb{R}^{p_2}$ é a saída controlada e $u(t) \in \mathbb{R}^m$ é o vetor de controle. Todas as matrizes são reais com dimensões apropriadas. Assume-se que A e B_2 pertencem aos domínios convexos definidos por:

$$\mathcal{D}_A = \left\{ A \in \mathbb{R}^{n \times n} : A = \sum_{i=1}^N \alpha_i A_i, \sum_{i=1}^N \alpha_i = 1, \alpha_i \geq 0 \right\}, \quad (3.50)$$

$$\mathcal{D}_B = \left\{ B_2 \in \mathbb{R}^{n \times m} : B_2 = \sum_{j=1}^N \beta_j B_{2j}, \sum_{j=1}^N \beta_j = 1, \beta_j \geq 0 \right\}. \quad (3.51)$$

Para uma lei de controle por realimentação de estado linear $u(t) = Kx(t)$, $K \in \mathbb{R}^{m \times n}$,

o sistema resultante em malha fechada é descrito pelas equações de estado:

$$\begin{cases} \dot{x}(t) = A_f x(t) + B_1 w_2(t) \\ z_2(t) = C_{2f} x(t) \\ y(t) = C_y x(t), \end{cases} \quad (3.52)$$

onde as A_f e C_{2f} são dadas por:

$$\begin{cases} A_f = A + B_2 K \\ C_{2f} = C_2 + D_{22} K. \end{cases} \quad (3.53)$$

com $A \in \mathcal{D}_A$ e $B_2 \in \mathcal{D}_B$ definidos em (3.88) e (3.89), respectivamente. A matriz de transferência entre z_2 e w_2 é dada por:

$$T_{z_2 w_2}(s) = C_{2f} [s\mathbf{I} - A_f]^{-1} B_1. \quad (3.54)$$

O problema de controle robusto \mathcal{H}_2 pode então ser formulado como:

Encontrar um controlador robusto K , admissível e próprio, que minimize a norma \mathcal{H}_2 de $\|T_{z_2 w_2}\|_2$, $\forall A \in \mathcal{D}_A$ e $\forall B_2 \in \mathcal{D}_B$.

Considerando-se a convexidade da estrutura LMI, o conceito de estabilidade quadrática e a notação da teoria de otimização, o problema de controle robusto \mathcal{H}_2 por realimentação de estado pode ser escrito como:

$$\begin{aligned} \min \quad & \text{tr}(W_3) \\ & W_1 = W_1^T > 0, W_2, W_3 \end{aligned} \quad (3.55)$$

sujeito a

$$A_i W_1 + W_1 A_i^T + B_{2j} W_2 + W_2^T B_{2j}^T + B_1 B_1^T \leq 0, \quad (3.56)$$

$$\begin{bmatrix} W_1 & W_1 C_2^T + W_2^T D_{22}^T \\ C_2 W_1 + D_{22} W_2 & W_3 \end{bmatrix} \geq 0, \quad (3.57)$$

ou

$$\begin{bmatrix} A_i W_1 + W_1 A_i^T + B_{2j} W_2 + W_2^T B_{2j}^T & (C_2 W_1 + D_{22} W_2)^T \\ C_2 W_1 + D_{22} W_2 & -\mathbf{I} \end{bmatrix} \leq 0 \quad (3.58)$$

$$\begin{bmatrix} W_3 & B_1^T \\ B_1 & W_1 \end{bmatrix} \geq 0, \quad (3.59)$$

com $i = 1, 2, \dots, N$, $j = 1, 2, \dots, M$. A solução ótima do problema é dada por $K = W_2 W_1^{-1}$.

Nesta forma, encontra-se a solução ótima para o problema de controle robusto \mathcal{H}_2 e a norma resultante representa um custo garantido para o sistema incerto. A formulação mais geral deste problema, por realimentação dinâmica de saída, é tratada em conjunto o problema \mathcal{H}_∞ na seção 3.6. A solução proposta é formulada com base nos algoritmos evolutivos e é descrita no capítulo 6.

3.5 Problema \mathcal{H}_∞

Um grande esforço da comunidade científica tem sido aplicado no desenvolvimento da teoria de controle \mathcal{H}_∞ , que trata da síntese de controladores que minimizam a norma \mathcal{H}_∞ de uma determinada matriz de transferência para sistemas lineares sujeitos a distúrbios desconhecidos e incertezas no modelo. Para um sistema contínuo, linear e invariante no tempo, a norma \mathcal{H}_∞ da matriz de transferência corresponde ao máximo valor singular sobre todas as frequências.

O problema de controle \mathcal{H}_∞ foi inicialmente formulado por George Zames [135], no início da década de 80, no contexto da redução da sensibilidade em processos lineares. Zames o formulou como um problema de otimização matemática utilizando o operador norma \mathcal{H}_∞ . Inicialmente escrito no domínio da frequência, as principais ferramentas utilizadas no desenvolvimento do problema \mathcal{H}_∞ foram a fatoração espectral, a parametrização de Youla e a teoria de aproximação. O uso destas ferramentas resultava em um problema de síntese de controladores \mathcal{H}_∞ ótimos ou sub-ótimos complicado e de dimensão elevada.

Os trabalhos que se seguiram, em meados da década de 80, demonstraram, porém, que a caracterização do problema no domínio do tempo resultava na solução através de equações algébricas de Riccati. Esta descoberta proporcionou um grande desenvolvimento na teoria de controle \mathcal{H}_∞ , culminando com as soluções no espaço de estados que permitiram a formulação de problemas gerais, incluindo processos variantes no tempo [10].

Nas décadas de 70 e 80, o problema foi relacionado com a teoria de jogos dinâmicos e foi abordado na forma de um problema *minimax*, onde o objetivo reside na minimização dos distúrbios e maximização do índice de desempenho adotado [10]. Uma descrição aprofundada da teoria \mathcal{H}_∞ pode ser encontrada em [89].

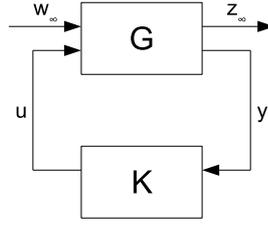


Figura 4: Sistema genérico realimentado - caso \mathcal{H}_∞ .

Seja o sistema linear S representado pelo diagrama na figura 4, descrito pelas equações de estado

$$S : \begin{cases} \dot{x}(t) = A_f x(t) + B_{1_f} w_\infty(t) \\ z_\infty(t) = C_{1_f} x(t) + D_{11_f} w_\infty(t) \\ y(t) = C_{y_f} x(t) + D_{y1_f} w_\infty(t), \end{cases} \quad (3.60)$$

onde $x(t) \in \mathbb{R}^n$ é o vetor de estados, $u(t) \in \mathbb{R}^m$ é o vetor de controle, $w_\infty(t) \in \mathbb{R}^l$ é o vetor de distúrbios externos, $y(t) \in \mathbb{R}^p$ é o vetor de saída do sensor, $z_\infty(t) \in \mathbb{R}^{p_1}$ é a saída controlada e o sub-índice f indica que as matrizes consideradas estão associadas ao sistema em malha fechada para uma lei de controle $u(t) = Ky(t)$. Por hipótese, assume-se que S seja assintoticamente estável.

Seja $T_{z_\infty w_\infty}(s)$ a função transferência entre a saída z_∞ e a entrada w_∞ . Como, por hipótese, S é estável, $T_{z_\infty w_\infty}(s)$ é analítica e limitada para todo $s \in \mathbb{C}$ com parte real positiva. Ou seja, o valor singular máximo σ_{\max} de $T_{z_\infty w_\infty}(s)$ é finito para todo $s \in \mathbb{C}$ com $\text{Re}(s) > 0$ e $T_{z_\infty w_\infty}(s)$ pertencente ao espaço \mathcal{H}_∞ . Assim, a norma \mathcal{H}_∞ de $T_{z_\infty w_\infty}(s)$ é dada pela equação (2.13) e pode-se definir o problema de controle ótimo \mathcal{H}_∞ como:

Encontrar um controlador K , racional, próprio e admissível, tal que a matriz de transferência $T_{z_\infty w_\infty}(s)$ seja estável e possua norma $\|T_{z_\infty w_\infty}(s)\|_\infty$ mínima, definida por (2.13).

A relação entre a norma \mathcal{H}_∞ no domínio do tempo e no domínio da frequência se dá pelo resultado seguinte, extraído de [117].

Seja $T_{z_\infty w_\infty}(s) \in \mathcal{H}_\infty$ a função de transferência do sistema assintoticamente estável S . Para todas as condições iniciais $x(0) = x_0$, $w_\infty \in \mathcal{L}_2$ implica que $z_\infty \in \mathcal{L}_2$ e para a condição $x(0) = 0$ é verdade que

$$\|T_{z_\infty w_\infty}(s)\|_\infty = \sup_{w_\infty \in \mathcal{L}_2} \frac{\|z_\infty\|_2}{\|w_\infty\|_2} \leq \gamma, \quad (3.61)$$

onde o lado esquerdo desta equação é a chamada norma induzida \mathcal{L}_2 do sistema S . Assim,

pode-se interpretar que, para um sistema estável S dado por (3.60), a norma \mathcal{H}_∞ de sua função transferência é o ganho \mathcal{L}_2 do operador entrada-saída associado ao sistema [117].

Este fato resulta em uma relação simples e estreita entre a norma \mathcal{H}_∞ do sistema e o *Bounded Real Lemma* (seção 2.6.4). Desta relação extrai-se que, se o sistema S é assintoticamente estável, então $\|T_{z_\infty w_\infty}(s)\|_\infty \leq \gamma$ se e somente se existir uma matriz solução $P = P^T > 0$ que satisfaz a desigualdade matricial

$$\begin{bmatrix} A_f^T P + P A_f + C_{1f}^T C_{1f} & P B_{1f} + C_{1f}^T D_{11f} \\ B_{1f}^T P + D_{11f}^T C_{1f} & D_{11f}^T D_{11f} - \gamma^2 \mathbf{I} \end{bmatrix} \leq 0. \quad (3.62)$$

Com isso, pode-se redefinir o problema de controle ótimo \mathcal{H}_∞ como:

Encontrar um controlador K , racional, próprio e admissível, e uma matriz P , que satisfaçam a desigualdade matricial (3.62), tais que a norma $\|T_{z_\infty w_\infty}(s)\|_\infty$, dada por γ , seja mínima.

Para um sistema com uma entrada e uma saída, no qual somente um valor singular existe, tem-se que $\sigma(T_{z_\infty w_\infty}(j\omega)) = |T_{z_\infty w_\infty}(j\omega)|$. Assim, o comportamento do valor singular na frequência pode ser obtido com o módulo da função transferência senoidal, ou então, a partir do gráfico do ganho de $T_{z_\infty w_\infty}(s)$ no diagrama de Bode. Neste diagrama, a norma \mathcal{H}_∞ de $T_{z_\infty w_\infty}(s)$ corresponde ao valor mais elevado do módulo, identificando portanto uma medida de pior caso.

Apesar de eficaz na análise da norma \mathcal{H}_∞ do sistema, a inequação (3.62) não é, porém, adequada para a síntese. Para isto, são necessárias algumas modificações algébricas. Ao se aplicar o complemento de Schur na desigualdade (3.62) obtém-se

$$\begin{aligned} & A_f^T P + P A_f + C_{1f}^T C_{1f} \\ & + \left(P B_{1f} + C_{1f}^T D_{11f} \right) \left(\gamma^2 \mathbf{I} - D_{11f}^T D_{11f} \right)^{-1} \left(B_{1f}^T P + D_{11f}^T C_{1f} \right) \leq 0. \end{aligned} \quad (3.63)$$

Multiplicando-se ambos os lados da equação (3.63) por $X_\infty = P^{-1}$ tem-se

$$\begin{aligned} & A_f X_\infty + X_\infty A_f^T + X_\infty C_{1f}^T C_{1f} X_\infty \\ & + \left(B_{1f} + X_\infty C_{1f}^T D_{11f} \right) \left(\gamma^2 \mathbf{I} - D_{11f}^T D_{11f} \right)^{-1} \left(B_{1f}^T + D_{11f}^T C_{1f} X_\infty \right) \leq 0. \end{aligned} \quad (3.64)$$

Aplicando-se o complemento de Schur em (3.64) tem-se

$$\begin{bmatrix} A_f X_\infty + X_\infty A_f^T + X_\infty C_{1f}^T C_{1f} X_\infty & B_{1f} + X_\infty C_{1f}^T D_{11f} \\ B_{1f}^T + D_{11f}^T C_{1f} X_\infty & D_{11f}^T D_{11f} - \gamma^2 \mathbf{I} \end{bmatrix} \leq 0. \quad (3.65)$$

O lado esquerdo da desigualdade (3.65) equivale à soma

$$\begin{bmatrix} A_f X_\infty + X_\infty A_f^T & B_{1_f} \\ B_{1_f}^T & -\gamma^2 \mathbf{I} \end{bmatrix} + \begin{bmatrix} X_\infty C_{1_f}^T C_{1_f} X_\infty & X_\infty C_{1_f}^T D_{11_f} \\ D_{11_f}^T C_{1_f} X_\infty & D_{11_f}^T D_{11_f} \end{bmatrix} \leq 0 \quad (3.66)$$

e (3.66), por sua vez, é idêntico a

$$\begin{bmatrix} A_f X_\infty + X_\infty A_f^T & B_{1_f} \\ B_{1_f}^T & -\gamma^2 \mathbf{I} \end{bmatrix} + \begin{bmatrix} X_\infty C_{1_f}^T \\ D_{11_f}^T \end{bmatrix} \begin{bmatrix} C_{1_f} X_\infty & D_{11_f} \end{bmatrix} \leq 0. \quad (3.67)$$

Finalmente, aplicando-se o complemento de Schur em (3.67) chega-se a

$$\begin{bmatrix} A_f X_\infty + X_\infty A_f^T & B_{1_f} & X_\infty C_{1_f}^T \\ B_{1_f}^T & -\gamma^2 \mathbf{I} & D_{11_f}^T \\ C_{1_f} X_\infty & D_{11_f} & -\mathbf{I} \end{bmatrix} \leq 0, \quad (3.68)$$

que é uma desigualdade matricial mais apropriada para a síntese de controladores sujeita à restrição de norma \mathcal{H}_∞ . Pode-se, assim, redefinir o problema de controle ótimo \mathcal{H}_∞ como:

Encontrar um controlador K , racional, próprio e admissível, tal que exista uma matriz simétrica e definida positiva, $X_\infty = X_\infty^T$ e $X_\infty > 0$, que satisfaz a desigualdade matricial (3.68) e a norma $\|T_{z_\infty w_\infty}(s)\|_\infty$, dada por γ , seja mínima.

Este problema pode ser escrito, na forma de um problema de otimização, como:

$$\min_{K \in \mathcal{K}, X_\infty = X_\infty^T > 0} \gamma \quad (3.69)$$

sujeito a

$$\begin{bmatrix} A_f X_\infty + X_\infty A_f^T & B_{1_f} & X_\infty C_{1_f}^T \\ B_{1_f}^T & -\gamma^2 \mathbf{I} & D_{11_f}^T \\ C_{1_f} X_\infty & D_{11_f} & -\mathbf{I} \end{bmatrix} \leq 0, \quad (3.70)$$

onde \mathcal{K} representa o conjunto de todos os controladores K que estabilizam o sistema S (3.60). Diversas soluções foram apresentadas para o problema (3.69). De forma análoga ao problema \mathcal{H}_2 , estas soluções são aplicáveis somente no caso precisamente conhecido. Para os sistemas de controle onde o processo a ser controlado não é precisamente conhecido, não existe solução geral LMI para a lei de controle de realimentação de saída. Para o problema de realimentação de estado, onde a lei de controle é dada por $u(t) = Kx(t)$, a solução LMI é direta e é dada por uma simples transformação de variáveis, como no caso \mathcal{H}_2 , apresentado na seção 3.4. O problema LMI de otimização \mathcal{H}_∞ por realimentação de

estado, para sistemas precisamente conhecidos, é apresentado na seqüência.

Seja o sistema linear genérico apresentado na figura 4 e descrito pelas equações de estado

$$S : \begin{cases} \dot{x}(t) = Ax(t) + B_1w_\infty(t) + B_2u(t) \\ z_\infty(t) = C_1x(t) + D_{11}w_\infty + D_{12}u(t) \\ y(t) = C_yx(t) + D_{y1}w_\infty(t), \end{cases} \quad (3.71)$$

onde $x(t) \in \mathbb{R}^n$ é o vetor de estados, $u(t) \in \mathbb{R}^m$ é a entrada de controle, $y(t) \in \mathbb{R}^p$ é a saída do sensor, $z_\infty(t) \in \mathbb{R}^{p_1}$ é a saída controlada e $w_\infty(t) \in \mathbb{R}^{l_1}$ é a entrada exógena. Todas as matrizes são reais com dimensões apropriadas.

O sistema em malha fechada resultante, para uma lei de controle por realimentação de estado $u(t) = Kx(t)$, $K \in \mathbb{R}^{m \times n}$, é descrito pelas equações de estado

$$S_f : \begin{cases} \dot{x}(t) = A_fx(t) + B_1w_\infty(t) \\ z_\infty(t) = C_{1_f}x(t) + D_{11}w_\infty(t) \\ y(t) = C_yx(t) + D_{y1}w_\infty(t), \end{cases} \quad (3.72)$$

onde as matrizes em malha fechada A_f e C_{1_f} são dadas por

$$\begin{cases} A_f = A + B_2K \\ C_{1_f} = C_1 + D_{12}K. \end{cases} \quad (3.73)$$

A equação (3.68) para o sistema em malha fechada S_f , dado por (3.80), torna-se:

$$\begin{bmatrix} A_fX_\infty + X_\infty A_f^T & B_1 & X_\infty C_{1_f}^T \\ B_1^T & -\gamma^2 \mathbf{I} & D_{11}^T \\ C_{1_f}X_\infty & D_{11} & -\mathbf{I} \end{bmatrix} \leq 0. \quad (3.74)$$

Substituindo-se (3.73) em (3.74) e aplicando-se as transformações $KW_1 = W_2$, $X_\infty = W_1$ e $r = \gamma^2$, chega-se à LMI resultante:

$$\begin{bmatrix} AW_1 + W_1A^T + B_2W_2 + W_2^T B_2^T & B_1 & W_1C_1^T + W_2^T D_{12}^T \\ B_1^T & -r\mathbf{I} & D_{11}^T \\ C_1W_1 + D_{12}W_2 & D_{11} & -\mathbf{I} \end{bmatrix} \leq 0. \quad (3.75)$$

A LMI (3.86) é utilizada na a síntese de controladores ótimos \mathcal{H}_∞ por realimentação de estado. O problema de controle ótimo \mathcal{H}_∞ torna-se:

Encontrar as matrizes W_1 e W_2 tais que r seja mínimo e a LMI (3.86) seja satisfeita.

Na forma de um problema de otimização LMI, o problema de controle ótimo \mathcal{H}_∞ por realimentação de estado corresponde a

$$\min_{W_1 > 0, W_2} r \quad (3.76)$$

sujeito a (3.86) onde $W_1 = W_1^T \in \mathbb{R}^{n \times n}$, $W_1 > 0$, $W_2 \in \mathbb{R}^{m \times n}$. Se a solução ótima $[W_1, W_2]$ para o problema é encontrada, então o controlador por realimentação de estado que minimiza a norma \mathcal{H}_∞ para o sistema na figura 4 é dado por $K = W_2 W_1^{-1}$ e a norma \mathcal{H}_∞ da função $T_{z_\infty w_\infty}(s)$ é dada por $\|T_{z_\infty w_\infty}(s)\|_\infty = \gamma = \sqrt{r}$. Este problema é convexo e facilmente solucionado com os métodos de otimização LMI.

De forma análoga à apresentada para o problema de controle \mathcal{H}_2 , a solução LMI para o problema \mathcal{H}_∞ pode ser estendida para o caso robusto utilizando-se o conceito de estabilidade quadrática. Com isso, garante-se um limitante superior para a norma \mathcal{H}_∞ , γ , do sistema incerto em malha fechada. A solução para o problema de controle robusto \mathcal{H}_∞ por realimentação de estado, utilizando-se o conceito de estabilidade quadrática, é descrita a seguir.

Seja o sistema linear, invariante no tempo e sujeito a incertezas do tipo politopo, descrito pelas equações de estado:

$$S : \begin{cases} \dot{x}(t) = Ax(t) + B_1 w_\infty(t) + B_2 u(t) \\ z_\infty(t) = C_1 x(t) + D_{11} w_\infty + D_{12} u(t) \\ y(t) = C_y x(t) + D_{y1} w_\infty(t), \end{cases} \quad (3.77)$$

onde $x(t) \in \mathbb{R}^n$ é o vetor de estados, $u(t) \in \mathbb{R}^m$ é a entrada de controle, $y(t) \in \mathbb{R}^p$ é a saída do sensor, $z_\infty(t) \in \mathbb{R}^{p_1}$ é a saída controlada e $w_\infty(t) \in \mathbb{R}^{l_1}$ é a entrada exógena. Todas as matrizes são reais com dimensões apropriadas. Assume-se que as matrizes A e B_2 pertencem aos domínios convexos definidos por:

$$\mathcal{D}_A = \left\{ A \in \mathbb{R}^{n \times n} : A = \sum_{i=1}^N \alpha_i A_i, \sum_{i=1}^N \alpha_i = 1, \alpha_i \geq 0 \right\}, \quad (3.78)$$

$$\mathcal{D}_B = \left\{ B_2 \in \mathbb{R}^{n \times m} : B_2 = \sum_{j=1}^N \beta_j B_{2j}, \sum_{j=1}^N \beta_j = 1, \beta_j \geq 0 \right\}. \quad (3.79)$$

Para uma lei de controle por realimentação de estado linear $u(t) = Kx(t)$, $K \in \mathbb{R}^{m \times n}$,

o sistema resultante em malha fechada é descrito pelas equações de estado:

$$S_f : \begin{cases} \dot{x}(t) = A_f x(t) + B_1 w_\infty(t) \\ z_\infty(t) = C_{1f} x(t) + D_{11} w_\infty(t) \\ y(t) = C_y x(t) + D_{y1} w_\infty(t), \end{cases} \quad (3.80)$$

onde as A_f e C_{1f} são dadas por:

$$\begin{cases} A_f = A + B_2 K \\ C_{1f} = C_1 + D_{11} K. \end{cases} \quad (3.81)$$

com $A \in \mathcal{D}_A$ e $B_2 \in \mathcal{D}_B$ definidos em (3.88) e (3.89), respectivamente. A matriz de transferência de z_∞ para w_∞ é dada por:

$$T_{z_\infty w_\infty}(s) = C_{1f} [s\mathbf{I} - A_f]^{-1} B_1 + D_{11}. \quad (3.82)$$

Define-se o problema de controle robusto ótimo \mathcal{H}_∞ por realimentação de estado como:

Encontrar um controlador robusto K , admissível e próprio, que minimize a norma \mathcal{H}_∞ , $\|T_{z_\infty w_\infty}\|_\infty = \gamma$, $\forall A \in \mathcal{D}_A$ e $\forall B_2 \in \mathcal{D}_B$.

Considerando-se a convexidade da estrutura LMI, o conceito de estabilidade quadrática e a notação da teoria de otimização, o problema de controle robusto \mathcal{H}_∞ por realimentação de estado pode ser formulado como:

$$\begin{aligned} \min \quad & r \\ & W_1 > 0, W_2 \end{aligned} \quad (3.83)$$

sujeito a

$$\begin{bmatrix} A_i W_1 + W_1 A_i^T + B_{2j} W_2 + W_2^T B_{2j}^T & B_1 & W_1 C_1^T + W_2^T D_{12}^T \\ & B_1^T & -r\mathbf{I} & D_{11}^T \\ & C_1 W_1 + D_{12} W_2 & D_{11} & -\mathbf{I} \end{bmatrix} \leq 0. \quad (3.84)$$

onde $W_1 = W_1^T \in \mathbb{R}^{n \times n}$, $W_1 > 0$, $W_2 \in \mathbb{R}^{m \times n}$, $\gamma = \sqrt{r}$ e $i = 1, 2, \dots, N$, $j = 1, 2, \dots, M$. A solução ótima deste problema é dada por $K = W_2 W_1^{-1}$.

Em diversos casos, porém, buscam-se otimizar outros critérios de desempenho do controlador, como por exemplo a norma \mathcal{H}_2 , considerando-se um limite superior admissível para a norma \mathcal{H}_∞ , γ . Para estes casos, define-se o problema de controle robusto sub-ótimo \mathcal{H}_∞ por realimentação de estado como:

Encontrar um controlador robusto K , admissível e próprio, que satisfaça a condição

de norma \mathcal{H}_∞ , $\|T_{z_\infty w_\infty}\|_\infty \leq \gamma$, para um dado limite superior γ e $\forall A \in \mathcal{D}_A$ e $\forall B_2 \in \mathcal{D}_B$.

Na forma LMI, este problema pode é descrito como:

$$\begin{aligned} &\text{encontrar } W_1, W_2 \\ &W_1 > 0, W_2 \end{aligned} \quad (3.85)$$

tal que

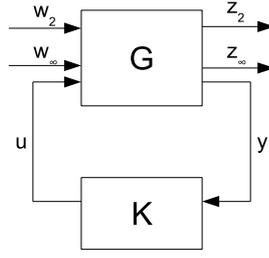
$$\begin{bmatrix} A_i W_1 + W_1 A_i^T + B_{2j} W_2 + W_2^T B_{2j}^T & B_1 & W_1 C_1^T + W_2^T D_{12}^T \\ & B_1^T & -\gamma^2 \mathbf{I} & D_{11}^T \\ & C_1 W_1 + D_{12} W_2 & D_{11} & -\mathbf{I} \end{bmatrix} \leq 0. \quad (3.86)$$

onde $W_1 = W_1^T \in \mathbb{R}^{n \times n}$, $W_1 > 0$, $W_2 \in \mathbb{R}^{m \times n}$ e $i = 1, 2, \dots, N$, $j = 1, 2, \dots, M$. Para este problema, γ é conhecido e representa um limite superior para a norma \mathcal{H}_∞ . O controlador, solução do problema, é dado por $K = W_2 W_1^{-1}$.

A exemplo do caso \mathcal{H}_2 , descrito na seção 3.4, apenas o problema de controle \mathcal{H}_∞ por realimentação de estado foi descrito, pois apresenta solução matemática LMI. A formulação geral do problema para realimentação dinâmica de saída é tratada na seção 3.6 em conjunto com o problema \mathcal{H}_2 . A solução proposta para o problema geral é formulada através dos algoritmos evolutivos e está descrita no capítulo 5.

3.6 Problema de Controle Robusto Misto $\mathcal{H}_2/\mathcal{H}_\infty$ Sujeito à \mathcal{D} -estabilidade

Esta seção apresenta o problema foco deste trabalho. Consideram-se processos lineares, invariantes no tempo e sujeitos a incertezas paramétricas do tipo politopo. Assume-se a lei de controle de realimentação dinâmica de saída e a ordem do controlador é definida *a priori*, podendo ser completa ou reduzida. A formulação deste problema considera a minimização da norma \mathcal{H}_2 com uma restrição de norma \mathcal{H}_∞ como critérios de desempenho. Além disso, admitem-se restrições de alocação de pólos, responsáveis por garantir um desempenho satisfatório em regime transitório para o sistema em malha fechada. Embora se considere uma matriz de Lyapunov comum para o domínio de incerteza do problema, a formulação apresentada nesta seção não pressupõe a mesma matriz de Lyapunov para todas as restrições do problema. Este fato libera parcialmente a restrição imposta pelo conceito de estabilidade quadrática e permite um espaço de busca mais amplo, podendo-se encontrar melhores resultados. O problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$ sujeito à \mathcal{D} -estabilidade é descrito a seguir.

Figura 5: Sistema genérico realimentado - caso $\mathcal{H}_2/\mathcal{H}_\infty$.

Seja o sistema S apresentado na figura 5 e descrito pelas equações de estado:

$$S : \begin{cases} \dot{x}(t) = Ax(t) + B_{1_2}w_2(t) + B_{1_\infty}w_\infty(t) + B_2u(t) \\ z_\infty(t) = C_1x(t) + D_{11_2}w_2(t) + D_{11_\infty}w_\infty(t) + D_{12}u(t) \\ z_2(t) = C_2x(t) + D_{21_\infty}w_\infty(t) + D_{22}u(t) \\ y(t) = C_yx(t) + D_{y1_\infty}w_\infty(t), \end{cases} \quad (3.87)$$

com $u(t) = Ky(t)$, onde $x(t) \in \mathbb{R}^n$ é o vetor de estados, $u(t) \in \mathbb{R}^m$ é a entrada de controle, $y(t) \in \mathbb{R}^p$ é a saída do sensor, $z_\infty(t) \in \mathbb{R}^{p_1}$ e $z_2(t) \in \mathbb{R}^{p_2}$ são as saídas controladas e $w_\infty(t) \in \mathbb{R}^{l_1}$ e $w_2(t) \in \mathbb{R}^{l_2}$ são entradas exógenas. Todas as matrizes são reais com dimensões apropriadas. Assume-se que A e B_2 pertencem aos domínios de incerteza convexos definidos por:

$$\mathcal{D}_A = \left\{ A \in \mathbb{R}^{n \times n} : A = \sum_{i=1}^N \alpha_i A_i, \sum_{i=1}^N \alpha_i = 1, \alpha_i \geq 0 \right\}, \quad (3.88)$$

$$\mathcal{D}_B = \left\{ B_2 \in \mathbb{R}^{n \times m} : B_2 = \sum_{j=1}^N \beta_j B_{2j}, \sum_{j=1}^N \beta_j = 1, \beta_j \geq 0 \right\}. \quad (3.89)$$

Considera-se que todos os pares (A, B_2) e (C_y, A) são estabilizáveis e detectáveis, respectivamente. Seja o controlador por realimentação de saída L , linear, dinâmico e invariante no tempo, descrito pelas equações:

$$L : \begin{cases} \dot{\eta}(t) = A_K\eta(t) + B_Ky(t) \\ u(t) = C_K\eta(t) + D_Ky(t), \end{cases} \quad (3.90)$$

onde $A_K \in \mathbb{R}^{n_c \times n_c}$. Nesta abordagem, o controlador pode ser de ordem reduzida ($n_c < n$) ou de ordem completa ($n_c \geq n$). A ordem n_c é fixa e dada *a priori*. Tanto S quanto L são racionais e próprios.

Com o controlador L definido por (3.90), o sistema resultante em malha fechada na

figura 5 é dado por:

$$\begin{cases} \dot{x}_f(t) = A_f x_f(t) + B_{1f_2} w_2(t) + B_{1f_\infty} w_\infty(t) \\ z_\infty(t) = C_{1f} x_f(t) + D_{11f_2} w_2(t) + D_{11f_\infty} w_\infty(t) \\ z_2(t) = C_{2f} x_f(t) + D_{21f_\infty} w_\infty(t) \\ y(t) = C_{yf} x_f(t) + D_{y1f_\infty} w_\infty(t), \end{cases} \quad (3.91)$$

onde

$$\begin{aligned} A_f &= \begin{bmatrix} A + B_2 D_K C_y & B_2 C_K \\ B_K C_y & A_K \end{bmatrix}, \quad B_{1f_2} = \begin{bmatrix} B_{1_2} \\ 0 \end{bmatrix}, \\ B_{1f_\infty} &= \begin{bmatrix} B_{1_\infty} + B_2 D_K D_{y1_\infty} \\ B_K D_{y1_\infty} \end{bmatrix}, \quad C_{yf} = \begin{bmatrix} C_y & 0 \end{bmatrix}, \\ C_{if} &= \begin{bmatrix} C_i + D_{i2} D_K C_y & D_{i2} C_K \end{bmatrix}, \quad i = 1, 2, \\ D_{i1f_\infty} &= \begin{bmatrix} D_{i1_\infty} + D_{i2} D_K & D_{y1_\infty} \end{bmatrix}, \quad i = 1, 2, \\ D_{11f_2} &= [D_{11_2}], \end{aligned} \quad (3.92)$$

com $A \in \mathcal{D}_A$ e $B_2 \in \mathcal{D}_B$ definidos em (3.88) e (3.89), respectivamente.

O sistema (3.87) pode ser reescrito como:

$$\begin{cases} \dot{x}_f(t) = \tilde{A} x_f(t) + \tilde{B}_{1_2} w_2(t) + \tilde{B}_{1_\infty} w_\infty(t) + \tilde{B}_2 u_s(t) \\ z_\infty(t) = \tilde{C}_1 x_f(t) + D_{11_2} w_2(t) + D_{11_\infty} w_\infty(t) + \tilde{D}_{12} u_s(t) \\ z_2(t) = \tilde{C}_2 x_f(t) + D_{21_\infty} w_\infty(t) + \tilde{D}_{22} u_s(t) \\ y_s(t) = \tilde{C}_y x_f(t) + \tilde{D}_{y1_\infty} w_\infty(t) \\ y(t) = C_{yf} x_f(t) + D_{y1f_\infty} w_\infty(t), \end{cases} \quad (3.93)$$

onde

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} A & 0 \\ 0 & 0_{n_c \times n_c} \end{bmatrix}, \quad \tilde{B}_{1_i} = \begin{bmatrix} B_{1_i} \\ 0 \end{bmatrix}, \quad i = 2, \infty, \\ \tilde{B}_2 &= \begin{bmatrix} B_2 & 0 \\ 0 & \mathbf{I}_{n_c \times n_c} \end{bmatrix}, \\ \tilde{C}_i &= \begin{bmatrix} C_i & 0 \end{bmatrix}, \quad \tilde{D}_{i2} = \begin{bmatrix} D_{i2} & 0 \end{bmatrix}, \quad i = 1, 2, \\ \tilde{C}_y &= \begin{bmatrix} C_y & 0 \\ 0 & \mathbf{I}_{n_c \times n_c} \end{bmatrix}, \quad \tilde{D}_{y1} = \begin{bmatrix} D_{y1} \\ 0 \end{bmatrix}, \end{aligned} \quad (3.94)$$

e a lei de controle investigada torna-se $u_s(t) = L_K y_s(t)$, onde

$$L_K = \begin{bmatrix} D_K & C_K \\ B_K & A_K \end{bmatrix} \in \mathbb{R}^{(m+n_c) \times (p+n)}. \quad (3.95)$$

Assim, o problema de realimentação dinâmica de saída pode ser tratado como um problema de realimentação estática de saída.

Definindo-se $T_{z_\infty w_\infty}(s)$ como a matriz de transferência em malha fechada entre w_∞ e z_∞ e $T_{z_2 w_2}(s)$ entre w_2 e z_2 , tem-se:

$$\begin{aligned} T_{z_\infty w_\infty}(s) &= (\tilde{C}_1 + \tilde{D}_{12} L_K \tilde{C}_y) [s\mathbf{I} - (\tilde{A} + \tilde{B}_2 L_K \tilde{C}_y)]^{-1} \\ &\quad (\tilde{B}_{1_\infty} + \tilde{B}_2 L_K \tilde{D}_{y1}) + (D_{11_\infty} + \tilde{D}_{12} L_K \tilde{D}_{y1_\infty}), \\ T_{z_2 w_2}(s) &= (\tilde{C}_2 + \tilde{D}_{22} L_K \tilde{C}_y) [s\mathbf{I} - (\tilde{A} + \tilde{B}_2 L_K \tilde{C}_y)]^{-1} \\ &\quad (\tilde{B}_{1_2} + \tilde{B}_2 L_K \tilde{D}_{y1}). \end{aligned} \quad (3.96)$$

Considera-se uma região genérica de \mathcal{D} -estabilidade $M_{\mathcal{D}}(A, X_D)$ dada pela equação (2.33). O problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$, sujeito à \mathcal{D} -estabilidade, pode ser formulado como:

Encontrar um controlador robusto L_K , admissível, próprio e \mathcal{D} -estável, que minimize a norma \mathcal{H}_2 de $\|T_{z_2 w_2}\|_2$ sujeito à restrição de norma \mathcal{H}_∞ de $\|T_{z_\infty w_\infty}\|_\infty < \gamma$, para um dado limite da norma \mathcal{H}_∞ γ , $\forall A \in \mathcal{D}_A$ e $\forall B_2 \in \mathcal{D}_B$.

Lembra-se que a matriz de transferência $T_{z_2 w_2}(s)$ é estritamente própria. É importante observar que a restrição de norma \mathcal{H}_∞ mantém o sistema internamente estável para todo o conjunto de incertezas ($\forall A \in \mathcal{D}_A$ e $\forall B_2 \in \mathcal{D}_B$).

Utilizando-se a notação da teoria de otimização e o conteúdo apresentado nas seções 3.4 e 3.5, o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$ restrito, ou seja, sujeito à \mathcal{D} -estabilidade, resultante, é dado por:

$$\begin{aligned} \min \quad & \mathbf{tr}(C_{2f} X_2 C_{2f}^T) \\ & X_2 > 0, X_\infty > 0, X_D > 0, L_K \end{aligned} \quad (3.97)$$

sujeito a:

$$\tilde{A}_f X_2 + X_2 \tilde{A}_f^T + B_{1_2} B_{1_2}^T < 0, \quad (3.98)$$

$$\begin{bmatrix} \tilde{A}_f X_\infty + X_\infty \tilde{A}_f^T & \tilde{B}_{1f_\infty} & X_\infty \tilde{C}_{1f}^T \\ \tilde{B}_{1f_\infty}^T & -\gamma^2 \mathbf{I} & \tilde{D}_{11f_\infty}^T \\ \tilde{C}_{1f} X_\infty & \tilde{D}_{11f_\infty} & -\mathbf{I} \end{bmatrix} < 0, \quad (3.99)$$

$$M_D(A, X) = \left[t_{rs}X_D + \tau_{rs}\tilde{A}_fX_D + \tau_{rs}X_D\tilde{A}_f^T \right] < 0, \quad 1 \leq r, s \leq q, \quad (3.100)$$

onde

$$\begin{aligned} \tilde{A}_f &= \tilde{A}_i + \tilde{B}_{2j}L_K\tilde{C}_y \\ \tilde{B}_{1f\infty} &= \tilde{B}_{1\infty} + \tilde{B}_{2j}L_K\tilde{D}_{y1} \\ \tilde{C}_{1f} &= \tilde{C}_1 + \tilde{D}_{12}L_K\tilde{C}_y \\ \tilde{D}_{11f\infty} &= \tilde{D}_{12}L_K\tilde{D}_{y1\infty}, \end{aligned} \quad (3.101)$$

com $i = 1, 2, \dots, N$, $j = 1, 2, \dots, M$ e t_{rs} e τ_{rs} definidos em (2.32).

O critério de desempenho apresentado considera um limite superior ótimo para o desempenho \mathcal{H}_2 , sujeito à restrição de norma \mathcal{H}_∞ e às restrições de alocação de pólos. Este problema misto é um problema de otimização matricial bi-linear (BMI). As soluções para problemas BMI não são triviais e apenas problemas bastante específicos possuem soluções diretas, como a solução para o caso de realimentação de estado apresentado nas seções 3.4 e 3.5. Observa-se, porém, que apesar deste problema não ser conjuntamente convexo nas variáveis $(X_2, X_\infty, X_D, L_K)$, é convexo para um controlador dado L_K ou para matrizes de Lyapunov (X_2, X_∞, X_D) conhecidas. Este fato será explorado na solução do problema proposta no capítulo 5.

Nesta forma, a hipótese de matrizes de Lyapunov comuns, $X_2 = X_\infty = X_D$, não é considerada. Relaxando-se esta restrição, aumenta-se o espaço de busca, permitindo-se melhores soluções. Na troca de variáveis introduzida em [115], utilizando-se as matrizes R e S , as matrizes A e B_2 do sistema são utilizadas, gerando limitações para a extensão de tal resultado para a síntese de controle para processos com incertezas politópicas. Relaxando estas hipóteses, a síntese de controladores, tanto de ordem reduzida quanto completa, pode ser considerada em uma estrutura única através desta abordagem.

3.7 Conclusão

Este capítulo apresentou a definição de sistemas incertos e o conceito de incertezas paramétricas do tipo politopo, aos quais os processos estudados neste trabalho estão sujeitos. O conceito de estabilidade quadrática é simples e permite o tratamento direto das incertezas politópicas, através da solução de um conjunto de LMIs. A principal limitação da utilização deste conceito é a imposição de uma matriz comum de Lyapunov tanto para o domínio de incerteza quanto para as diferentes restrições do problema, tornando a

abordagem restritiva.

As soluções LMI dos problemas \mathcal{H}_2 e \mathcal{H}_∞ apresentadas, para o caso de realimentação de estado, formam a base da solução proposta para o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$ restrito, descrita no capítulo 5. A motivação principal para este trabalho reside no parcial relaxamento da hipótese de estabilidade quadrática, admitindo-se matrizes de Lyapunov diferentes para cada restrição do problema.

4 Algoritmos Evolutivos

Neste capítulo são descritos os fundamentos dos Algoritmos Evolutivos (AEs) com ênfase nos Algoritmos Genéticos (AGs), na Evolução Diferencial (ED) e no algoritmo de Salomon. Na seção 4.1, uma breve evolução histórica dos AEs, como ferramentas de otimização, é apresentada. Na seção 4.2, os algoritmos genéticos são estudados. A abordagem de Michalewicz é apresentada na seção 4.3. Os algoritmos de Evolução Diferencial e de Salomon são descritos nas seções 4.4 e 4.5, respectivamente. A seção 4.6 ressalta os principais aspectos deste capítulo.

4.1 Introdução

A Computação Evolutiva (CE), ramo da computação no qual os algoritmos evolutivos estão inseridos, teve seu início na década de 60. Entretanto, a maioria dos trabalhos nesta área permaneceu desconhecida por mais de uma década, devido à limitada capacidade computacional então existente. Entre os autores pioneiros citam-se Bremermann [18], Friedberg [52, 53] e Box [16].

A evolução tecnológica da década de 70 permitiu que os trabalhos de Holland [69] e Fogel [49] iniciassem uma lenta revolução na computação evolutiva. Desde então, observou-se um crescimento exponencial no número de publicações na área, fato que demonstra sua relevância não só acadêmica, mas também prática [23].

Muito mais abrangente do que um conjunto de algoritmos estabelecidos e prontos para serem utilizados, a CE apresenta conceitos para a resolução de problemas. Sua principal aplicação são problemas complexos de otimização, pois os conceitos da CE são flexíveis e podem ser adaptados a problemas das mais diversas naturezas. Esta flexibilidade é demonstrada nos exemplos apresentados em [20, 11], que vão da arte à biologia.

O princípio dos algoritmos evolutivos é a evolução biológica, processo natural responsável pelo surgimento de espécies extremamente complexas e bem adaptadas. De forma

simples, a evolução é o resultado da interação entre espécies e seu meio, que resulta na troca e na evolução das suas informações genéticas. O conceito fundamental da evolução através da seleção natural foi inicialmente estudado por Charles Darwin em seu trabalho pioneiro “A origem das espécies” [34]. A interação genética e a transmissão hereditária das características das espécies tiveram sua origem nos estudos de Johann Gregor Mendel, que formulou as famosas leis da genética de Mendel. A mecânica básica da evolução é formada pelo conjunto seleção natural, procriação e adaptação (ou mutação). A seleção natural garante maior probabilidade de sobrevivência aos indivíduos mais aptos a superar as dificuldades ambientais. A troca de informações genéticas entre os diferentes indivíduos se dá através da procriação, que possibilita a perpetuação das características genéticas dos indivíduos mais aptos. As mutações são modificações aleatórias no código genético do indivíduo e acontecem pela influência de fatores ambientais. Estas mutações permitem a evolução do código genético, adaptando o indivíduo a adversidades do seu meio.

Estes conceitos biológicos foram, analogamente, aplicados na solução de problemas computacionais complexos. A idéia subjacente é gerar um conjunto de soluções iniciais para o problema (população inicial) e evoluí-lo através de mecanismos computacionais paralelos à evolução natural das espécies.

Na biologia, todas as características do indivíduo, seu fenótipo, estão contidas no seu código genético, ou genótipo, formado pelo ácido desoxirribonucléico (DNA). A mesma idéia pode ser aplicada na computação. Assim, para se evoluir computacionalmente as soluções de um problema, os fenótipos, estas são codificadas formando-se um “código genético” de cada solução, ou seu genótipo. A evolução acontece através do genótipo das soluções, na CE, também denominados cromossomos.

A aptidão em sobreviver é medida na biologia por uma série de fatores que tornam o indivíduo mais apto a superar as dificuldades ambientais. Na computação evolutiva, a aptidão de um cromossomo corresponde à medida de quão boa cada solução é para determinado problema. Desta forma, a aptidão de cada solução da população corresponde, em geral, à avaliação das funções as quais se deseja otimizar, denominadas funções objetivo.

A exemplo da biologia, a procriação dos indivíduos é computacionalmente efetuada através de métodos de cruzamento (ou recombinação¹), que trocam informações geralmente entre duas soluções distintas. Na adaptação ou mutação, a informação das soluções

¹Em geral, o termo cruzamento indica a utilização da representação binária dos indivíduos. O termo recombinação é preferível para as abordagens que utilizam representações em números reais. Neste trabalho, adota-se este padrão, com exceção para o algoritmo de Michalewicz, que é apresentado na seção 4.3 e possui nomenclatura própria.

é arbitrariamente modificada, como na mutação genética biológica.

A estrutura geral dos algoritmos evolutivos é dada pela seqüência de passos:

1. gerar a população inicial;
2. selecionar heurísticamente alguns indivíduos da população;
3. efetuar o cruzamento (ou recombinação);
4. efetuar a mutação;
5. atualizar a população através de regras de substituição dos indivíduos selecionados pelos indivíduos recém gerados;
6. retornar ao passo 2 até que um critério de parada seja alcançado.

A cada iteração do algoritmo dá-se o nome de geração. Os métodos de seleção, recombinação (ou cruzamento), mutação e as regras de substituição variam entre os diferentes algoritmos evolutivos existentes. O conceito geral discutido não sofre, porém, modificações significativas. A geração da população inicial é variada e dependente do problema. Em muitos casos a geração aleatória dos indivíduos da população inicial é suficiente.

Os AEs são fundamentalmente métodos heurísticos e não determinísticos de busca global. Pela característica probabilística dos AEs, suas aplicações concentram-se em problemas de difícil solução, nos quais métodos determinísticos de busca falham ou produzem soluções insatisfatórias. As principais diferenças entre os AEs e os métodos de busca clássicos e determinísticos, geralmente baseados em informações de gradiente, são:

- os AEs realizam a busca a partir de um conjunto de pontos, a população, e não a partir de um único ponto;
- os AEs utilizam somente a função objetivo como informação para a busca, não necessitando de informações adicionais como o cálculo da sua derivada ou hessiana;
- os AEs utilizam regras de transição probabilísticas.

Atualmente, a teoria dos AEs é composta de três abordagens que se relacionam intimamente, mas que foram desenvolvidas independentemente: os Algoritmos Genéticos, a Programação Evolutiva e as Estratégias Evolutivas [24, 20, 23, 21]. Neste trabalho, o foco de atenção são os AGs, uma variante específica de Evolução Diferencial e o algoritmo Híbrido de Salomon.

4.2 Algoritmos Genéticos

Os Algoritmos Genéticos estão provavelmente entre os mais difundidos algoritmos da Computação Evolutiva. Recebem grande atenção, tanto da comunidade científica quanto prática, pela sua simplicidade e amplo horizonte de aplicação. Os trabalhos predecessores dos atuais AGs foram desenvolvidos por Fraser, um biólogo que buscava simular o processo de evolução, como sumariza Goldberg em [61] e cita Bäck em [20].

A estrutura moderna dos AGs foi inicialmente proposta por Holland [70] em seu trabalho pioneiro na Universidade de Michigan. Psicólogo e cientista da computação, Holland reuniu no seu livro todo o trabalho sobre planos adaptativos e reprodutivos, marcando o início de praticamente todas as aplicações e implementações dos AGs.

O aumento da capacidade computacional na década de 80 permitiu um proeminente impulso no desenvolvimento dos AGs, que passaram a receber atenção não só do meio acadêmico, mas também das indústrias. As principais razões para este vertiginoso crescimento da utilização dos AGs são sua facilidade de implementação e eficácia na solução de problemas complexos. Além disso, os AGs, ao contrário dos métodos de busca determinísticos, não pressupõem quaisquer hipóteses restritivas no espaço de busca, como diferenciabilidade, continuidade, convexidade etc.

Conforme a estrutura geral dos algoritmos evolutivos, a idéia básica dos AGs consiste em: partindo de um conjunto inicial de soluções, ou população inicial, evoluí-las através de mecanismos análogos à reprodução biológica (recombinação e mutação) até que uma solução satisfatória seja encontrada. A figura 6 apresenta o fluxograma de um algoritmo genético básico.

Do grupo de AEs, os AGs são provavelmente aqueles que mais se aproximam da evolução biológica [20]. A proposição inicial de Holland utiliza a representação binária (ou canônica) de cada solução como forma de codificação. Assim, cada solução numérica do problema, o indivíduo ou fenótipo, é representado por um conjunto de bits que carrega sua informação genética, o genótipo, a exemplo do DNA para os seres vivos. Por analogia à biologia, na computação evolutiva utilizam-se também os termos cromossomo para a codificação da solução, e genes para os bits componentes do cromossomo. Uma função de avaliação da aptidão (ou simplesmente função aptidão) positiva é utilizada para se mensurar e se comparar os indivíduos da população, identificando-se aqueles “mais aptos” para a solução do problema.

A população inicial de cromossomos é geralmente gerada de forma aleatória dentro

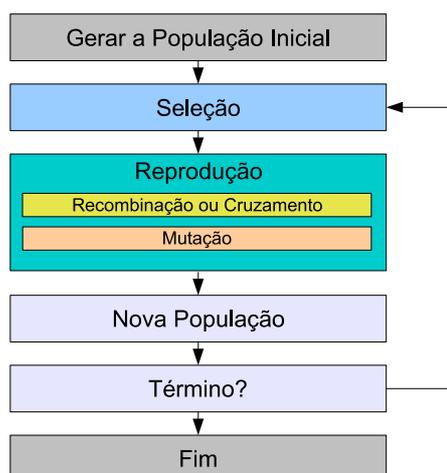


Figura 6: Fluxograma de um algoritmo gen tico b sico.

do dom nio de busca do algoritmo. O tamanho da populaç o utilizada pode variar entre problemas, embora exista um certo consenso quanto ao n mero razo vel de indiv duos. Geralmente, consideram-se 30 elementos uma populaç o pequena, 100 elementos uma populaç o grande e valores entre 70 e 100 suficientes para a maioria dos problemas [120, 85]. A partir da populaç o inicial, a cada geraç o, uma populaç o derivada   criada a partir da populaç o atual, atrav s da aplicaç o dos operadores gen ticos: seleç o, recombinaç o e mutaç o. Neste processo, um subconjunto da populaç o   selecionado atrav s de regras espec ficas e os seus indiv duos s o recombinados para a criaç o dos novos indiv duos, seguindo o esquema b sico de um AE apresentado na seç o 4.1.

O mecanismo de seleç o mais conhecido   o algoritmo da roleta. Nesse algoritmo, cada indiv duo da populaç o recebe uma probabilidade de ser selecionado diretamente proporcional ao valor da respectiva funç o aptid o. Assim, respeitando-se as regras da seleç o natural de Darwin, quanto “melhor” for o indiv duo, ou mais elevado for o valor da sua funç o aptid o, maior a sua probabilidade de ser escolhido para sofrer o cruzamento e a mutaç o. Existem diversas outras maneiras de se efetuar a seleç o, preservando-se as caracter sticas da seleç o natural, por m cada qual com suas peculiaridades. Uma descriç o resumida dos principais m todos utilizados encontra-se na seç o 4.2.3.

Ap s a seleç o dos indiv duos, o operador de cruzamento   respons vel pelo interc mbio de material gen tico entre as diferentes soluç es. No algoritmo proposto por Holland, o cruzamento   efetuado atrav s da permutaç o das cadeias bin rias de duas soluç es, a partir de um ponto de corte, e gera dois novos indiv duos, como ilustrado na figura 7. Ao cruzamento   atribu do uma probabilidade de ocorr ncia p_c . Esta probabilidade indica o percentual de elementos da populaç o aos quais o cruzamento ser  aplicado. Na pr tica,

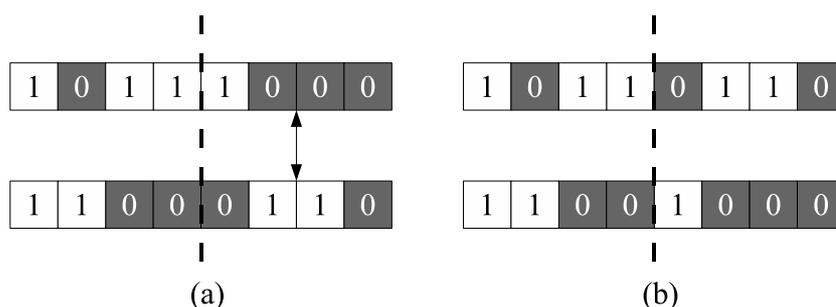


Figura 7: Diagrama do cruzamento.

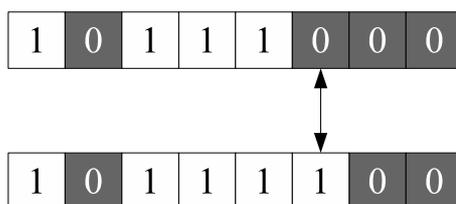


Figura 8: Diagrama da mutação.

recomenda-se probabilidade de cruzamento 0,9 para populações grandes (100 elementos) e 0,6 para populações pequenas (30 elementos) [85, 120].

A última fase da geração é a mutação, que é aplicada individualmente a cada elemento após a recombinação. Este operador modifica o cromossomo através da inversão de um bit selecionado aleatoriamente, gerando um novo indivíduo. À mutação também é atribuída uma probabilidade de ocorrência p_m , que indica o percentual dos indivíduos selecionados que sofrerá mutação. Os valores típicos de probabilidade de mutação, encontrados na literatura, variam entre 0,01 para populações pequenas (30 elementos) e 0,001 para populações grandes (100 elementos) [85, 120]. O mecanismo clássico de mutação, proposto por Holland, é ilustrado na figura 8.

Após aplicados os operadores de seleção, cruzamento e mutação, os elementos novos são reinseridos na população através de uma regra definida. Em geral substitui-se os indivíduos geradores e o processo é reiniciado, repetindo-o até que um critério de parada seja alcançado. A regra de parada pode variar desde um número máximo de gerações até um valor limite para o desvio padrão dos valores da função aptidão na população.

Os parâmetros do algoritmo, como o número de indivíduos N da população, a probabilidade de cruzamento p_c e a probabilidade de mutação p_m , não são de fácil escolha. A natureza do problema tem grande influência nestes parâmetros e não existem regras determinísticas para sua seleção. Os valores observados na literatura mencionados ante-

riormente podem ser uma boa escolha inicial. Alguns ajustes são, em geral, necessários.

4.2.1 Representação

A codificação binária para o genótipo de cada indivíduo, inicialmente proposta por Holland [70], é o mecanismo de codificação mais utilizado, devido à sua simplicidade de conversão. As desvantagens dessa estratégia são o custo computacional extra de codificação e decodificação e a dificuldade de operação com pontos flutuantes.

Michalewicz [93, 91] propôs uma alternativa ao método clássico, utilizando diretamente a codificação em valores reais. Assim, para um problema mono-variável, o fenótipo e o genótipo de cada solução se confundem, não sendo necessárias as etapas de codificação e decodificação. Além disso, para problemas numéricos, o tratamento direto através de números reais não introduz erros de conversão e discretização associados ao problema de codificação e decodificação.

Não existe consenso quanto à melhor forma de representação dos indivíduos. Por possuir um tratamento mais direto, a abordagem de Michalewicz [91], detalhada na seção 4.3, será utilizada neste trabalho. As elevadas dimensão e precisão necessárias nos cromossomos do problema estudado dificultam a utilização da representação canônica.

4.2.2 Função Objetivo e Função Aptidão

A função objetivo é a função para a qual procura-se a solução ótima. A função objetivo varia de problema para problema e pode possuir características extremamente não-lineares influenciando diretamente no funcionamento do AG. Para garantir a uniformidade e a flexibilidade do algoritmo para qualquer função objetivo, esta função é, em geral, escalonada de forma a possuir uma variação linear entre dois extremos positivos, geralmente 0 e 1. A função resultante desse escalonamento, denominada função aptidão (ou no inglês *fitness function*) do elemento x , representada por $\text{apt}(x)$, é utilizada diretamente na avaliação dos indivíduos pelo algoritmo.

O melhor escalonamento da função objetivo constitui um processo crítico na solução do problema, uma vez que uma função objetivo mal escalonada pode gerar indivíduos “muito fortes”, ou seja, com aptidão muito acima da média da população. Este fato, em conjunto com a escolha inadequada do método de seleção, pode ocasionar a convergência prematura da população, fenômeno detalhado na próxima seção.

4.2.3 Seleção

O mecanismo de seleção possui um papel determinante na convergência do algoritmo, pois é responsável por garantir a sobrevivência dos indivíduos mais aptos sem permitir que uma única solução domine a população. O princípio da seleção é, através de uma regra específica, priorizar os indivíduos “mais aptos” da população na escolha de quais deles passarão pela reprodução. A regra mais direta é atribuir uma probabilidade de seleção diretamente proporcional ao valor da função aptidão. Este processo é denominado seleção por roleta [120].

Quando a função objetivo é não-linear e difícil de ser adequadamente escalonada, a evolução do algoritmo pode gerar um indivíduo com aptidão muito elevada em comparação com a média da população. Quando isso acontece, o método de seleção por roleta prioriza este indivíduo proporcionalmente, atribuindo-o uma probabilidade de seleção muito maior do que para os outros elementos da população. Com isso, este mesmo indivíduo é sempre escolhido para a reprodução e sua informação genética domina a população. Este fato é indesejável, pois nestes casos, a diversidade da população se reduz em poucas gerações. A este fenômeno denomina-se convergência prematura, pois o algoritmo converge a população para uma única solução em poucas gerações e esta solução, em geral, não é a solução ótima do problema.

Pode-se contornar o problema de convergência prematura do algoritmo através da escolha de outro mecanismo de seleção que não o da roleta. Um mecanismo bastante difundido é a seleção por *rank*. Neste mecanismo, ordenam-se os elementos através dos valores da função aptidão de maneira crescente. A seguir, a cada indivíduo é atribuída uma posição no *rank*, sendo que para uma população com N indivíduos, o menos apto recebe 1 e o mais apto, o valor N . A probabilidade de seleção é aplicada proporcionalmente a essa posição no *rank*. Assim, mesmo que a função aptidão seja mal escalonada, a diferença de probabilidade de seleção entre o indivíduo mais apto e o menos apto não excede o fator N . A tabela 1 apresenta um exemplo da diferença entre os algoritmos de seleção por roleta e por *rank*. A população do exemplo possui 5 indivíduos e o objetivo é minimizar a função $f(x) = x^2, \forall x \geq 0$. Observa-se que, neste caso, quando se utiliza o método de seleção por roleta, a probabilidade de seleção atribuída ao indivíduo x_1 é muito superior à probabilidade média dos outros indivíduos da população, o que pode causar a convergência prematura. Quando, porém, se utiliza o método de seleção por *rank*, observa-se que as probabilidades dos diferentes indivíduos $P_{rank}(x_i)$ são mais bem distribuídas, evitando-se a convergência da população para o elemento x_1 .

Tabela 1: Exemplo de algoritmos de seleção.

	$\text{apt}(x_i)$	$P_{\text{roleta}}(x_i)$	Valor do rank	$P_{\text{rank}}(x_i)$
x_1	4,04	96,75%	5	33,33%
x_2	304,54	1,28%	4	26,67%
x_3	431,68	0,90%	3	20,00%
x_4	623,72	0,63%	2	13,33%
x_5	890,23	0,44%	1	6,67%

Além da seleção por rank , pode-se selecionar os indivíduos por torneio. Este mecanismo se dá através de N seleções prévias aleatórias de N_T indivíduos da população, onde N_T é uma parcela de N que corresponde ao número de elementos na população. De cada um dos N subconjuntos aleatórios da população seleciona-se o indivíduo mais apto dentre os N_T indivíduos. O conjunto resultante forma a população de indivíduos selecionados. Este processo, por sua característica aleatória na seleção das sub-populações de tamanho N_T , é, como a seleção por rank , mais eficaz no contorno do problema de convergência prematura. A desvantagem da utilização da seleção por torneio, em comparação com a seleção por rank , é a escolha do tamanho do torneio N_T . Este é um parâmetro de projeto e sua melhor seleção depende do problema abordado.

4.2.4 Probabilidades de Cruzamento e Mutação

A escolha das probabilidades ou taxas de cruzamento p_c e de mutação p_m é uma tarefa bastante delicada. Não existe uma regra determinística para a escolha destes valores. Observa-se que um aumento excessivo na taxa de recombinação eleva a troca de informações entre os bons indivíduos, porém também eleva a “quebra” ou deterioração dos mesmos. Por outro lado, um aumento excessivo na taxa de mutação transforma o processo de busca em um método puramente estocástico. Alguns dos métodos, como descrito em [85], sugerem taxas variáveis de probabilidade ao longo das gerações. A justificativa reside no fato de que à medida em que a população converge para um sub-espço do espaço de busca, uma taxa de mutação elevada aliada a uma reduzida taxa de recombinação pode transportar o algoritmo para outro sub-espço mais próximo do ótimo global da função, evitando assim sua convergência prematura.

Como citado na introdução deste capítulo, em geral, recomendam-se probabilidades de cruzamento em torno de 0,9 para populações grandes (100 elementos) e 0,6 para populações pequenas (30 elementos). Os valores típicos de probabilidade de mutação,

encontrados na literatura, variam entre 0,01 para populações pequenas (30 elementos) e 0,001 para populações grandes (100 elementos) [85, 120].

4.2.5 Reinservação e Elitismo

Após as operações de seleção, cruzamento e mutação, uma nova população é formada. No algoritmo clássico, a população da geração anterior é simplesmente substituída pela população recentemente formada. Este mecanismo não garante que o melhor elemento da população seja armazenado. Assim, pode ocorrer que o melhor indivíduo da população tenha sido deteriorado através da recombinação e/ou da mutação, tornando mais lenta a evolução do algoritmo. Para evitar que isto ocorra, outras estratégias de substituição dos elementos podem ser utilizadas. A estratégia elitista, proposta por Goldberg [61], substitui o “pior” indivíduo² da população na geração g pelo indivíduo “mais apto”³ da geração $g - 1$. Com isso, o melhor elemento de cada geração é mantido, acelerando-se a convergência do algoritmo.

A estratégia elitista, quando aliada a estratégias de escalonamento e de seleção inadequadas, pode contribuir para a convergência prematura do algoritmo. Em geral, porém, a sua utilização melhora consideravelmente o desempenho do algoritmo.

4.3 Algoritmo de Michalewicz

Esta seção discute o algoritmo genético desenvolvido por Michalewicz em [93, 91]. As principais características deste algoritmo são: a representação direta em ponto flutuante e o tratamento do problema de otimização restrita para o caso de restrições convexas sem o uso de penalidade [38]. O problema abordado neste trabalho envolve a otimização multi-objetivo com restrições de desigualdades matriciais. Assim, a abordagem de Michalewicz adequa-se ao objetivo a ser alcançado.

A representação em ponto flutuante é preferível nos problemas em que o espaço de busca é amplo ou desconhecido e quando elevada precisão na solução é necessária. Isto acontece porque, nestes casos, a representação binária pode-se tornar computacionalmente ineficaz, devido à elevada dimensão do genótipo necessária para a representação do espaço de busca com a precisão adequada. Além disso, os processos de codificação e decodificação

²O “pior” ou “menos apto” indivíduo ou cromossomo é aquele que possui pior avaliação da função aptidão em relação aos outros indivíduos da população.

³O “melhor” ou “mais apto” indivíduo ou cromossomo é aquele que possui melhor avaliação da função aptidão em relação aos outros indivíduos da população.

podem aumentar significativamente o tempo de execução do algoritmo. Assim, o problema que surge é como definir os operadores de recombinação e mutação, de tal maneira que as características do AG clássico de Holland sejam mantidas. Uma opção é a abordagem descrita em [93], na qual Michalewicz propõe três operadores de mutação e outros três de cruzamento⁴ utilizando a representação em números reais. Esta abordagem é descrita na seqüência.

A teoria de AGs tem sido largamente utilizada em problemas de otimização irrestritos. Na prática, porém, é conveniente que o algoritmo seja capaz de tratar restrições, pois diversas aplicações reais se traduzem em problemas restritos. Diversas implementações dos AGs para problemas com restrições utilizam o conceito de penalidade. Gen e Cheng descrevem em [56] as principais técnicas de penalização utilizadas nos AGs. O conceito de penalidade se refere à adição de um termo à função aptidão, de tal forma que, quando a solução gerada é inactível, esta seja “penalizada”, forçando o algoritmo a retornar ao espaço de busca factível.

Outra forma de se abordar os problemas restritos é garantir que os indivíduos resultantes das operações genéticas pertençam ao espaço de factibilidade do problema. Isto, porém, nem sempre é possível, devido a forma geométrica complexa do espaço de soluções do problema. Para um tipo especial de restrições, as restrições convexas, porém, a teoria já estabelecida permite garantir a geração apenas de indivíduos factíveis [93].

Michalewicz [93, 91] propõe um AG que dispensa o uso das penalidades partindo do pressuposto que a população inicial seja factível e que o conjunto de restrições seja convexo. Considera-se o problema de otimização restrita:

$$\begin{aligned} & \text{otimizar } f(x) && (4.1) \\ \text{s.a } & x \in \Omega \subset \mathbb{R}^n \end{aligned}$$

onde Ω é o espaço de factibilidade da função $f(x)$, definido por um conjunto de restrições convexas. Michalewicz desenvolveu três operadores de mutação e outros três de recombinação que são aplicados sobre as componentes dos vetores x com representação em ponto flutuante. Estes operadores garantem que as soluções geradas a partir da população factível também sejam factíveis. Os operadores descritos por Michalewicz são detalhados a seguir.

⁴Apesar de não utilizar a representação canônica, Michalewicz utiliza o termo cruzamento para os operadores de recombinação. Optou-se por se manter a notação original do autor.

4.3.1 Mutação Uniforme

Seja um vetor factível $x = [x_1, \dots, x_k, \dots, x_n] \in \Omega$. Seja $[left(x_k), right(x_k)]$ o intervalo composto pelo conjunto de valores y que a componente x_k do vetor x pode assumir, tais que

$$y \in [left(x_k), right(x_k)] \Leftrightarrow [x_1, \dots, x_{k-1}, y, x_{k+1}, \dots, x_n] \in \Omega$$

onde $x_i, i = 1, \dots, n; i \neq k$ permanecem constantes. O operador mutação uniforme seleciona aleatoriamente uma componente $k = 1, \dots, n$, do vetor $x = [x_1, \dots, x_k, \dots, x_n]$ e produz o vetor $x' = [x_1, \dots, x'_k, \dots, x_n]$, onde x'_k é uma variável aleatória uniformemente distribuída no intervalo $[left(x_k), right(x_k)]$. Em particular, este operador é importante quando a população inicial é formada por diversas cópias do mesmo indivíduo. Isto acontece em problemas complexos, nos quais frequentemente o usuário deve indicar o ponto de partida do algoritmo [93].

4.3.2 Mutação de Contorno

Este operador gera elementos sobre os limites do espaço de busca factível Ω . Analogamente à mutação uniforme, seleciona-se aleatoriamente uma componente $k = 1, \dots, n$ do vetor $x = [x_1, \dots, x_k, \dots, x_n]$ e produz o vetor $x' = [x_1, \dots, x'_k, \dots, x_n]$, onde $x'_k = left(k)$ ou $x'_k = right(k)$, com mesma probabilidade para ambas.

Este operador tem importante função quando as soluções para o problema de otimização se encontram nos limites do espaço de busca. O espaço de busca do problema abordado neste trabalho é desconhecido, fato que limita a aplicação direta da mutação de contorno.

4.3.3 Mutação Não-Uniforme

Apesar da elevada capacidade de exploração do espaço de busca do problema, uma das características indesejáveis dos AGs clássicos é sua limitada capacidade de busca local, ou o refinamento da solução. O operador de mutação não-uniforme é um método de mutação adaptativo e localizado. Adaptativo, pois aumenta a probabilidade de mutação, e localizado pois este aumento ocorre nas proximidades do cromossomo selecionado. Assim, o espaço de busca da mutação é reduzido gradativamente permitindo o refinamento da solução.

Seleciona-se aleatoriamente uma componente $k = 1, \dots, n$, do vetor $x = [x_1, \dots, x_k, \dots, x_n]$

e produz-se o vetor $x' = [x_1, \dots, x'_k, \dots, x_n]$ onde x'_k é dado por:

$$x'_k = \begin{cases} x_k + \Delta(t, right(k) - x_k) & \text{se um dado número aleatório é igual a 0} \\ x_k - \Delta(t, x_k - left(k)) & \text{se um dado número aleatório é igual a 1,} \end{cases} \quad (4.2)$$

onde

$$\Delta(t, y) = yr \left(1 - \frac{t}{T}\right)^b, \quad (4.3)$$

r é um número real aleatório entre 0 e 1, T é o número máximo de gerações definido e b é um parâmetro de não-uniformidade. A função $\Delta(t, y)$ retorna um valor no intervalo $[0, y]$ tal que a probabilidade de $\Delta(t, y)$ se aproximar de 0 aumenta à medida que o número de gerações t aumenta. Essa propriedade faz com que o operador busque em uma vizinhança de x cada vez menor, localizando a busca e refinando o resultado [93]. Nos algoritmos implementados, o conceito de mutação uniforme é utilizado para o desenvolvimento de estratégias semelhantes de localização.

4.3.4 Cruzamento Aritmético

O operador cruzamento aritmético combina dois cromossomos x_1 e x_2 gerando dois novos indivíduos x'_1 e x'_2 através da combinação convexa entre os mesmos:

$$\begin{cases} x'_1 = \alpha x_1 + (1 - \alpha)x_2 \\ x'_2 = (1 - \alpha)x_1 + \alpha x_2, \end{cases} \quad (4.4)$$

onde α é uma variável aleatória no intervalo $[0, 1]$. As propriedades de convexidade do espaço factível garantem que, através da combinação convexa dos elementos x_1 e x_2 , os elementos x'_1 e x'_2 pertencem ao conjunto factível Ω .

4.3.5 Cruzamento Simples

O cruzamento simples é o operador que mais se aproxima do cruzamento proposto por Holland [70], no algoritmo de representação binária. Sejam os cromossomos $x_1 = [x_1, \dots, x_n]$ e $x_2 = [y_1, \dots, y_n]$ e uma posição k escolhida aleatoriamente na

seqüência das componentes dos vetores. Os vetores x'_1 e x'_2 são definidos por:

$$\begin{cases} x'_1 = [x_1, \dots, x_k, y_{k+1}, \dots, y_n] \\ x'_2 = [y_1, \dots, y_k, x_{k+1}, \dots, x_n] \end{cases} \quad (4.5)$$

Nenhuma propriedade garante que x'_1 e x'_2 pertençam ao conjunto Ω . Quando ocorre a infactibilidade desses elementos, uma propriedade particular dos conjuntos convexos é utilizada para modificar os vetores obtidos. Esta propriedade garante que existe $\alpha \in [0, 1]$ tal que:

$$\begin{cases} x'_1 = [x_1, \dots, x_k, y_{k+1}\alpha + x_{k+1}(1 - \alpha), \dots, y_n\alpha + x_n(1 - \alpha)] \\ x'_2 = [y_1, \dots, y_k, x_{k+1}\alpha + y_{k+1}(1 - \alpha), \dots, x_n\alpha + y_n(1 - \alpha)] \end{cases} \quad (4.6)$$

sejam elementos factíveis. O problema se reduz a encontrar o maior valor possível de α tal que x'_1 e x'_2 sejam elementos factíveis [93].

4.3.6 Cruzamento Heurístico

O operador de cruzamento heurístico destaca-se dos demais por algumas particularidades, sejam elas:

- utilização dos valores da função aptidão na determinação da direção de busca;
- geração de apenas 1 elemento a partir da combinação de outros 2;
- o operador pode, eventualmente, não produzir nenhum elemento.

O elemento x_3 resultante do cruzamento heurístico dos elementos x_1 e x_2 é dado por:

$$x_3 = r(x_2 - x_1) + x_2, \quad (4.7)$$

onde r é um número aleatório entre 0 e 1, e $f(x_2) \geq f(x_1)$ para problemas de maximização e $f(x_2) \leq f(x_1)$ para problemas de minimização. Quando x_3 não é factível, o fator r é dividido por 2 e tentativas sucessivas são efetuadas até se produzir um elemento x_3 factível ou até se alcançar um número máximo de tentativas. Quando nenhum elemento factível x_3 é gerado, aplica-se um dos outros dois operadores de cruzamento, garantindo a factibilidade dos elementos gerados.

Por utilizar a informação de direção, o cruzamento heurístico contribui para a aceleração do processo de busca, bem como melhora a precisão da solução encontrada [93].

4.4 Evolução Diferencial

O algoritmo de Evolução Diferencial (ED) foi proposto inicialmente por Reiner Storn e Kenneth Price, em [122, 123], com o intuito de minimizar uma função de variáveis reais. O algoritmo de ED é extremamente robusto na busca por mínimos globais [2]. O algoritmo de ED, assim como os AGs, parte de um conjunto de soluções candidatas iniciais e as evolui através de um mecanismo de reprodução (cruzamento e mutação) e uma avaliação final de aceitação. Ao contrário dos AGs, que podem evoluir apenas uma parcela da população a cada geração, no algoritmo de ED todos os elementos da população passam pelo processo de reprodução e aceitação. Assim, a cada geração, a população de soluções candidatas é inteiramente substituída.

Em seu trabalho original [122], Storn e Price sugerem duas variantes de Evolução Diferencial. Outras sugestões do algoritmo são descritas em [123, 2]. O princípio básico dos diferentes algoritmos de ED é a alteração de um vetor solução x_1 a partir da utilização de dois outros vetores, x_2 e x_3 , todos selecionados aleatoriamente [122]. O algoritmo inicial de Storn e Price parte de uma população inicial construída aleatoriamente, e, para cada elemento $x_i, i = 1, \dots, N$, da população:

1. gera-se um vetor temporário v tal que:

$$v = x_1 + F \times (x_2 - x_3), \quad (4.8)$$

onde x_1, x_2 e x_3 são elementos da população atual escolhidos aleatoriamente, distintos entre si e diferentes do elemento x . F é um fator de escala real, constante e positivo;

2. gera-se um vetor temporário u , tal que:

$$u_j = \begin{cases} v_j & \text{para } j = \Phi \text{ ou } R \leq p_c \\ x_j & \text{nos outros casos,} \end{cases} \quad (4.9)$$

onde o índice j representa cada elemento do vetor correspondente, $\Phi \in [0, D - 1]$ e $R \in [0, 1]$ são selecionados aleatoriamente, D é a dimensão do vetor x e $p_c \in [0, 1]$ é a probabilidade de cruzamento;

3. se $\text{apt}(u) > \text{apt}(x)$ então substitui-se x por u ;

Em uma analogia grosseira com os algoritmos genéticos, pode-se entender o passo 1 como um método de recombinação que gera v e o passo 2 como um operador de mutação que forma o vetor u . O passo 1 lembra o operador de cruzamento heurístico de Michlewicz (seção 4.3.6). A diferença entre os dois operadores encontra-se nos elementos x_1 e x_3 , que correspondem ao mesmo elemento no caso do operador de Michalewicz. A exemplo do cruzamento heurístico, o passo 1 da ED utiliza o vetor direção $(x_2 - x_3)$ para modificar a solução aleatória x_1 , o que traz maior eficiência ao algoritmo.

O passo 2 da ED aumenta a diversidade dos elementos do vetor solução gerado. A regra de aceitação, definida pelo passo 3, elimina a possibilidade do algoritmo piorar a aptidão média da população atual. Por outro lado, essa característica do algoritmo pode confinar as soluções em um vale local da função a ser otimizada.

Diversos trabalhos, dentre eles [122, 123, 1, 2], apresentam a ED como um método não só robusto, mas também confiável na solução de problemas complexos de busca. Ressaltam os autores que uma importante característica deste algoritmo é o reduzido número de parâmetros de projeto, permitindo mais fácil ajuste do método para cada problema específico.

4.5 Algoritmo de Salomon

O algoritmo de Salomon, *Evolutionary-Gradient Search* (EGS)⁵, foi proposto por Ralf Salomon, em [114]. Seu algoritmo explora as semelhanças entre os algoritmos clássicos de busca, baseados na utilização das informações de gradiente, e os métodos heurísticos da computação evolutiva. O EGS procura mesclar as vantagens de ambas classes.

A minimização de uma função contínua de n variáveis é um dos problemas clássicos de otimização não-linear [86]. O problema típico é especificado por um vetor x de n parâmetros $\{x_1, x_2, \dots, x_n\}$ e por uma função objetivo $f(x)$ a qual se deseja otimizar. No contexto dos algoritmos evolutivos, a função $f(x)$ é denominada função aptidão. O objetivo do problema é encontrar um vetor x_{opt} tal que o valor da função $f(x_{opt})$ seja ótimo. Sem perda de generalidade, é suficiente considerar apenas os problemas de minimização, uma vez que minimizar $f(x)$ corresponde a maximizar $-f(x)$.

Existem diversas maneiras de se solucionar um problema de minimização. Para pro-

⁵Por questões de conveniência, optou-se manter a sigla original em inglês.

blemas de otimização envolvendo funções reais, contínuas e diferenciáveis, Newton desenvolveu um método baseado no gradiente da função a ser otimizada $f(x)$. O método de Newton (também conhecido como método da máxima descida⁶) é um método iterativo que pode resumidamente ser descrito como:

1. seja uma função objetivo $f(x)$ real, contínua e diferenciável;
2. dado um vetor, ou iterando, x_k , encontrar uma direção d de descida para $f(x)$;
3. encontrar o iterando x_{k+1} , tal que

$$x_{k+1} = x_k + \eta d, \quad (4.10)$$

onde η denota uma constante e é conhecida como tamanho de passo⁷;

4. voltar ao passo 2 até que uma solução satisfatória tenha sido encontrada.

Newton observou que o gradiente da função $f(x)$ em um ponto x aponta para a direção de máximo crescimento da função neste ponto. Com essa propriedade, Newton adotou o negativo do gradiente da função objetivo como direção de descida d , ou seja:

$$d = -\nabla f(x) = -\left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n}\right)^T. \quad (4.11)$$

O método de Newton inicia-se com um vetor x_0 e então, repetidamente, subtrai do iterando x_k uma pequena fração η do gradiente da função $\nabla f(x)$. É garantido que, para passos η suficiente pequenos, o método de máxima descida converge para um mínimo local da função.

Dentre as principais vantagens dos algoritmos de máxima descida encontram-se a prova de convergência, sua velocidade e seu detalhamento matemático. As mais importantes desvantagens desta classe de métodos de otimização são a falta de convergência para o ótimo global da função e as hipóteses restritivas sobre a função objetivo. Para maiores detalhes sobre os métodos baseados em gradiente, pode-se consultar [109, 84, 86].

Nos problemas em que a respectiva função objetivo não cumpre os requisitos de continuidade e diferenciabilidade, faz-se necessária a utilização de uma técnica alternativa, como os AEs apresentados neste capítulo. Lembra-se que as principais diferenças entre os AEs e os métodos clássicos de otimização são: os AEs são heurísticos; em geral, não

⁶Do inglês *steepest descent method*.

⁷Do inglês *step size*.

possuem um detalhamento matemático formal, como por exemplo prova de convergência e; no processo de busca, os AEs utilizam o conhecimento não só de um único ponto, mas de um conjunto de soluções candidatas. Outras características relevantes dos AEs podem ser encontradas na seção 4.1 e em [61, 118, 127, 114].

À primeira vista, induz-se à conclusão de que os AEs e os métodos baseados em gradientes possuem características bem distintas. Acredita-se que o potencial dos AEs origina-se da busca paralela por meio de uma população de soluções candidatas. Em especial, a habilidade de escapar de um ótimo local, uma característica que dificilmente é encontrada nos métodos clássicos de otimização, é atribuída ao uso de populações. Salomon argumenta porém, que as diferenças entre os métodos clássicos de otimização e os AEs residem apenas no ponto de vista do observador e que ambas técnicas são mais semelhantes do que se possa parecer [114].

Alguns autores argumentam que os AEs, a exemplo dos métodos clássicos, são métodos que seguem uma direção de busca [110], e como exemplo citam as Estratégias Evolutivas (ES). Outros autores ressaltam, porém, que os métodos evolutivos não podem ser apropriadamente descritos como métodos que seguem uma direção de busca [15]. Salomon defende que ambos pontos de vista possuem seus méritos. Em seu artigo [114], Salomon descreve o comportamento de um algoritmo de estratégia evolutiva. Ao efetuar a projeção do vetor solução no espaço bidimensional, Salomon resalta que o algoritmo estudado apresenta um comportamento que “segue” a direção do gradiente. Entretanto, se, ao invés da projeção, dois parâmetros quaisquer x_i e x_j do vetor x forem selecionados para investigação, observa-se claramente um comportamento aleatório dos parâmetros durante o processo de otimização. Salomon resalta que experimentos demonstraram que, para problemas de dimensão n reduzida e um algoritmo de estratégia evolutiva $(1, \lambda)$ - para cada solução candidata selecionada da população são geradas outras λ soluções - o algoritmo referido demonstra um comportamento de busca que segue a direção do gradiente. O comportamento aleatório da busca é observado, porém, se o módulo dos vetores resultantes da mutação são maiores do que a distância do ponto atual até o ótimo. Semelhantemente, nos métodos clássicos, quando o tamanho de passo η é superestimado, a busca pode não convergir e possuir um comportamento aleatório. Maiores detalhes podem ser encontrados em [15, 114].

Outro ponto discutido por Salomon diz respeito à busca multi-ponto dos AEs em contraste à busca através de um único ponto dos métodos clássicos. Muitos autores, como por exemplo Golberg, Bäck e Beyer, em [61, 24, 22, 15], argumentam que a busca através

da população é a maior vantagem dos AEs, proporcionando-lhes a interessante característica de conseguir “escapar” de um ótimo local. Essa característica está amparada em uma grande quantidade de evidências empíricas colecionadas na maioria das publicações que dizem respeito ao assunto. Salomon argumenta que, em algoritmos de estratégia evolutiva nos quais um operador intermediário de recombinação é aplicado aos pais antes da geração das soluções filhas, o conceito multi-ponto dos algoritmos evolutivos parece ser contraditório. Isso porque o operador de recombinação intermediário calcula o centro de massa dos elementos pais selecionados, gerando apenas uma solução intermediária na qual o processo de recombinação é aplicado. Essa única solução intermediária, utilizada pelo algoritmo para a geração dos elementos filhas, parece contradizer a busca multi-ponto dos algoritmos evolutivos. Salomon ressalta que essa visão está em consonância com os resultados apresentados por Schwefel, em [119].

Salomon argumenta também, que os algoritmos de máxima descida não utilizam a informação de apenas um ponto na busca pelo ótimo, como se sugere à primeira vista. A determinação do gradiente, passo 3 do algoritmo descrito, é realizada através das derivadas parciais da função objetivo, dadas por

$$\frac{\partial f(x)}{\partial x_i} = \lim_{\Delta x \rightarrow 0} \left[\frac{f(\dots, x_i + \Delta x, \dots) - f(\dots, x_i, \dots)}{\Delta x} \right], \quad (4.12)$$

sobre cada coordenada x_i , $i = 1, \dots, n$. Assim, pode-se considerar que a equação (4.11) utiliza informações de n diferentes pontos para a determinação da direção de busca. Estes n pontos definem um conjunto, ou população, de indivíduos distribuídos ortogonalmente com distâncias infinitesimais. Desta forma, o algoritmo de Newton também possui a característica de busca multi-ponto. Nos casos em que a função objetivo não é fornecida, o gradiente da função objetivo pode ser estimado através de aproximações discretas das derivadas parciais, dadas por

$$\frac{\partial f(\vec{x})}{\partial x_i} \approx \left[\frac{f(\dots, x_i + \Delta x, \dots) - f(\dots, x_i, \dots)}{\Delta x} \right]. \quad (4.13)$$

Salomon [114] argumenta que, se a aproximação discreta do gradiente utilizar um valor Δx apropriado, a direção de busca pode diferir significativamente do gradiente real, introduzindo um caráter aleatório à busca. Essa propriedade aleatória pode trazer ao algoritmo a desejável característica de escapar de mínimos locais. Assim, as diferenças entre as duas classes de algoritmos, os Algoritmos Evolutivos e os métodos clássicos de busca por gradiente, podem ser reduzidas a pontos de vista distintos.

Com base nesses argumentos, Salomon desenvolveu seu algoritmo apresentado em

[114]. A estrutura básica do EGS⁸ é apresentada a seguir:

1. para cada iteração, ou geração, k :
 - (a) estimar a direção do gradiente, inspirado nos algoritmos evolutivos;
 - (b) calcular o próximo iterando de forma análoga ao algoritmo de Newton.

Desta forma, Salomon procurou mesclar as interessantes propriedades das duas classes distintas de algoritmos de otimização. O algoritmo de Salomon (EGS) é iterativo e, para cada geração k , segue os seguintes passos:

1. para um dado vetor solução x_k , gerar λ vetores candidatos (*offspring*) $t_1 = x_k + z_1, t_2 = x_k + z_2, \dots, t_\lambda = x_k + z_\lambda$ onde z_i são vetores de mutação aleatórios com componentes normalmente distribuídos com média zero e desvio padrão σ_k/\sqrt{n} , e n é número de componentes do vetor x_k ;
2. calcular o vetor gradiente estimado

$$\tilde{g}_k = \sum_{i=1}^{\lambda} (f(t_i) - f(x_k)) (t_i - x_k); \quad (4.14)$$

3. calcular o vetor unitário de direção

$$e_k = \frac{\tilde{g}_k}{\|\tilde{g}_k\|}; \quad (4.15)$$

4. adaptar o tamanho de passo

$$\sigma_{k+1} = \begin{cases} \sigma_k \zeta, & \text{se } f(x_k - \sigma_k \zeta e_k) \leq f(x_k - (\sigma_k/\zeta) e_k), \\ \sigma_k/\zeta, & \text{caso contrário,} \end{cases} \quad (4.16)$$

onde ζ é um parâmetro do algoritmo;

5. calcular o próximo ponto

$$x_{k+1} = x_k - \sigma_{k+1} e_k. \quad (4.17)$$

O passo 1 gera λ vetores z que, para $n \gg 1$, estão igualmente distribuídos em uma hipersfera com raio σ_k ($\|z_i\| \approx \sigma_k$). Quanto maior o número de vetores λ , maior a aproximação do gradiente da função objetivo. A escolha adequada de λ se faz necessária

⁸Do inglês *Evolutionary-Gradient-Search Procedure*, nome dado por Salomon ao seu algoritmo.

para garantir ao mesmo tempo a velocidade de convergência e um erro de estimação capaz de desviar a evolução do algoritmo dos mínimos locais. O somatório do passo 2 resulta na estimativa do gradiente da função objetivo $f(x_k)$. O processo é semelhante à diferenciação por aproximação discreta e sua prova encontra-se em [114]. No passo 3, o vetor unitário na direção do gradiente estimado é calculado. Este é utilizado para o cálculo do próximo iterando em 5, a exemplo dos algoritmos clássicos de otimização. O passo 4, a exemplo de alguns algoritmos de máxima descida, adapta o tamanho de passo na tentativa de se maximizar a eficiência do algoritmo.

Na forma mais simples do algoritmo, Salomon propõe que o valor de σ_0 seja calculado aleatoriamente através de uma distribuição lognormal com média um e $\zeta \approx 1,8$. Estes valores são empíricos, baseados em experimentos e podem ser modificados conforme a natureza do problema a ser tratado.

A exemplo dos algoritmos evolutivos, Salomon defende que o EGS gera λ vetores através da mutação (passo 1). Ao contrário da maioria dos AEs, que seleciona apenas o indivíduo mais apto gerado na reprodução, o EGS utiliza a informação de todos os elementos t_i gerados para o cálculo do gradiente estimado. O EGS possui um caráter híbrido que combina os procedimentos aleatórios dos AEs com a estrutura determinística dos algoritmos clássicos de otimização. Neste trabalho, o EGS é utilizado como um método de se direcionar a busca do algoritmo apresentado no capítulo 5.

4.6 Conclusão

Este capítulo apresenta as três técnicas escolhidas para a solução do problema de controle robusto. Os Algoritmos Genéticos são bastante difundidos pela sua facilidade de implementação, por sua robustez e confiabilidade. O algoritmo de Michalewicz possui a característica de permitir o tratamento direto de restrições convexas sem a utilização de funções de penalização, além de dispensar a codificação dos cromossomos, que são representados por números reais. A Evolução Diferencial, amplamente explorada na literatura por sua robustez, demonstra ótimos resultados com sua aplicação [122, 123, 129]. O algoritmo inicial, proposto por Storn e Price [122, 123], foi selecionado por sua facilidade de compreensão e implementação. O algoritmo de Salomon, por sua vez, é a técnica que mais se aproxima dos métodos clássicos de otimização e tem como característica a possibilidade de estimar a direção do gradiente da função objetivo. No problema de controle abordado, o algoritmo de Salomon aparece como uma poderosa ferramenta na estimativa

do gradiente da função objetivo, diminuindo o caráter estritamente aleatório da busca.

5 Algoritmo Híbrido

Neste capítulo é apresentado um conjunto de algoritmos evolutivos para a solução do problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$, baseada na teoria apresentada no capítulo 4. Unindo-se as propriedades dos métodos LMI e as técnicas de computação evolutiva apresentadas, busca-se um conjunto de algoritmos capaz de obter soluções satisfatórias para diferentes problemas de controle. Nesse sentido, implementou-se em uma estrutura unificada os diferentes algoritmos evolutivos, no intuito de mesclá-los, preservando as peculiaridades de cada um. O algoritmo é híbrido no sentido que utiliza não somente diferentes algoritmos evolutivos, mas também o algoritmo de otimização LMI.

A seção 5.1 apresenta a estrutura geral do método de solução proposto e o conjunto de algoritmos evolutivos desenvolvido. A função objetivo utilizada, bem como os operadores de seleção implementados são descritos nas seções 5.2 e 5.3, respectivamente. O método de elitismo implementado é apresentado na seção 5.4. O algoritmo proposto para a geração da população inicial é discutido na seção 5.5. Os diferentes grupos de operadores de recombinação utilizados no algoritmo são descritos na seção 5.6. Na seção 5.7 apresentam-se os três grupos de operadores de mutação implementados. Finalmente, uma breve conclusão do capítulo é realizada na seção 5.8.

5.1 Estrutura Geral

O algoritmo proposto combina as propriedades da busca heurística e a confiabilidade dos AGs [85, 91] e DEs [122, 2, 129] com a acurácia e eficiência dos métodos de solução através de busca por gradiente [1] e de otimização LMI [17, 95].

Baseado na estrutura geral de um algoritmo evolutivo, descrito no capítulo 4, e apresentada na figura 9, o algoritmo busca um controlador robusto ótimo \mathcal{D} -estável L_K (3.95) e, conseqüentemente, determina X_2 , X_∞ e X_D que solucionam o problema de otimização (3.97), satisfazendo as equações (3.98)-(3.100).

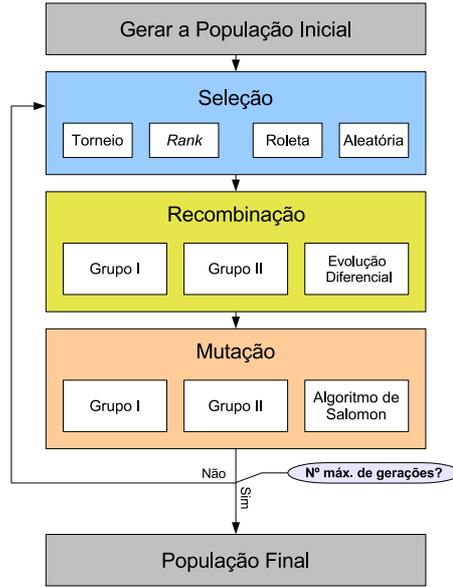


Figura 9: Algoritmo evolutivo h brido

O cromossomo   formado pelo conjunto de matrizes $\begin{bmatrix} L_K & X_2 & X_\infty \end{bmatrix}$. Esta opç o foi feita com o objetivo de aumentar a efic cia do algoritmo e a otimizaç o   realizada atrav s dos operadores de seleç o, mutaç o e recombinaç o sobre este cromossomo. Nenhuma operaç o   efetuada sobre a matriz X_D uma vez que esta n o tem relaç o direta com a funç o objetivo. A matriz X_D aparece como resultado da aplicaç o da otimizaç o LMI sobre cada cromossomo.

A cada geraç o o problema de otimizaç o (3.97)   solucionado para todos os cromossomos da populaç o de tamanho p_s , utilizando-se a caixa de ferramentas LMI-Lab [55] do Matlab. O crit rio de parada do algoritmo   o n mero m ximo de geraç es n_{gen} .   importante ressaltar que, tanto para um controlador fixo $L_K = [l_{ij}]_{(m+n_c) \times (p+n_c)}$ quanto para um par de matrizes (X_2, X_∞) dado, o problema de controle robusto $\mathcal{H}_2/\mathcal{H}_\infty$ (3.97) sujeito  s restriç es (3.98)-(3.100)   convexo. A restriç o de norma \mathcal{H}_∞ , γ ,   fornecida ao algoritmo *a priori*, por m o problema de controle robusto mais geral, definido em [55], com o crit rio de desempenho

$$\min_{L_K} \alpha \|T_{z_\infty w_\infty}\|_\infty^2 + \beta \|T_{z_2 w_2}\|_2^2, \quad (5.1)$$

onde $\alpha > 0$ e $\beta > 0$ com as restriç es (3.98)-(3.100) tamb m pode ser resolvido com este algoritmo.

5.2 Função Objetivo e Função Aptidão

A função objetivo $J(L_K, X_2, X_\infty)$ proporciona um mecanismo de comparação entre cada cromossomo $\begin{bmatrix} L_K & X_2 & X_\infty \end{bmatrix}$. Neste algoritmo a função objetivo corresponde à norma \mathcal{H}_2 (veja seção 3.4) do sistema, uma vez que é esta a função que se busca otimizar. Assim

$$J(L_K, X_2, X_\infty) = \min_{X_2 > 0, X_\infty > 0, X_D > 0} \text{tr}(C_{2f} X_2 C_{2f}^T). \quad (5.2)$$

Com o objetivo de se manter a uniformidade sobre diferentes problemas, a função aptidão é escalonada da seguinte maneira:

$$\text{apt}(L_K, X_2, X_\infty) = \frac{1}{1 + J(L_K, X_2, X_\infty)}. \quad (5.3)$$

Minimizar a função objetivo $J(L_K, X_2, X_\infty)$ é equivalente a obter o valor de aptidão máximo na busca evolutiva. Na tentativa de minimizar o esforço computacional necessário para encontrar uma solução, este algoritmo não emprega funções de penalidade e, a exemplo do algoritmo de Michalewicz [93], armazena apenas cromossomos factíveis.

5.3 Seleção

Para garantir a sobrevivência dos elementos mais aptos da população, estes são selecionados para produzir a população seguinte através dos operadores genéticos. Os métodos de seleção por roleta, *rank* e torneio foram implementados.

O número de cromossomos selecionados depende do operador de cruzamento empregado, variando de 1 a 3 entre os diferentes grupos de operadores implementados.

5.4 Estratégia Elitista

Neste algoritmo é implementada uma estratégia elitista, na qual o cromossomo menos apto da geração formada i é substituído pelo cromossomo mais apto da geração anterior $i - 1$.

5.5 População Inicial

Devido à complexidade do problema e à sua geometria desconhecida, optou-se por se manter somente elementos factíveis na população. Portanto, pressupõe-se que a população inicial p_0 também seja factível [93, 91]. Para a geração de uma população inicial factível através de tentativa e erro, com cromossomos aleatórios, seria necessário um esforço computacional alto, pois o espaço de busca do problema não é convexo e não se conhecem os limites do mesmo. Além da complexidade do próprio problema, a dimensão dos cromossomos é elevada e a geração aleatória de cromossomos factíveis se torna uma estratégia inviável.

Na tentativa de se reduzir o custo computacional para a geração da população inicial factível, utilizou-se uma condição suficiente para a geração de elementos factíveis descrita em [35]. Esta condição utiliza apenas a desigualdade de Lyapunov associada à norma \mathcal{H}_2 para o problema de controle robusto com realimentação de estados. Assim, é necessário que o problema de controle robusto com realimentação estática de saída descrito pelas equações (3.93)-(3.94) e pela lei de controle $u = L_K y$, onde L_K é dado por (3.95), seja escrito na forma de um problema de controle por realimentação de estado. Como foi visto na seção 3.6, o sistema com realimentação de saída dinâmica pode ser reescrito como um sistema com realimentação de saída estática. Seja o sistema obtido a partir de (3.93), considerando-se o problema de otimização \mathcal{H}_2 :

$$\begin{cases} \dot{x}_f = \tilde{A}x_f + \tilde{B}_{12}w_2 + \tilde{B}_2u_s \\ z_2 = \tilde{C}_2x_f + \tilde{D}_{22}u_s \\ y_s = \tilde{C}_y x_f. \end{cases} \quad (5.4)$$

A lei de controle por realimentação estática de saída é então dada por:

$$u_s = L_K y_s. \quad (5.5)$$

Se o posto de \tilde{C}_y não for completo, gera-se uma matriz aleatória N tal que a matriz

$$M = \begin{bmatrix} N \\ C_y \end{bmatrix} \quad (5.6)$$

tenha posto completo. A matriz N é responsável pela diversidade da população inicial, uma vez que para cada matriz M gerada, uma solução candidata inicial é encontrada.

Com a matriz M , definida em 5.5, realiza-se a transformação de similaridade $\bar{x} = Mx_f$

e o sistema (5.4) se torna:

$$\begin{cases} \dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}_{12}w_2 + \bar{B}_2u_s \\ z_2 = \bar{C}_2\bar{x} + D_{22}u_s \\ y_s = \bar{C}_y\bar{x}, \end{cases} \quad (5.7)$$

onde $\bar{A} = M\tilde{A}M^{-1}$, $\bar{B}_{12} = M\tilde{B}_{12}$, $\bar{B}_2 = M\tilde{B}_2$, $\bar{C}_2 = \tilde{C}_2M^{-1}$ e $\bar{C}_y = \tilde{C}_yM^{-1}$. A lei de controle resultante é dada por

$$u_s = L\bar{C}_yM^{-1}\bar{x} = \bar{L}\bar{x}, \quad (5.8)$$

onde o novo controlador \bar{L} possui a seguinte estrutura:

$$\bar{L} = \begin{bmatrix} 0 & L_K \end{bmatrix}. \quad (5.9)$$

Para tornar o problema convexo e encontrar os controladores factíveis \bar{L} , as desigualdades matriciais (3.36) e (3.38) podem ser convertidas em LMI utilizando-se a transformação de variáveis $\bar{L}\bar{W}_1 = \bar{W}_2$, com matrizes de estrutura fixa dadas por

$$\bar{W}_1 = \begin{bmatrix} \bullet & 0 \\ 0 & \bullet \end{bmatrix}; \quad \bar{W}_2 = \begin{bmatrix} 0 & \bullet \end{bmatrix}. \quad (5.10)$$

Assume-se que $X_2 = X_D$, a exemplo do exposto na seção 3.4. Este problema reduzido é convexo e pode ser solucionado com a caixa de ferramentas LMI-Lab do Matlab [55].

O controlador L_K é extraído de \bar{L} e então calcula-se X_2 , X_∞ e X_D , resolvendo-se o problema convexo (3.97)-(3.100).

Se uma solução factível para o problema é encontrada, armazena-se o cromossomo $\begin{bmatrix} L_K & X_2 & X_\infty \end{bmatrix}$. A partir de diferentes matrizes M (5.6) gera-se a população inicial.

Ressalta-se que a imposição das restrições estruturais em \bar{W}_1 e \bar{W}_2 , conforme a equação (5.10), insere um conservadorismo extra na geração da população inicial. Esta é uma maneira simples de se obter uma população inicial factível, porém, estas restrições podem se tornar severas, fazendo com que o procedimento não encontre soluções factíveis para determinados problemas. Além disso, estas restrições podem contribuir para que a população inicial esteja confinada em uma região restrita do espaço de busca, comprometendo a diversidade da população inicial. A vantagem de se utilizar este método, contudo, está na garantia de que, quando as soluções são encontradas, estas são factíveis para todo o domínio de incerteza do problema.

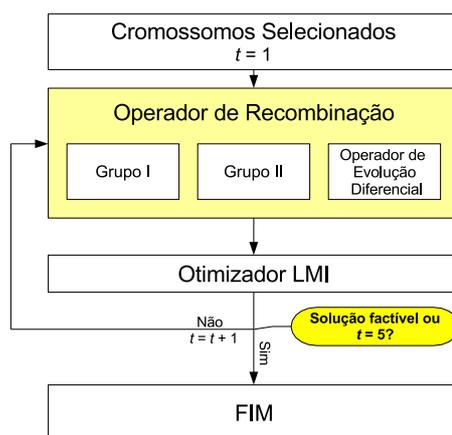


Figura 10: Operadores de recombinação.

5.6 Operadores de Recombinação

Os operadores de recombinação têm como objetivo o intercâmbio de informações entre cromossomos distintos. Para aumentar a eficácia do algoritmo, três diferentes grupos de operadores de recombinação foram implementados: grupo de operadores I, estruturados sobre a teoria proposta por Michalewicz; grupo II de operadores e o operador de evolução diferencial. A estrutura geral de recombinação é apresentada na figura 10. A variável t é um contador e indica o número da repetição. Após a seleção, os cromossomos são submetidos ao grupo de operadores de recombinação selecionado pelo usuário. Se nenhuma solução factível for encontrada na primeira tentativa, o mesmo operador de recombinação é repetido, até um máximo de 5 vezes. Se, quando $t = 5$, nenhuma solução for encontrada, o método retorna os cromossomos selecionados. O parâmetro máximo $t = 5$ foi selecionado com base em diversas simulações buscando-se um adequado compromisso entre o custo computacional do algoritmo e a adequada evolução das soluções. Cada operador de recombinação dentro do seu grupo possui uma probabilidade de ocorrência distinta, e apenas um operador de recombinação é aplicado a cada par de cromossomos selecionados por geração. Os diferentes grupos de operadores são descritos na seqüência.

5.6.1 Grupo de Operadores de Recombinação I

Os operadores deste grupo são os operadores de Michalewicz, descritos na seção 4.3. Por questão de simplicidade, optou-se por aplicar os operadores de Michalewicz somente no controlador $L_K \in \mathbb{R}^{m \times n}$. Os operadores de cruzamento são descritos a seguir.

5.6.1.1 Cruzamento Aritmético

1. Aplica-se a combinação convexa em L_K tal que:

$$\begin{aligned}\hat{L}_{K1} &= \alpha L_{K1} + (1 - \alpha)L_{K2} \\ \hat{L}_{K2} &= (1 - \alpha)L_{K1} + \alpha L_{K2},\end{aligned}\quad (5.11)$$

onde $0 \leq \alpha \leq 1$ e \hat{L}_{Kj} , $j = 1, 2$, representam os controladores gerados;

2. para cada \hat{L}_{Kj} calcula-se \hat{X}_{2j} , $\hat{X}_{\infty j}$ e X_{Dj} , solucionando-se o problema convexo (3.97)-(3.100).

5.6.1.2 Cruzamento Simples

1. Seleciona-se aleatoriamente um ponto de corte τ tal que $1 < \tau < mn$;
2. a partir dos cromossomos selecionados

$$L_{K1} = \begin{bmatrix} L_{K1}^1 \\ L_{K1}^2 \\ \vdots \\ L_{K1}^m \end{bmatrix}, \quad L_{K2} = \begin{bmatrix} L_{K2}^1 \\ L_{K2}^2 \\ \vdots \\ L_{K2}^m \end{bmatrix}, \quad (5.12)$$

criam-se as representações vetoriais, dadas por

$$\begin{aligned}L_1 &= [L_{K1}^1 \mid L_{K1}^2 \mid \dots \mid L_{K1}^m] \\ L_2 &= [L_{K2}^1 \mid L_{K2}^2 \mid \dots \mid L_{K2}^m];\end{aligned}\quad (5.13)$$

3. com os vetores $L_1 = [l_1^{(1)} \dots l_1^{(mn)}]$ e $L_2 = [l_2^{(1)} \dots l_2^{(mn)}]$, criam-se os vetores

$$\begin{aligned}\hat{L}_1 &= \left[\begin{array}{cccc} l_1^{(1)} & \dots & l_1^{(\tau)} & l_2^{(\tau+1)} & \dots & l_2^{(mn)} \end{array} \right], \\ \hat{L}_2 &= \left[\begin{array}{cccc} l_2^{(1)} & \dots & l_2^{(\tau)} & l_1^{(\tau+1)} & \dots & l_1^{(mn)} \end{array} \right];\end{aligned}\quad (5.14)$$

4. geram-se as representações vetoriais de \hat{L}_1 e \hat{L}_2 , formando os cromossomos filhos \hat{L}_{K1} e \hat{L}_{K2} ;
5. para cada \hat{L}_{Kj} , $j = 1, 2$, calcula-se \hat{X}_{2j} , $\hat{X}_{\infty j}$ e X_{Dj} , solucionando-se o problema convexo (3.98)-(3.100).

5.6.1.3 Cruzamento Heurístico

1. Selecciona-se aleatoriamente $r \in (0, 1)$;
2. cria-se o controlador

$$\hat{L}_K = \begin{cases} L_{K1} + r \times (L_{K2} - L_{K3}), & \text{se } \mathbf{apt}(L_{K2}) > \mathbf{apt}(L_{K3}) \\ L_{K1} + r \times (L_{K3} - L_{K2}), & \text{caso contrário;} \end{cases} \quad (5.15)$$

3. para \hat{L}_K , calcula-se \hat{X}_2 , \hat{X}_∞ e X_D solucionando-se o problema convexo (3.98)-(3.100).

5.6.2 Grupo de Operadores de Recombinação II

Com o intuito de se buscar um melhor desempenho do algoritmo, implementa-se quatro diferentes mecanismos de recombinação. Este segundo grupo de operadores se caracteriza pela troca de informações não somente entre os controladores, mas também entre as matrizes de Lyapunov. Os procedimentos deste grupo são aplicados em diferentes variáveis e são baseados no cruzamento aritmético de Michalewicz e nas propriedades de convexidade do problema (seção 4.3.4).

5.6.2.1 Operador de Recombinação 1

1. Aplica-se a combinação convexa em L_K tal que:

$$\begin{aligned} \hat{L}_{K1} &= \alpha L_{K1} + (1 - \alpha)L_{K2}, \\ \hat{L}_{K2} &= (1 - \alpha)L_{K1} + \alpha L_{K2}, \end{aligned} \quad (5.16)$$

onde $0 \leq \alpha \leq 1$ e \hat{L}_{Kj} , $j = 1, 2$, representam os controladores gerados;

2. para cada \hat{L}_{Kj} , $j = 1, 2$, calcula-se \hat{X}_{2j} , $\hat{X}_{\infty j}$ e X_{Dj} , solucionando-se o problema convexo (3.98)-(3.100).

5.6.2.2 Operador de Recombinação 2

1. Aplica-se a combinação convexa em X_2 tal que:

$$\begin{aligned} \hat{X}_{21} &= \alpha X_{21} + (1 - \alpha)X_{22}, \\ \hat{X}_{22} &= (1 - \alpha)X_{21} + \alpha X_{22}, \end{aligned} \quad (5.17)$$

onde $0 \leq \alpha \leq 1$ e \hat{X}_{2j} , $j = 1, 2$, representam as matrizes de Lyapunov geradas;

2. para cada par $\left[\begin{array}{cc} \hat{X}_{2j} & X_{\infty j} \end{array} \right]$, calcula-se \hat{L}_{Kj} e X_{Dj} , solucionando-se o problema convexo (3.98)-(3.100).

5.6.2.3 Operador de Recombinação 3

1. Aplica-se a combinação convexa em X_{∞} tal que:

$$\begin{aligned} \hat{X}_{\infty 1} &= \alpha X_{\infty 1} + (1 - \alpha) X_{\infty 2}, \\ \hat{X}_{\infty 2} &= (1 - \alpha) X_{\infty 1} + \alpha X_{\infty 2}, \end{aligned} \quad (5.18)$$

onde $0 \leq \alpha \leq 1$ e $\hat{X}_{\infty j}$, $j = 1, 2$, representam as matrizes de Lyapunov geradas;

2. para cada par $\left[\begin{array}{cc} X_{2j} & \hat{X}_{\infty j} \end{array} \right]$, calcula-se \hat{L}_K e X_{Dj} , solucionando-se o problema convexo (3.98)-(3.100).

5.6.2.4 Operador de Recombinação 4

1. Aplica-se a combinação convexa em X_2 e X_{∞} tal que:

$$\begin{aligned} \hat{X}_{21} &= \alpha X_{21} + (1 - \alpha) X_{22}, \\ \hat{X}_{22} &= (1 - \alpha) X_{21} + \alpha X_{22}, \\ \hat{X}_{\infty 1} &= \alpha X_{\infty 1} + (1 - \alpha) X_{\infty 2}, \\ \hat{X}_{\infty 2} &= (1 - \alpha) X_{\infty 1} + \alpha X_{\infty 2}, \end{aligned} \quad (5.19)$$

onde $0 \leq \alpha \leq 1$ e \hat{X}_{ij} , $i = 2, \infty$, $j = 1, 2$, representam as matrizes de Lyapunov geradas;

2. para cada par $\left[\begin{array}{cc} \hat{X}_{ij} & \hat{X}_{ij} \end{array} \right]$ calcula-se \hat{L}_K e X_{Dj} , solucionando-se o problema convexo (3.98)-(3.100).

5.6.3 Operador de Evolução Diferencial

O operador de evolução diferencial foi inspirado no algoritmo proposto por Storn e Price, descrito na seção 4.4. O operador desenvolvido é descrito a seguir.

Para cada controlador L_K da população P :

1. calcula-se $F = \frac{\alpha}{10^{2t}}$, onde $\alpha \in [0, 10)$ é um valor aleatório e $t \in (1, 5)$ é o contador do número da tentativa de cruzamento indicado na figura 10;
2. seleciona-se $L_{K1} \in P$ pelo método de seleção heurístico adotado no início do algoritmo;
3. seleciona-se aleatoriamente L_{K2} e L_{K3} tal que $L_{K1} \neq L_{K2} \neq L_{K3}$;
4. calcula-se

$$L_{K4} = \begin{cases} L_{K1} + F \times (L_{K2} - L_{K3}), & \text{se } \text{apt}(L_{K2}) > \text{apt}(L_{K3}), \\ L_{K1} + F \times (L_{K3} - L_{K2}), & \text{caso contrário;} \end{cases} \quad (5.20)$$

5. a partir da notação

$$L_{\bullet} = \begin{bmatrix} l_{\bullet}^{1,1} & l_{\bullet}^{1,2} & \dots & l_{\bullet}^{1,n} \\ l_{\bullet}^{2,1} & l_{\bullet}^{2,2} & \dots & l_{\bullet}^{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{\bullet}^{m,1} & l_{\bullet}^{m,2} & \dots & l_{\bullet}^{m,n} \end{bmatrix}, \quad (5.21)$$

onde o índice \bullet é substituído pelo índice do respectivo controlador, determina-se

$$l_{K5}^{i,j} = \begin{cases} l_{K4}^{i,j}, & \text{se } R > p_c \text{ ou se } (i * j) = \Phi, \\ l_{K1}^{i,j}, & \text{caso contrário,} \end{cases} \quad (5.22)$$

onde $R \in (0, 1)$ e $\Phi \in (1, mn)$ são valores aleatórios e p_c indica a probabilidade de cruzamento;

6. para L_{K5} , soluciona-se o problema convexo (3.98)-(3.100). Se L_{K5} não for factível incrementa-se t e retorna-se ao passo 1 até o máximo de 5 vezes, senão segue-se para o passo 7. Caso nenhum L_{K5} seja encontrado retorna-se ao passo 1 para o próximo controlador da população P ;
7. se $\text{apt}(L_{K5}) > \text{apt}(L_K)$ substitui-se L_K por L_{K5} , senão, incrementa-se t até o máximo 5 e retorna-se ao passo 1;

As principais modificações efetuadas em relação ao algoritmo proposto na seção 4.4 são:

- a variável F é calculada de forma aleatória, porém é decrementada exponencialmente no decorrer das tentativas de cruzamento. A razão para tal fato reside na tentativa de se reduzir gradativamente a influência da parcela $(L_{K2} - L_{K3})$ ou $(L_{K3} - L_{K2})$ no elemento L_{K4} . Essa estratégia de localização permite um aumento gradativo da

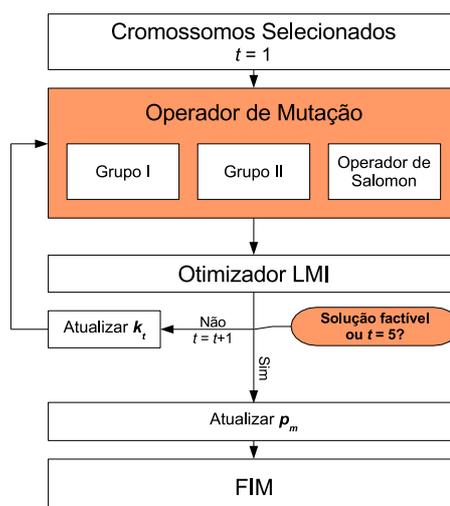


Figura 11: Operadores de mutação.

probabilidade do controlador L_{K5} permanecer no espaço de factibilidade do problema;

- a seleção do elemento L_{K1} é efetuada pelo método estocástico de seleção escolhido no início do algoritmo e não de forma aleatória direta como no algoritmo de Storn e Price. Observa-se que a convergência é mais rápida quando um método estocástico é utilizado. A ideia foi inspirada no método de cruzamento heurístico proposto por Michalewicz;
- a construção do controlador L_{K5} é realizada a partir dos elementos L_{K4} e L_{K1} e não a partir dos elementos L_{K4} e L_K , como propõe o algoritmo de Storn e Price. Observa-se que ao se utilizar o controlador L_K na formação de L_{K5} , uma grande parcela dos elementos L_{K5} formados são infactíveis. O uso de L_K , como proposto no algoritmo original, introduz uma variação muito significativa no vetor L_{K5} , excluindo-o do espaço de factibilidade do problema. A utilização de L_{K1} produz um percentual significativamente maior de controladores L_{K5} factíveis, melhorando a evolução do algoritmo.

5.7 Operadores de Mutação

De forma similar aos operadores de cruzamento, três diferentes grupos de operadores de mutação são implementados. A estrutura geral da mutação é apresentada na figura 11.

No intuito de reduzir a probabilidade de convergência prematura do algoritmo, uma estratégia que aumenta a probabilidade de mutação no decorrer das gerações é implementada. Dada a probabilidade de mutação inicial, esta é linearmente incrementada até atingir 20% a mais do seu valor inicial na última geração. Em outras palavras,

$$pm_i = pm + 0,2 \frac{n_i}{n_{gen}}, \quad (5.23)$$

onde pm_i representa a probabilidade de mutação na geração i , pm representa a probabilidade de mutação inicial, n_i é o número da geração atual e n_{gen} é o número total de gerações. O incremento em 20% foi escolhido com base em diversas simulações do algoritmo, buscando-se o melhor compromisso entre a taxa de convergência e a capacidade do algoritmo de não fazer a população convergir prematuramente.

Para contornar a deficiência dos algoritmos genéticos de se aproximar da solução, mas nunca encontrá-la, desenvolve-se uma estratégia de localização responsável por introduzir um maior refinamento no algoritmo aumentando a probabilidade de produzir um elemento factível. Essa estratégia é inspirada no algoritmo de mutação não-uniforme de Michalewicz. Para tanto, cria-se um parâmetro aleatório e constante k_t tal que

$$\pm k_t = \frac{k}{p^t}, \quad (5.24)$$

onde k_t representa um incremento/decremento aleatório da t -ésima tentativa de mutação na geração i . O valor $k \in [0, 10]$ é selecionado aleatoriamente e $p = 10$ é um parâmetro de projeto empírico. A variável t representa o índice da t -ésima tentativa de mutação e possui valor máximo 5. Este parâmetro k_t é utilizado de diferentes maneiras nos algoritmos de mutação para produzir o efeito desejado. Os valores dos parâmetros k , p e t foram dimensionados através de diversas simulações do algoritmo na tentativa de se manter uma velocidade de convergência adequada em conjunto com o maior refinamento possível das soluções. Isso representa uma tentativa de ponderação das propriedades de busca global e local do algoritmo.

Cada operador de mutação dentro do seu grupo possui uma probabilidade de ocorrência distinta e somente um operador é aplicado para cada cromossomo de cada geração.

5.7.1 Grupo de Operadores de Mutação I

Este grupo de operadores é baseado nos operadores de mutação propostos por Michalewicz (seção 4.3). Algumas modificações são necessárias, devido à natureza do pro-

blema. A mutação de contorno não pode ser aplicada diretamente devido ao desconhecimento dos limites do espaço de busca. Assim, estes limites são substituídos pelos limites do conjunto $\Omega(L_K)$, formado por todos os elementos da matriz L_K , e definido por $\Omega(L_K) = \{ l_K^{(i,j)}, \dots, l_K^{(m,n)} \}$, $1 < i < m$ e $1 < j < n$. A determinação dos conjuntos *right()* e *left()*, definidos na seção 4.3.3, não é possível. Portanto, a importante propriedade gerada pelo conhecimento destes conjuntos é ignorada e utiliza-se simplesmente o conjunto $\Omega(L_K)$.

5.7.1.1 Mutação Uniforme

1. Seleciona-se aleatoriamente um elemento $l_K^{i,j}$ da matriz L_K . Substitui-se $l_K^{i,j}$ por $\hat{l} \in [\text{mínimo}(\Omega(L_K)), \text{máximo}(\Omega(L_K))]$, aleatório e uniformemente distribuído, formando \hat{L}_K ;
2. para o controlador \hat{L}_K , calcula-se $\left[\hat{X}_2 \quad \hat{X}_\infty \quad \hat{X}_D \right]$, solucionando-se o problema convexo (3.98)-(3.100)

5.7.1.2 Mutação de Contorno

1. Seleciona-se aleatoriamente um elemento $l_K^{i,j}$ da matriz L_K . Substitui-se $l_K^{i,j}$ por um valor aleatório \hat{l} formando \hat{L}_K , tal que $\hat{l} = \text{mínimo}(\Omega(L_K))$ ou $\hat{l} = \text{máximo}(\Omega(L_K))$;
2. para o controlador \hat{L}_K , calcula-se $\left[\hat{X}_2 \quad \hat{X}_\infty \quad \hat{X}_D \right]$, solucionando-se o problema convexo (3.98)-(3.100).

5.7.1.3 Mutação Não-Uniforme

1. Seleciona-se aleatoriamente um componente $l_K^{i,j}$ da matriz L_K e o substitui por \hat{l} produzindo-se a matriz \hat{L}_K onde \hat{l} é dado por:

$$\hat{l} = \begin{cases} l_K^{i,j} + \Delta (\text{máximo}(\Omega(L_K)) - l_K^{i,j}), & \text{se um dado número aleatório é igual a 0,} \\ l_K^{i,j} - \Delta (l_K^{i,j} - \text{mínimo}(\Omega(L_K))), & \text{se um dado número aleatório é igual a 1,} \end{cases} \quad (5.25)$$

onde

$$\Delta(y) = yr \left(1 - \frac{n}{n_{gen}} \right)^2, \quad (5.26)$$

n indica o número da geração atual e n_{gen} indica o número máximo de gerações;

2. para o controlador \hat{L}_K calcula-se $\left[\hat{X}_2 \quad \hat{X}_\infty \quad \hat{X}_D \right]$, solucionando-se o problema convexo (3.98)-(3.100).

5.7.2 Grupo de Operadores de Mutação II

Os algoritmos de mutação deste grupo são baseados na mutação uniforme de Michalewicz (seção 4.3.1), nos procedimentos de mutação apresentados em [120, 46, 47] e nos algoritmos de recombinação.

O objetivo dos operadores 1 e 2 é alterar o cromossomo de forma que suas características não sejam drasticamente alteradas, assim garantindo maior probabilidade de se gerar elementos factíveis. O algoritmo 3 é baseado na estratégia de mutação clássica (seção 4.2 e [120, 46, 47]) e possui uma maior probabilidade de escapar do espaço de factibilidade do problema. Possui, porém, a característica de reduzir a probabilidade do algoritmo convergir prematuramente. O objetivo do algoritmo de mutação 4 é reduzir a norma Euclideana dos elementos do cromossomo. Os operadores são descritos a seguir.

5.7.2.1 Operador de Mutação 1

1. Calcula-se as matrizes P e Λ_2 tal que $X_2 = P\Lambda_2P^{-1}$ e Λ_2 representa a matriz diagonal dos autovalores de X_2 ;
2. seleciona-se aleatoriamente um autovalor de X_2 e soma-se k_t a ele gerando-se $\hat{\Lambda}_2$;
3. calcula-se a matriz $\hat{X}_2 = P\hat{\Lambda}_2P^{-1}$;
4. para o conjunto de matrizes $\begin{bmatrix} \hat{X}_2 & X_\infty & X_D \end{bmatrix}$, calcula-se \hat{L}_K , solucionando-se o problema convexo (3.98)-(3.100).

5.7.2.2 Operador de Mutação 2

1. calcula-se as matrizes P e Λ_∞ tal que $X_\infty = P\Lambda_\infty P^{-1}$ e Λ_∞ representa a matriz diagonal dos autovalores de X_∞ ;
2. seleciona-se aleatoriamente um autovalor de X_∞ e soma-se k_t a ele gerando-se $\hat{\Lambda}_\infty$;
3. calcula-se a matriz $\hat{X}_\infty = P\hat{\Lambda}_\infty P^{-1}$;
4. para o conjunto de matrizes $\begin{bmatrix} X_2 & \hat{X}_\infty & X_D \end{bmatrix}$, calcula-se \hat{L}_K , solucionando-se o problema convexo (3.98)-(3.100).

5.7.2.3 Operador de Mutação 3

1. Seleciona-se aleatoriamente um elemento da matriz L_K e soma-se a ele k_t gerando-se \hat{L}_K ;
2. para o controlador \hat{L}_K , calcula-se $\begin{bmatrix} \hat{X}_2 & \hat{X}_\infty & \hat{X}_D \end{bmatrix}$, solucionando-se o problema convexo (3.98)-(3.100)

5.7.2.4 Operador de Mutação 4

1. Gera-se um valor α aleatório tal que $0 \leq \alpha \leq 1$;
2. calcula-se $\begin{bmatrix} \hat{X}_2 \\ \hat{X}_\infty \end{bmatrix} = \alpha \begin{bmatrix} X_2 \\ X_\infty \end{bmatrix}$;
3. para o conjunto de matrizes $\begin{bmatrix} \hat{X}_2 & \hat{X}_\infty & X_D \end{bmatrix}$, calcula-se \hat{L}_K , solucionando-se o problema convexo (3.98)-(3.100).

5.7.3 Operador de Salomon

O operador de Salomon inspira-se no algoritmo de Salomon (seção 4.5) e na possibilidade de, a exemplo da aproximação discreta da derivada, aproximar o gradiente da função objetivo do problema. O algoritmo é descrito a seguir.

Para cada elemento L_K da população P :

1. gera-se λ candidatos L_i , tais que

$$L_i = L_K + Z_i, \quad i = 1, \dots, \lambda, \quad (5.27)$$

onde Z_i é uma matriz de elementos aleatórios distribuídos normalmente com média 0 e desvio padrão σ_k/\sqrt{n} e n é número de componentes do vetor;

2. calcula-se o vetor gradiente estimado

$$G = \sum_{i=1}^{\lambda} (\text{apt}(L_i) - \text{apt}(L_K)) (L_i - L_K); \quad (5.28)$$

3. calcula-se a matriz unitária de direção

$$E = \frac{G}{\|G\|_2}; \quad (5.29)$$

4. adapta-se o tamanho de passo

$$\sigma_{k+1} = \begin{cases} \zeta, & \text{se } \text{apt}(L_K - \sigma_k \zeta E) \leq \text{apt}(L_K - (\sigma_k/\zeta)E), \\ \sigma_k/\zeta, & \text{caso contrário;} \end{cases} \quad (5.30)$$

5. calcula-se o cromossomo resultante \hat{L}_K , tal que

$$\hat{L}_K = L_K - \sigma_{k+1}E \quad (5.31)$$

6. para o controlador \hat{L}_K , calcula-se $\left[\hat{X}_2 \quad \hat{X}_\infty \quad \hat{X}_D \right]$, solucionando-se o problema convexo (3.98)-(3.100).

O algoritmo de Salomon como operador de mutação tem a vantagem de produzir uma busca “direcionada” reduzindo o caráter estritamente aleatório dos outros operadores de mutação. O efeito desta busca é a geração de um maior percentual de elementos factíveis pelo algoritmo aumentando a velocidade de convergência.

5.8 Conclusão

Este capítulo apresenta o algoritmo proposto como solução para o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$. O algoritmo é baseado na teoria de computação evolutiva e utiliza como principais fontes o algoritmo genético de Michalewicz, Evolução Diferencial e a abordagem de Salomon. O algoritmo de Michalewicz foi escolhido para ser implementado por permitir a abordagem direta de problemas com restrições convexas. O grupo II de operadores de recombinação e de mutação foi desenvolvido no intuito de se verificar a eficiência de se efetuar operações genéticas sobre as diferentes matrizes de Lyapunov do problema. O algoritmo de ED é conhecido por seu comportamento de busca robusto. Por fim, o algoritmo de Salomon foi implementado com o intuito de se direcionar a busca genética, proporcionando-se melhores resultados. Além disso, a abordagem via LMIs permite grande flexibilidade ao algoritmo que pode ser configurado para diferentes regiões de alocação de pólos e estruturas de controle.

6 Resultados

O objetivo deste capítulo é validar a solução para o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$ proposta no capítulo 5. O escopo deste capítulo é restrito ao estudo da capacidade de cada algoritmo de encontrar soluções satisfatórias do ponto de vista de norma \mathcal{H}_2 com restrições de norma \mathcal{H}_∞ . Para isto, diversos exemplos foram extraídos e adaptados da literatura, aos quais diferentes configurações do algoritmo proposto foram aplicadas. Estas configurações são caracterizadas por três grupos distintos de operadores:

- *grupo I*: forma-se com os operadores de recombinação e de mutação derivados do algoritmo de Michalewicz, apresentados nas seções 5.6.1 e 5.7.1;
- *grupo II*: utiliza-se o grupo II de operadores de recombinação em conjunto com o grupo II de operadores de mutação, apresentados nas seções 5.6.2 e 5.7.2, respectivamente;
- *grupo III*: compõe-se do operador de evolução diferencial (seção 5.6.3) como estratégia de recombinação, juntamente com o operador de Salomon (seção 5.7.3) como estratégia de mutação.

Através de diversas simulações prévias, buscou-se encontrar um conjunto único de parâmetros do AE adequado aos diversos exemplos estudados. A busca do valor ótimo de cada parâmetro do AE, para cada exemplo, fuge do escopo deste trabalho.

Utilizaram-se 100 gerações por simulação com populações de 20 elementos, exceto quando explicitado o contrário. O estudo do desvio padrão da população resultante de cada simulação sugere que 100 gerações são, em geral, suficientes para a convergência dos diferentes grupos de operadores para uma única solução. Além disso, na busca do comportamento médio de cada grupo de operadores, efetuaram-se 10 simulações, por exemplo, por algoritmo. Cabe ressaltar que 10 simulações são insuficientes para se garantir estatisticamente qualquer conclusão realizada. Entretanto, o elevado custo computacional médio dos algoritmos, 2, 6 horas por execução, impossibilitou tanto a utilização de maiores

populações e mais gerações por simulação, quanto um número maior de simulações por algoritmo.

Nas simulações prévias, percebeu-se que pequenas variações nos parâmetros de cada operador não influenciam o resultado final encontrado pelos diferentes algoritmos. Assim, em todas as simulações utilizou-se o conjunto de parâmetros apresentado nas tabelas 2, 3 e 4 para os diferentes grupos de operadores. Este conjunto demonstrou resultados satisfatórios em todos os exemplos estudados. As elevadas taxas de mutação, resultantes da análise prévia dos algoritmos, são frutos da utilização de populações de tamanho bastante reduzido, fato coerente com os indícios apresentados em [120].

As diversas simulações de teste dos algoritmos sugeriram forte influência do método de seleção escolhido na convergência média dos diferentes algoritmos. O estudo detalhado da eficácia de cada método de seleção, porém, não é foco principal deste trabalho. Contudo, observou-se que a convergência média, atingida com os métodos de seleção por roleta e por torneio, está intimamente relacionada com o correto escalonamento da função aptidão e com o parâmetro do torneio, respectivamente. O método de seleção por *rank*, entretanto, demonstrou taxa de convergência satisfatória e independente do exemplo estudado. Por esta característica, utilizou-se o método de seleção por *rank* em todos os exemplos deste capítulo.

A comparação estatística da eficiência computacional de cada algoritmo não é foco deste trabalho. A validação de conclusões sobre a eficiência de cada algoritmo implicaria a necessidade de um número maior de simulações por algoritmo, fato inviabilizado pelo elevado custo computacional associado. Assim, os tempos de execução apresentados são apenas ilustrativos.

Tabela 2: Probabilidades do grupo I de operadores.

Cruzamento		Mutação	
p_c	0,8	p_m	0,3
Aritmético	0,33	Uniforme	0,33
Simplex	0,33	De Contorno	0,33
Heurístico	0,33	Não-Uniforme	0,33

Tabela 3: Probabilidades do grupo II de operadores.

Recombinação		Mutação	
p_c	0,8	p_m	0,3
Op. 1	0,4	Op. 1	0,4
Op. 2	0,2	Op. 2	0,2
Op. 3	0,2	Op. 3	0,2
Op. 4	0,2	Op. 4	0,2

Tabela 4: Probabilidades do grupo III de operadores.

E.D.		Op. Salomon	
p_c	1	p_m	0,35
F	$\frac{\alpha}{10^{2t}}$	σ_0	1
		λ	80
		ζ	2

6.1 Exemplo 1

Este exemplo foi extraído de [59] e representa um modelo linearizado de um helicóptero VTOL. Três parâmetros incertos são considerados e o seu conjunto forma um poliedro convexo de 8 vértices, representando os domínios de incerteza \mathcal{D}_A e \mathcal{D}_B do problema. A condição de sub-otimalidade utilizada foi $\gamma \leq 30$. As matrizes do problema são dadas por:

$$A = \begin{bmatrix} -0,0366 & 0,0271 & 0,0188 & -0,4555 \\ 0,0482 & -1,0100 & 0,0024 & -4,0208 \\ 0,1002 & a_1 & -0,7070 & a_2 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

onde os parâmetros incertos a_1 e a_2 pertencem aos intervalos:

$$a_1 \in [-0,6319; 1,3681], \quad a_2 \in [1,22; 1,62],$$

$$B_2 = \begin{bmatrix} -0,4422 & -0,1761 \\ b_1 & 7,5922 \\ 5,5200 & -4,4900 \\ 0 & 0 \end{bmatrix},$$

onde o parâmetro incerto $b_1 \in [2, 7446; 4, 3446]$ e

$$\begin{aligned}
 C_2 &= \begin{bmatrix} \mathbf{I}_4 \\ 0_{1 \times 4} \end{bmatrix}, & C_y &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, & D_{22} &= \begin{bmatrix} 0 \\ \mathbf{I}_2 \end{bmatrix}, \\
 B_{12} &= \mathbf{I}_4, & B_{1\infty} &= \mathbf{I}_4, & C_1 &= \mathbf{I}_4, \\
 D_{112} &= 0, & D_{11\infty} &= 0, & D_{12} &= 0, \\
 D_{21\infty} &= 0, & D_{y1\infty} &= 0.
 \end{aligned}$$

6.1.1 Caso 1: Realimentação Estática de Saída

Nesta subseção simulam-se os diferentes grupos de operadores na síntese de controladores por realimentação estática de saída. A figura 12 é utilizada como padrão para a apresentação da evolução de cada grupo de operadores, nos diferentes exemplos deste capítulo. Esta figura está dividida em quatro gráficos que foram gerados a partir de 10 simulações de cada grupo distinto de operadores. Com isso, o gráfico da média do melhor elemento mostra o valor médio da norma \mathcal{H}_2 do melhor cromossomo da população a cada geração, para o conjunto de 10 simulações. Este gráfico indica se o algoritmo é capaz de evoluir a população na busca por melhores resultados. A mesma lógica é utilizada para o gráfico da média do pior elemento, no qual utilizou-se, ao invés da norma \mathcal{H}_2 do melhor cromossomo, o valor da norma \mathcal{H}_2 do pior elemento da população. O gráfico da média da população, por sua vez, considera a média da norma \mathcal{H}_2 média de todos os elementos da população, a cada geração, no conjunto de 10 simulações. Por fim, para a geração do gráfico do desvio padrão médio, utilizou-se o cálculo do desvio padrão da norma H_2 de todos os elementos da população a cada geração e calculou-se a média deste desvio para as 10 simulações. Estes três últimos gráficos permitem identificar se o conjunto de operadores genéticos utilizado é capaz de convergir a população para uma única região do espaço de factibilidade do problema. Além disso, pode-se comparar a velocidade média de convergência dos diferentes grupos de operadores.

Neste exemplo, o gráfico da média do melhor elemento, apresentados na figura 12, sugere que o grupo I de operadores foi capaz de encontrar os controladores com mais baixo índice de desempenho norma \mathcal{H}_2 . Por outro lado, o conjunto dos gráficos da média do pior elemento, da média da população e do desvio padrão médio, indicam que o grupo I de operadores foi menos eficaz na convergência da população para uma única solução no espaço de soluções factíveis.

Outra figura que é utilizada com padrão de apresentação dos resultados neste capítulo

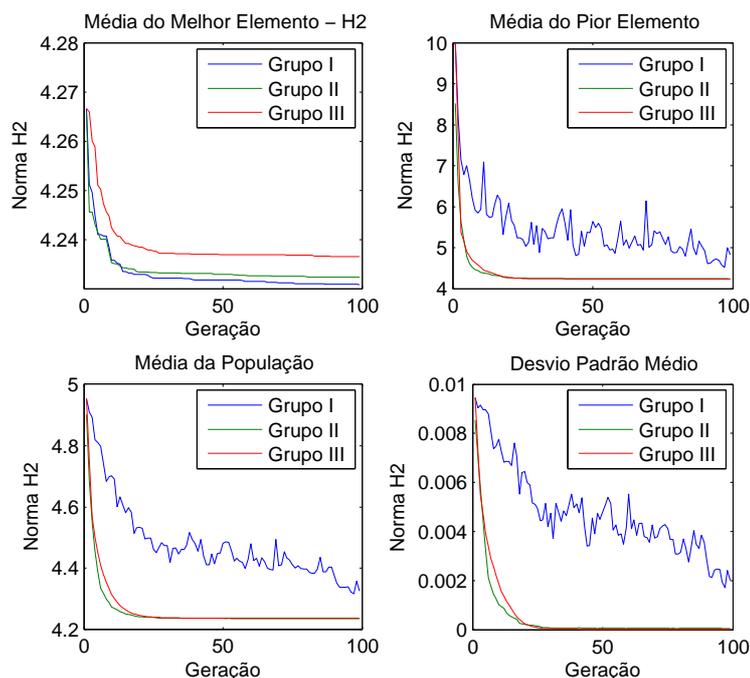


Figura 12: Evolução dos algoritmos no exemplo 1 - caso 1.

é a figura 13. Nela são mostradas as normas \mathcal{H}_2 e \mathcal{H}_∞ , calculadas para o sistema em malha fechada, com melhor controlador encontrado nas 10 simulações com cada grupo de operadores. Lembra-se que o objetivo dos problemas apresentados neste capítulo é a minimização da norma \mathcal{H}_2 . Assim, no lado direito da figura 13, estão apresentados os valores da norma \mathcal{H}_∞ para o sistema em malha fechada com o controlador que apresentou a norma \mathcal{H}_2 mínima.

Apesar das diferentes evoluções médias observadas na figura 12, os melhores elementos, encontrados pelos diferentes algoritmos, mostrados na figura 13, possuem critérios de desempenho praticamente iguais. A melhor solução encontrada para este exemplo foi obtida com o grupo II de operadores e é dada por:

$$K = \begin{bmatrix} 2,2818 & -17,4041 \end{bmatrix}^T. \quad (6.1)$$

Além da análise do valor da norma \mathcal{H}_2 , buscou-se identificar a eficiência de cada grupo de operadores na geração de cromossomos factíveis. Para isso, contou-se o número de cromossomos infactíveis gerados pelos diferentes grupos de operadores a cada geração. Estes resultados são apresentados na figura 14. O lado esquerdo desta figura mostra o número médio de cromossomos infactíveis resultante a cada geração, para o total de

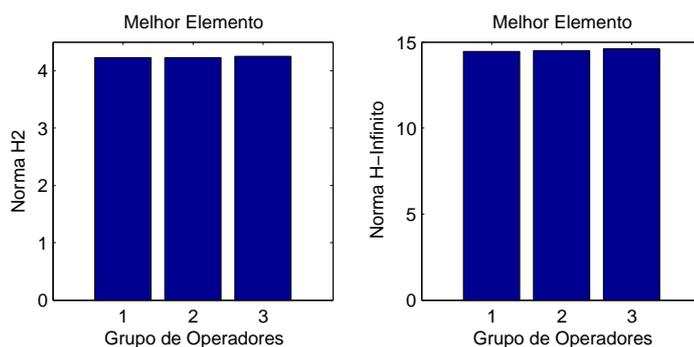


Figura 13: Comparação dos melhores resultados do exemplo 1 - caso 1.

10 simulações. O número médio total de elementos inactivos, lado direito da figura 14, corresponde à média do número médio de elementos inactivos resultantes de cada geração, para o total de 100 gerações.

Neste exemplo, a figura 14 mostra que o grupo III de operadores foi o grupo mais eficaz na geração de cromossomos activos. Este fato é um resultado provável da utilização de informações de direção pelos operadores de Evolução Diferencial e de Salomon. Por outro lado, o grupo II de operadores, que não utiliza informações de direção, e é aplicado sobre as diferentes matrizes de Lyapunov, gerou um número expressivo de elementos inactivos.

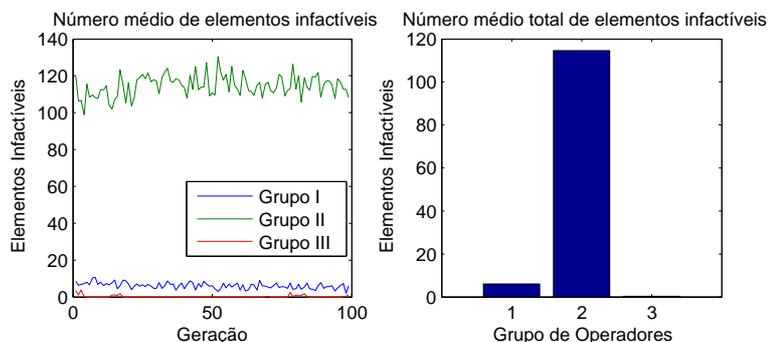


Figura 14: Número médio de elementos inactivos gerados no exemplo 1 - caso 1.

As nuvens de pólos em malha fechada, geradas pelo melhor controlador encontrado com cada grupo de operadores, estão representadas na figura 15.

A tabela 5 mostra as normas \mathcal{H}_2 e \mathcal{H}_∞ calculadas para o sistema em malha fechada com o melhor controlador encontrado nas 10 simulações de cada grupo de operadores.

Observa-se que as três abordagens estudadas apresentaram, para este exemplo, resultados superiores aos encontrados na literatura, descritos na tabela 6, validando a aborda-

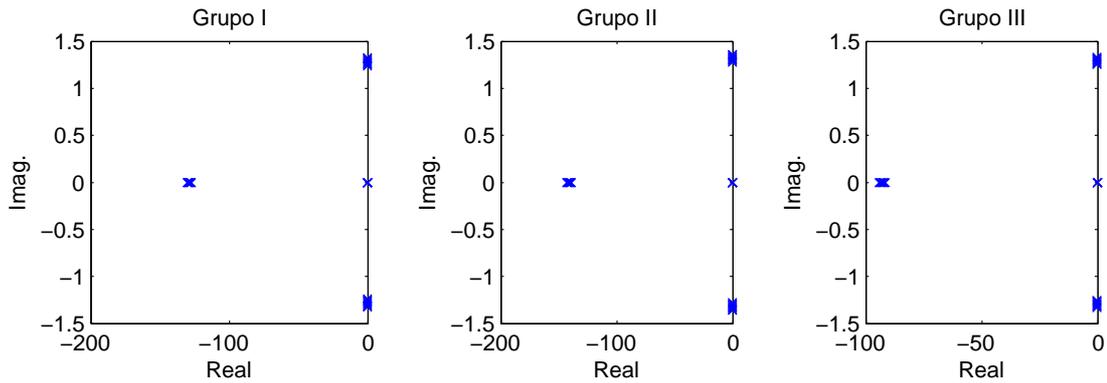


Figura 15: Pólos em malha fechada do exemplo 1 - caso 1.

Tabela 5: Comparação dos resultados encontrados no exemplo 1 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	4, 2300	4, 2303	4, 2471
$\ T_{z_\infty w_\infty}\ _\infty$	14, 4472	14, 4902	14, 6133

gem proposta. O tempo médio de execução deste exemplo foi de 48 minutos por simulação em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

Tabela 6: Resultados encontrados na literatura para o exemplo 1 - caso 1.

	Resultados de [101]	Resultados de [59]
$\ T_{z_2 w_2}\ _2$	4, 3383	4, 5001
$\ T_{z_\infty w_\infty}\ _\infty$	-	-

6.1.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subseção, projetaram-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 2$), utilizando-se os três diferentes grupos de operadores genéticos. A evolução dos algoritmos está representada na figura 16.

Para este caso, observa-se que o grupo I de operadores não foi eficaz na evolução dos cromossomos da população inicial. Contudo, a convergência média dos grupos II e III se demonstrou satisfatória, como verificado no gráfico da média do melhor elemento, na figura 16. A tendência crescente do desvio padrão médio do grupo I de operadores sugere que este não foi eficaz em fazer convergir a população para um único ponto solução, mantendo-se

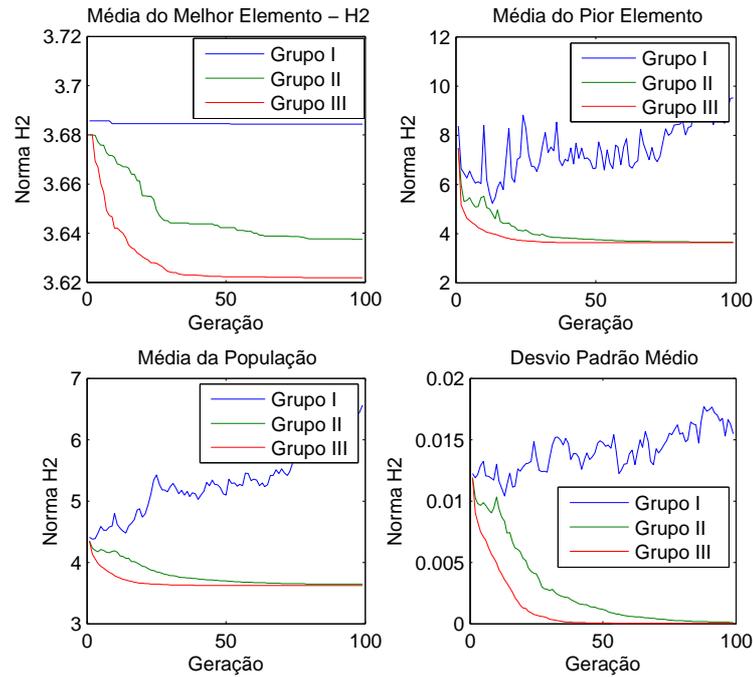


Figura 16: Evolução dos algoritmos no exemplo 1 - caso 2.

elevada diversidade na população final gerada. A melhor taxa de convergência do grupo III de operadores indica que a utilização das informações de direção de busca contribuiu na evolução do AE, conforme discutido em [19].

Ao se observar a figura 17, nota-se que, apesar do grupo I de operadores não apresentar convergência média satisfatória, o melhor controlador encontrado por este grupo nas 10 simulações possui critérios de desempenho bastante próximos dos resultados dos outros dois grupos de operadores. O melhor controlador dinâmico sintetizado, com a utilização do grupo III de operadores, é dado por:

$$A_K = \begin{bmatrix} -1,4693 & 0,2951 \\ -2,6762 & 4,2367 \end{bmatrix}, \quad B_K = \begin{bmatrix} -0,1297 \\ 4,8651 \end{bmatrix},$$

$$C_K = \begin{bmatrix} -1,9838 & 2,9132 \\ 4,0264 & -8,0881 \end{bmatrix}, \quad D_K = \begin{bmatrix} 2,9902 \\ -5,3166 \end{bmatrix}.$$

Ao se observar a figura 18, percebe-se que, novamente, o grupo II de operadores gerou um elevado número de elementos inactíveis indicando menor eficácia destes operadores na permanência da busca no espaço de factibilidade do problema. Avaliando-se os operadores genéticos que formam o grupo II, conclui-se que esta elevada taxa de elementos

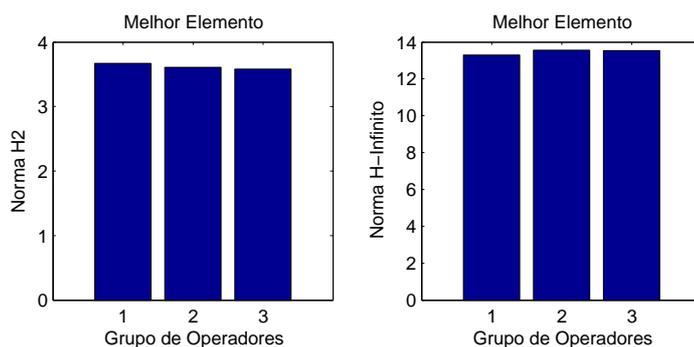


Figura 17: Comparação dos melhores resultados do exemplo 1 - caso 2.

infectíveis gerada é proveniente dos operadores de mutação, discutidos na seção 5.7.2. Isto porque os algoritmos de recombinação deste grupo são estruturados utilizando-se o conceito de combinação convexa, que possui como propriedade principal a garantia de que os cromossomos gerados estão dentro do espaço de busca factível do problema.

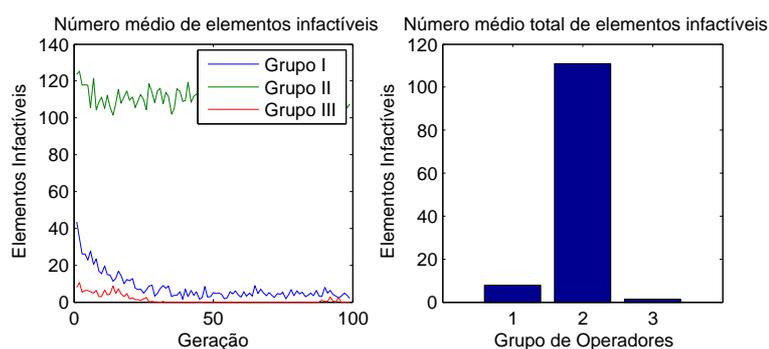


Figura 18: Número médio de elementos infectíveis gerados no exemplo 1 - caso 2.

A nuvem de pólos em malha fechada do melhor controlador encontrado com cada grupo de operadores é exposta na figura 19.

A tabela 7 oferece uma comparação dos melhores resultados encontrados nas 10 simulações realizadas com cada grupo de operadores com os resultados descritos em [101]. Verifica-se que, apesar da convergência média do grupo I ser insatisfatória, este grupo de operadores apresentou um resultado final próximo aos outros dois grupos de operadores.

Tabela 7: Comparação dos resultados encontrados no exemplo 1 - caso 2.

	Grupo I	Grupo II	Grupo III	Resultados de [101]
$\ T_{z_2 w_2}\ _2$	3,6663	3,6085	3,5779	4,3576
$\ T_{z_\infty w_\infty}\ _\infty$	13,2996	13,5598	13,5410	18,9714

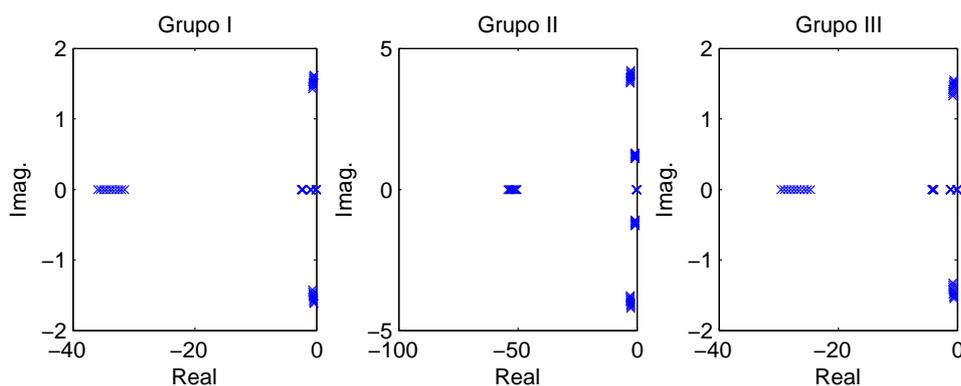


Figura 19: Pólos em malha fechada do exemplo 1 - caso 2.

A mudança do tipo de controlador, de realimentação estática de saída para realimentação dinâmica de saída de ordem 2, com o conseqüente aumento da dimensão do cromossomo, prejudicou o comportamento do grupo I de operadores. Os grupos II e III resultaram em comportamentos semelhantes em ambos casos, como pode ser observado na comparação da evolução do desvio padrão médio das figuras 12 e 16. O tempo médio de execução deste exemplo foi de 2 horas e 35 minutos, por simulação, em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

6.1.3 Caso 3: Realimentação Dinâmica de Saída de Ordem Completa

Nesta seção, projetaram-se controladores por realimentação dinâmica de saída de ordem completa ($n_c = 4$). A evolução dos três grupos de operadores, nesta síntese, é apresentada na figura 20. Os três grupos de operadores mostraram eficácia na evolução da população inicial, sendo que, na média, melhores resultados foram obtidos com os grupos II e III. A avaliação do comportamento do desvio padrão médio da população indica que, neste exemplo, a convergência dos grupos I e II de operadores foi mais lenta do que a convergência do grupo III. Este comportamento não prejudicou, porém, a eficácia do grupo II de operadores na busca de melhores soluções.

Embora com evoluções médias diferentes, os melhores resultados encontrados pelos três grupos de operadores são bastante próximos, como pode ser visto na figura 21.

O grupo II de operadores, mais uma vez, se destacou pela elevada geração de elementos inactivos, conforme se verifica na figura 22, indício de que os seus operadores de mutação não são eficazes na geração de elementos factíveis.

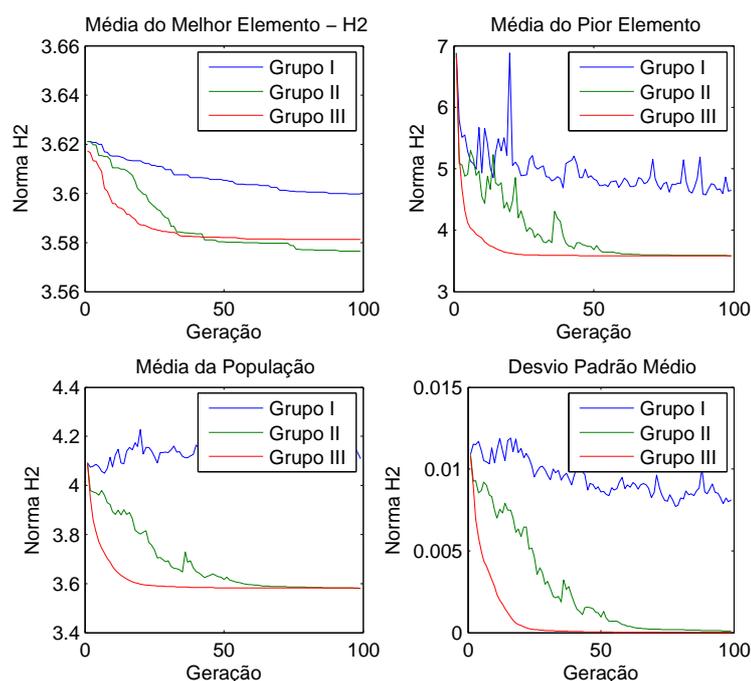


Figura 20: Evolução dos algoritmos no exemplo 1 - caso 3.

A nuvem de pólos em malha fechada para o melhor controlador encontrado com cada grupo de operadores é demonstrada na figura 23.

A tabela 8 apresenta a comparação entre os resultados encontrados pelos diferentes grupos de operadores. Observa-se a superioridade do resultado do grupo III, tanto com relação à norma \mathcal{H}_2 , quanto \mathcal{H}_∞ . A solução encontrada com este grupo de operadores, mostrada na figura 8, foi resultado do controlador dinâmico dado pelas matrizes:

$$A_K = \begin{bmatrix} -0,9829 & 2,8728 & 2,8945 & 4,4745 \\ 0,2316 & -0,1969 & 1,0088 & 0,7271 \\ -0,4868 & 0,0975 & -0,01254 & 0,8459 \\ 0,2657 & -1,9102 & -1,2569 & -2,9219 \end{bmatrix}, \quad B_K = \begin{bmatrix} 5,5477 \\ -2,9249 \\ 8,4659 \\ -0,3625 \end{bmatrix},$$

$$C_K = \begin{bmatrix} -0,2485 & 1,3422 & 1,1497 & 1,9671 \\ -0,6870 & -2,2508 & -1,9646 & -2,4559 \end{bmatrix}, \quad D_K = \begin{bmatrix} 1,5290 \\ -2,0408 \end{bmatrix}.$$

Buscando-se um ponto de referência para se avaliar o desempenho dos controladores projetados, sintetizaram-se controladores por realimentação dinâmica de saída de ordem completa para o sistema nominal, precisamente conhecido, onde os parâmetros incertos

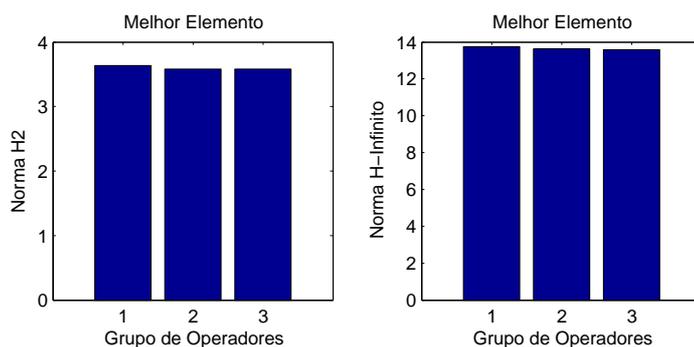


Figura 21: Comparação dos melhores resultados do exemplo 1 - caso 3.

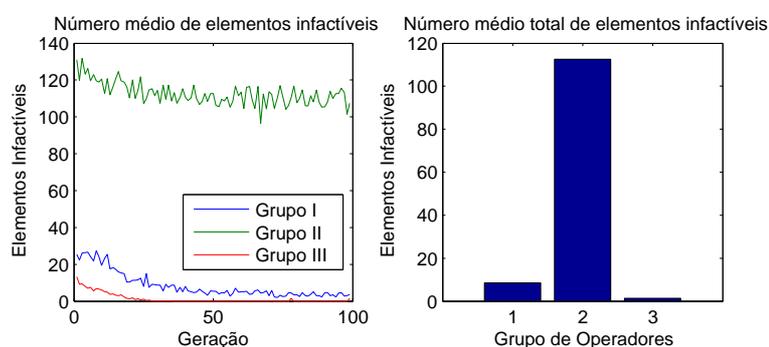


Figura 22: Número médio de elementos inactivos gerados no exemplo 1 - caso 3.

a_1 , a_2 e b_1 assumem os valores:

$$a_1 = 0,3681, \quad a_2 = 1,42, \quad b_1 = 3,5446.$$

Dois controladores foram sintetizados para o sistema nominal, o controlador que minimiza a norma \mathcal{H}_2 e o controlador que minimiza a norma \mathcal{H}_∞ . Para isso, utilizou-se o comando `hinfmix` para a síntese multi-objetivo de controladores via LMIs do Matlab. A coluna esquerda da tabela 9 mostra os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ para o controlador que minimiza a norma \mathcal{H}_2 , enquanto a coluna direita apresenta estes valores para o controlador que minimiza a norma \mathcal{H}_∞ . A comparação entre as tabelas 8 e 9 sugere que os resultados encontrados pelos algoritmos são satisfatórios tanto do ponto de vista de norma \mathcal{H}_2 , quanto \mathcal{H}_∞ , quando considerado que as incertezas do sistema estudado formam um poliedro convexo com 8 vértices. O tempo médio de execução deste exemplo foi de 8 horas e 28 minutos por simulação em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

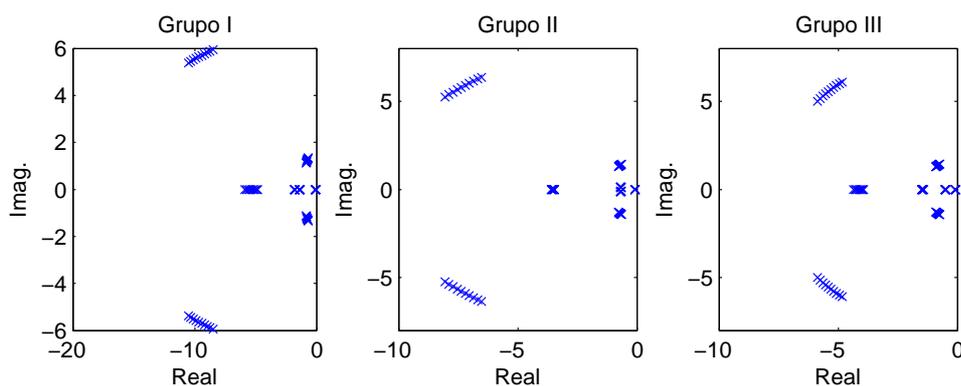


Figura 23: Pólos em malha fechada do exemplo 1 - caso 3.

Tabela 8: Comparação dos resultados do exemplo 1 - caso 3.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	3,6386	3,5800	3,5793
$\ T_{z_\infty w_\infty}\ _\infty$	13,7557	13,6215	13,5792

6.1.4 Análise do Exemplo 1

A figura 24 mostra o número de elementos inactíveis para as três estruturas de controlador utilizadas. Destaca-se a expressiva taxa média de elementos inactíveis gerada pelo grupo II de operadores. Considerando-se que os operadores de recombinação deste grupo são formados sobretudo por combinações convexas, o alto número de inactibilidades sugere que os algoritmos de mutação não são eficazes na geração de elementos factíveis. Este fato se comprova, uma vez que do número total de inactibilidades gerado por este grupo, verificou-se que apenas $\frac{1}{5}$ destas são formadas pelos operadores de recombinação. Isto indica que os operadores de mutação do grupo II, modificando de maneira aleatória os autovalores das matrizes de Lyapunov, se demonstram “agressivos”, geralmente resultando em elementos não factíveis.

Observou-se neste exemplo que o aumento da ordem do controlador, com o conseqüente aumento do número de parâmetros a serem otimizados, prejudicou sobretudo o desempenho do grupo I de operadores. A convergência do grupo II de operadores foi afetada em menor grau, como se pode verificar na análise da evolução do desvio padrão médio de cada caso. Por fim, o grupo III de operadores foi o menos influenciado pela ordem do controlador. Sugere-se que a utilização de informações de direção neste grupo de operadores contribuiu para sua convergência satisfatória, independente do número de parâmetros a ser otimizado.

Tabela 9: Controladores ótimos para o sistema nominal do exemplo 1 - caso 3.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	3,3405	187,1467
$\ T_{z_\infty w_\infty}\ _\infty$	12,4716	10,0642

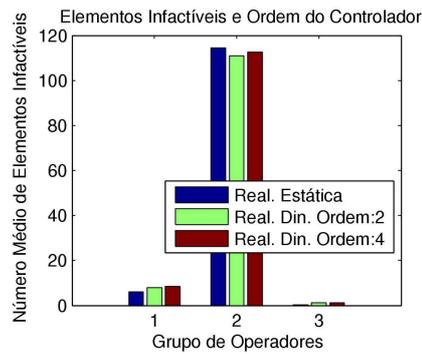


Figura 24: Número médio de elementos ineficazes gerados no exemplo 1.

Ressalta-se a influência direta da ordem do controlador no tempo médio de execução, que passou de 48 minutos, para o caso de realimentação estática, para 2 horas e 35 minutos para o projeto de controladores por realimentação dinâmica de saída de ordem reduzida. E ao se sintetizar controladores de ordem completa, o tempo de execução saltou para 8 horas e 28 minutos. Isto sugere que o aumento da ordem do controlador influencia diretamente no desempenho do algoritmo de otimização LMI de pontos interiores, pois aumenta a ordem do conjunto de LMIs utilizado.

6.2 Exemplo 2

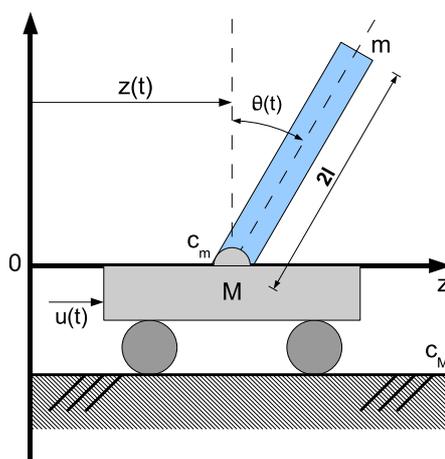


Figura 25: Sistema de pêndulo invertido.

Este exemplo foi extraído de [98] e descreve o pêndulo invertido ilustrado na figura 25. A massa M representa a massa do carrinho que possui um pêndulo de massa m uniforme e com comprimento $2l$ fixado no seu centro. O sistema é controlado pela aplicação da força escalar $u(t)$. A posição do carrinho é dada por $z(t)$, e $\theta(t)$ é o ângulo entre o pêndulo e o eixo vertical. Os coeficientes de atrito c_m e c_M relacionam o pêndulo e o carrinho e o carrinho e o solo, respectivamente.

Assume-se que θ seja pequeno, tal que:

$$\text{sen}\theta \cong \theta, \quad \text{cos}\theta \cong 1, \quad \dot{\theta}^2\theta \cong 0. \quad (6.2)$$

O modelo do sistema na figura 25 é dado por [98]:

$$\dot{x}_p(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_1 & a_2 & a_3 \\ 0 & a_4 & a_5 & a_6 \end{bmatrix} x_p(t) + \begin{bmatrix} 0 \\ 0 \\ b_1 \\ b_2 \end{bmatrix} u(t)$$

onde $x_p(t) \triangleq \begin{bmatrix} z & \theta & \dot{z} & \dot{\theta} \end{bmatrix}^T$ e

$$\begin{aligned} a_1 &= -\frac{1}{d}m^2gl^2 & a_2 &= -\frac{1}{d}c_M(J_m + ml^2) \\ a_3 &= \frac{1}{d}c_mml & a_4 &= \frac{1}{d}mgl(M + m) \\ a_5 &= \frac{1}{d}c_Mml & a_6 &= -\frac{1}{d}c_m(M + m) \\ b_1 &= \frac{1}{d}c_1(J_m + ml^2) & b_2 &= -\frac{1}{d}c_1ml \\ d &= (M + m)J_m + mMl^2. \end{aligned}$$

O vetor de saída observado $y(t)$ é dado por

$$y(t) = \begin{bmatrix} c_2 & 0 & 0 & 0 \\ 0 & c_3 & 0 & 0 \end{bmatrix} x(t).$$

A tabela 10 mostra os valores das constantes do sistema físico investigado.

Tabela 10: Constantes do pêndulo invertido.

Parâmetro	Valor	Dimensão
M	6,65	[kg]
m	$7,00 \times 10^{-2}$	[kg]
c_M	$7,20 \times 10^{-1}$	[kg/s]
c_m	$1,27 \times 10^{-3}$	[kg.m ² /s]
l	$1,15 \times 10^{-1}$	[m]
J_m	$1,65 \times 10^{-3}$	[kg.m ²]
c_1	$3,30 \times 10$	[N/V]
c_2	$2,33 \times 10$	[V/m]
c_3	3,32	[V/rad]
g	9,81	[m/s ²]

As incertezas do problema estão ligadas aos coeficientes de atrito c_M e c_m . Assume-se que ambos estejam contidos no intervalo de -10% a $+10\%$ do seu valor nominal. Com isso, o politopo de incertezas \mathcal{D}_A deste problema é formado por um conjunto de 4 matrizes vértices, dadas por:

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_1 & (1 - \delta)a_2 & (1 - \delta)a_3 \\ 0 & a_4 & (1 - \delta)a_5 & (1 - \delta)a_6 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_1 & (1 + \delta)a_2 & (1 + \delta)a_3 \\ 0 & a_4 & (1 + \delta)a_5 & (1 + \delta)a_6 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_1 & (1 + \delta)a_2 & (1 - \delta)a_3 \\ 0 & a_4 & (1 + \delta)a_5 & (1 - \delta)a_6 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_1 & (1 - \delta)a_2 & (1 + \delta)a_3 \\ 0 & a_4 & (1 - \delta)a_5 & (1 + \delta)a_6 \end{bmatrix},$$

onde $\delta = 0, 1$.

Calculando-se os parâmetros $a_1, \dots, a_6, b_1, b_2, c_2$ e c_3 para o sistema discutido, tem-se que as matrizes resultantes são dadas por:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_1 & a_2 & a_3 \\ 0 & a_4 & a_5 & a_6 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad C_y = \mathbf{I}_4,$$

$$B_{12}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \quad D_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} I_4 \\ 0_{1 \times 4} \end{bmatrix}, \quad B_{1\infty} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix},$$

$$D_{11_2} = 0, \quad C_2 = 0, 1 * \mathbf{I}_4, \quad D_{11_\infty} = 0,$$

$$D_{y1_\infty} = 0, \quad D_{21_\infty} = 0, \quad D_{22} = 0,$$

onde $a_1 = -0,0369$, $a_2 \in [-0,1183; -0,0968]$, $a_3 \in 10^{-3}[0,5336; 0,6522]$, $a_4 = 30,7744$, $a_5 \in [0,3025; 0,3697]$ e $a_6 \in [-0,5444; -0,4454]$. Os parâmetros a_2 e a_5 , assim como os parâmetros a_3 e a_6 , são derivados de somente duas variáveis incertas [98]. Assim, apenas quatro vértices formam o poliedro de incertezas \mathcal{D}_A , associado à matriz dinâmica do sistema. A região de alocação de pólos desejada é um disco de raio $r = 9$, com centro em $(-9.8, 0)$. A condição de sub-otimalidade utilizada foi $\gamma \leq 5$.

6.2.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subseção projetaram-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 2$). A evolução resultante da aplicação dos diferentes grupos de operadores evolutivos é apresentada na figura 26.

O grupo III de operadores obteve a melhor convergência entre os três grupos, com relação à norma média \mathcal{H}_2 , seguido pelo grupo II. Apesar de evoluções distintas dos diferentes grupos de operadores, observa-se que os três algoritmos foram capazes de fazer a população convergir para uma única solução, como refletido pela curva do desvio padrão

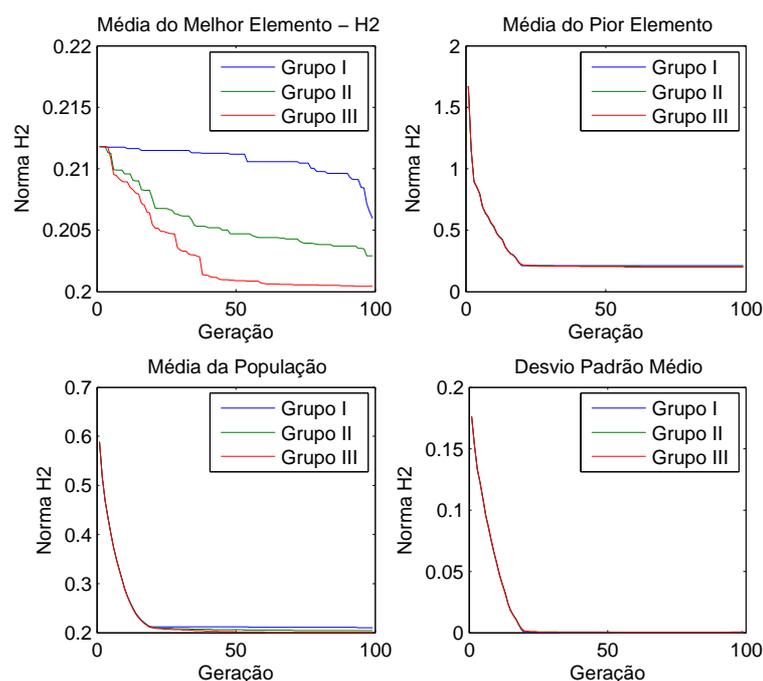


Figura 26: Evolução dos algoritmos no exemplo 2 - caso 1.

médio na figura 26. Além disso, na figura 27, observa-se que não houve significativa diferença com relação à norma \mathcal{H}_2 do melhor cromossomo encontrado pelos diferentes grupos de operadores, nas 10 simulações.

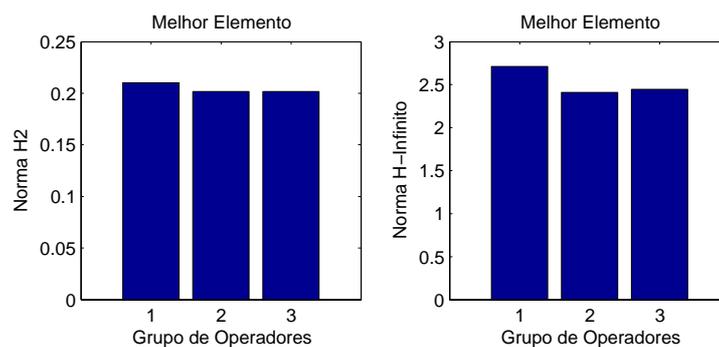


Figura 27: Comparação dos melhores resultados do exemplo 2 - caso 1.

A nuvem de pólos em malha fechada com o melhor cromossomo encontrado pelos algoritmos é apresentada na figura 28. Em detalhe, pode-se observar na figura 29 que os pólos em malha fechada respeitam a restrição de alocação de pólos do problema.

A tabela 11 mostra os melhores resultados encontrados pelos diferentes grupos de algoritmos. Do ponto de vista do objetivo adotado, a minimização da norma \mathcal{H}_2 , encontrou-se

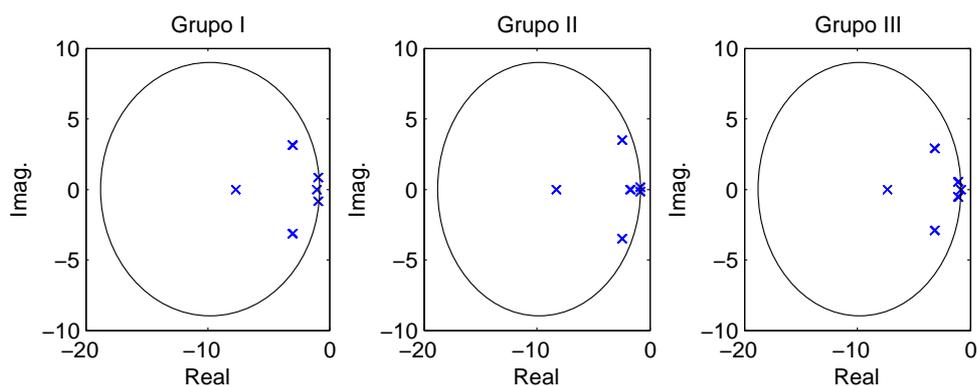


Figura 28: Pólos em malha fechada do exemplo 2 - caso 1.

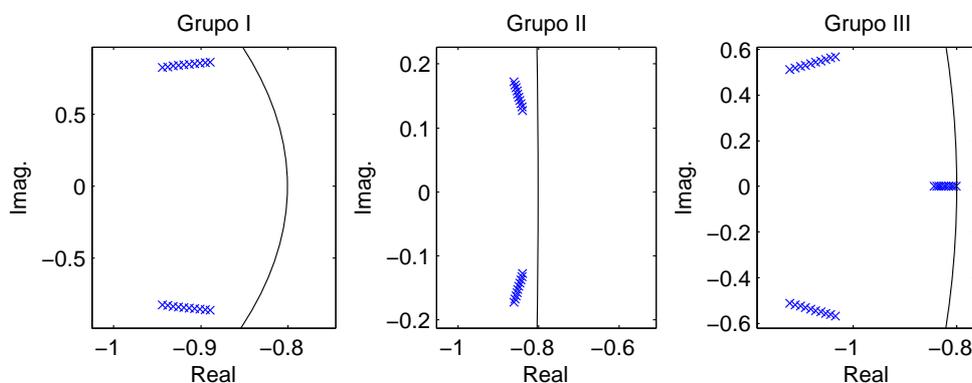


Figura 29: Detalhe dos pólos em malha fechada do exemplo 2 - caso 1.

o melhor controlador com o grupo II de operadores. As matrizes do controlador que representa a melhor solução encontrada são dadas por:

$$A_K = \begin{bmatrix} 85,6950 & 90,8026 \\ -96,1028 & -101,6581 \end{bmatrix}, \quad B_K = \begin{bmatrix} -5,8210 & -186,5870 \\ 6,5893 & 197,2523 \end{bmatrix}, \\ C_K = \begin{bmatrix} -1,0670 & -1,1130 \end{bmatrix}, \quad D_K = \begin{bmatrix} 0,0825 & 2,3998 \end{bmatrix}.$$

O tempo médio de execução, neste exemplo, foi de 6 horas e 16 minutos, por simulação, em um Pentium IV 2,6 GHz com 1,0 GB de memória RAM.

Tabela 11: Comparação dos melhores resultados do exemplo 2 caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	0,2102	0,2016	0,2018
$\ T_{z_\infty w_\infty}\ _\infty$	2,7125	2,4085	2,4471

6.2.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Completa

Nesta subseção sintetizaram-se controladores por realimentação dinâmica de saída de ordem completa ($n_c = 4$). A evolução dos diferentes grupos de operadores é apresentada na figura 30. A exemplo do ocorrido na síntese de controladores de ordem reduzida, o grupo III de operadores demonstra maior eficácia com relação à norma \mathcal{H}_2 . Neste caso, a norma média dos melhores elementos do grupo III foi cerca de 20% inferior à do grupo I e 9% inferior à norma média do grupo II de operadores. Os três grupos de operadores fizeram a população convergir para uma única solução no espaço de busca do problema, como pode ser observado pela curva do desvio padrão médio.

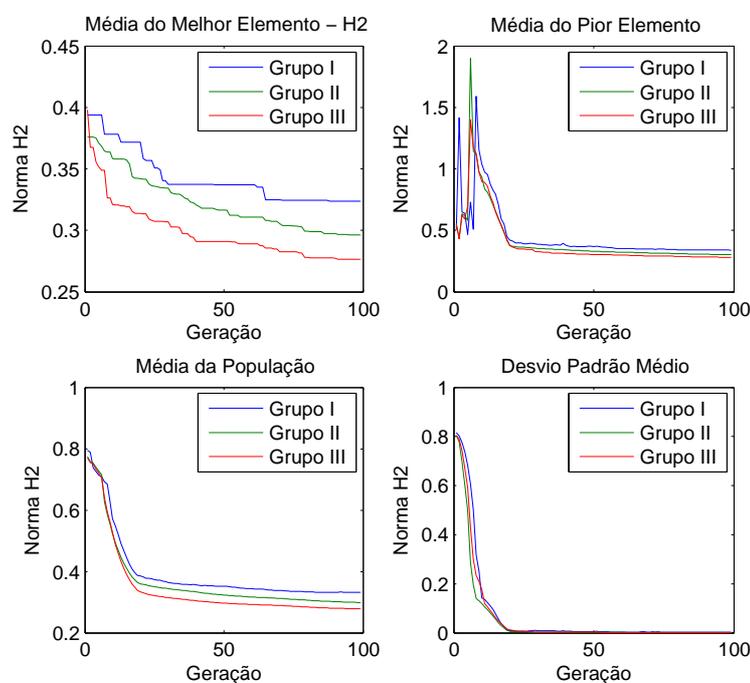


Figura 30: Evolução dos algoritmos no exemplo 2 - caso 2.

Os grupos I e III de operadores também foram superiores ao grupo II, quando considerada a norma \mathcal{H}_∞ , como pode ser verificado na figura 31.

Os pólos em malha fechada do melhor controlador encontrado com cada algoritmo estão desenhados nas figuras 32. Em detalhe, a figura 33 demonstra que a restrição de alocação de pólos não foi violada.

A tabela 12 exhibe os melhores resultados encontrados pelo algoritmo. O controlador, cuja norma \mathcal{H}_2 é a mais baixa, resultado da aplicação do grupo III de operadores, é dado

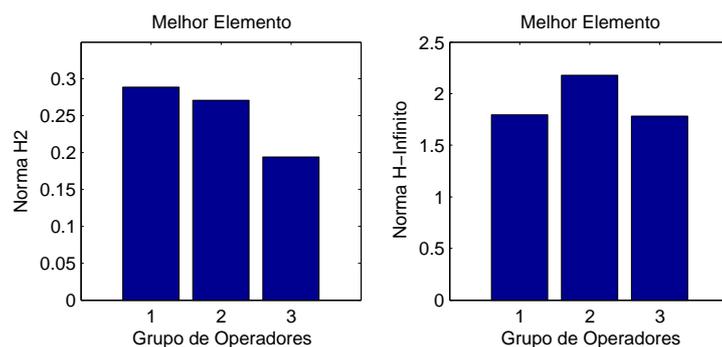


Figura 31: Comparação dos melhores resultados do exemplo 2 - caso 2.

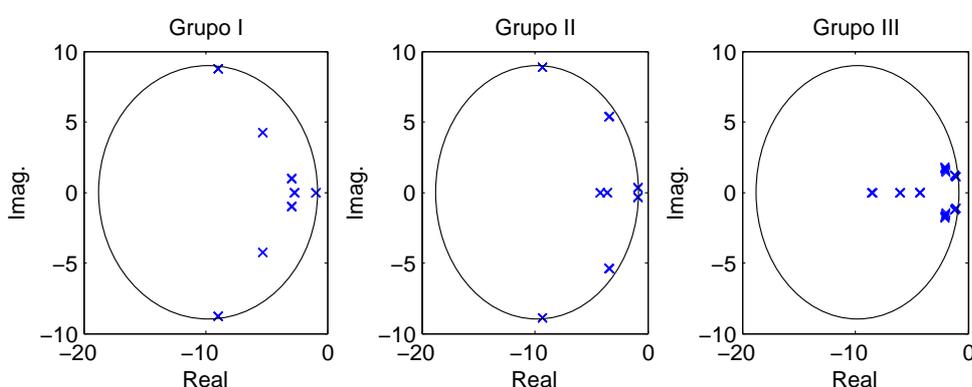


Figura 32: Pólos em malha fechada do exemplo 2 - caso 2.

pelas matrizes:

$$A_K = \begin{bmatrix} -10,7222 & -0,3819 & 2,3484 & -7,0453 \\ -2,7720 & -4,0182 & 2,3384 & -2,0189 \\ -5,0469 & 1,0550 & 1,6685 & -2,3598 \\ -12,6763 & 0,2841 & 7,7226 & -15,7121 \end{bmatrix}, \quad B_K = \begin{bmatrix} 1,7653 & 19,8589 \\ 0,6553 & 8,8968 \\ 1,0710 & 16,7618 \\ 3,2347 & 39,9706 \end{bmatrix}, \\
 C_K = \begin{bmatrix} -1,4343 & -0,0060 & 0,7436 & -1,3904 \end{bmatrix}, \quad D_K = \begin{bmatrix} 0,3300 & 4,8369 \end{bmatrix}.$$

Os resultados encontrados na literatura, para o problema incerto, e as soluções ótimas, calculadas para o sistema nominal são apresentados na tabela 13. Observa-se que a abordagem utilizada neste trabalho permitiu grande melhoria nos resultados, em comparação com o resultado de [98]. Os resultados da tabela 12 também são satisfatórios em comparação com as normas mínimas provenientes da síntese de controladores para o sistema nominal, considerando-se que o sistema analisado possui 2 parâmetros incertos.

O tempo médio de execução deste exemplo foi de 19 horas e 30 minutos por simulação em um Pentium IV 2,6 GHz com 1,0 GB de memória RAM.

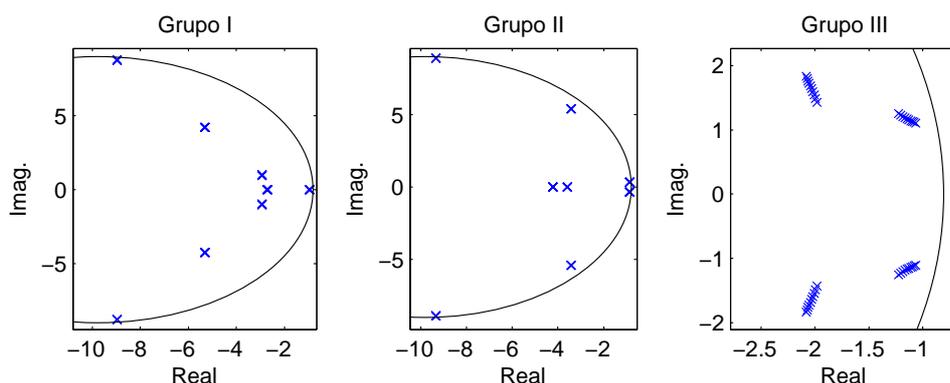


Figura 33: Detalhe dos pólos em malha fechada do exemplo 2 - caso 2.

Tabela 12: Comparação dos melhores resultados do exemplo 2 - caso 2.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	0,2887	0,2710	0,1938
$\ T_{z_\infty w_\infty}\ _\infty$	1,7940	2,1768	1,7808

6.2.3 Análise do Exemplo 2

Neste exemplo, o grupo III de operadores apresentou eficácia bastante superior aos outros dois grupos, quando analisada a evolução média da população para os casos de síntese de controladores por realimentação dinâmica de saída de ordem reduzida e completa. Em comparação com os resultados encontrados em [98], os três grupos de operadores foram eficazes. A elevação da ordem do controlador, de 2 para 4, resultou em um aumento de custo computacional de cerca de 212%. Este fato indica uma limitação do algoritmo apresentado, em função do elevado custo computacional, na utilização de sistemas ou controladores de ordens elevadas.

6.3 Exemplo 3

Este exemplo foi extraído e adaptado da biblioteca COMPlib [82] e corresponde a uma aeronave supersônica de transporte, em vôo Mach 2,7. A condição de sub-otimalidade

Tabela 13: Resultados da literatura e ótimos para o sistema nominal do exemplo 2 - caso 2.

	Resultados de [98]	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	3,6433	0,1260	0,1671
$\ T_{z_\infty w_\infty}\ _\infty$	—	1,7808	0,2732

$\gamma \leq 60$ foi utilizada e a matriz dinâmica do sistema é dada por:

$$A = \begin{bmatrix} -0,037 & 0,0123 & 0,00055 & -1 \\ 0 & 0 & 1 & 0 \\ -0,37 & 0 & -0,23 & 0,0618 \\ a_1 & 0 & 0,016 & -0,0457 \end{bmatrix},$$

onde o parâmetro incerto a_1 assume os valores $a_1 \in [0,75; 1,75]$, formando um poliedro convexo de incertezas \mathcal{D}_A com dois vértices. O restante das matrizes do sistema é dado por:

$$B_2 = \begin{bmatrix} 0,00084 & 0,000236 \\ 0 & 0 \\ 0,08 & 0,804 \\ -0,0862 & -0,0665 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1,3 & 0 & 0 & 0 \\ 0 & 1,3 & 0 & 0 \\ 0 & 0 & 1,3 & 0 \\ 0 & 0 & 0 & 1,3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad C_y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$D_{22} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_{11_2} = 0, \quad D_{11_\infty} = 0, \quad B_{1_\infty} = 1,2 * \mathbf{I}_4,$$

$$D_{21_\infty} = 0, \quad D_{y1_\infty} = 0, \quad B_{1_2} = \mathbf{I}_4.$$

6.3.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subseção sintetizaram-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 2$). A figura 34 apresenta a evolução dos diferentes grupos de operadores. A análise desta figura permite identificar a superioridade do grupo III de

operadores, quando observada a evolução da média do melhor elemento. Além disso, os gráficos da média do pior elemento e da média da população indicam que o grupo I de operadores piorou a norma \mathcal{H}_2 média da população. Neste caso, indica-se que os operadores do grupo I, baseados no algoritmo de Michalewicz, “deterioraram” os cromossomos selecionados, permitindo que a média da norma \mathcal{H}_2 da população se elevasse.

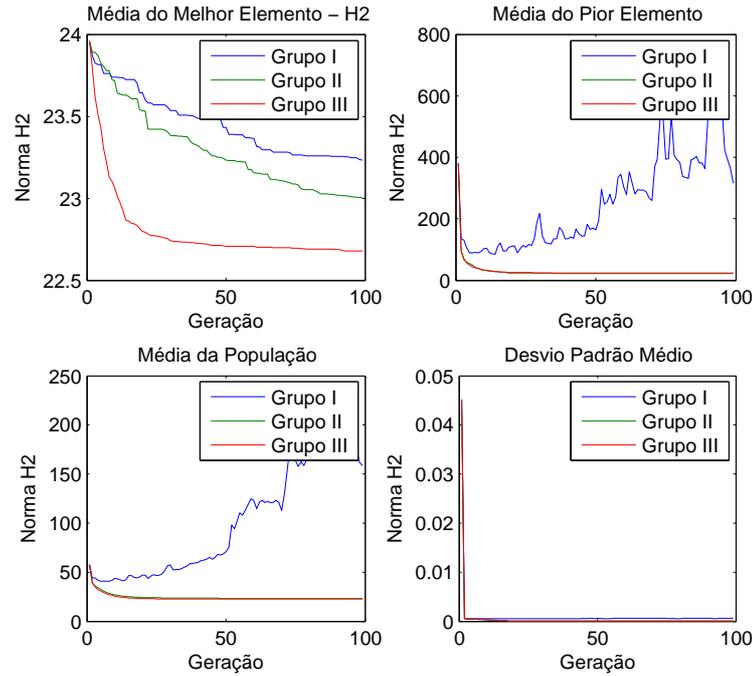


Figura 34: Evolução média dos algoritmos no exemplo 3 - caso 1.

As normas calculadas com o melhor cromossomo encontrado pelos diferentes grupos de operadores são mostradas na figura 35. Observa-se que os três grupos de operadores resultam em controladores com normas bastante próximas, quando analisada a melhor solução encontrada nas 10 simulações. O melhor resultado é dado pelo controlador formado pelas matrizes:

$$A_K = \begin{bmatrix} -0,9632 & 1,3932 \\ 0,3405 & -26,2650 \end{bmatrix}, \quad B_K = \begin{bmatrix} 1,4963 & 1,9046 & 2,2925 \\ 1,1910 & 0,7764 & 1,2131 \end{bmatrix},$$

$$C_K = \begin{bmatrix} 0,8088 & 9,2137 \\ -14,6867 & -6,5519 \end{bmatrix}, \quad D_K = \begin{bmatrix} 10,5435 & 11,7560 & 22,1648 \\ -9,8844 & -9,2480 & -7,9968 \end{bmatrix},$$

e foi encontrado pelo grupo III de operadores.

O número médio de elementos ineficazes gerado pelos diferentes algoritmos é apresentado na figura 36. Observa-se que o grupo III de operadores apresentou menor número

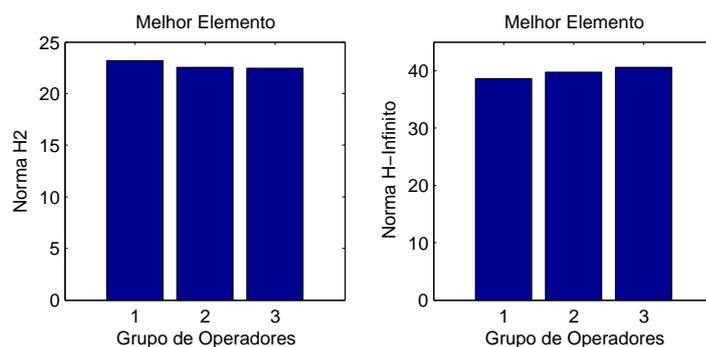


Figura 35: Comparação dos melhores resultados do exemplo 3 - caso 1.

de elementos inactivos gerados. Este fato é resultante da utilização das informações de direção na busca, tanto na recombinação através da evolução diferencial, quanto na mutação com o operador de Salomon. O grupo II de operadores, como observado nos exemplos anteriores, se destacou com um elevado número médio de elementos inactivos gerados, resultado da estrutura dos operadores de mutação deste grupo.

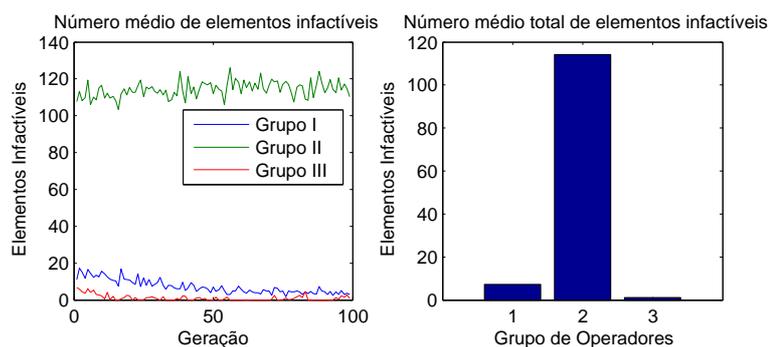


Figura 36: Número médio de elementos inactivos gerados no exemplo 3 - caso 1.

A figura 37 mostra a nuvem de pólos dos melhores controladores encontrados pelos diferentes algoritmos. Observa-se que as soluções encontradas geram nuvens bastante distintas. Isto sugere que regiões diferentes no espaço de busca foram exploradas pelos diferentes algoritmos.

A tabela 14 apresenta um resumo dos resultados do melhor cromossomo encontrado com cada grupo de operadores. Observa-se que o conjunto das melhores soluções encontradas forma um grupo de soluções ótimas de Pareto. O tempo médio de execução deste exemplo foi de 37 minutos, por simulação, em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

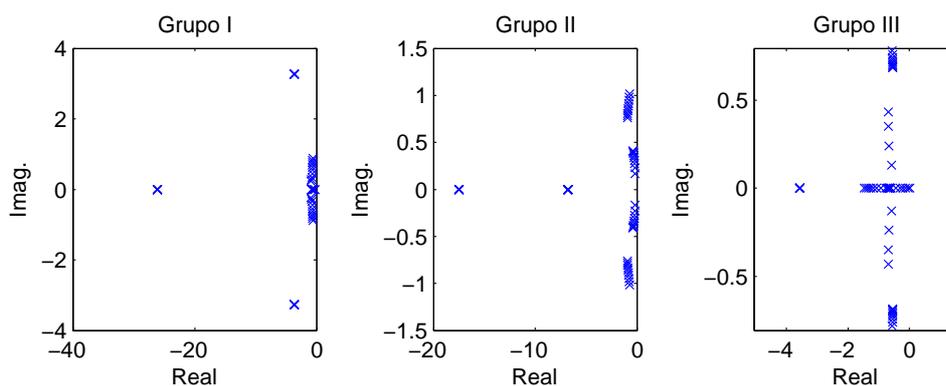


Figura 37: Pólos em malha fechada do exemplo 3 - caso 1.

Tabela 14: Comparação dos resultados do exemplo 3 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	23, 1928	22, 5320	22, 4758
$\ T_{z_\infty w_\infty}\ _\infty$	38, 6098	39, 7865	40, 5776

6.3.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Completa

Nesta subseção, projetaram-se controladores por realimentação dinâmica de saída de ordem completa ($n_c = 4$). O resultado da aplicação dos diferentes algoritmos é mostrado na figura 38. O grupo III de operadores se destaca pela eficácia na convergência da média dos melhores elementos, com relação à norma \mathcal{H}_2 . Além disso, o formato desta curva lembra o comportamento de convergência atingido por métodos de gradiente. Isto sugere que a utilização das informações de direção efetivamente contribuíram para a busca de melhores elementos no espaço de factibilidade do problema, fato observado também em diversos outros exemplos analisados.

A aplicação do grupo II de operadores resultou em uma evolução média melhor do que o conjunto de operadores baseados no algoritmo de Michalewicz. Os melhores elementos encontrados pelos três algoritmos são, entretanto, bastante próximos, do ponto de vista de norma \mathcal{H}_2 , como mostrado na figura 39.

Um comportamento recorrente, observado nos diversos exemplos, é grande quantidade média de elementos ineficazes gerada pela aplicação do grupo II de operadores, como pode ser visto na figura 40. Ressalta-se que este fato se deve às características dos operadores de mutação do grupo II, bastante “agressivos” em relação às matrizes de Lyapunov. Neste

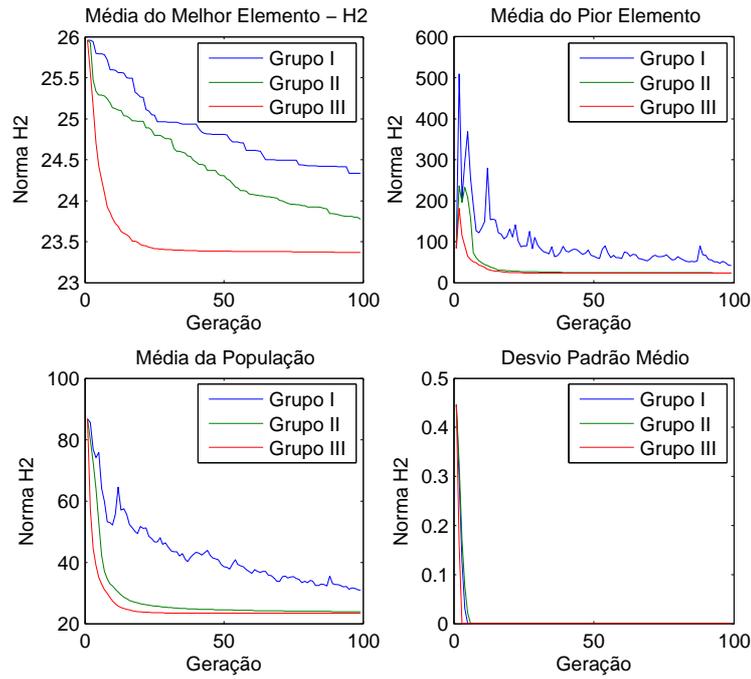


Figura 38: Evolução média dos algoritmos no exemplo 3 - caso 2.

exemplo, em média, apenas 23% dos elementos inactivos foram resultantes dos operadores de recombinação do grupo II. O restante das inactividades foram geradas pelos operadores de mutação deste grupo.

A tabela 15 apresenta um resumo dos resultados do melhor cromossomo encontrado com cada grupo de operadores. Do ponto de vista da função a ser otimizada (norma \mathcal{H}_2), o grupo de operadores III obteve o melhor resultado, dado pelo controlador:

$$A_K = \begin{bmatrix} -1,0785 & 0,4049 & 0,5756 & -0,7955 \\ 0,5264 & -0,8272 & 0,8304 & 0,2137 \\ 0,4144 & 0,8597 & 0,0935 & 0,2556 \\ 0,1607 & 0,2256 & 0,8486 & -1,3200 \end{bmatrix},$$

$$B_K = \begin{bmatrix} 0,7310 & 0,7035 & 0,8452 \\ 0,8353 & 0,7345 & 0,9410 \\ 1,4437 & 1,3725 & 1,4858 \\ 0,3117 & 0,4929 & 0,2476 \end{bmatrix},$$

$$C_K = \begin{bmatrix} 5,2543 & 2,3754 & 14,5334 & 4,1151 \\ -7,1420 & -9,5255 & -13,7142 & -4,1018 \end{bmatrix},$$

$$D_K = \begin{bmatrix} -0,6775 & 4,1854 & 16,4928 \\ -11,4282 & -16,6085 & -11,5967 \end{bmatrix}.$$

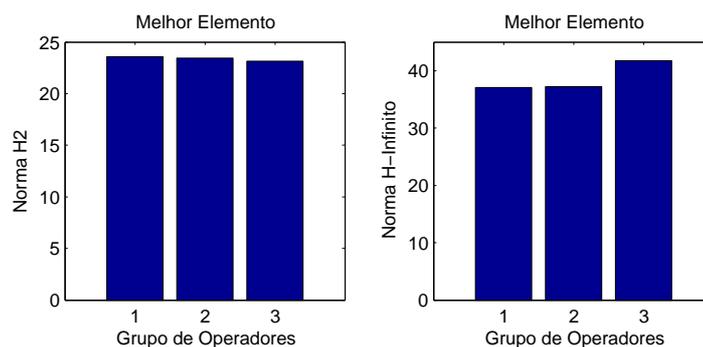


Figura 39: Comparação dos melhores resultados do exemplo 3 - caso 2.

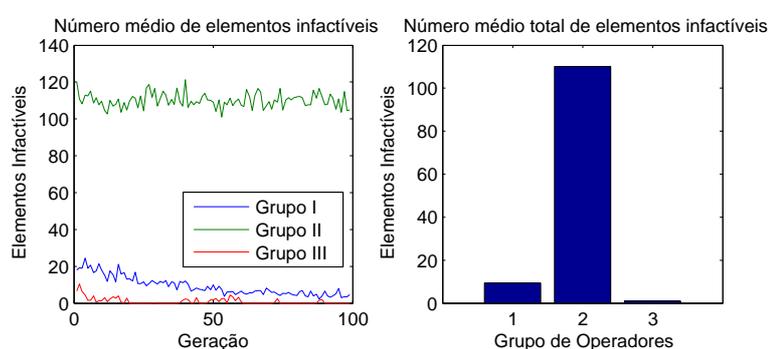


Figura 40: Número de elementos inactivos gerados no exemplo 3 - caso 2.

Entretanto, considerando-se o critério misto, observa-se que as três melhores soluções formam um grupo de soluções ótimas de Pareto.

Tabela 15: Comparação dos resultados do exemplo 3 - caso 2.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	23, 5811	23, 4583	23, 1382
$\ T_{z_\infty w_\infty}\ _\infty$	37, 0518	37, 2177	41, 7280

A tabela 16 mostra os valores ótimos de norma \mathcal{H}_2 e \mathcal{H}_∞ , quando sintetizados controladores de ordem completa para o sistema nominal, onde o parâmetro incerto $a_1 = 1, 25$. Neste caso, as soluções encontradas pelos algoritmos estão mais distantes da solução ótima para o modelo nominal.

As nuvens de pólos em malha fechada, formadas com o melhor controlador encontrado com cada grupo de operadores, estão apresentadas na figura 41. Observa-se que estas nuvens são diferentes entre si, sugerindo uma exploração de regiões distintas do espaço de busca pelos diferentes algoritmos. O tempo médio de execução deste exemplo foi de 1

Tabela 16: Controladores ótimos para o sistema nominal do exemplo 3 - caso 2.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	13,9267	225,3829
$\ T_{z_\infty w_\infty}\ _\infty$	24,4635	17,8443

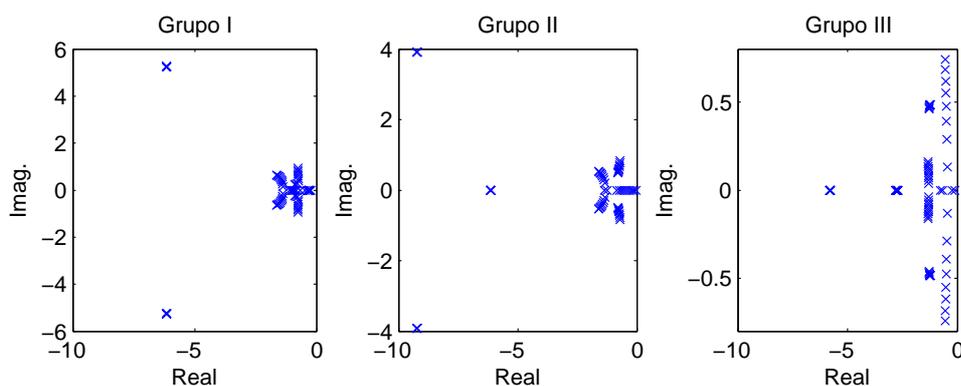


Figura 41: Pólos em malha fechada do exemplo 3 - caso 2.

hora e 16 minutos por simulação em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

6.3.3 Análise do Exemplo 3

A evolução média do grupo III de operadores apresentou significativa superioridade em relação à dos outros grupos, tanto na síntese de controladores de ordem reduzida quanto completa. O tempo médio de execução sugere uma relação polinomial entre a ordem do controlador e a eficiência computacional média dos algoritmos. O aumento da ordem do controlador de 2 para 4 resultou em um aumento de 105% no tempo de execução. A figura 42 não indica uma correlação direta entre número de elementos ineficazes gerados e a ordem do controlador.

O comportamento "suave", recorrente na evolução do grupo III de operadores com relação à media do melhor elemento, em conjunto com os bons resultados apresentados, sugere que este grupo de operadores fornece boas soluções com um número reduzido de simulações. Neste sentido, pode-se dizer que o algoritmo de ED apresenta um comportamento de busca mais robusto, em relação aos outros grupos de operadores, fato coerente com as características da ED e do algoritmo de Salomon ressaltadas em [122, 114].

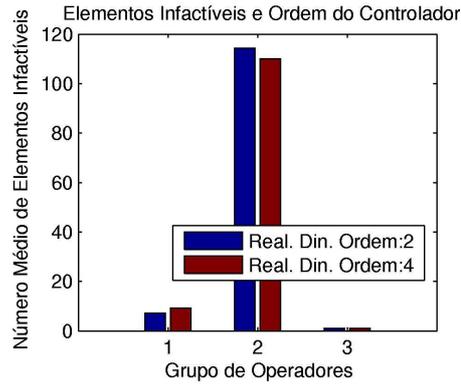


Figura 42: Número médio de elementos ineficazes do exemplo 3.

6.4 Exemplo 4

Este exemplo foi extraído e adaptado da biblioteca COMPlib [82] e corresponde ao modelo linearizado do piloto automático de um míssil ar-ar. A restrição de norma \mathcal{H}_∞ utilizada foi $\gamma \leq 100$ e adicionou-se um parâmetro incerto à matriz dinâmica do sistema, dada por:

$$A = \begin{bmatrix} -0,876 & 1 & -0,1209 & 0 \\ a_1 & 0 & -130,75 & 0 \\ 0 & 0 & -150 & 0 \\ -1 & 0 & 0 & -0,05 \end{bmatrix},$$

onde $a_1 \in [3,9117; 13,9117]$ forma o domínio de incerteza \mathcal{D}_A do problema. As outras matrizes do sistema são dadas por:

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 150 \\ 0 \end{bmatrix}, \quad C_y = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} -0,5 & 0 & 0 & 3 \\ 0 & 0 & -1,5 & 0 \end{bmatrix},$$

$$B_{1_2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_{1_\infty} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1,2 \end{bmatrix}, \quad C_1 = \begin{bmatrix} -0,25 & 0 & 0 & 3,487 \\ 0 & -3 & 0 & 0 \end{bmatrix},$$

$$D_{11_2} = 0, \quad D_{11_\infty} = \begin{bmatrix} 0,25 \\ 0 \end{bmatrix}, \quad D_{12} = \begin{bmatrix} 0 \\ 3 \end{bmatrix},$$

$$D_{21_\infty} = 0, \quad D_{y1_\infty} = 0, \quad D_{22} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

6.4.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subseção, considera-se a síntese de controladores robustos por realimentação dinâmica de ordem reduzida ($n_c = 2$). A evolução dos diferentes algoritmos é apresentada na figura 43. Não se verificam diferenças significativas na convergência média dos algoritmos, uma vez que a evolução não é perceptível até a quarta casa decimal.

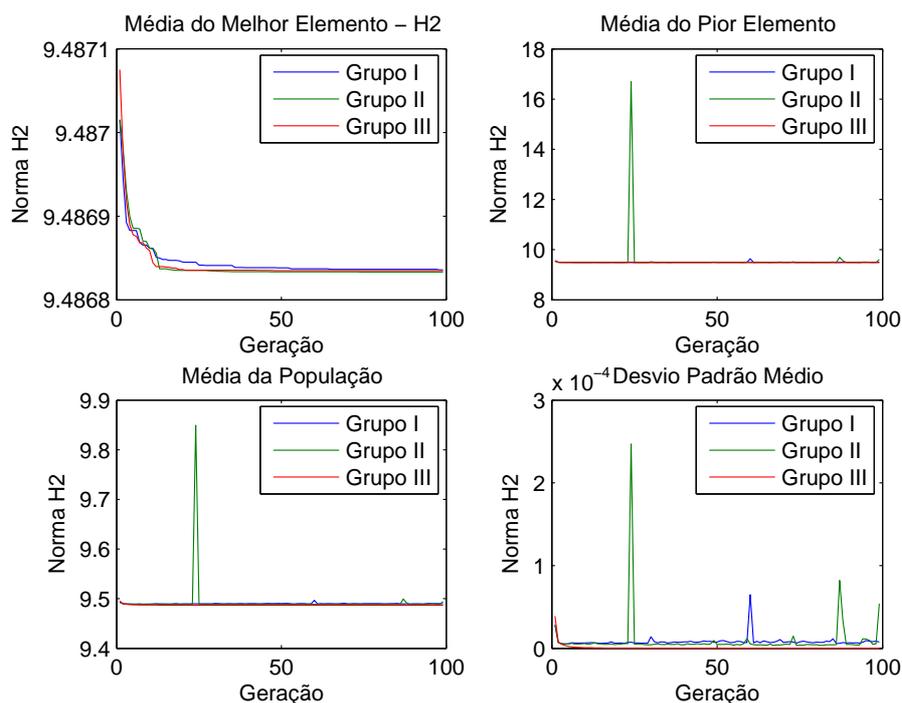


Figura 43: Evolução média dos algoritmos no exemplo 4 - caso 1.

A figura 44 mostra as normas dos melhores elementos obtidos com cada algoritmo. Os valores da norma \mathcal{H}_2 , obtidos com os melhores controladores, sintetizados a partir dos diferentes grupos de operadores, são idênticos. Considerando-se a norma \mathcal{H}_∞ , observa-se que o grupo II de operadores foi superior aos outros dois grupos e o controlador resultante desta síntese é dado por:

$$A_K = \begin{bmatrix} -206,5674 & -232,1344 \\ -151,3151 & -215,4429 \end{bmatrix}, \quad B_K = \begin{bmatrix} 278,2279 & 163,7945 \\ 147,3315 & 140,3374 \end{bmatrix}, \\ C_K = \begin{bmatrix} 9,9715 & 12,6213 \end{bmatrix}, \quad D_K = \begin{bmatrix} -12,7972 & -12,1918 \end{bmatrix}.$$

Ressalta-se que os valores diferentes de norma \mathcal{H}_∞ não se devem a qualquer característica de minimização do algoritmo, uma vez que a função objetivo não considera a

norma \mathcal{H}_∞ .

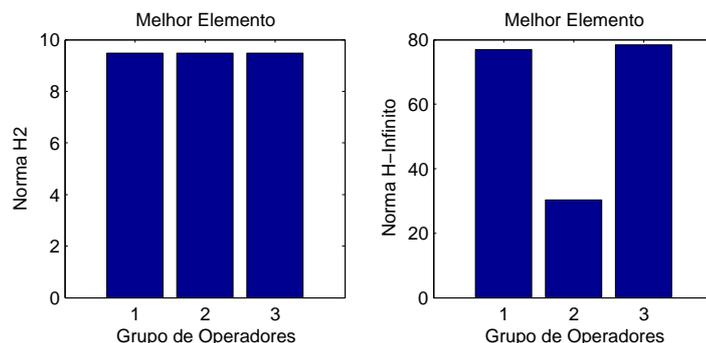


Figura 44: Comparação dos melhores resultados do exemplo 4 - caso 1.

O número de elementos inactivos gerados pelos diferentes grupos de operadores é mostrado na figura 45. Como observado nos diversos exemplos estudados, o grupo II se destaca pela elevada taxa de elementos inactivos gerada, fruto da estrutura dos seus operadores de mutação.

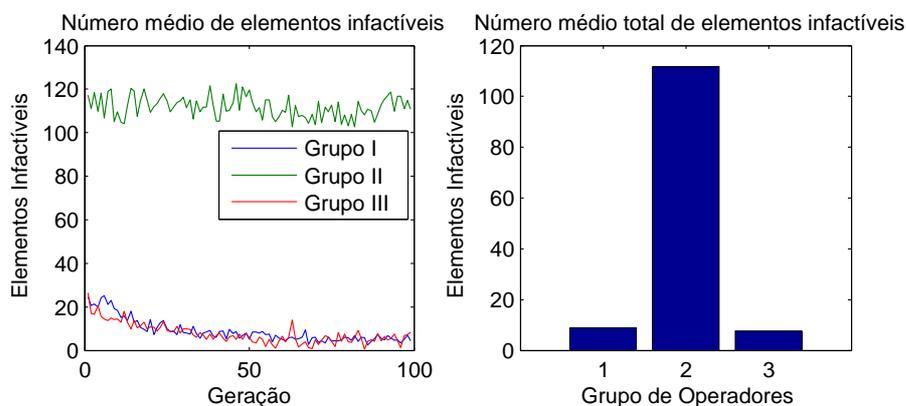


Figura 45: Número de elementos inactivos gerados no exemplo 4 - caso 1.

As nuvens de pólos em malha fechada, formadas com o melhor controlador de cada algoritmo, são demonstradas na figura 46. Ressalta-se que, apesar do valor idêntico das normas \mathcal{H}_2 , resultantes dos melhores controladores projetados, as nuvens de pólos desenhadas na figura 46 são distintas. A tabela 17 mostra a comparação dos resultados dos diferentes grupos de operadores. O tempo médio de execução deste exemplo foi de 53 minutos por simulação em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

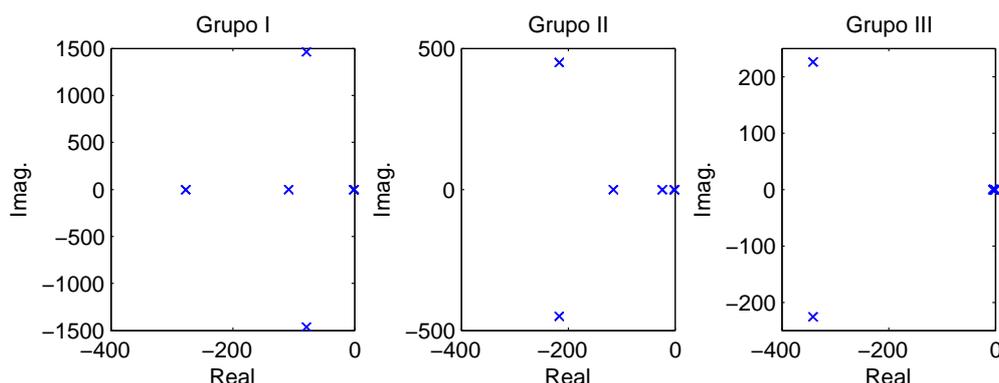


Figura 46: Pólos em malha fechada do exemplo 4 - caso 1.

Tabela 17: Comparação dos melhores resultados do exemplo 4 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	9,4868	9,4868	9,4868
$\ T_{z_\infty w_\infty}\ _\infty$	76,9104	30,3800	78,4414

6.4.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida com Alocação de Pólos

Nesta seção, projetaram-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 2$), adicionando-se uma restrição de alocação de pólos. O objetivo da alocação de pólos é reduzir o esforço de controle e, para isso, foi adicionada uma faixa vertical com limites esquerdo $h_1 = -70$ e direito $h_2 = 0$. A figura 47 mostra a convergência média dos algoritmos neste caso. Como na subseção anterior, não se verificam diferenças significativas na convergência média dos algoritmos, uma vez que a evolução não é perceptível até a quarta casa decimal.

A figura 48 apresenta as normas dos melhores elementos obtidos com cada algoritmo. Não se verifica diferenças nas normas \mathcal{H}_2 . Considerando-se a norma \mathcal{H}_∞ , observa-se que o grupo II de operadores foi superior aos outros dois grupos de operadores, como ocorrido na subseção anterior. O melhor controlador encontrado, resultante da utilização do grupo II de operadores, é dado por:

$$A_K = \begin{bmatrix} -117,4396 & 76,9693 \\ -483,0042 & 104,8242 \end{bmatrix}, \quad B_K = \begin{bmatrix} -110,5334 & -0,6939 \\ -146,8350 & -101,1469 \end{bmatrix},$$

$$C_K = \begin{bmatrix} 0,2821 & -1,1787 \end{bmatrix}, \quad D_K = \begin{bmatrix} -0,7082 & -0,5997 \end{bmatrix}.$$

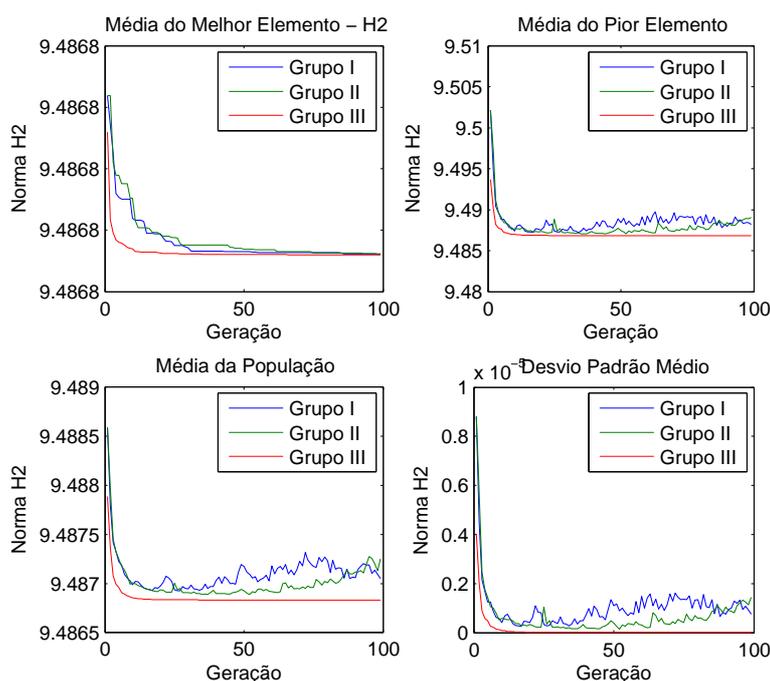


Figura 47: Evolução média dos algoritmos no exemplo 4 - caso 2.

Observando-se a figura 49, em comparação com a figura 45 da subseção anterior, observa-se que o número de elementos inactíveis gerados pelos três grupos de algoritmos se elevou. Este fato é decorrente da adição da restrição de alocação de pólos. Esta restrição diminui o espaço de busca factível do problema, aumentando-se a probabilidade de geração de elementos inactíveis pelos diferentes grupos de operadores.

As nuvens de pólos em malha fechada, formadas com o melhor controlador de cada algoritmo são demonstradas na figura 50. Apesar de gerar nuvens de pólos distintas, os diferentes controladores obedecem à restrição de alocação de pólos imposta ao sistema e possuem valores idênticos de norma \mathcal{H}_2 .

A tabela 18 reflete os resultados obtidos com cada grupo distinto de operadores. Observa-se a superioridade do grupo de II de operadores no resultado da norma \mathcal{H}_∞ . Lembra-se que o melhor valor de norma \mathcal{H}_2 não se deve, contudo, a qualquer característica de busca do algoritmo, uma vez que a função objetivo considerada não inclui a minimização da norma \mathcal{H}_∞ . Observa-se, entretanto, que a adição da restrição de alocação de pólos contribuiu para reduzir o valor máximo da norma \mathcal{H}_∞ encontrado com os três diferentes grupos de operadores. O tempo médio de execução deste exemplo foi de 51 minutos por simulação em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

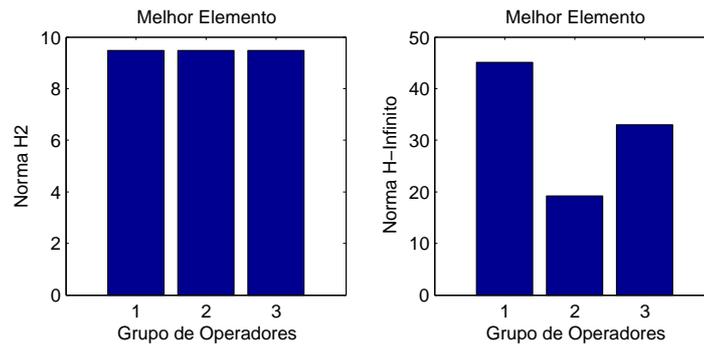


Figura 48: Comparação dos melhores resultados do exemplo 4 - caso 2.

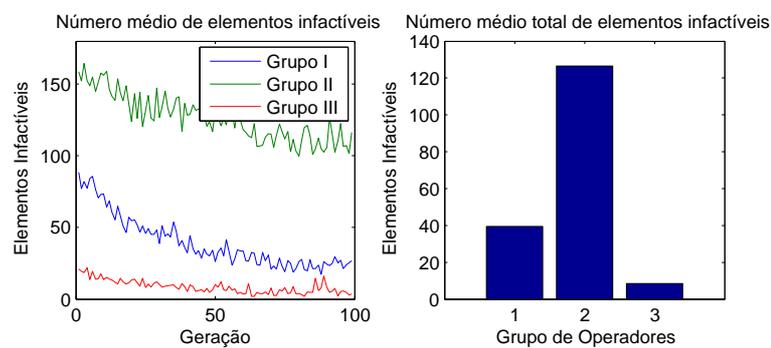


Figura 49: Número de elementos inactivos gerados no exemplo 4 - caso 2.

6.4.3 Caso 3: Realimentação Dinâmica de Saída de Ordem Completa

Nesta subseção projetaram-se controladores por realimentação dinâmica de saída de ordem completa ($n_c = 4$). Nenhuma restrição de alocação de pólos foi considerada. As evoluções dos diferentes grupos de operadores são mostradas na figura 51. Como nas seções anteriores deste exemplo, não se verificam diferenças significativas na convergência média dos algoritmos, que se dá apenas na quarta casa decimal.

Os melhores resultados, encontrados com os diferentes grupos de operadores, são mostrados na figura 52. Novamente, os três algoritmos foram igualmente eficazes na otimização da norma \mathcal{H}_2 . Neste caso, os valores de norma \mathcal{H}_∞ encontrados também são muito próximos. O melhor controlador encontrado pelo grupo III de operadores é dado

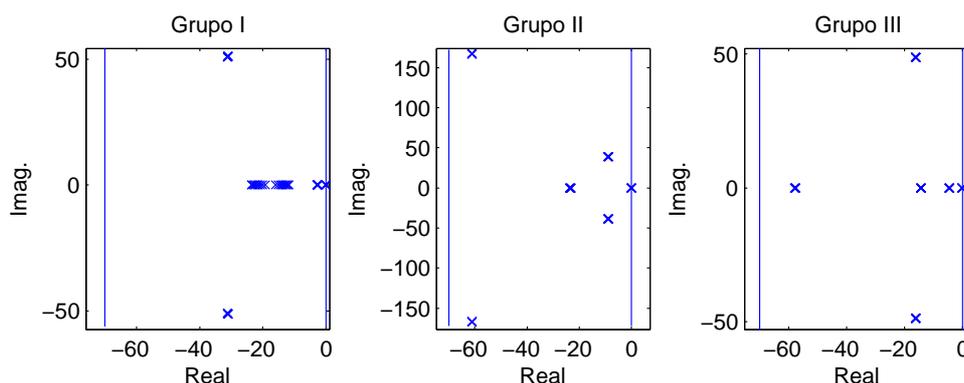


Figura 50: Pólos em malha fechada do exemplo 4 - caso 2.

Tabela 18: Comparação dos melhores resultados do exemplo 4 - caso 2.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	9,4868	9,4868	9,4868
$\ T_{z_\infty w_\infty}\ _\infty$	45,0988	19,2271	32,9908

por:

$$A_K = \begin{bmatrix} -1,4291 & -27,0162 & -19,2909 & -28,6704 \\ -2,1552 & -50,6019 & -35,3873 & -52,8098 \\ -1,2100 & -23,9871 & -17,5137 & -25,2740 \\ -2,0182 & -43,4900 & -30,7174 & -46,4866 \end{bmatrix}, \quad B_K = \begin{bmatrix} 28,2528 & 9,3080 \\ 55,2129 & 17,8091 \\ 27,2690 & 8,6917 \\ 48,6578 & 15,6122 \end{bmatrix}, \\
 C_K = \begin{bmatrix} -0,0124 & -0,4424 & -0,3164 & -0,4698 \end{bmatrix}, \quad D_K = \begin{bmatrix} 0,2571 & 0,0741 \end{bmatrix}.$$

A figura 53 mostra o número de elementos ineficazes gerados com a utilização dos diferentes grupos de operadores. Em comparação com a figura 45, respectiva à síntese de controladores por realimentação dinâmica de saída de ordem reduzida e sem restrições de alocação de pólos, não se verificam significativas diferenças com relação ao número de elementos ineficazes gerados. Isto sugere que o aumento da ordem do controlador a ser projetado não possui influência na eficiência dos algoritmos, quando analisado o número de cromossomos ineficazes gerados.

As nuvens de pólos em malha fechada, formadas com o melhor controlador de cada algoritmo, são demonstradas na figura 54. É interessante notar que, apesar dos valores muito próximos de norma \mathcal{H}_2 e \mathcal{H}_∞ , as nuvens de pólos em malha fechada são significativamente diferentes.

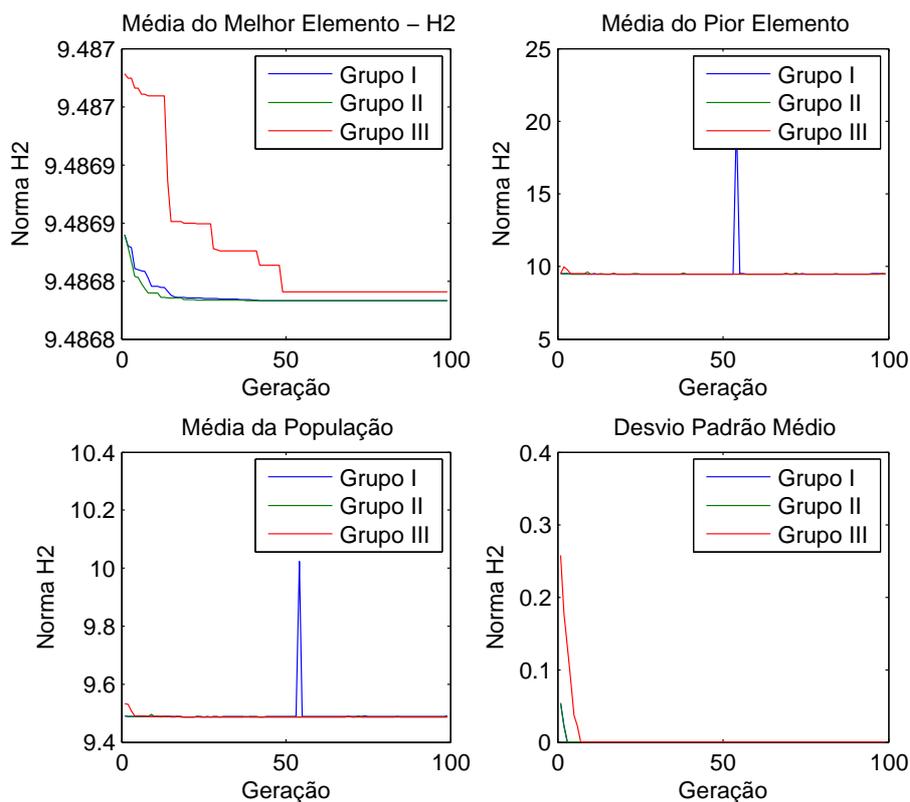


Figura 51: Evolução média dos algoritmos no exemplo 4 - caso 3.

A tabela 19 apresenta os melhores resultados obtidos com cada grupo de algoritmos distinto. Observa-se que os resultados dos grupos II e III são idênticos até a quarta casa decimal. A tabela 20 mostra as normas ótimas encontradas para o sistema nominal, onde

Tabela 19: Comparação dos melhores resultados do exemplo 4 - caso 3.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	9,4868	9,4868	9,4868
$\ T_{z_\infty w_\infty}\ _\infty$	83,9460	83,9383	83,9383

o parâmetro a_1 assume o valor dado por $a_1 = 8,9117$. O resultado encontrado pelos grupos II e III são iguais e correspondem às normas ótimas encontradas com a síntese de controladores \mathcal{H}_2 para o sistema nominal. O tempo médio de execução deste exemplo foi de 3 horas e 43 minutos por simulação em um AMD Athlon 2,4 GHz com 512 MB de memória RAM.

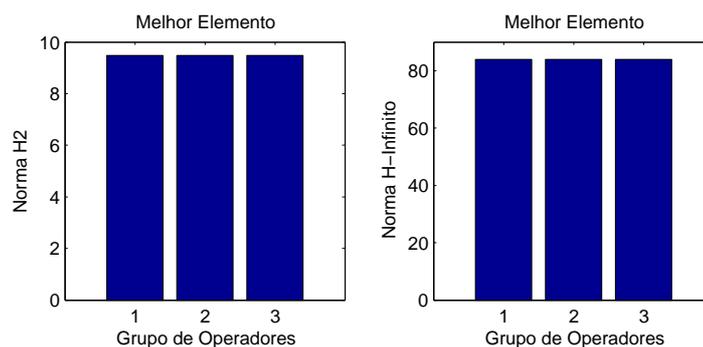


Figura 52: Comparação dos melhores resultados do exemplo 4 - caso 3.

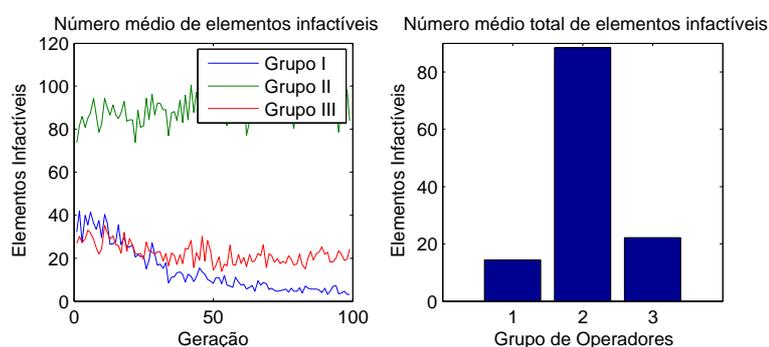


Figura 53: Número de elementos inefectivos gerados no exemplo 4 - caso 3.

6.4.4 Análise do Exemplo 4

A figura 55 mostra o número médio de elementos inefectivos gerados para este exemplo. O lado esquerdo desta figura considera a síntese de controladores por realimentação dinâmica de saída de ordem reduzida, comparando-se a síntese com e sem alocação de pólos. Observa-se que a inclusão da restrição de alocação de pólos resultou em maior número de elementos inefectivos pelos três algoritmos. Este fato é resultante da redução do espaço factível de busca, ao se inserir a restrição de alocação de pólos. O lado direito da figura 55, por sua vez, oferece a comparação do número médio de elementos inefectivos gerados entre a síntese de controladores por realimentação dinâmica de saída de ordem reduzida e completa. Este gráfico não sugere qualquer relação entre o aumento da ordem do controlador e o número de elementos inefectivos gerados. O tempo médio de execução dos algoritmos não foi influenciado pela adição da restrição de alocação de pólos. Entretanto, o aumento da ordem do controlador, de 2 para 4, resultou em um custo computacional aproximadamente 320% mais elevado. Isto indica que a inclusão de uma LMI de alocação de pólos possui menor influência no custo computacional do algoritmo do que o aumento da ordem das LMIs do problema, quando se aumenta a ordem do controlador.

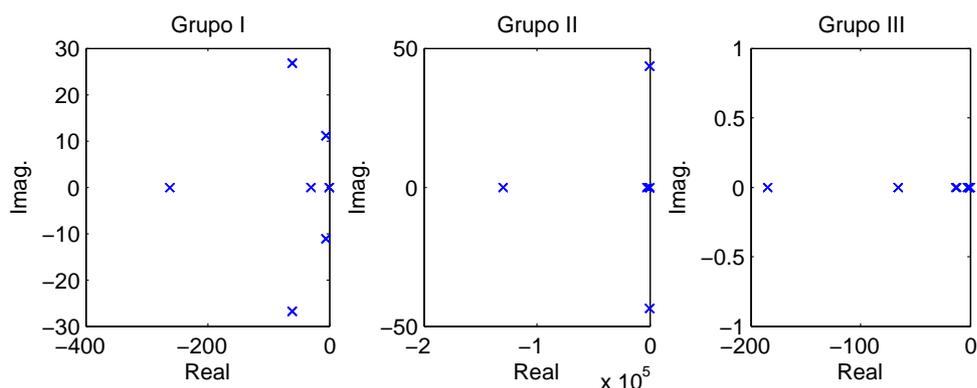


Figura 54: Pólos em malha fechada do exemplo 4 - caso 3.

Tabela 20: Controladores ótimos para o sistema nominal do exemplo 4 - caso 3.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	9,4868	9,4871
$\ T_{z_\infty w_\infty}\ _\infty$	83,9383	83,9380

6.5 Exemplo 5

Este exemplo clássico na literatura foi extraído de [4] e corresponde a um *benchmark* da teoria controle. Nenhuma incerteza foi adicionada ao sistema. A condição de suboptimalidade $\gamma \leq 1,5$ foi utilizada. As matrizes são dadas por:

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad C_y = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D_{22} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad B_{1_2} = \mathbf{I}_2, \quad B_{1_\infty} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D_{11_2} = 0, \quad D_{11_\infty} = 0,$$

$$D_{12} = 0, \quad D_{21_\infty} = 0, \quad D_{y1_\infty} = 0.$$

6.5.1 Caso 1: Realimentação Estática de Saída

Nesta subseção projetam-se controladores por realimentação estática de saída. A figura 56 apresenta a evolução dos diferentes grupos de operadores. A evolução média do

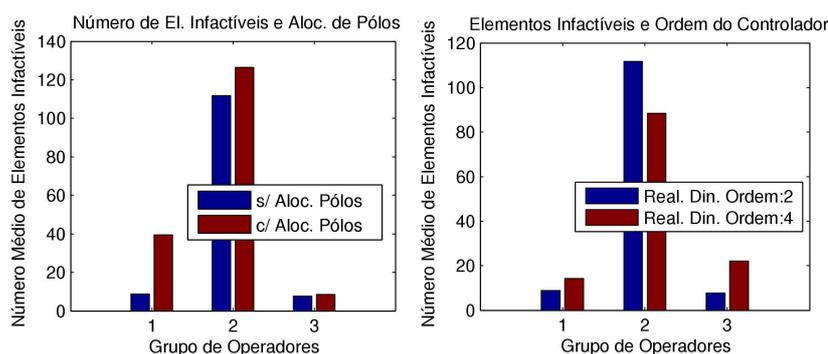


Figura 55: Número médio de elementos ineficazes gerados no exemplo 4.

grupo III de operadores apresentou desempenho ligeiramente superior aos outros dois grupos de operadores. A análise do desvio padrão médio da população indica uma mais veloz convergência da população para uma única solução atingida pelo grupo III de operadores. Este fato pode ser justificado pela utilização das informações de direção pelos algoritmos de Evolução Diferencial e algoritmo de Salomon. O grupo I de operadores, por outro lado, não foi capaz de evoluir a população para uma única solução e aumentou a diversidade da população, conforme observado não somente no gráfico do desvio padrão médio, mas também nos gráficos da média da norma \mathcal{H}_2 da população e da média do pior elemento.

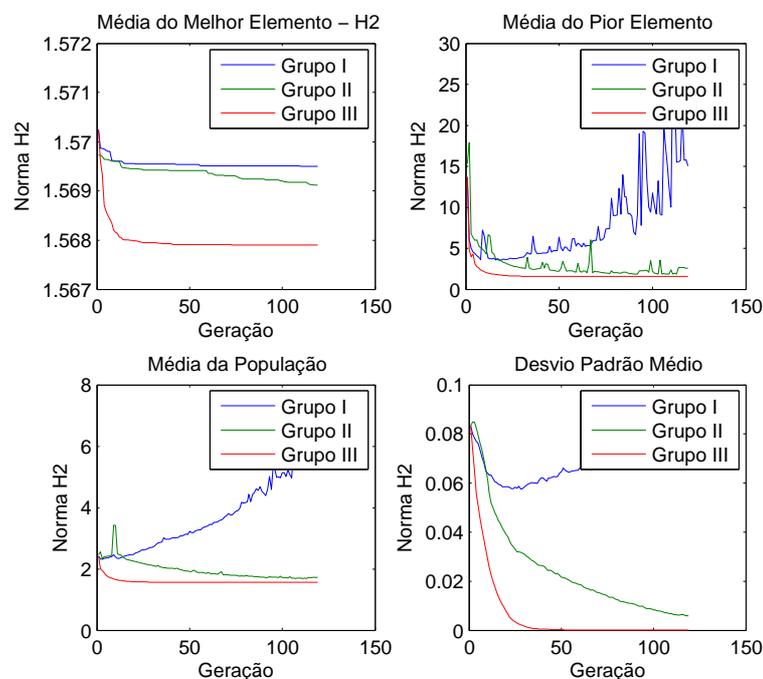


Figura 56: Evolução dos algoritmos no exemplo 5 - caso 1.

Os melhores resultados, encontrados com cada grupo de algoritmos, não indicam significativa superioridade de qualquer um dos grupos de operadores, como observado na figura 57. O melhor resultado foi encontrado com o grupo III de operadores e é dado pelo controlador

$$K = [-1, 4956].$$

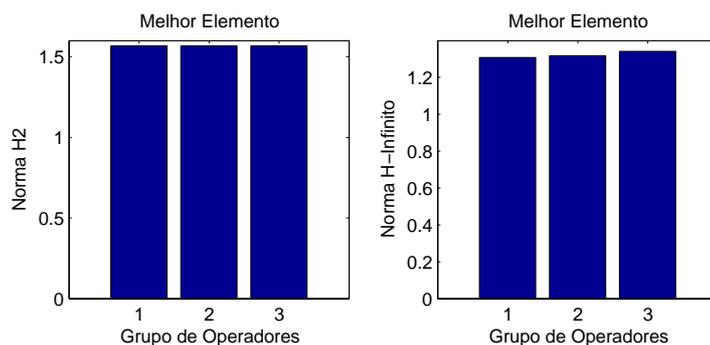


Figura 57: Comparação dos melhores resultados do exemplo 5 - caso 1.

Os pólos em malha fechada, resultantes da utilização dos melhores controladores encontrados pelos diferentes grupos de operadores são demonstrados na figura 58 e sugerem que os três grupos de operadores fizeram a população convergir para a mesma região do espaço de busca.

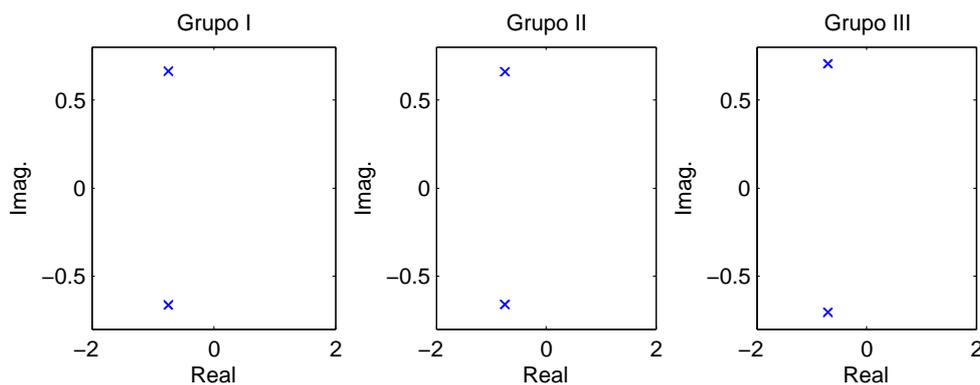


Figura 58: Pólos em malha fechada do exemplo 5 - caso 1.

A tabela 21 apresenta um resumo dos resultados calculados com o melhor cromossomo encontrado com cada grupo de operadores. Os resultados encontrados na literatura para este exemplo são mostrados na figura 22. A comparação entre as tabelas 21 e 22 sugere que os diferentes grupos de operadores proporcionam resultados satisfatórios. Observa-se

que estas três soluções formam um conjunto de soluções ótimas de Pareto e a diferença encontra-se apenas na terceira casa decimal para a norma \mathcal{H}_2 e segunda casa decimal para a norma \mathcal{H}_∞ . O tempo médio de execução deste exemplo foi de 10 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 21: Comparação dos melhores resultados do exemplo 5 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	1,5670	1,5668	1,5653
$\ T_{z_\infty w_\infty}\ _\infty$	1,3082	1,3172	1,3405

Tabela 22: Resultados encontrados na literatura para o exemplo 5 - caso 1.

	Resultados de [101]	Resultados de [4]
$\ T_{z_2 w_2}\ _2$	1,5651	1,5651
$\ T_{z_\infty w_\infty}\ _\infty$	1,0000	1,3416

6.5.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subseção sintetizaram-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 1$). A evolução dos diferentes grupos de operadores é apresentada na figura 59. Neste caso, observa-se que a convergência média do grupo III de operadores é significativamente superior à convergência média dos outros grupos.

A figura 60 mostra as normas do sistema em malha fechada com os melhores controladores encontrados com cada grupo de operadores. Neste caso, observa-se que os grupos II e III foram mais eficazes na minimização da norma \mathcal{H}_2 . O melhor resultado encontrado pelo grupo III de operadores é dado pelo controlador formado pelas matrizes:

$$A_K = [-12, 5773], \quad B_K = [0, 5319], \\ C_K = [0, 5514], \quad D_K = [-1, 4226].$$

Os pólos em malha fechada para cada um dos melhores controladores sintetizados com os três diferentes grupos de operadores estão desenhados na figura 61.

A tabela 23 apresenta um resumo dos resultados do melhor cromossomo encontrado

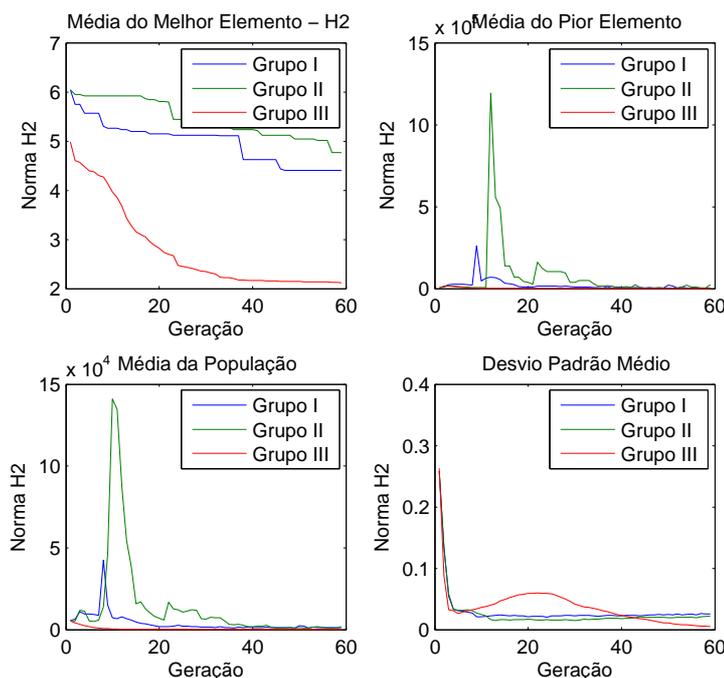


Figura 59: Evolução média dos algoritmos no exemplo 5 - caso 2.

com cada grupo de operadores. O tempo médio de execução deste exemplo foi de 9 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 23: Comparação dos melhores resultados do exemplo 5 - caso 2.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	3,4398	1,5789	1,5625
$\ T_{z_\infty w_\infty}\ _\infty$	1,0005	1,3387	1,3319

6.5.3 Caso 3: Realimentação Dinâmica de Saída de Ordem Completa

Nesta subseção projetaram-se controladores por realimentação dinâmica de saída de ordem completa ($n_c = 2$). Na figura 62 é apresentada a evolução média de cada grupo de operadores. Observa-se que, neste caso, a média da norma dos melhores elementos encontrada pelo grupo III de operadores é cerca de 45% inferior à norma média resultante da simulação dos grupos I e II de operadores.

As normas calculadas com o melhor controlador encontrado com cada grupo de operadores são mostradas na figura 63. Observa-se que o resultado encontrado com o grupo III

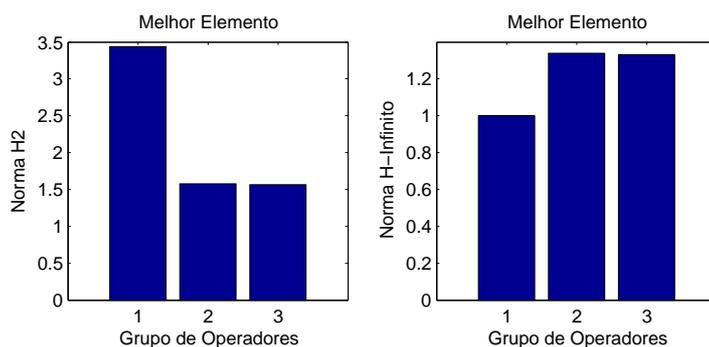


Figura 60: Comparação dos melhores resultados do exemplo 5 - caso 2.

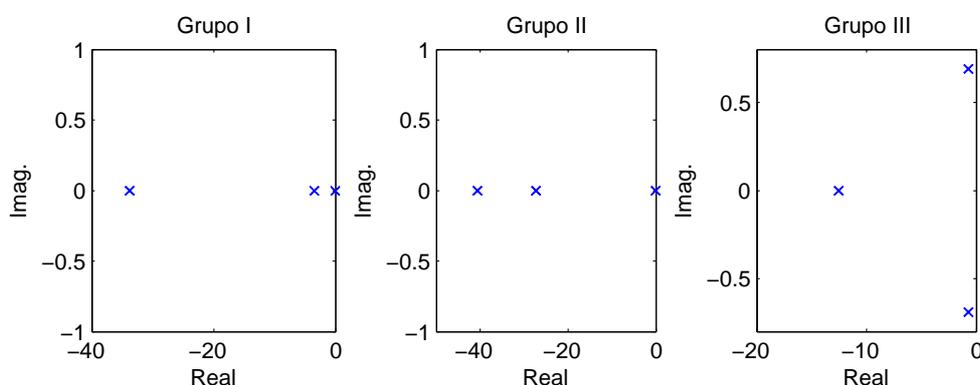


Figura 61: Pólos em malha fechada do exemplo 5 - caso 2.

de operadores foi superior aos resultados dos outros grupos, quando analisado o critério de desempenho norma \mathcal{H}_2 . O melhor controlador encontrado nas 10 simulações do grupo III de operadores é dado por:

$$A_K = \begin{bmatrix} -160,6582 & -191,2822 \\ 144,6731 & 6,2073 \end{bmatrix}, \quad B_K = \begin{bmatrix} -1,8815 \\ -28,8795 \end{bmatrix}, \\ C_K = \begin{bmatrix} -64,7502 & 46,8729 \end{bmatrix}, \quad D_K = [20,4524].$$

Na figura 64 estão desenhados os pólos em malha fechada resultantes da utilização dos melhores controladores encontrados pelos diferentes grupos de operadores.

A tabela 24 apresenta as normas calculadas com o melhor controlador encontrado com cada grupo de operadores. A tabela 25 mostra as normas \mathcal{H}_2 e \mathcal{H}_∞ ótimas calculadas para este exemplo, minimizando-se cada uma das normas em separado. A comparação das tabelas 24 e 25 indica que o grupo III de operadores foi eficaz em aproximar a solução encontrada do ótimo para o problema, sob o ponto de vista de norma \mathcal{H}_2 . O melhor resultado encontrado por este grupo de operadores pode ser atribuído à

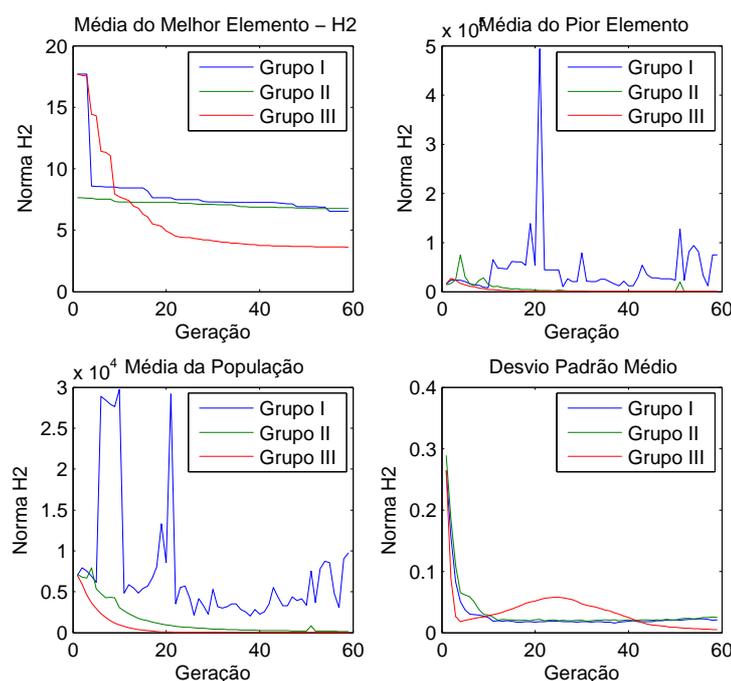


Figura 62: Evolução média dos algoritmos no exemplo 5 - caso 3.

utilização das informações de direção, fato que contribui para suas melhores características de busca local. O tempo médio de execução deste exemplo foi de 21 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 24: Comparação dos melhores resultados do exemplo 5 - caso 3.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	5,8591	3,8398	1,5762
$\ T_{z_\infty w_\infty}\ _\infty$	1,0000	1,0001	1,3281

6.5.4 Análise do Exemplo 5

Neste exemplo bastante simples, o grupo III de operadores se destacou por apresentar os melhores resultados nas três estruturas de controle estudadas. Isto sugere que a correta utilização das informações de direção podem efetivamente contribuir para a melhor convergência dos algoritmos evolutivos, conforme observado em [19]. O aumento da ordem do controlador de 1 para 2 resultou em um aumento de 133% no tempo médio de execução dos algoritmos, reforçando a relação direta entre a ordem do controlador e o custo computacional associado, também identificada em outros exemplos.

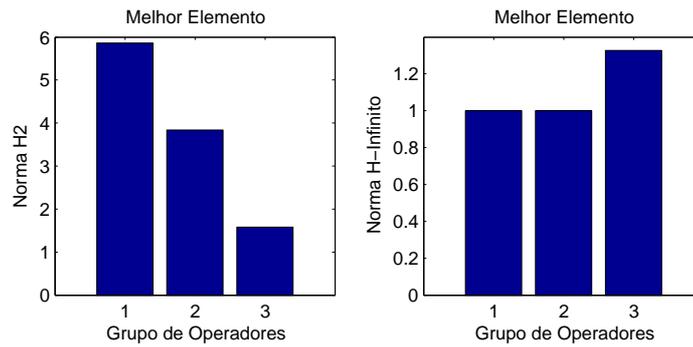


Figura 63: Comparação dos melhores resultados do exemplo 5 - caso 3.

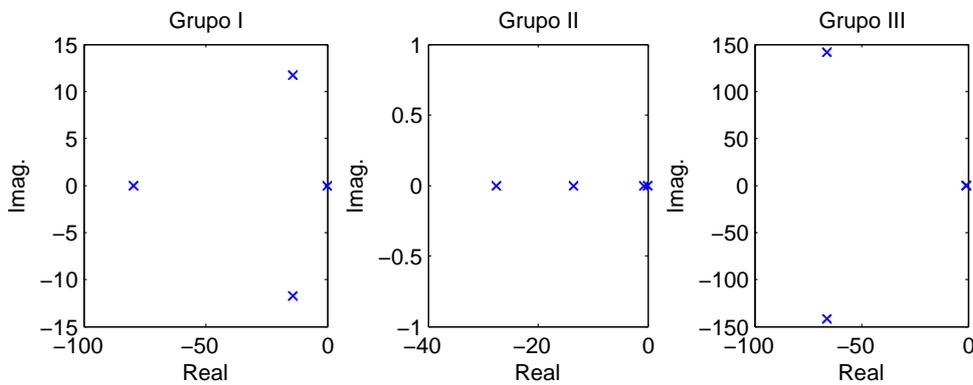


Figura 64: Pólos em malha fechada do exemplo 5 - caso 3.

6.6 Exemplo 6

Este exemplo foi extraído e adaptado da biblioteca COMPlib [82] e corresponde a uma aeronave espacial. Nenhuma incerteza foi adicionada ao problema. O limite superior de norma \mathcal{H}_∞ , $\gamma \leq 2,5$ foi utilizado. As matrizes do sistema são dadas por:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -79,285 & -0,113 & 0 \\ 28,564 & 0,041 & 0 \end{bmatrix},$$

Tabela 25: Controladores ótimos para o exemplo 5 - caso 3.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	1,5392	144,1486
$\ T_{z_\infty w_\infty}\ _\infty$	1,2346	1,0000

$$\begin{aligned}
B_2 &= \begin{bmatrix} 0 & 0 \\ 0,041 & -0,0047 \\ -0,03 & -0,0016 \end{bmatrix}, & C_2 &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & C_y &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \\
B_{1_2} &= \begin{bmatrix} 0 \\ 0,041 \\ -0,03 \end{bmatrix}, & B_{1_\infty} &= \begin{bmatrix} 0 \\ 0,025 \\ -0,01 \end{bmatrix}, & D_{22} &= \begin{bmatrix} 0 & 0 \\ 0,3 & 0 \\ 0 & 0,3 \end{bmatrix}, \\
C_1 &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & D_{12} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0,1 & 0 \\ 0 & 0,1 \end{bmatrix}, & D_{11_2} &= 0, \\
D_{11_\infty} &= 0, & D_{21_\infty} &= 0, & D_{y1_\infty} &= 0.
\end{aligned}$$

6.6.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subseção projetam-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 2$). A evolução média da população, resultante da aplicação dos diferentes grupos de operadores, é mostrada na figura 65. Pode-se observar que a convergência média grupo III de operadores foi ligeiramente superior à convergência dos outros grupos de operadores.

Na figura 66, os melhores resultados encontrados com os três grupos de operadores são apresentados. Os três grupos produziram resultados próximos, com ligeira superioridade do grupo III. O melhor controlador, resultante da aplicação do grupo III de operadores é dado por:

$$\begin{aligned}
A_K &= \begin{bmatrix} -0,3260 & 0,1839 \\ 0,2019 & -0,3333 \end{bmatrix}, & B_K &= \begin{bmatrix} -0,0411 & 18,8942 \\ -0,0089 & 21,8830 \end{bmatrix}, \\
C_K &= \begin{bmatrix} -1,3022 & -1,4088 \\ -0,4885 & -0,2296 \end{bmatrix}, & D_K &= \begin{bmatrix} 2,6610 & -4,0551 \\ 0,2580 & -0,1072 \end{bmatrix}.
\end{aligned}$$

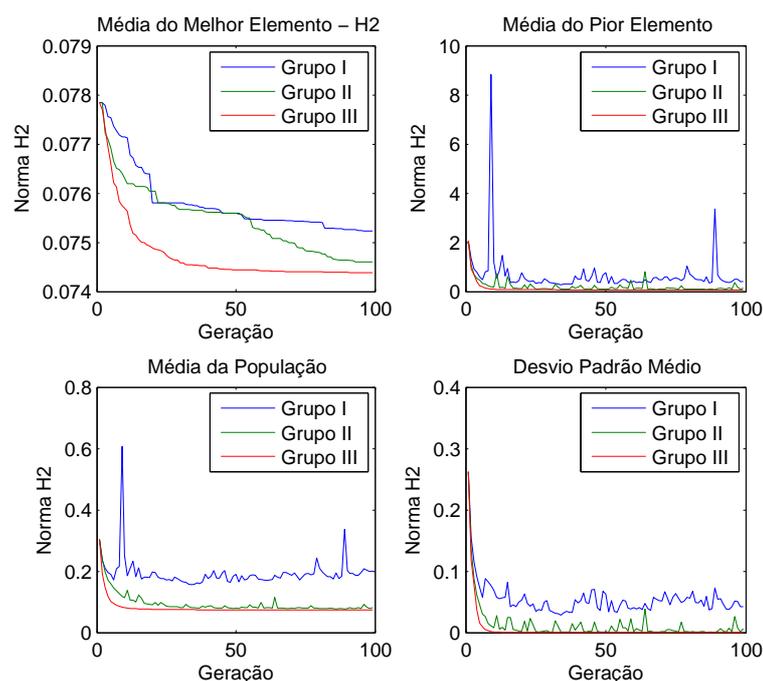


Figura 65: Evolução média dos algoritmos do exemplo 6 - caso 1.

O número médio de elementos inactivos, gerado pelos diferentes grupos de operadores, é apresentado na figura 67. O grupo II de operadores foi o menos eficiente na geração de elementos factíveis, fato observado em todas as simulações efetuadas. As características dos operadores de mutação deste grupo de operadores são as causas para esta elevada taxa de geração de elementos inactivos.

Os pólos em malha fechada, obtidos com a utilização dos melhores controladores sintetizados com os diferentes grupos de operadores, estão desenhados na figura 68.

As normas dos melhores cromossomos, encontrados pelos grupos de operadores, estão mostradas na tabela 26. Observa-se que o conjunto de melhores controladores sintetizados forma um conjunto ótimo de Pareto. O tempo médio de execução deste exemplo foi de 28 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 26: Comparação dos melhores resultados do exemplo 6 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	0,0761	0,0752	0,0747
$\ T_{z_\infty w_\infty}\ _\infty$	0,0662	0,0750	0,0821

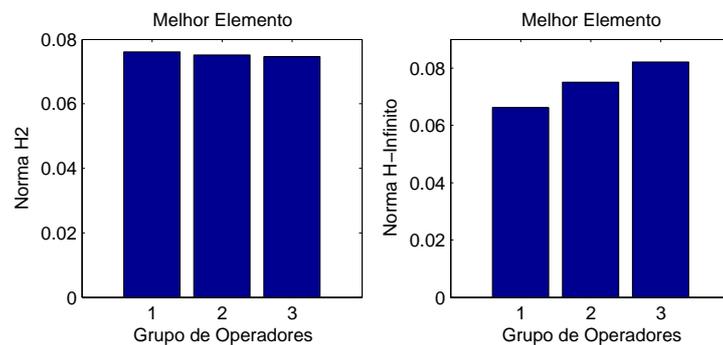


Figura 66: Comparação dos melhores resultados do exemplo 6 - caso 1.

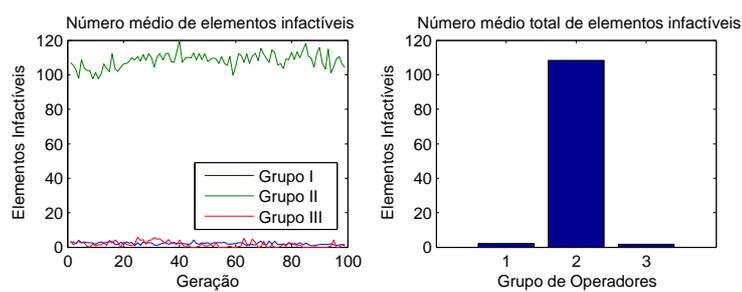


Figura 67: Número de elementos inactivos gerado no exemplo 6 - caso 1.

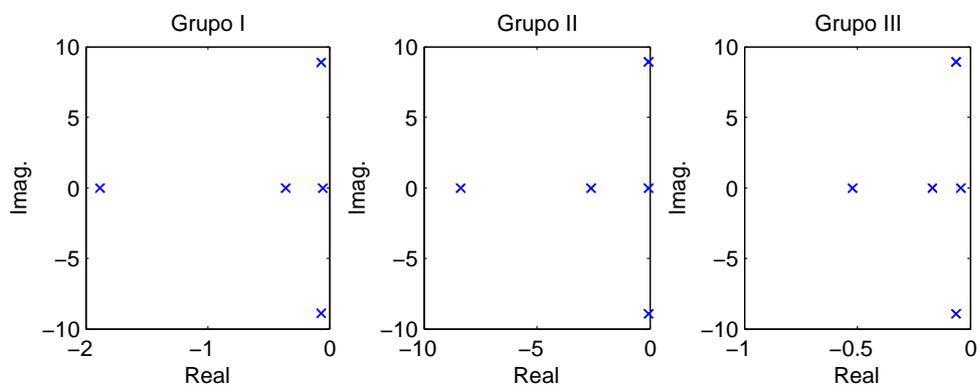


Figura 68: Pólos em malha fechada do exemplo 6 - caso 1.

6.7 Exemplo 7

Este exemplo foi extraído e adaptado da biblioteca COMLib [82] e corresponde à mesma aeronave espacial do exemplo 6.6. A condição de sub-otimalidade $\gamma \leq 50$ foi utilizada e um parâmetro incerto foi adicionado à matriz dinâmica do sistema, dada por:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -79,285 & -0,113 & 0 \\ a_1 & 0,041 & 0 \end{bmatrix},$$

onde $a_1 \in [26,0640; 31,0640]$ é o parâmetro incerto do sistema e caracteriza o domínio de incertezas \mathcal{D}_A , correspondente a um poliedro convexo com dois vértices. O restante das matrizes do sistema é dado por:

$$\begin{aligned} B_2 &= \begin{bmatrix} 0 & 0 \\ 0,041 & -0,0047 \\ -0,03 & -0,0016 \end{bmatrix}, & C_2 &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & C_y &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \\ B_{1_2} &= \begin{bmatrix} 0 \\ 0,041 \\ -0,03 \end{bmatrix}, & B_{1_\infty} &= \begin{bmatrix} 0 \\ 0,025 \\ -0,01 \end{bmatrix}, & D_{22} &= \begin{bmatrix} 0 & 0 \\ 0,3 & 0 \\ 0 & 0,3 \end{bmatrix}, \\ C_1 &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & D_{12} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0,1 & 0 \\ 0 & 0,1 \end{bmatrix}, & D_{11_2} &= 0, \\ D_{11_\infty} &= 0, & D_{21_\infty} &= 0, & D_{y1_\infty} &= 0. \end{aligned}$$

6.7.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subseção são projetados controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 2$). A convergência média dos diferentes grupos de operadores é apresentada na figura 69. Neste exemplo, observa-se que o grupo I apresentou um desempenho bastante superior aos outros dois grupos de algoritmos, quando analisada a norma H_2 média do melhor elemento da população.

A figura 70 demonstra os resultados encontrados com o melhor cromossomo encontrado pelos diferentes grupos de operadores. Observa-se a significativa diferença percentual entre os diferentes algoritmos com expressiva superioridade para o grupo I de

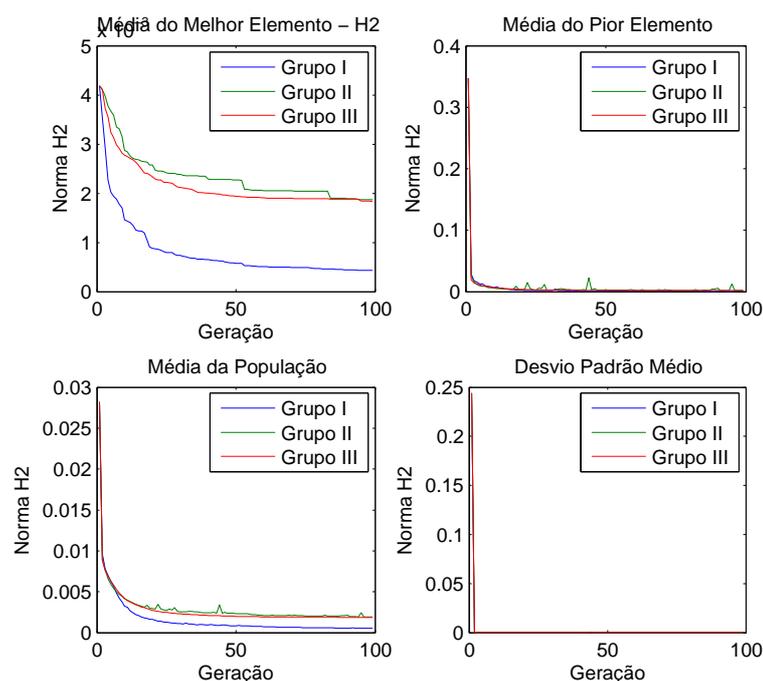


Figura 69: Evolução média dos algoritmos no exemplo 7 - caso 1.

operadores. O melhor controlador, gerado pelo grupo I de operadores, é dado por:

$$\begin{aligned}
 A_K &= \begin{bmatrix} -628,2472 & 1599,5606 \\ 90,1117 & -942,9974 \end{bmatrix}, & B_K &= \begin{bmatrix} 2822,2796 & -1654,5685 \\ 209,7138 & -260,9279 \end{bmatrix}, \\
 C_K &= \begin{bmatrix} 2430,9645 & 1447,6146 \\ 17,7602 & -1475,4351 \end{bmatrix}, & D_K &= \begin{bmatrix} 1152,8241 & -1764,0699 \\ 2590,0837 & 57,2695 \end{bmatrix}.
 \end{aligned}$$

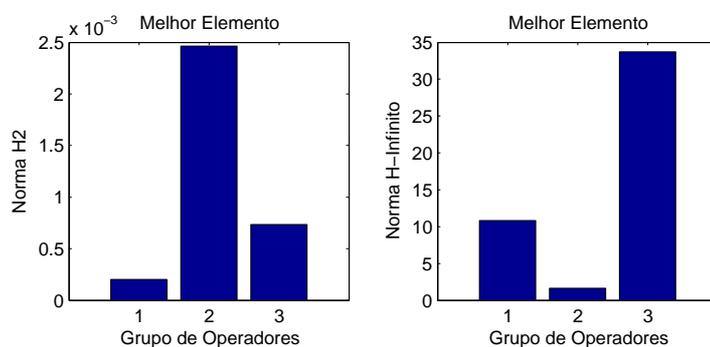


Figura 70: Comparação dos melhores resultados do exemplo 7 - caso 1.

O número de elementos ineficazes gerados pela aplicação dos diferentes grupos de operadores é apresentado na figura 71. Em comparação com o caso precisamente conhecido,

demonstrado na figura 67, observa-se um aumento percentual relativamente expressivo no número de elementos ineficazes gerados: 67%, 6% e 53%, para os grupos I, II e III, respectivamente. Este fato sugere que a adição do parâmetro incerto reduz o espaço de busca factível do problema, aumentando o número de indivíduos ineficazes gerado.

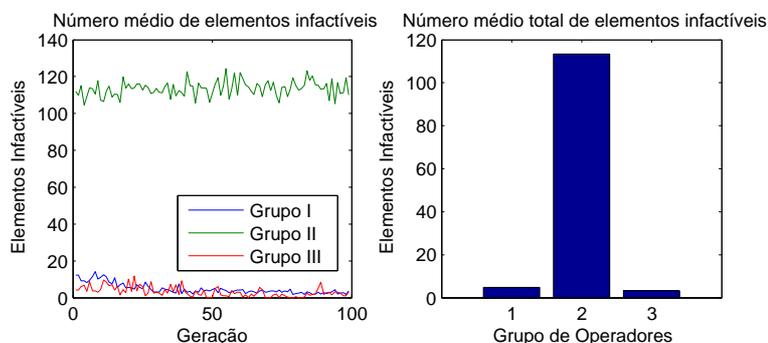


Figura 71: Número de elementos ineficazes gerados no exemplo 7 - caso 1.

Os pólos em malha fechada, gerados utilizando-se o melhor controlador encontrado com cada grupo de operadores, são mostrados na figura 72. Estas nuvens diferem significativamente daquelas desenhadas na figura 68, diferenças atribuídas à inserção do parâmetro incerto.

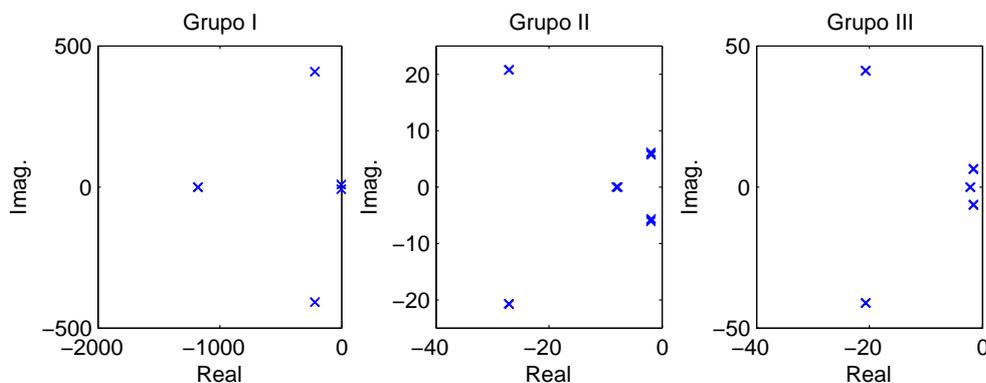


Figura 72: Pólos em malha fechada do exemplo 7 - caso 1.

A tabela 27 mostra os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ calculados com o melhor controlador encontrado com cada grupo de operadores. O tempo médio de execução deste exemplo foi de 56 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 27: Comparação dos melhores resultados do exemplo 7 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2} \ _2$	$203 \cdot 10^{-6}$	$2466 \cdot 10^{-6}$	$737 \cdot 10^{-6}$
$\ T_{z_\infty w_\infty} \ _\infty$	10,8195	1,6652	33,7018

6.7.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Reduzida com Alocação de Pólos

Nesta seção, adicionou-se uma região de alocação de pólos formada por um cone com ângulo interno $\theta = 120^\circ$ e sintetizaram-se controladores por realimentação dinâmica de saída de ordem reduzida. A convergência média gerada pelos diferentes grupos de operadores é apresentada na figura 73. O desempenho dos três grupos de operadores é bastante próximo, quando observada a norma \mathcal{H}_2 média do melhor elemento das populações.

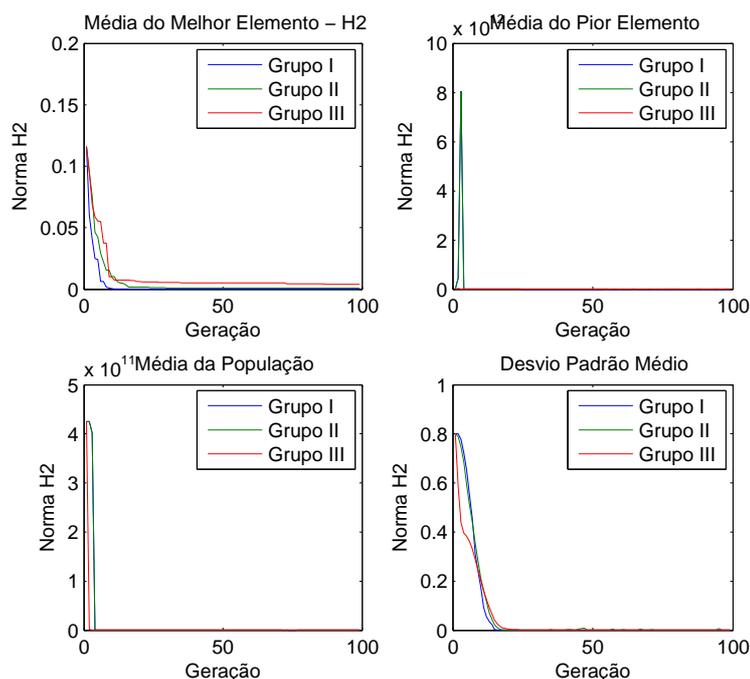


Figura 73: Evolução média dos algoritmos no exemplo 7 - caso 2.

A figura 74 mostra os resultados provenientes do cálculo das normas \mathcal{H}_2 e \mathcal{H}_∞ em malha fechada com o uso do melhor controlador encontrado com cada grupo de operadores. Observa-se a significativa diferença percentual dos resultados dos algoritmos, com expressiva vantagem para o grupo I de operadores, cujo melhor controlador sintetizado é

dado por:

$$A_K = 10^3 \begin{bmatrix} -27271,3677 & -233,7952 \\ -17992,1612 & -552,2719 \end{bmatrix}, \quad B_K = 10^3 \begin{bmatrix} -5,7355 & -12148,0535 \\ -6,6729 & -28,6539 \end{bmatrix},$$

$$C_K = 10^3 \begin{bmatrix} 923,5665 & -16800,1286 \\ -5470,5589 & 246,5398 \end{bmatrix}, \quad D_K = 10^3 \begin{bmatrix} 274,6638 & -23791,3110 \\ 2,3540 & 93,3234 \end{bmatrix}.$$

Este comportamento é semelhante ao comportamento encontrado na subseção anterior.

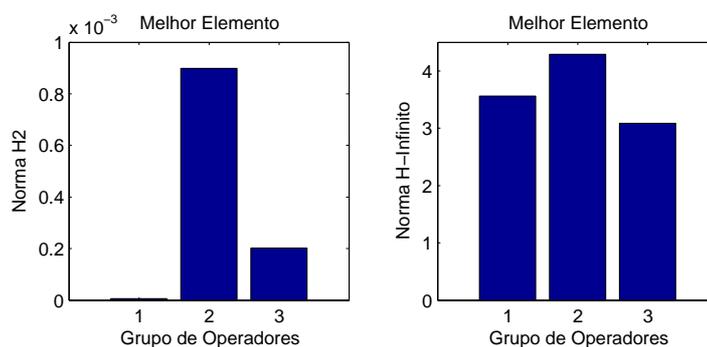


Figura 74: Comparação dos melhores resultados do exemplo 7 - caso 2.

O número de elementos ineficazes gerados pela aplicação dos diferentes grupos de operadores é apresentado na figura 75. Os três grupos de operadores obtiveram um sensível incremento percentual no número de elementos ineficazes gerados, em comparação com o resultado da subseção anterior, apresentado na figura 71. Os grupos I, II e III resultaram em uma média 263%, 13% e 160% maior no número de elementos ineficazes, respectivamente. Este aumento é resultado da adição da restrição de alocação de pólos ao exemplo, que reduz o espaço de busca factível do problema.

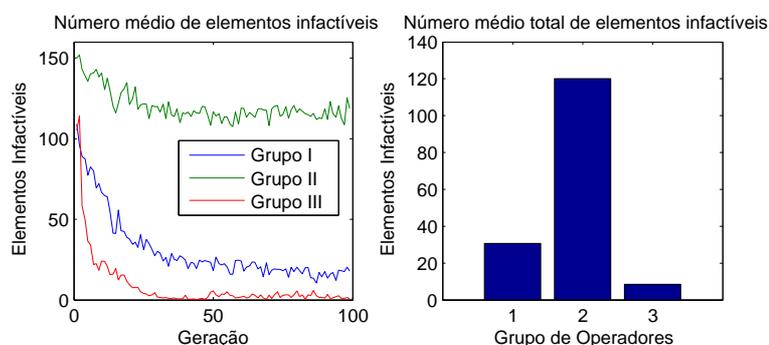


Figura 75: Número médio de elementos ineficazes gerados no exemplo 7 - caso 2.

Os pólos em malha fechada e a restrição de alocação de pólos estão desenhados na figura 76. O detalhe da figura 76, mostrado na figura 77, indica que a restrição de alocação de pólos foi respeitada.

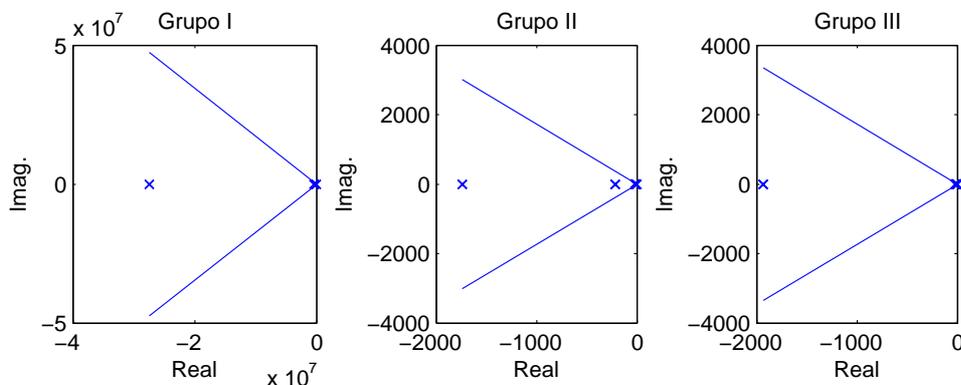


Figura 76: Pólos em malha fechada do exemplo 7 - caso 2.

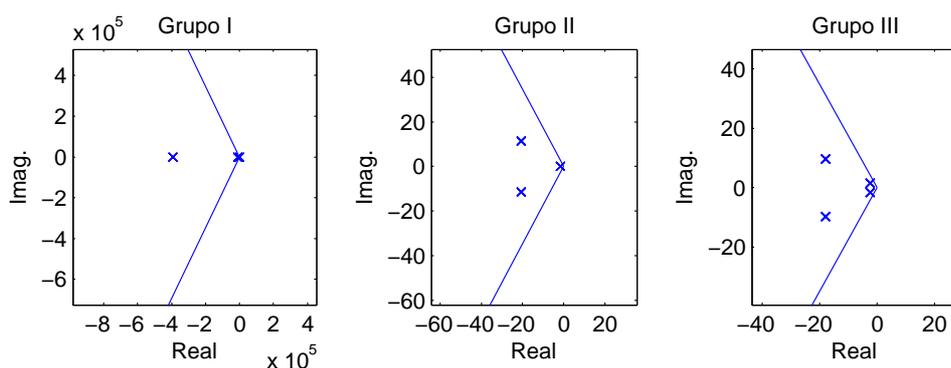


Figura 77: Detalhe dos pólos em malha fechada do exemplo 7 - caso 2.

A tabela 28 mostra os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ calculados utilizando-se o melhor controlador sintetizado com cada grupo de operadores. O tempo médio de execução deste exemplo foi de 1 hora e 2 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 28: Comparação dos melhores resultados do exemplo 7 - caso 2.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	$6 \cdot 10^{-6}$	$899 \cdot 10^{-6}$	$201 \cdot 10^{-6}$
$\ T_{z_\infty w_\infty}\ _\infty$	3,5630	4,2876	3,0869

6.7.3 Caso 3: Realimentação Dinâmica de Saída de Ordem Reduzida com Alocação de Pólos e População de 40 Elementos

Nesta subseção projetaram-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 2$). Manteve-se a restrição de alocação de pólos, descrita na subseção anterior e dada por um cone com ângulo interno $\theta = 120^\circ$. Além disso, nesta subseção utilizou-se como parâmetro para o algoritmo híbrido uma população de 40 elementos. Lembra-se que a população utilizada nos outros exemplos possui 20 cromossomos. A convergência média dos diferentes grupos de operadores é apresentada na figura 78. O desempenho dos três grupos de operadores é bastante próximo, quando observada a média do melhor elemento das populações.

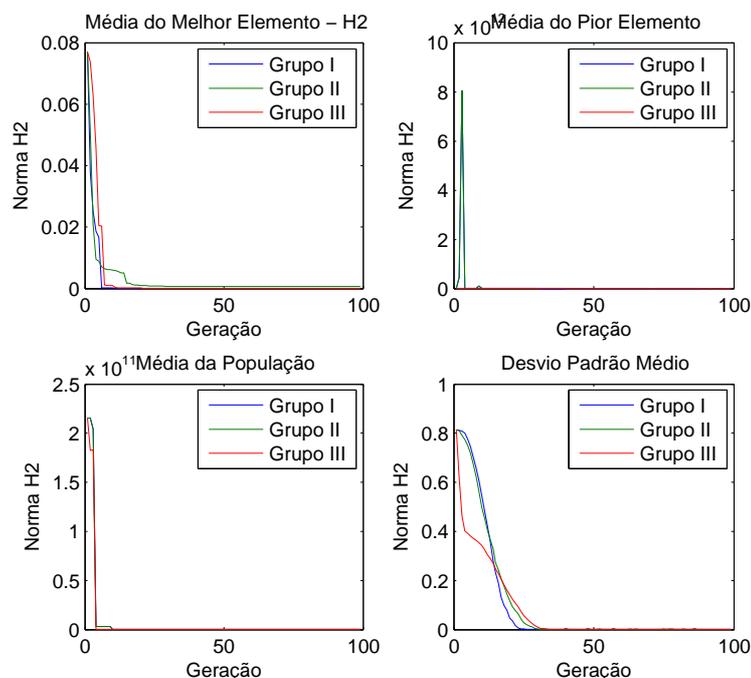


Figura 78: Evolução média dos algoritmos no exemplo 7 - caso 3.

A figura 79 demonstra os melhores resultados encontrados com cada grupo de operadores. Observa-se a significativa diferença percentual entre os diferentes algoritmos com expressiva vantagem para o grupo I de operadores, mantendo-se o comportamento identificado na subseção anterior. O controlador sintetizado pelo grupo I de operadores, cujo

resultado encontra-se na figura 79, é dado pelas matrizes:

$$A_K = 10^6 \begin{bmatrix} -407,8766 & -125,9580 \\ -0,0262 & -0,0246 \end{bmatrix}, \quad B_K = 10^6 \begin{bmatrix} -777,8410 & -38,1309 \\ -0,0106 & 4,5734 \end{bmatrix},$$

$$C_K = 10^6 \begin{bmatrix} 3,5869 & 3,3918 \\ -778,9034 & -219,5900 \end{bmatrix}, \quad D_K = 10^6 \begin{bmatrix} 1,2453 & -682,5888 \\ -108,8031 & -8,2057 \end{bmatrix}.$$

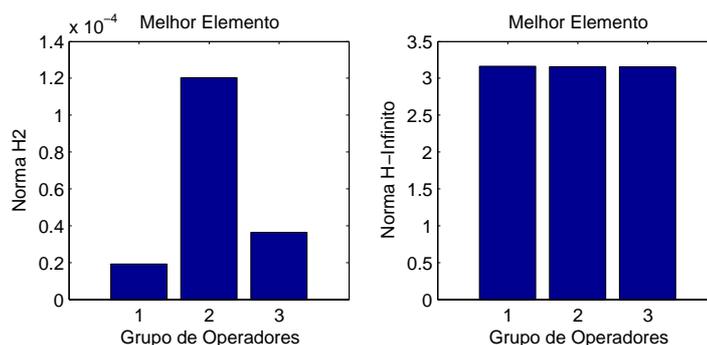


Figura 79: Comparação dos melhores resultados do exemplo 7 - caso 3.

O número de elementos ineficazes gerados pela aplicação dos diferentes grupos de operadores é apresentado na figura 80. Os três grupos de operadores obtiveram um incremento percentual significativo no número de elementos ineficazes gerados, em comparação com o resultado apresentado na figura 75, relativo a uma população de 20 cromossomos. Os grupos de operadores I, II e III geraram, respectivamente, 62%, 104% e 92% de elementos ineficazes a mais do que na subseção anterior. Esta diferença é resultado direto do aumento do tamanho população.

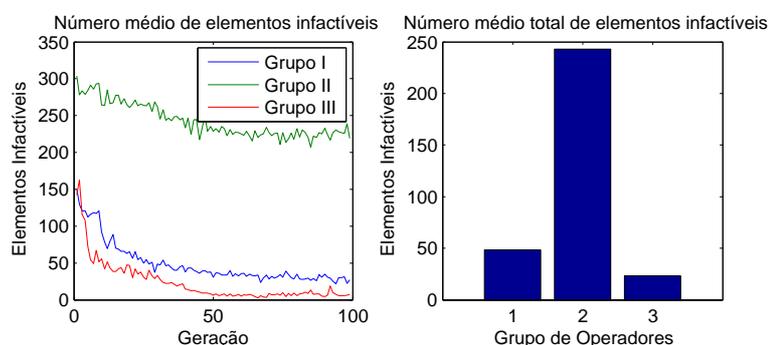


Figura 80: Número médio de elementos ineficazes gerados no exemplo 7 - caso 3.

Os pólos em malha fechada e a restrição de alocação de pólos são mostradas na figura 81. O detalhe da figura 81, exibido na figura 82, indica que a restrição de alocação de

pólos foi respeitada.

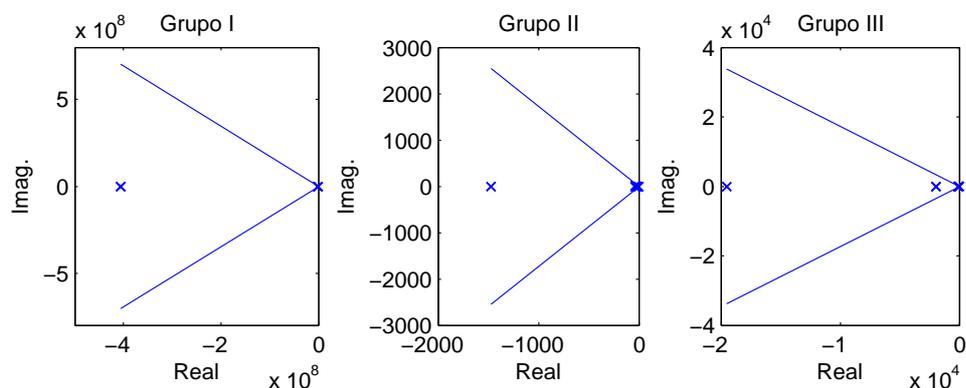


Figura 81: Pólos em malha fechada do exemplo 7 - caso 3.

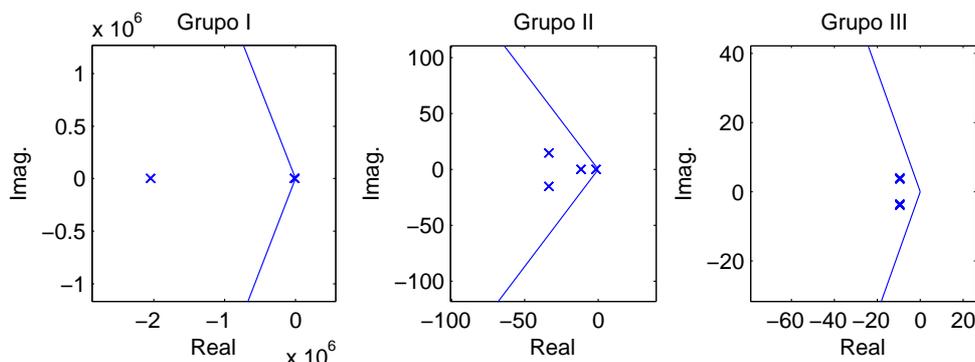


Figura 82: Detalhe dos pólos em malha fechada do exemplo 7 - caso 3.

O melhor resultado encontrado com a utilização do melhor controlador gerado com cada grupo de operadores é apresentado na tabela 29. Ressalta-se que, apesar de elevada diferença percentual entre os resultados da norma \mathcal{H}_2 , as diferenças ocorrem apenas na quinta casa decimal. O tempo médio de execução deste exemplo foi de 2 horas e 16 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 29: Comparação dos melhores resultados do exemplo 7 - caso 3.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	$19 \cdot 10^{-6}$	$120 \cdot 10^{-6}$	$36 \cdot 10^{-6}$
$\ T_{z_\infty w_\infty}\ _\infty$	3,1619	3,1571	3,1484

6.7.4 Análise do Exemplo 7

A figura 83 mostra o número médio de elementos ineficazes gerados para cada subseção deste exemplo: síntese sem alocação de pólos; síntese com alocação de pólos e; síntese com alocação de pólos e população de 40 elementos. Tanto a adição da restrição de alocação de pólos, quanto o aumento do tamanho da população influenciaram diretamente o número de elementos ineficazes gerados pelos algoritmos.

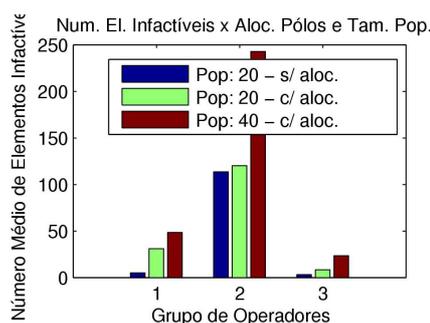


Figura 83: Número de elementos ineficazes gerado no exemplo 7.

Com relação ao custo computacional, o tempo médio de execução dos algoritmos para este sistema aumentou de 28 minutos para o sistema nominal (exemplo 6.6) para 56 minutos para o sistema incerto (exemplo 6.7). A adição da restrição de alocação de pólos, entretanto, não resultou em uma diferença significativa no custo computacional, que se elevou de 56 para 62 minutos. E, por fim, o aumento do número de elementos na população praticamente dobrou o tempo de execução médio dos algoritmos. Estes dados sugerem que a inclusão de um parâmetro incerto, aumentando-se o número de vértices do poliedro e conseqüentemente o número de LMIs consideradas, possui influência direta no custo computacional dos algoritmos. Por outro lado, a simples adição de uma LMI de alocação de pólos não aumenta significativamente o custo associado. A população maior, contudo, também representou um aumento significativo no custo computacional médio dos algoritmos. Esta relação é direta por conta do maior número de cálculos necessários para se evoluir uma população com mais cromossomos.

6.8 Exemplo 8

Este exemplo foi extraído e adaptado de [67]. Um parâmetro incerto foi adicionado à matriz dinâmica do sistema, dada por:

$$A = \begin{bmatrix} -0,1122 & -1,5614 & 1,3569 & 1,3079 \\ -0,2794 & 1,3766 & 1,0666 & 0,4676 \\ -0,6081 & a_1 & -1,6049 & 1,3048 \\ 0,7840 & -0,0367 & 0,2163 & 0,0915 \end{bmatrix},$$

onde o conjunto $a_1 \in [0,1581; 0,6581]$ forma o domínio de incertezas, \mathcal{D}_A , do problema.

O restante das matrizes do sistema é dado por:

$$\begin{aligned} B_2 &= \begin{bmatrix} 1,1355 & 0,3258 \\ -0,8003 & -0,1583 \\ 0,1339 & -0,0169 \\ 0,7579 & 0,5475 \end{bmatrix}, & C_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0,2792 & 0,3457 & 0 & 0 \\ 1,3730 & 1,1088 & 0 & 0 \end{bmatrix}, & D_{22} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \\ B_{1_2} &= \begin{bmatrix} 0,2168 & 0,6674 \\ 0,6518 & 0,3109 \\ 0,0528 & 0,3066 \\ 0,2293 & 0,7207 \end{bmatrix}, & B_{1_\infty} &= \begin{bmatrix} 0,9544 & 0,1662 \\ 0,1311 & 0,9114 \\ 0,0683 & 0,1363 \\ 0,1252 & 0,6170 \end{bmatrix}, & D_{11_2} &= 0,, \\ C_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0,2959 & 0,5296 & 0 & 0 \\ 1,5011 & 0,8386 & 0 & 0 \end{bmatrix}, & C_y &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, & D_{12} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \\ D_{11_\infty} &= 0, & D_{21_\infty} &= 0, & D_{y1_\infty} &= 0. \end{aligned}$$

6.8.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Completa e $\gamma \leq 100$

Nesta seção, simularam-se os algoritmos na síntese de controladores por realimentação dinâmica de saída de ordem completa ($n_c = 4$). A condição de sub-otimalidade de norma \mathcal{H}_∞ , $\gamma \leq 100$, foi utilizada. A figura 84 apresenta a evolução média dos diferentes grupos de operadores. Os grupos II e III apresentaram evoluções médias bastante próximas, quando analisada a norma \mathcal{H}_2 média do melhor elemento. Entretanto, o grupo I de operadores demonstrou convergência média lenta, em comparação com os outros dois grupos de operadores. A análise do desvio padrão médio, apresentado na figura 84, sugere

que o algoritmo mais eficaz na convergência da população para uma única solução foi o grupo III de operadores. Por outro lado, sugere-se que esta convergência do grupo III foi prematura, evoluindo toda a população para uma solução que ainda pode ser melhorada. Este fato não ocorreu com o grupo II de operadores.

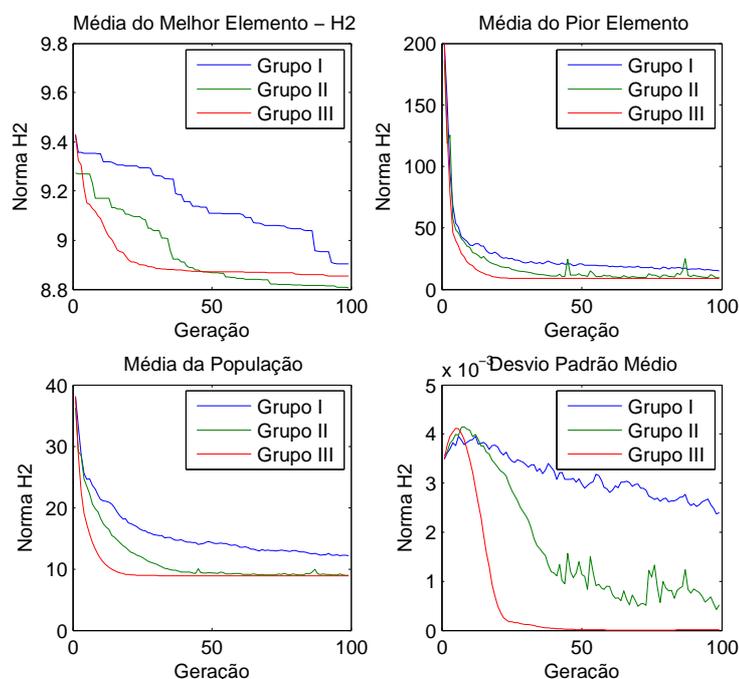


Figura 84: Evolução média dos algoritmos no exemplo 8 - caso 1.

Apesar de apresentar a melhor taxa de convergência entre os diferentes grupos de operadores, do ponto de vista da média da norma \mathcal{H}_2 do melhor elemento, o grupo III de operadores resultou no pior cromossomo encontrado nas 10 simulações, como pode ser observado na figura 85. O melhor controlador encontrado entre os diferentes algoritmos é dado por:

$$A_K = 10^6 \begin{bmatrix} -57,8590 & -112,1373 & -85,8222 & -237,4672 \\ 100,0415 & 197,4800 & 151,2491 & 419,9646 \\ -92,7811 & -184,1292 & -141,1190 & -392,2666 \\ -9,0889 & -17,8122 & -13,6389 & -37,8187 \end{bmatrix},$$

$$B_K = 10^6 \begin{bmatrix} -39,4463 & 14,5397 \\ 71,8649 & -28,1076 \\ -67,7282 & -26,9454 \\ -6,3986 & 2,4476 \end{bmatrix},$$

$$C_K = 10^3 \begin{bmatrix} -2,6537 & -5,1570 & -3,9450 & -10,9220 \\ 8,2518 & 16,0093 & 12,2474 & 33,9005 \end{bmatrix},$$

$$D_K = 10^3 \begin{bmatrix} -1,8183 & 0,6784 \\ 5,6089 & -2,0783 \end{bmatrix},$$

e foi obtido com o grupo II de operadores. A figura 86 mostra o número médio de

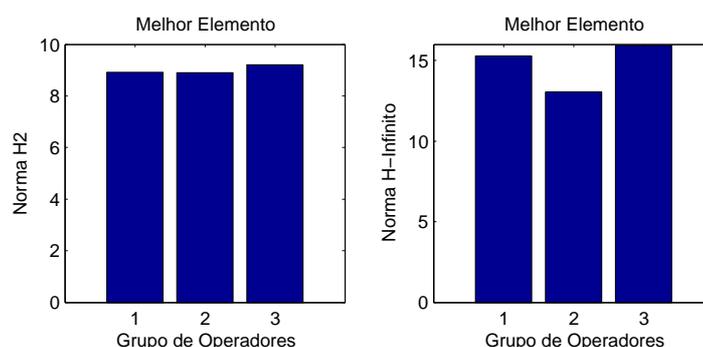


Figura 85: Comparação dos melhores resultados do exemplo 8 - caso 1.

elementos ineficazes gerados pelos diferentes grupo de operadores. O elevado número de elementos ineficazes gerados pelo grupo II pode ser, mais uma vez, observado neste exemplo. A causa deste fato é a estrutura dos algoritmos de mutação do grupo II de operadores, responsáveis por 76% dos elementos ineficazes gerados com este grupo neste exemplo.

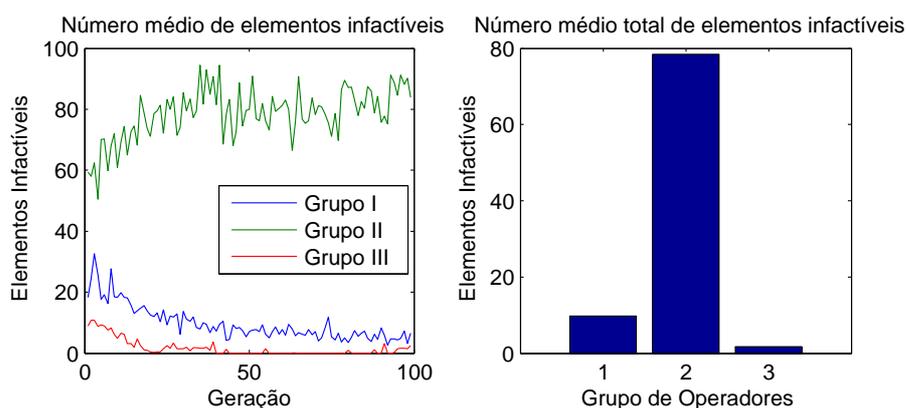


Figura 86: Número médio de elementos ineficazes gerados no exemplo 8 - caso 1.

As nuvens de pólos em malha fechada, desenhadas utilizando-se o melhor cromossomo encontrado com cada grupo de operadores, são apresentadas na figura 87.

A tabela 30 apresenta os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ , calculadas em malha fechada com o melhor controlador encontrado com cada grupo de operadores. O melhor resultado,

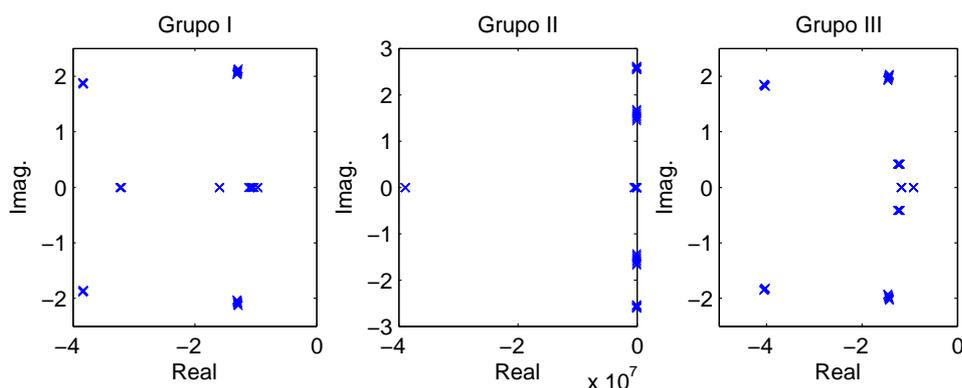


Figura 87: Pólos em malha fechada do exemplo 8 - caso 1.

encontrado com o grupo II de operadores, é dado pelo controlador:

$$A_K = 10^6 \begin{bmatrix} -57,8590 & -112,1373 & -85,8222 & -237,4672 \\ 100,0415 & 197,4800 & 151,2491 & 419,9646 \\ -92,7811 & -184,1292 & -141,1190 & -392,2666 \\ -9,0889 & -17,8122 & -13,6389 & -37,8187 \end{bmatrix},$$

$$B_K = 10^6 \begin{bmatrix} -39,4463 & 14,5397 \\ 71,8649 & -28,1076 \\ -67,7282 & -26,9454 \\ -6,3986 & 2,4476 \end{bmatrix},$$

$$C_K = 10^3 \begin{bmatrix} -2,6537 & -5,1570 & -3,9450 & -10,9220 \\ 8,2518 & 16,0093 & 12,2474 & 33,9005 \end{bmatrix},$$

$$D_K = 10^3 \begin{bmatrix} -1,8183 & 0,6784 \\ 5,6089 & -2,0783 \end{bmatrix}.$$

A tabela 31 mostra os resultados ótimos encontrados para o modelo nominal da planta, no qual o parâmetro incerto assume o valor $a_1 = 0,4081$. A comparação entre os resultados das tabelas 30 e 31 indicam um bom desempenho do algoritmo, considerando-se que o problema possui um parâmetro incerto. O tempo médio de execução deste exemplo foi de 2 horas e 27 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 30: Comparação dos melhores resultados do exemplo 8 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	8,9124	8,9043	9,2049
$\ T_{z_\infty w_\infty}\ _\infty$	15,2713	13,0560	15,9294

Tabela 31: Controladores ótimos para o sistema nominal do exemplo 8 - caso 1.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	7,8553	148,0799
$\ T_{z_\infty w_\infty}\ _\infty$	12,9402	5,1088

6.8.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Completa e $\gamma \leq 12,5$

Nesta seção, considerou-se o problema de síntese de controladores por realimentação dinâmica de saída de ordem completa ($n_c = 4$), porém restringiu-se o espaço de busca factível considerando-se $\gamma \leq 12,5$. A evolução média dos três grupos de operadores é mostrada na figura 88. Neste caso, o comportamento médio de convergência dos três algoritmos foi similar.

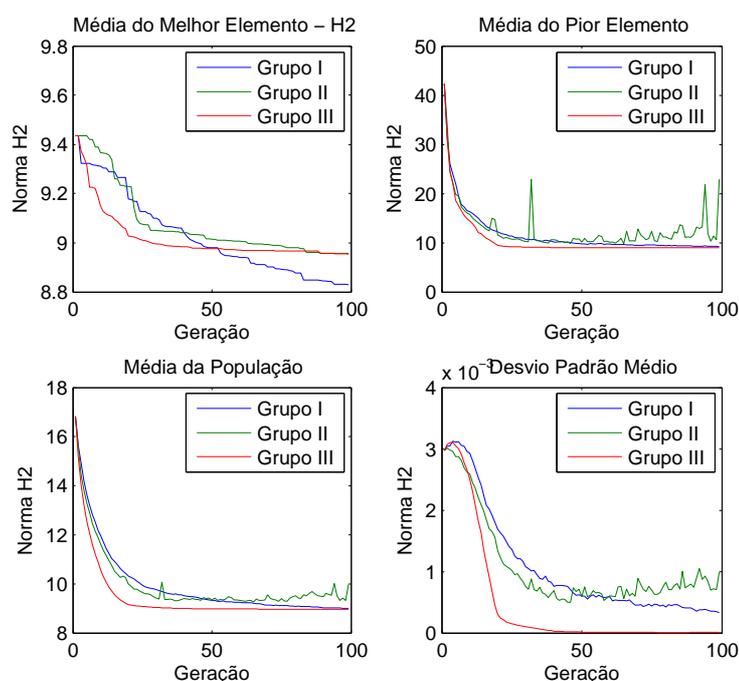


Figura 88: Evolução média dos algoritmos no exemplo 8 - caso 2.

A figura 89 apresenta os melhores elementos encontrados com os diferentes grupos de operadores. Os resultados encontrados são bastante próximos, considerando-se ambos critérios de desempenho. O grupo II de operadores, porém, resultou na melhor solução encontrada, dada pelas matrizes do controlador dinâmico:

$$A_K = 10^3 \begin{bmatrix} 201,7613 & 92,8200 & 263,6298 & 291,6901 \\ -586,3444 & -269,7441 & -766,0792 & -846,8013 \\ 12,5234 & 5,7625 & 16,3556 & 18,0925 \\ 4,9013 & 2,2567 & 6,4048 & 7,0827 \end{bmatrix},$$

$$B_K = 10^3 \begin{bmatrix} 67,3210 & -88,2200 \\ -195,6514 & 256,3170 \\ 4,1803 & -5,4705 \\ 1,6382 & -2,1549 \end{bmatrix},$$

$$C_K = \begin{bmatrix} -458,0765 & -210,7388 & -607,9408 & -644,8698 \\ -7,8991 & -4,0302 & 13,8096 & -7,3350 \end{bmatrix},$$

$$D_K = \begin{bmatrix} -152,4743 & 217,3407 \\ -9,9693 & -28,2444 \end{bmatrix}.$$

O número médio de elementos ineficazes gerados com cada grupo de operadores está exposto na figura 90. O grupo II de operadores se destaca pelo elevado número de elementos ineficazes nos diversos exemplos estudados.

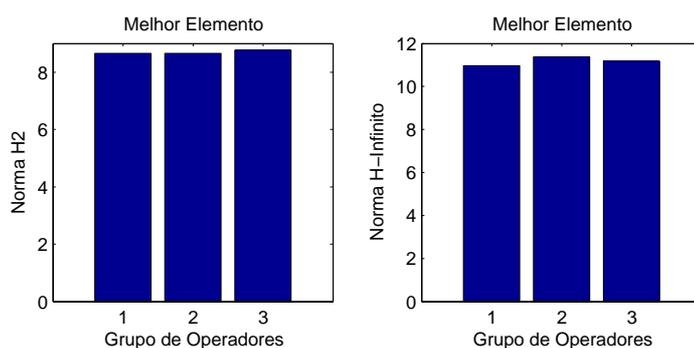


Figura 89: Comparação dos melhores resultados do exemplo 8 - caso 2.

A figura 91 mostra as nuvens de pólos em malha fechada utilizando-se o melhor cromossomo encontrado com cada grupo distinto de operadores. O formato distinto das nuvens mostradas nesta figura sugere que diferentes regiões do espaço de busca foram exploradas pelos diferentes algoritmos.

Os melhores resultados, encontrados com os diferentes grupos de operadores, estão

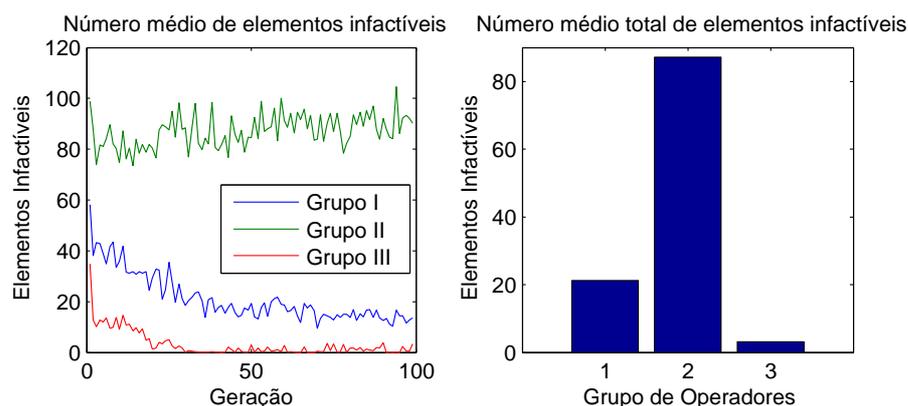


Figura 90: Número médio de elementos ineficazes gerados no exemplo 8 - caso 2.

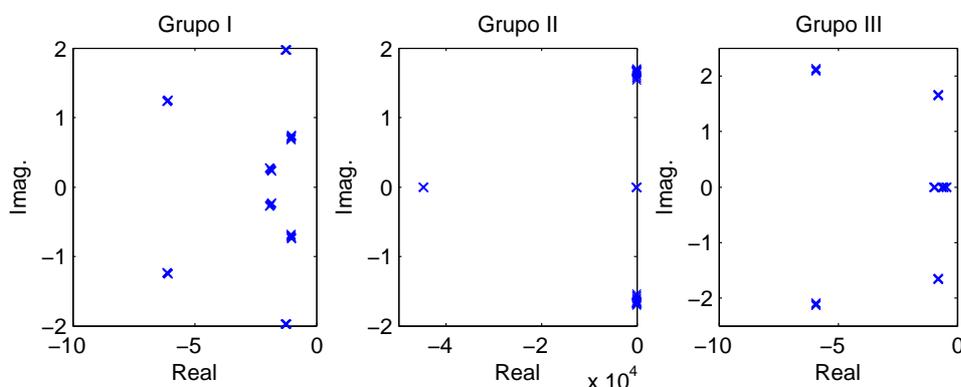


Figura 91: Pólos em malha fechada do exemplo 8 - caso 2.

demonstrados na figura 32 e formam conjunto de soluções de Pareto. A tabela 33 mostra os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ , quando sintetizados os controladores ótimos para cada norma em separado, utilizando-se o sistema nominal, onde o parâmetro $a_1 = 0,4081$. A comparação dos resultados das tabelas 32 e 33 sugere que as três abordagens foram capazes de aproximar a solução ótima para o sistema nominal. O tempo médio de execução deste exemplo foi de 2 horas e 46 minutos por simulação em um Pentium IV de 2,6 GHz com 1 GB de memória RAM.

Tabela 32: Comparação dos melhores resultados do exemplo 8 - caso 2.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	8,6558	8,6522	8,7613
$\ T_{z_\infty w_\infty}\ _\infty$	10,9751	11,3682	11,1767

Tabela 33: Comparação dos resultados ótimos para o sistema nominal do exemplo 8 - caso 2.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	8,4706	148,0799
$\ T_{z_\infty w_\infty}\ _\infty$	11,2633	5,1088

6.8.3 Análise do Exemplo 8

A análise das tabelas 30 e 32 indica que, com o espaço de busca mais restrito, os três algoritmos foram capazes de encontrar melhores soluções. Além disso, os grupos I e II apresentaram melhor convergência média para o caso onde $\gamma \leq 12,5$ do que para o caso irrestrito, quando $\gamma \leq 100$. A adição da condição mais restritiva, $\gamma \leq 12,5$, indica que o espaço de busca dos algoritmos foi reduzido, localizando a busca. Isto justifica o resultado final mais próximo, encontrado pelos três algoritmos, quando $\gamma \leq 12,5$.

Por outro lado, a adição da restrição em γ gerou um aumento significativo do número médio de elementos inactiváveis nos três algoritmos: 150%, 8% e 133% a mais para os grupos I, II e III respectivamente. Estes valores são mostrados na figura 92 e se devem à redução do espaço de soluções factíveis do problema, que aumenta a probabilidade do algoritmo gerar um elemento inactivável.

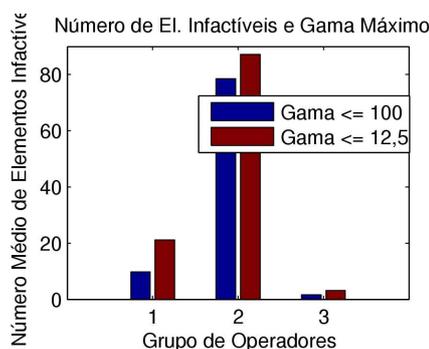


Figura 92: Número médio de elementos inactiváveis gerados no exemplo 8.

O grupo II de operadores se destaca pelo significativo número de elementos inactiváveis gerado, justificado pelos seus operadores de mutação. Neste exemplo, 74% dos elementos inactiváveis foram gerados pelos algoritmos de mutação do grupo II.

6.9 Exemplo 9

Este exemplo foi extraído e adaptado da biblioteca COMPlib [82] e corresponde a um modelo acadêmico de testes com matriz D_{12} de posto deficiente. A restrição de norma \mathcal{H}_∞ , $\gamma = 70$, foi utilizada e adicionou-se um parâmetro incerto à matriz dinâmica do sistema, dada por:

$$A = \begin{bmatrix} 0 & -1 & 2 \\ 1 & -2 & a_1 \\ 0 & 1 & 0 \end{bmatrix},$$

onde os limites do conjunto $a_1 \in [2, 9; 3, 1]$ formam os vértices do domínio de incertezas \mathcal{D}_A , representado por um polítopo com 2 vértices. As outras matrizes do sistema são dadas por:

$$\begin{aligned} C_2 &= \begin{bmatrix} 1,4 & 0 & 0 \end{bmatrix}, & C_y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, & C_1 &= \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \\ B_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, & B_{1_2} &= \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, & B_{1_\infty} &= \begin{bmatrix} 1 \\ -0,5 \\ 0 \end{bmatrix}, \\ D_{11_2} &= 0, & D_{11_\infty} &= 0, & D_{22} &= 0, \\ D_{12} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, & D_{21_\infty} &= 0, & D_{y1_\infty} &= 0. \end{aligned}$$

6.9.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Completa

Nesta subseção, projetaram-se controladores por realimentação dinâmica de saída de ordem completa ($n_c = 3$). A evolução dos diferentes grupos de operadores é apresentada na figura 93. A evolução da média da norma \mathcal{H}_2 dos melhores cromossomos mostra que o grupo I de operadores obteve um desempenho superior ao desempenho dos outros dois grupos. Este gráfico sugere que os grupos II e III não foram eficazes em “escapar” dos ótimos locais, indicando a convergência prematura destes do algoritmo.

Os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ , calculados com o melhor controlador sintetizado com cada grupo de operadores, são apresentados na figura 94. Pode-se observar que o resultado do grupo I foi superior aos resultados dos outros grupos de operadores, considerando-se ambos critérios de desempenho utilizados. O melhor controlador encontrado pelo grupo I

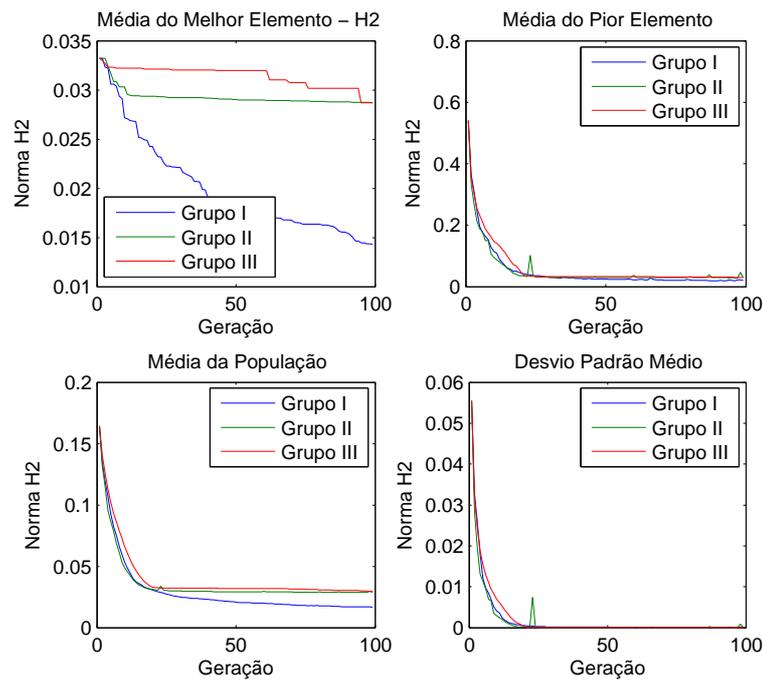


Figura 93: Evolução média dos algoritmos no exemplo 9 - caso 1.

de operadores é dado pelas matrizes:

$$A_K = \begin{bmatrix} -1317,5001 & 727,2368 & 703,2764 \\ 13,6837 & -1336,6491 & -217,0534 \\ 57,6206 & 1454,7312 & -336,4972 \end{bmatrix}, \quad B_K = \begin{bmatrix} 2887,1152 \\ 34,6927 \\ 1,2840 \end{bmatrix},$$

$$C_K = \begin{bmatrix} -820,6174 & 325,5949 & 3637,9215 \\ 1990,9009 & 3038,1087 & 4599,6349 \end{bmatrix}, \quad D_K = \begin{bmatrix} -1336,6490 \\ 1423,8262 \end{bmatrix}.$$

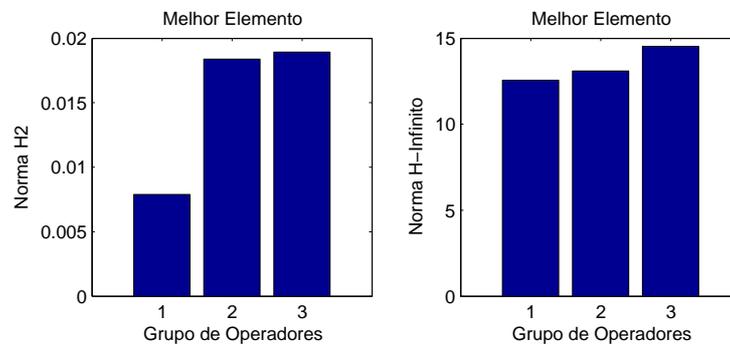


Figura 94: Comparação dos melhores resultados do exemplo 9 - caso 1.

Os números médios de elementos inactiváveis gerados com a utilização dos diferentes

grupos de operadores são apresentados na figura 95. Como ocorrido em todos os exemplos estudados, o grupo II de operadores se destaca pelo elevado número de elementos ineficazes gerados. Esta característica se deve ao conjunto de algoritmos de mutação do grupo II de operadores que, neste exemplo, gerou 78% do total de cromossomos ineficazes.

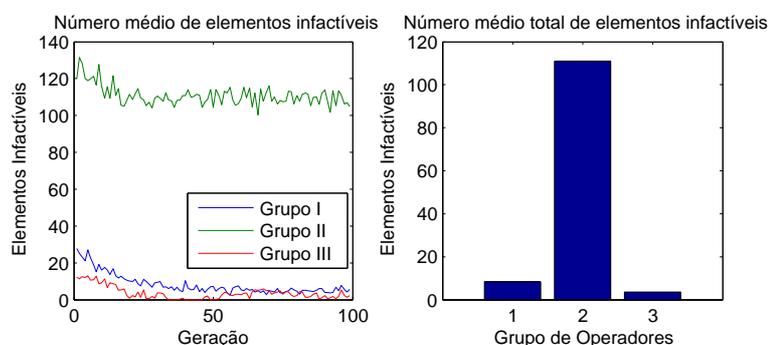


Figura 95: Número médio de elementos ineficazes gerados no exemplo 9 - caso 1.

A figura 96 mostra a nuvem de pólos em malha fechada para o melhor controlador encontrado com cada algoritmo.

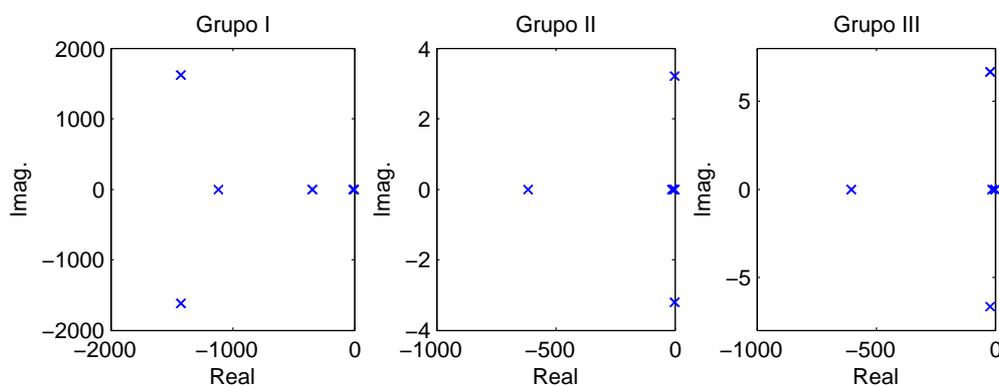


Figura 96: Pólos em malha fechada do exemplo 9 - caso 1.

Os valores da norma \mathcal{H}_2 e \mathcal{H}_∞ , calculados a partir do melhor cromossomo projetado com cada grupo de operadores são apresentados na tabela 34. A tabela 35, por sua vez, apresenta os controladores ótimos, sintetizados a partir de minimização das normas \mathcal{H}_2 e \mathcal{H}_∞ em separado para o sistema nominal, onde o parâmetro incerto $a_1 = 3,0$. Considerando-se o valor da norma \mathcal{H}_2 das tabelas 34 e 35, o grupo I foi o mais eficaz em encontrar um controlador satisfatório. O tempo médio de execução deste exemplo foi de 37 minutos por simulação em um Pentium IV de 1,4 GHz com 512MB de memória RAM.

Tabela 34: Comparação dos melhores resultados do exemplo 9 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	0,0078	0,0184	0,0189
$\ T_{z_\infty w_\infty}\ _\infty$	12,5703	13,0852	14,5234

Tabela 35: Resultados ótimos obtidos para o sistema nominal do exemplo 9 - caso 1.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	0,0062	93,4632
$\ T_{z_\infty w_\infty}\ _\infty$	7,2263	2,3122

6.9.2 Caso 2: Realimentação Dinâmica de Saída de Ordem Completa com População de 40 Elementos

Nesta subseção, projetaram-se controladores por realimentação dinâmica de saída de ordem completa. Em comparação com a seção anterior, na qual a população utilizada continha 20 elementos, usou-se nesta subseção uma população com 40 indivíduos. A evolução dos diferentes grupos de operadores, para este caso, é apresentada na figura 97. A exemplo da seção anterior, o grupo I de operadores foi o mais eficiente entre grupos na evolução média do melhor indivíduo da população. Além disso, pode-se também observar na figura 97 a convergência prematura dos grupos de operadores II e III.

A figura 98 mostra os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ , calculados com o melhor controlador encontrado com os diferentes grupos de operadores. Pode-se observar que, do ponto de vista de minimização da norma \mathcal{H}_2 , o grupo de operadores I apresentou o melhor resultado, formado com o controlador dinâmico:

$$A_K = \begin{bmatrix} -1588,9561 & -40,2805 & -2356,3594 \\ -1749,8986 & -2356,3594 & -2356,3594 \\ 2335,7390 & -52,1376 & -1935,6177 \end{bmatrix}, \quad B_K = \begin{bmatrix} -23,1694 \\ -29,4629 \\ -1094,9409 \end{bmatrix},$$

$$C_K = \begin{bmatrix} 8844,3890 & 210,5232 & 8844,4946 \\ -1318,6666 & -1490,7399 & -1158,9752 \end{bmatrix}, \quad D_K = \begin{bmatrix} -2356,3594 \\ 8844,8946 \end{bmatrix}.$$

O número médio de elementos ineficazes gerado pelos diferentes algoritmos é apresentado na figura 99 e segue o padrão encontrado nos demais exemplos estudados neste trabalho. Em comparação com a figura 94 da subseção anterior, pode-se observar que o número de elementos ineficazes praticamente dobrou, independentemente do grupo de

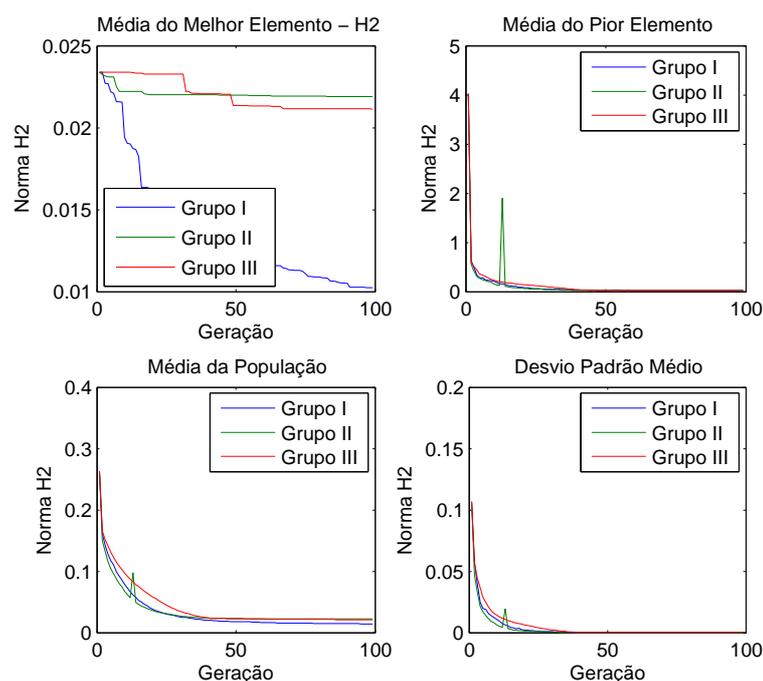


Figura 97: Evolução média dos algoritmos no exemplo 9 - caso 2.

operadores utilizado. Isto se deve ao aumento do tamanho da população.

As nuvens de pólos em malha fechada, calculadas com o melhor controlador encontrado com cada grupo de operadores, estão desenhadas na figura 100.

A tabela 36 exhibe os melhores resultados encontrados com os diferentes grupos de operadores, no conjunto de 10 simulações. A tabela 36, por sua vez, apresenta os controladores ótimos, sintetizados a partir de minimização das normas \mathcal{H}_2 e \mathcal{H}_∞ em separado, para o sistema nominal, onde o parâmetro incerto $a_1 = 3,0$. A exemplo da subseção anterior, considerando-se o valor da norma \mathcal{H}_2 das tabelas 36 e 37, o grupo I foi mais eficaz em encontrar um controlador satisfatório. O tempo médio de execução deste exemplo foi de 1 hora e 18 minutos por simulação em um Pentium IV de 1,4 GHz com 512 GB de memória RAM.

Tabela 36: Comparação dos melhores resultados do exemplo 9 - caso 2.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	0,01198	0,0237	0,01924
$\ T_{z_\infty w_\infty}\ _\infty$	47,2968	13,1887	13,7010

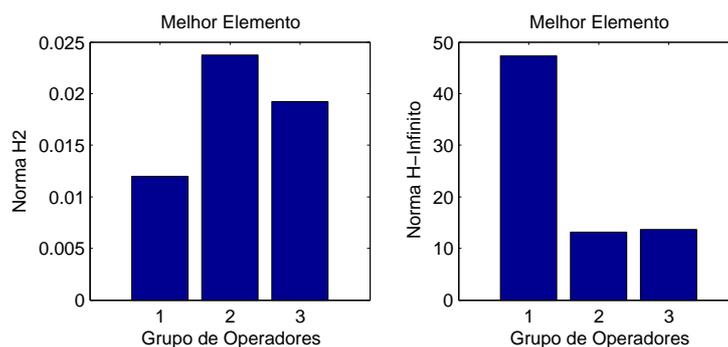


Figura 98: Comparação dos melhores resultados do exemplo 9 - caso 2.

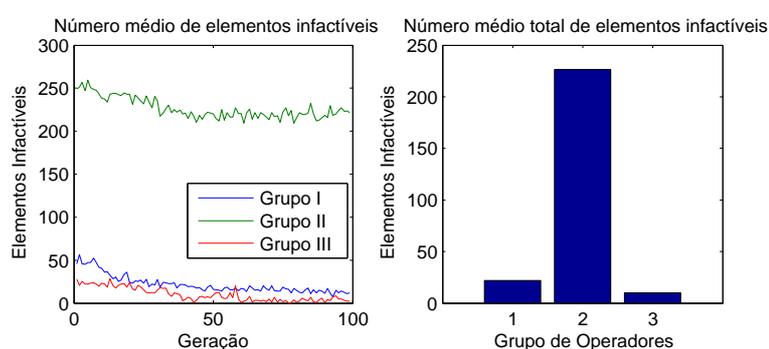


Figura 99: Número médio de elementos inactivos gerados no exemplo 9 - caso 2.

6.9.3 Análise do Exemplo 9

Este exemplo estudou o comportamento dos diferentes algoritmos quando o tamanho da população é aumentado. Este fato resultou em um impacto direto, tanto no tempo médio de execução dos grupos de operadores, quanto no número médio de elementos inactivos gerados pelos três grupos de operadores. O tempo médio de execução foi aumentado em 136% com a mudança do tamanho da população de 20 para 40 elementos. Os números médios de elementos inactivos gerados foram, respectivamente, aumentados em 172%, 112% e 166% para os grupos I, II e III de operadores, conforme se pode observar na figura 101. Além disso, a análise dos melhores resultados encontrados nas duas subseções deste exemplo, mostrados nas tabelas 34 e 36 sugerem que, neste exemplo, populações menores geraram melhores resultados finais. Isto indica que menores populações permitem a convergência mais rápida do algoritmo, em geral. O efeito colateral de pequenas populações é favorecer a convergência prematura dos algoritmos, uma vez que se reduz a diversidade de soluções na população. Neste exemplo, porém, o aumento do tamanho da população, de 20 para 40 elementos, não foi capaz de melhorar a convergência dos grupos II e III de operadores.

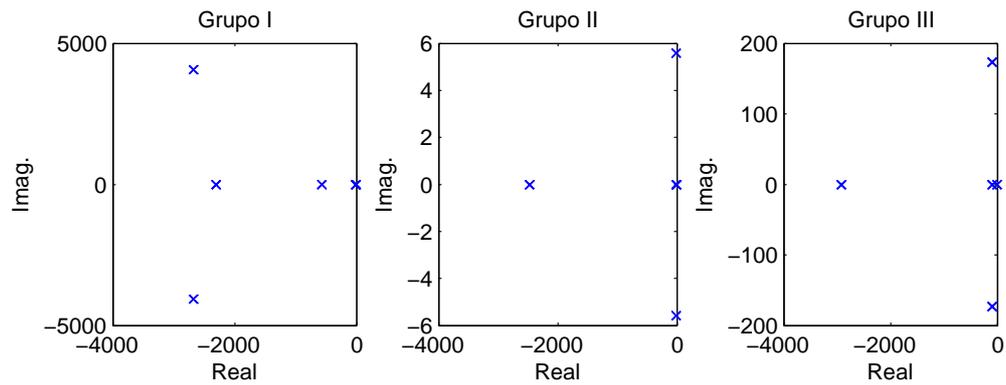


Figura 100: Pólos em malha fechada do exemplo 9 - caso 2.

Tabela 37: Resultados ótimos obtidos para o sistema nominal do exemplo 9 - caso 2.

	$\min \mathcal{H}_2$	$\min \mathcal{H}_\infty$
$\ T_{z_2 w_2}\ _2$	0,0062	93,4632
$\ T_{z_\infty w_\infty}\ _\infty$	7,2263	2,3122

6.10 Exemplo 10

Este exemplo de sexta ordem foi extraído e adaptado da biblioteca COMLib [82]. A restrição de norma \mathcal{H}_∞ , $\gamma \leq 100$, foi utilizada e adicionou-se um parâmetro incerto à matriz dinâmica do sistema, dada por:

$$A = \begin{bmatrix} -1 & 0 & 4 & 5 & -3 & -2 \\ -2 & 4 & -7 & -2 & 0 & 3 \\ -6 & 9 & -5 & 0 & 2 & -1 \\ -8 & 4 & 7 & -1 & -3 & 0 \\ 2 & a_1 & 8 & -9 & 1 & -4 \\ 3 & -5 & 8 & 0 & 2 & -6 \end{bmatrix},$$

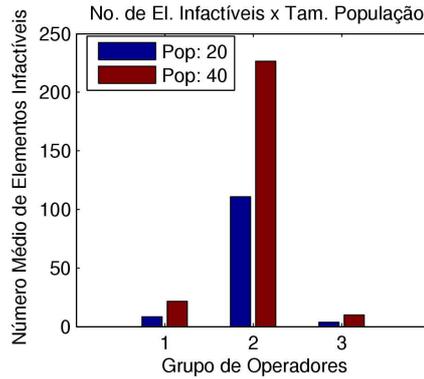


Figura 101: Número médio de elementos ineficazes gerados no exemplo 9.

onde o parâmetro $a_1 \in [2, 9; 3, 1]$ forma o poliedro convexo de incertezas, \mathcal{D}_A , do problema. O restante das matrizes do problema é dado por:

$$\begin{aligned}
 B_{1_2} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, & B_{1_\infty} &= \begin{bmatrix} -3 & -4 & -2 \\ 2 & 0 & 1 \\ -5 & -7 & 0 \\ 4 & -6 & 1 \\ -3 & 9 & -8 \\ 1 & -2 & 3 \end{bmatrix}, & B_2 &= \begin{bmatrix} 1 & 0 \\ -5 & 2 \\ 7 & -2 \\ 1 & -2 \\ 0 & 5 \\ -6 & -2 \end{bmatrix}, \\
 D_{y1_\infty} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & D_{12} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, & C_y &= \begin{bmatrix} 9 & -3 & 4 & 0 & 3 & 7 \\ 0 & 1 & -2 & 1 & -6 & 2 \end{bmatrix}, \\
 D_{11_2} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, & D_{11_\infty} &= \begin{bmatrix} 1 & -2 & -3 \\ 0 & 4 & 0 \\ 5 & -3 & -4 \end{bmatrix}, & C_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \\
 D_{22} &= 0, & D_{21_\infty} &= 0, & C_1 &= \begin{bmatrix} 1 & -1 & 2 & -4 & 0 & -3 \\ -3 & 0 & 5 & -1 & 1 & 1 \\ -7 & 5 & 0 & -8 & 2 & -2 \end{bmatrix}.
 \end{aligned}$$

6.10.1 Caso 1: Realimentação Dinâmica de Saída de Ordem Reduzida

Nesta subsecção projetaram-se controladores por realimentação dinâmica de saída de ordem reduzida ($n_c = 4$). A evolução dos diferentes grupos de operadores é apresentada na figura 102. A convergência média dos diferentes grupos de operadores é bastante distinta, sendo que o grupo I de operadores mostrou o melhor desempenho.

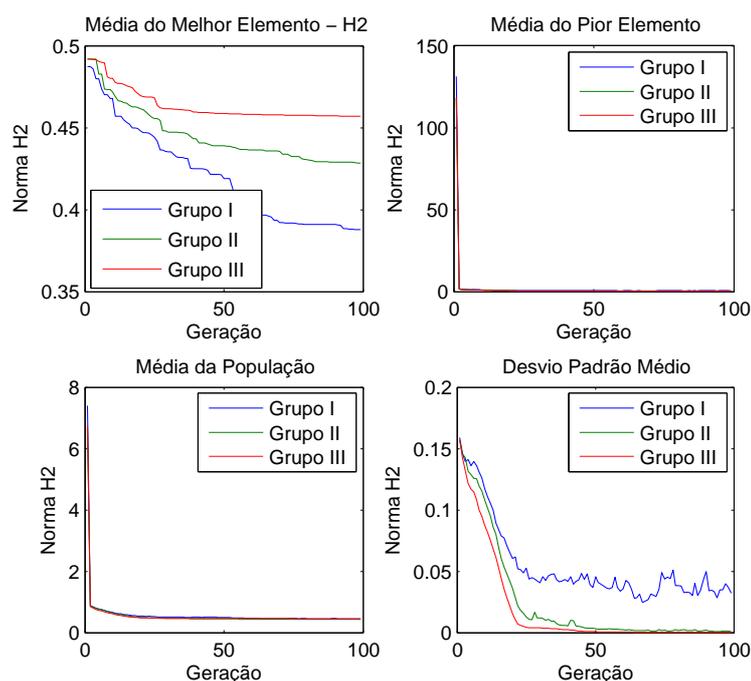


Figura 102: Evolução média dos algoritmos no exemplo 10 - caso 1.

A figura 103 apresenta os valores de norma \mathcal{H}_2 e \mathcal{H}_∞ calculados a partir do melhor controlador projetado pelos diferentes grupos de operadores. Pode-se verificar que, neste caso, o grupo I de operadores resultou na melhor solução encontrada, que é dada por:

$$A_K = \begin{bmatrix} -131,0224 & -489,7305 & -179,5080 & 10,5259 \\ 1388,9062 & 88,8219 & -53,0424 & -64,2754 \\ 615,5352 & -403,0315 & -317,4740 & -2,9702 \\ 2514,1329 & 35,2774 & -262,8192 & -191,1780 \end{bmatrix},$$

$$B_K = \begin{bmatrix} -4,9340 & 2513,8052 \\ -3,3229 & -528,0514 \\ -9,5509 & 446,7469 \\ -14,7286 & -43,7392 \end{bmatrix},$$

$$C_K = \begin{bmatrix} 20,1808 & -40,1648 & -27,8030 & 4,6906 \\ 2404,1898 & 2385,5487 & 1228,1411 & 2049,1945 \end{bmatrix},$$

$$D_K = \begin{bmatrix} -0,6476 & -1,1408 \\ -0,7243 & 2418,6914 \end{bmatrix}.$$

O número médio de elementos ineficazes gerados pelos diferentes algoritmos é apresentado na figura 104. Verifica-se, novamente, o elevado número médio de elementos

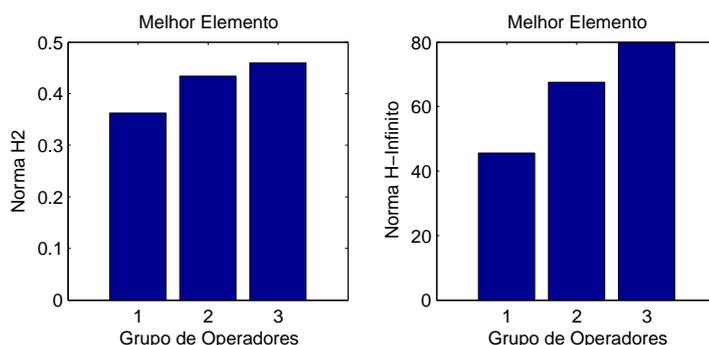


Figura 103: Comparação dos melhores resultados do exemplo 10 - caso 1.

inactivos gerados pelo grupo II de operadores. Neste exemplo, os operadores de mutação do grupo II foram responsáveis por 79% dos elementos inactivos gerados por este grupo. Por outro lado, a utilização de informações de direção na busca proporciona ao grupo III de operadores maior eficácia na geração de elementos activos.

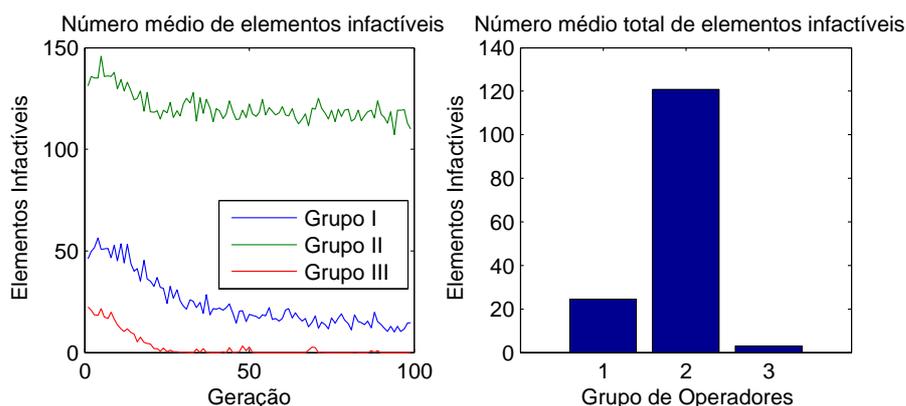


Figura 104: Número médio de elementos inactivos gerados no exemplo 10 - caso 1.

A figura 105 mostra os pólos em malha fechada para o melhor controlador encontrado com cada algoritmo. Os valores das normas \mathcal{H}_2 e \mathcal{H}_∞ , calculados com os melhores controladores encontrados com cada grupo de operadores, são exibidos na tabela 38. Pode-se verificar que, neste exemplo, o grupo I foi o mais eficaz na minimização da norma \mathcal{H}_2 . A análise da figura 102 sugere, todavia, que o grupo III de operadores, considerando a utilização das informações de direção, convergiu prematuramente e foi incapaz evoluir adequadamente a população.

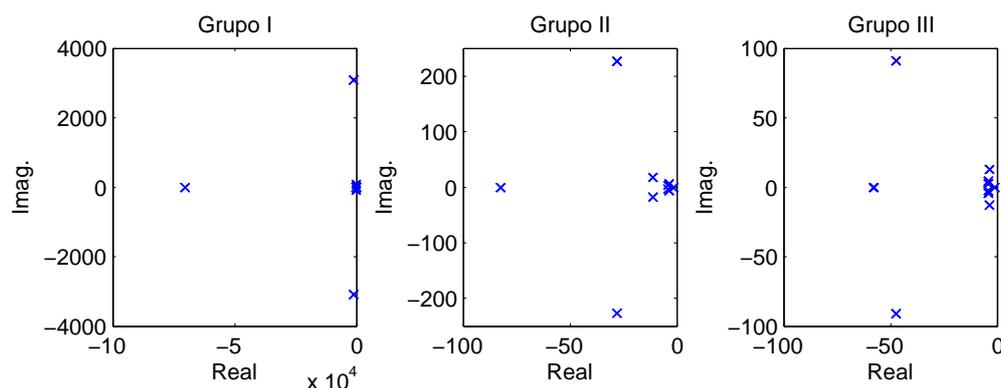


Figura 105: Pólos em malha fechada do exemplo 10 - caso 1.

Tabela 38: Comparação dos melhores resultados do exemplo 10 - caso 1.

	Grupo I	Grupo II	Grupo III
$\ T_{z_2 w_2}\ _2$	0,3622	0,4340	0,4594
$\ T_{z_\infty w_\infty}\ _\infty$	45,6440	67,5878	79,8484

6.11 Conclusão

Este capítulo apresentou a aplicação de três configurações básicas do algoritmo, discutido no capítulo 5, em diversos exemplos extraídos e adaptados da literatura. A tabela 39 mostra as estatísticas gerais de simulação para este capítulo. Apesar dos resultados

Tabela 39: Estatísticas gerais do capítulo.

Número Total de Execuções:	610
Tempo Médio por Execução:	2 horas e 40 minutos
Tempo Total de Execução:	70 dias, 9 horas e 38 minutos

inspiradores apresentados neste capítulo, a tabela 39 indica que o custo computacional desta abordagem é elevado, limitando sua aplicação em exemplos de ordens elevadas e na síntese de controladores em tempo real.

O lado esquerdo da figura 106 apresenta o percentual de vezes que cada grupo de operadores gerou o melhor resultado com relação à norma \mathcal{H}_2 . Considerou-se, para este cálculo, o melhor cromossomo encontrado a cada conjunto de 10 simulações. O melhor resultado foi encontrado em 62% das vezes pelo grupo III de operadores. O grupo I foi o mais eficaz em 31% dos casos e o grupo II em apenas 7%. Este resultado indica que o grupo III, com a utilização das informações de direção, é um bom ponto de partida

na síntese de controladores. Os resultados apresentados são, porém, insuficientes para se garantir estatisticamente a superioridade do grupo III de algoritmos.

O lado direito desta da figura 106, por sua vez, apresenta o percentual de vezes que cada grupo de operadores foi capaz de gerar o melhor resultado médio com relação à norma \mathcal{H}_2 . Para este cálculo, considerou-se a média da norma \mathcal{H}_2 dos melhores cromossomos de cada simulação no conjunto de 10 simulações. O comportamento observado no lado esquerdo da figura 106 se repetiu, quando se considerou a média do melhor cromossomo. Em 59% das vezes o grupo III de operadores foi o mais eficaz. Os grupos I e II geraram os melhores resultados médios em 36% e 5% das execuções, respectivamente. Este resultado reforça a superioridade do grupo III de operadores na solução do problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$.

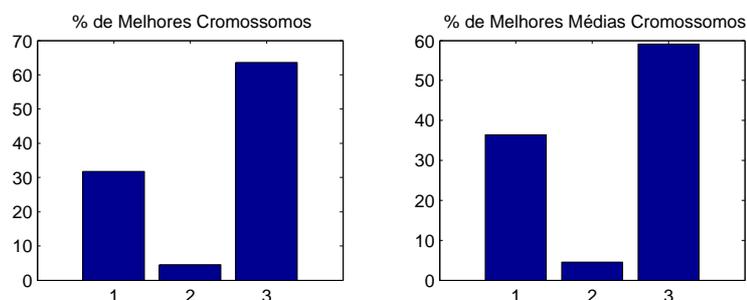


Figura 106: Percentual de melhores resultados x grupos de operadores.

A exemplo da figura 106, aplicou-se a mesma lógica no cálculo do percentual de vezes em que cada grupo de operadores gerou os piores resultados. O lado esquerdo da figura 107 mostra o percentual de vezes que cada grupo de operadores resultou no valor mais elevado da norma \mathcal{H}_2 entre os melhores elementos gerados. Pode-se observar que, dos melhores indivíduos encontrados com cada grupo de operadores no conjunto de 10 simulações, o grupo I gera, em 59% dos casos, os elementos com norma \mathcal{H}_2 mais elevada. O grupo II de operadores, por sua vez, é responsável por 27% da norma \mathcal{H}_2 mais elevada, entre os melhores cromossomos sintetizados. Por fim, o grupo III de operadores gera apenas 14% destes indivíduos.

O lado direito da figura 107 mostra o percentual de execuções nas quais cada grupo distinto de operadores gerou o valor médio da norma \mathcal{H}_2 mais elevado. Considerou-se para este cálculo a média dos 10 melhores elementos, sintetizados a cada conjunto de 10 simulações. Pode-se observar que o grupo I de operadores foi responsável por gerar o pior valor médio dos melhores cromossomos em 50% dos casos. O grupo II de operadores resultou no pior valor médio da norma \mathcal{H}_2 em 32% dos casos e o grupo III em apenas

18%.

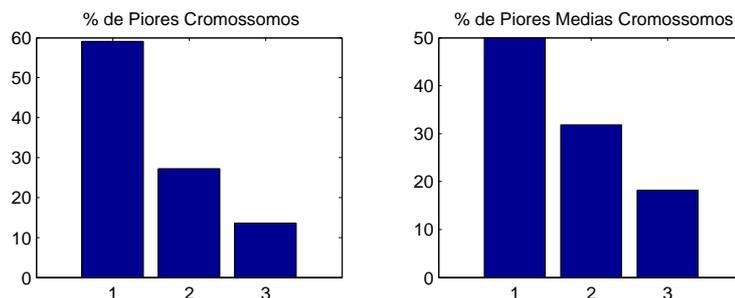


Figura 107: Percentual de piores resultados x grupos de operadores.

Estes percentuais, em conjunto com a análise das diversas simulações efetuadas neste capítulo, sugerem algumas conclusões:

1. o grupo III de operadores é um bom ponto de partida na solução do problema abordado neste trabalho, uma vez que, freqüentemente proporciona as melhores soluções e raramente produz os piores resultados entre os três diferentes grupos, indicando “robustez” na busca. Isto indica que a utilização das informações de busca por parte do grupo III de operadores efetivamente colabora na busca por melhores resultados;
2. o grupo I de operadores, apesar de gerar os melhores resultados em 31% dos casos estudados, correspondeu a 59% dos piores resultados. Este fato indica que os operadores derivados do algoritmo de Michalewicz freqüentemente oscilam entre a geração dos melhores e dos piores resultados entre os 3 grupos de operadores, sugerindo menor “robustez” e confiabilidade na busca;
3. a aplicação do grupo II de operadores resultou nos melhores resultados em apenas 7% dos problemas estudados. Por outro lado, este grupo gerou os piores valores de norma \mathcal{H}_2 , entre os melhores cromossomos dos três grupos, somente em 27% dos exemplos. Estes números indicam certa “robustez” de busca deste grupo de operadores. Todavia, sugerem que este grupo de operadores não é eficaz na geração dos melhores controladores;
4. o grupo III de operadores é o único que, em todos os exemplos analisados, efetivamente fez a população inicial convergir para uma única solução, como pode ser observado nos diversos gráficos do desvio padrão médio das populações. As prováveis causas desta característica são a utilização das informações de direção na busca,

tanto pelo algoritmo de Evolução Diferencial, quanto pelo algoritmo de Salomon e a mecânica de substituição dos cromossomos representada pelo passo 7 do operador de ED (seção 5.6.3);

5. não se identificaram classes entre os sistemas estudados que sugiram que para determinado tipo de problema um grupo específico de operadores seja o mais indicado. Como cada grupo de operadores pode resultar em evoluções e resultados bastante distintos, dependendo das restrições e características de cada problema, indica-se que o ideal na busca pela melhor solução de determinado problema é a aplicação dos três algoritmos. Este fato condiz com os teoremas de Wolpert [130], que pregam que qualquer desempenho elevado de um algoritmo para uma classe de problemas é penalizado com um desempenho ruim para outra classe de problemas. Assim, como não se identificaram classes entre os problemas estudados, justifica-se a utilização dos três grupos de operadores na busca pelos melhores resultados;
6. o grupo de operadores II gera um número consideravelmente superior de elementos ineficazes, comparado aos outros dois grupos de operadores, fato observado em todas as execuções dos algoritmos. Duas possíveis causas são atribuídas a esta característica:
 - (a) a não utilização de informações de direção pelo algoritmos do grupo II, que são utilizadas tanto pelo operador de cruzamento heurístico do grupo I, quanto pelos operadores de ED e de Salomon do grupo III;
 - (b) o caráter “agressivo” dos operadores de mutação do grupo II, que alteram diretamente os autovalores das matrizes de Lyapunov ou modificam aleatoriamente os componentes do controlador. Observou-se que, na média, 76% dos elementos ineficazes gerados pelo grupo II de operadores é resultado dos algoritmos de mutação deste grupo.
7. a reduzida eficácia média do grupo de operadores II, representada pela figura 106, indica que a estratégia dos operadores genéticos deste grupo, fundamentados sobretudo na operação sobre as matrizes de Lyapunov X_2 e X_∞ , não é tão eficiente quanto a operação direta sobre o controlador L_K , utilizada tanto pelo grupo I quanto pelo grupo III de operadores;
8. a inclusão de restrições, tanto de alocação de pólos quanto em γ , resulta em um maior número de elementos ineficazes gerados. A redução do espaço de busca com

a adição de restrições, bem como a não consideração das restrições na direção da busca são as possíveis causas deste efeito;

9. a eficiência dos três grupos de operadores é altamente influenciada pelo aumento da ordem do controlador. Esta relação é fruto da característica polinomial do algoritmo de pontos interiores, utilizado na otimização LMI;
10. não se identificou aparente correlação entre a ordem do controlador a ser sintetizado e o número de elementos inactíveis gerado pelos diferentes algoritmos.

Do ponto de vista de eficácia, considerando-se os valores da norma \mathcal{H}_2 encontrados com cada grupo de operadores, os diversos exemplos indicam que, as três abordagens, apesar apresentarem resultados diferentes entre si, são satisfatórias quando comparadas tanto aos resultados da literatura, quanto aos controladores ótimos sintetizados para os sistemas nominais. Além disso, os exemplos apresentados neste capítulo sugerem que o casamento entre os algoritmos evolutivos e a formulação LMIs resultam em soluções satisfatórias para o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$. Estas melhores soluções são, por sua vez, possibilitadas pela liberação da hipótese de uma matriz comum de Lyapunov para as diferentes restrições do problema.

7 Conclusão Geral

Este trabalho teve como objetivo principal a síntese de controladores robustos mistos $\mathcal{H}_2/\mathcal{H}_\infty$ \mathcal{D} -estáveis para sistemas lineares, invariantes no tempo e sujeitos a incertezas do tipo polítopo. Os algoritmos evolutivos foram estudados e, utilizando-se as desigualdades matriciais para a formulação do problema, um algoritmo de síntese foi proposto. A união dos algoritmos evolutivos com a formulação LMI proporciona grande flexibilidade ao projetista, que pode selecionar entre as leis de controle de realimentação de saída estática ou dinâmica. Neste último caso, pode-se ainda optar entre controladores de ordem reduzida ou completa, sendo que esta ordem é determinada *a priori*. Além disso, o algoritmo proposto não exige matrizes comuns de Lyapunov para todas as restrições. Este fato diminui o conservadorismo da solução, aumentando-se o espaço de busca factível. É importante ressaltar que para cada restrição, devido às incertezas paramétricas, a mesma matriz de Lyapunov é mantida, ou seja, o conceito de estabilidade quadrática ainda é utilizado para cada tipo de restrição.

Na busca de melhores soluções para o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$ com alocação de pólos, três grupos de operadores evolutivos foram implementados. O grupo I, formado pelos operadores apresentados nas seções 5.6.1 e 5.7.1, foi fundamentado no algoritmo genético de Michalewicz (seção 4.3). Estes operadores são aplicados apenas na matriz do controlador. O grupo II de operadores, conjunto dos algoritmos descritos nas seções 5.6.2 e 5.7.2, envolve a modificação não só da matriz do controlador, mas também das matrizes de Lyapunov associadas ao problema. Com o objetivo de se reforçar o uso de informações de direção de busca pelo algoritmo, implementaram-se os operadores de evolução diferencial e de Salomon, descritos nas seções 5.6.3 e 5.7.3, respectivamente. A evolução diferencial é amplamente conhecida por suas propriedades de robustez na busca. O operador de Salomon é um método de aproximação do gradiente que lembra a derivação por diferenças e permite direcionar a busca do algoritmo.

Diversos exemplos extraídos e adaptados da literatura foram estudados no capítulo 6. Os três grupos de operadores, propostos no capítulo 5, forneceram bons resultados

quando comparados com os resultados da literatura e os resultados ótimos para o sistema nominal. A análise dos exemplos estudados sugere que o grupo I de operadores, apesar de, em alguns casos, apresentar bons resultados finais, não possui um comportamento de busca confiável. Este grupo de operadores ora gera os melhores resultados, ora os piores. O grupo II, por sua vez, não se destaca na geração dos melhores resultados, porém apresenta resultados satisfatórios com um comportamento regular na maioria dos problemas estudados. O grupo III de operadores se destaca tanto pelos resultados obtidos, quanto pelo seu comportamento de busca. Este grupo de operadores freqüentemente gera os melhores resultados entre os três grupos e raramente produz as piores soluções. Ressalta-se que o número de exemplos estudados e o número de simulações efetuadas são insuficientes para garantir estatisticamente a superioridade do grupo III de operadores. Entretanto, os resultados sugerem que este grupo de operadores pode ser um bom ponto de partida. Uma das possíveis causas do bom desempenho do grupo III é o uso de informações de direção, tanto pelo algoritmo de ED, quanto pelo algoritmo de Salomon. As informações de busca, em geral, contribuem significativamente na melhoria do desempenho dos algoritmos evolutivos, conforme ressaltado em [19].

Na busca por melhores resultados, algumas modificações nos algoritmos originais foram realizadas e constituem as principais contribuições deste trabalho:

- a estruturação do grupo II de operadores com informações específicas do problema, utilizando-se as matrizes de Lyapunov e suas propriedades (autovalores) para as operações genéticas (seções 5.6.2 e 5.7.2);
- a adequação adaptativa do coeficiente F no operador de evolução diferencial, que permite restringir a região de busca possível do algoritmo, localizando-a (seção 5.6.3);
- a seleção heurística e não aleatória do elemento L_{K1} no algoritmo de evolução diferencial, proporcionando convergência mais rápida ao algoritmo (seção 5.6.3);
- a modificação da estratégia de construção do elemento L_{K5} no algoritmo de evolução diferencial (seção 5.6.3), reduzindo-se consideravelmente o percentual de elementos ineficazes gerados pelo algoritmo;
- a inclusão do fator pm_i (seção 5.7), responsável por aumentar gradativamente a probabilidade de mutação do algoritmo à medida em que este evolui. Esta estratégia possui influência direta na exploração do espaço de busca, diminuindo-se a probabilidade de convergência prematura dos diferentes operadores;

- a utilização do fator de localização de busca, k_t , nos operadores de mutação (seção 5.7). Este fator permite que, quando a aplicação dos operadores de mutação resulta em elementos infactíveis, a busca seja localizada, aumentando-se a probabilidade de geração de elementos factíveis;
- a adaptação dos operadores de mutação de Michalewicz (seção 5.7.1), incorporando-se o conjunto Ω , haja vista que o cálculo dos conjuntos $left()$ e $right()$ não é possível. O conjunto Ω permite que os operadores de mutação de Michalewicz, como, por exemplo, a mutação de contorno, sejam aplicados da forma mais direta possível no problema abordado, buscando-se a fidelidade ao algoritmo original;
- a utilização de um operador de estimação da direção de busca no problema, o operador de Salomon (seção 5.7.1), com o intuito de se direcionar a busca genética.

A geração da população inicial é uma das principais limitações deste trabalho. A utilização das restrições estruturais em W_1 e W_2 , apresentadas na equação (5.10), da seção 5.5, pode resultar na geração inicial de elementos confinados em uma região restrita do espaço de busca. Esta característica pode influenciar a busca do algoritmo ao gerar populações iniciais com pouca diversidade. Não se identificou, porém, outro possível algoritmo para a geração da população inicial. A complexidade do espaço de busca do problema estudado e as dimensões do cromossomo tornam a inicialização aleatória inviável, por conta do elevado custo computacional associado à análise das soluções. Com o intuito de se extrapolar a região do espaço de busca, na qual a população inicial foi gerada, utilizaram-se probabilidades de mutação elevadas e adaptativas. Além disso, tomou-se o cuidado de se permanecer conservador nos parâmetros do algoritmo de Salomon (seção 5.7.3) para que este fosse capaz de contribuir para a busca global e não somente local.

As principais dificuldades encontradas na realização deste trabalho são descritas na seqüência. No intuito de se explorar adequadamente o espaço de busca na geração da população inicial, tentou-se utilizar o teorema dos círculos de Gerschgorin, que fornece uma condição sobre os autovalores de uma matriz dada. Na tentativa de se gerar de uma população inicial factível, usou-se este teorema em conjunto com técnicas de programação linear para a síntese dos cromossomos iniciais. Esta tentativa foi frustrada, pois, para a aplicação do teorema de Gerschgorin, diversas hipóteses restritivas adicionais são necessárias, como por exemplo, a imposição de componentes positivos para a matriz em malha fechada, inviabilizando sua aplicação. Além da utilização do teorema de Gerschgorin, implementou-se o algoritmo de alocação de pólos MIMO, descrito em [31]. Como esta abordagem não considera sistemas incertos, a idéia era gerar populações de cromossomos

através da alocação aleatória de pólos para sistemas precisamente conhecidos, dentro do domínio de incertezas. Com estes cromossomos verificaria-se então a factibilidade robusta dos controladores encontrados. Este método se mostrou ineficaz, pois, o elevado esforço de controle necessário para a alocação dos pólos do sistema precisamente conhecido implica elevada sensibilidade com relação às incertezas da matriz dinâmica. Isso significa que, em geral, os controladores sintentizados através da abordagem de alocação de pólos, descrita em [31], não são factíveis do ponto de vista robusto, inviabilizando a utilização deste método. Por fim, a proposta inicial deste trabalho contemplava uma aplicação prática do algoritmo implementado. Para isso, o sistema *Coupled Drives*, formado por três polias e uma correia, descrito em [128], foi estudado. O objetivo era realizar o controle robusto do sistema, mantendo-se a velocidade e a tensão da correia constantes. Infelizmente, problemas técnicos com os equipamentos da universidade impossibilitaram a implementação do sistema de controle.

Como possíveis continuações deste trabalho, sugerem-se a incorporação de restrições na variável de controle, a adequação do algoritmo para tratar o caso discreto e o estudo de outras alternativas de geração da população inicial, como, por exemplo, a abordagem descrita em [63]. Além disso, na busca por uma solução ainda menos restritiva para o problema de controle robusto misto $\mathcal{H}_2/\mathcal{H}_\infty$, pode-se estudar a possível adequação do algoritmo aos conceitos de funções de Lyapunov dependentes de parâmetros, relaxando-se a hipótese de uma mesma matriz de Lyapunov para todo o domínio de incertezas. Esta opção incorporaria o caso apresentado e poderia contribuir na busca de melhores soluções. Com isso, porém, uma questão a ser avaliada, seria o aumento do número de LMIs a ser utilizado, fato que pode aumentar consideravelmente o custo computacional do algoritmo.

Referências

- [1] H. A. Abbass e R. Sarker, *The pareto differential evolution algorithm*, International Journal on Artificial Intelligence Tools **11** (2002), no. 4, 531–552.
- [2] M. M. Ali e A. Törn, *Population set-based global optimization algorithms: some modifications and numerical studies*, Computers and Operations Research **31** (2004), no. 10, 1703–1725.
- [3] D. Arzelier, J. Bernussou, e G. Garcia, *Pole assignment of linear uncertain systems in a sector via a lyapunov-type approach*, IEEE Transactions on Automatic Control **38** (1993), no. 7, 1128–1132.
- [4] D. Arzelier e D. Peaucelle, *An iterative method for mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis via static output feedback*, 41th Conference on Decision and Control **3** (2002), 3463–3469.
- [5] M. Athans, *Special issue on linear-quadratic-gaussian control*, IEEE Transactions on Automatic Control **16** (1971), no. 6.
- [6] O. Bachelier, J. Bernussou, M. C. de Oliveira, e J. C. Geromel, *Parameter dependant lyapunov control design: numerical evaluation*, 38th Conference on Decision and Control **1** (1999), 293–297.
- [7] R. Bambang, E. Shimemura, e K. Uchida, *Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control with pole placement, state-feedback case*, Proceedings of American Control Conference **3** (1993), 2777–2779, San Francisco, CA.
- [8] B. R. Barmish, *Stabilization of uncertain systems via linear control*, IEEE Transactions on Automatic Control **28** (1983), no. 8, 848–850.
- [9] ———, *Necessary and sufficient conditions for quadratic stabilizability of an uncertain system*, Journal of Optimization Theory and Applications **46** (1985), no. 4, 399–408.
- [10] T. Basar e P. Bernhard, *\mathcal{H}_∞ -optimal control and related minimax design problems: a dynamic game approach*, 2nd. ed., Birkhäuser, Boston, 1995.
- [11] P. J. Bentley, *Evolutionary design by computers*, Morgan Kaufmann Publishers, North Yorkshire, Great Britain, 1999.
- [12] D. S. Bernstein e W. M. Haddad, *The optimal projection equations with Petersen-Hollot bounds: Robust stability and performance via fixed order dynamic compensation*, IEEE Transactions on Automatic Control **33** (1988), no. 6, 578–582.
- [13] ———, *Robust stability and performance via fixed-order dynamic compensation with guaranteed cost bounds*, Mathematics of Control, Signals and Systems **3** (1990), 139–163.

- [14] J. Bernussou e G. Garcia, *Disk pole assignment for uncertain systems with norm bounded uncertainty*, IFAC International Workshop on Robust Control (1993).
- [15] H.-G. Beyer, N. Saravanan, D. E. Waagen, e A. E. Eiben, *On the 'explorative' power of ES/EP-like algorithms*, Proc. 7th Annual Conference on Evolutionary Programming (1998), 323–334.
- [16] G. E. P. Box, *Evolutionary operation: a method for increasing industrial productivity*, Applied Statistics **6** (1957), no. 2, 81–101.
- [17] S. Boyd, L.E. Ghaoui, E. Feron, e V. Balakrishnan, *Linear matrix inequalities in system and control theory*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.
- [18] H. J. Bremermann, *Optimization through evolution and recombination*, Self-Organizing Systems (M. Yovits, G. T. Jacobi, e G. D. Goldstine, eds.), Spartan Books, Washington, D.C., 1962, pp. 93–106.
- [19] M. Brown e R. E. Smith, *Effective use of directional information in multi-objective evolutionary computation*, Genetic and Evolutionary Computation **2723** (2003), no. 1, 778–789.
- [20] T. Bäck, *Evolutionary algorithms in theory and practice*, Oxford University Press, New York, NY, 1996.
- [21] T. Bäck, U. Hammel, e H.-P. Schwefel, *Evolutionary computation: Comments on the history and current state*, IEEE Transactions on Evolutionary Computation **1** (1997), no. 1, 3–17.
- [22] T. Bäck e F. Kursawe, *Evolutionary algorithms for fuzzy logic: a brief overview*, 5th Information Processing and Management of Uncertainty in Knowledge-Based Systems (1996), 659–664.
- [23] T. Bäck, Z. Michalewicz, D. B. Fogel, e S. Pidgeon, *Handbook of evolutionary computation*, Institute of Physics Publishing and Oxford University Press, Bristol, PH and New York, NY, 1997.
- [24] T. Bäck e H.-P. Schwefel, *An overview of evolutionary algorithms for parameter optimization*, Evolutionary Computation **1** (1993), no. 1, 1–23.
- [25] D. U. Campos-Delgado e Kemin Zhou, *A parametric optimization approach to \mathcal{H}_∞ and \mathcal{H}_2 strong stabilization*, Automatica (2003), 1205–1211.
- [26] S. S. L. Chang e T. K. C. Peng, *Adaptive guaranteed cost control of systems with uncertain parameters*, IEEE Transactions on Automatic Control **17** (1972), no. 4, 474–483.
- [27] B.S. Chen, Y.M. Cheng, e C.H. Lee, *A genetic approach to mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal PID control*, IEEE Control Systems Magazine **15** (1995), no. 5, 51–60.
- [28] C.-T. Chen, *Linear system theory and design*, CBS College Publishing, New York, NY, 1984.

- [29] M. Chilali e P. Gahinet, *\mathcal{H}_∞ design with pole placement constraints: An LMI approach*, IEEE Transactions on Automatic Control **41** (1996), no. 3, 358–367.
- [30] M. Chilali, P. Gahinet, e P. Apkarian, *Robust pole placement in LMIs regions*, IEEE Transactions on Automatic Control **44** (1999), no. 12, 2257–2270.
- [31] T. Clarke, S. J. Griffin, e J. Ensor, *Output feedback eigenstructure assignment using a new reduced orthogonality condition*, International Journal of Control **76** (2003), no. 4, 390–402.
- [32] L. S. Coelho, H. X. de Araújo, G. H. C. Oliveira, e N. Mendes, *Controle PI baseado em otimização por evolução diferencial aplicado a um sistema de conforto térmico*, Congresso Brasileiro de Automática - CBA04 (2004).
- [33] J. B. Cruz, *Feedback systems*, McGraw Hill, New York, NY, 1971.
- [34] C. Darwin, *The origin of species by means of natural selection*, John Murray, 1872.
- [35] H. X. de Araújo, *Sur la stabilité des systèmes incertains sujets à des contraintes sur l'état et la commande*, Ph.D. thesis, Université Paul Sabatier de Toulouse, Toulouse, January 1998.
- [36] H. X. de Araújo e D. Neumann, *Controle robusto $\mathcal{H}_2/\mathcal{H}_\infty$: Síntese via algoritmos genéticos*, Congresso Brasileiro de Automática - CBA (2004).
- [37] P. J. de Oliveira, R. C. L. F. Oliveira, V. J. S. Leite, V. F. Montagner, e P. L. D. Peres, *\mathcal{H}_2 guaranteed cost computation by means of parameter dependent Lyapunov functions*, International Journal of Systems Science **35** (2004), no. 5, 305–315.
- [38] E. C. de Paiva, *Otimização restrita de controladores robustos utilizando algoritmos genéticos*, Ph.D. thesis, UNICAMP, Campinas, Brazil, April 1997.
- [39] C. E. de Souza, U. Shaked, e M. Fu, *Robust \mathcal{H}_∞ filtering for continuous time varying uncertain systems with deterministic input signals*, IEEE Transactions on Signal Processing **43** (1995), no. 3, 709–719.
- [40] C. E. de Souza e L. Xie, *On the discrete-time bounded real lemma with application in the characterization of static state feedback \mathcal{H}_∞ controllers*, Systems and Control Letters **18** (1992), no. 1, 61–71.
- [41] P. Dorato, *Robust control*, IEEE Press, New York, NY, 1987.
- [42] J. C. Doyle, *Synthesis of robust controllers and filters*, Proceedings of the 22nd. IEEE Conference on Decision and Control (1983), 109–114, San Antonio, TX, USA.
- [43] J.C. Doyle, K. Glover, P.K. Khargonekar, e B.A. Francis, *State-space solutions to standard \mathcal{H}_2 and \mathcal{H}_∞ problems*, IEEE Transactions on Automatic Control **34** (1989), no. 8, 831–847.
- [44] J.C. Doyle, K. Glover K. Zhou, e B. Bodenheimer, *Mixed \mathcal{H}_2 and \mathcal{H}_∞ performance objectives: I. robust performance analysis. II. optimal control*, IEEE Transactions on Automatic Control **39** (1994), no. 8, 1564–1574.

- [45] E. Feron, P. Apkarian, e P. Gahinet, *Analysis and synthesis of robust control systems via parameter-dependent Lyapunov functions*, IEEE Transactions on Automatic Control **41** (1996), no. 7, 1041–1046.
- [46] J. R. Filho, P. C. Treleaven, e C. Alippi, *Genetic algorithm programming environments*, IEEE Computer **27** (1994), no. 6, 28–43.
- [47] J.L.R. Filho, *Tutorial: Algoritmos genéticos*, III Simpósio Brasileiro de Redes Neurais (1996).
- [48] P. J. Fleming e R. C. Purshouse, *Evolutionary algorithms in control systems engineering: a survey*, Control Engineering Practice **10** (2002), 1223–1241.
- [49] L. J. Fogel, *Autonomous automata*, Industrial Research **4** (1962), 14–19.
- [50] L. J. Fogel, A. J. Owens, e M. J. Walsh, *Artificial intelligence through simulated evolution*, John Wiley, New York, NY, 1966.
- [51] P. M. Frank, *Introduction to systems sensitivity theory*, Academic Press, New York, NY, 1978.
- [52] R. M. Friedberg, *A learning machine: part I*, IBM Journal **2** (1958), no. 1, 2–13.
- [53] R. M. Friedberg, B. Dunham, e J. H. North, *A learning machine: part II*, IBM Journal **3** (1959), no. 7, 282–287.
- [54] M. Fu, C. E. de Souza, e L. Xie, *\mathcal{H}_∞ estimation for uncertain systems*, International Journal of Robust and Nonlinear Control **2** (1992), 87–105.
- [55] P. Gahinet, A. Nemirovski, A.J. Laub, e M. Chilali, *The LMI control toolbox*, The Mathworks INC., 1995.
- [56] M. Gen e R. Cheng, *A survey of penalty techniques in genetic algorithms*, IEEE Conference on Evolutionary Computation (1996), 804–809.
- [57] J. C. Geromel, P. L. D. Peres, e S. R. de Souza, *\mathcal{H}_2 guaranteed cost control for uncertain continuous-time linear systems*, Systems and Control Letters **19** (1992), no. 1, 23–27.
- [58] ———, *\mathcal{H}_2 guaranteed cost control for uncertain discrete-time linear systems*, International Journal of Control **57** (1993), no. 4, 853–864.
- [59] ———, *Convex analysis of output feedback control problems: Robust stability and performance*, IEEE Transactions on Automatic Control **41** (1996), no. 7, 997–1003.
- [60] L. Ghaoui e S. Niculescu, *Advances in linear matrix inequality methods in control*, SIAM, Philadelphia, 2000.
- [61] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, Massachussetts, NY, 1989.
- [62] E. N. Gonçalves, R. M. Palhares, e R. H. C. Takahashi, *Otimização multi-objetivo aplicada a síntese de controladores robustos $\mathcal{H}_2/\mathcal{H}_\infty$ por realimentação de estado*, Congresso Brasileiro de Automática (2004).

- [63] ———, *Síntese de controladores robustos $\mathcal{H}_2/\mathcal{H}_\infty$ por realimentação dinâmica da saída para sistemas com incertezas politópicas*, Congresso Brasileiro de Automática (2004), 1–6.
- [64] S. Gutman e E. I. Jury, *A general theory for matrix root clustering in subregions of the complex plan*, IEEE Transactions on Automatic Control **26** (1981), 853–863.
- [65] S. Gutman e G. Leitmann, *Stabilizing feedback control for dynamical systems with bounded uncertainty*, Conference on Decision and Control (1976), 94–99.
- [66] W. M. Haddad e D. S. Bernstein, *Controller design with regional pole constraints*, IEEE Transactions on Automatic Control **37** (1992), no. 1, 54–69.
- [67] B. Halder e T. Kailtah, *LMI based design of mixed $\mathcal{H}_2/\mathcal{H}_\infty$ controllers: The state feedback case*, Proceedings of the American Control Conference **3** (1999), 1866–1870.
- [68] A. Herreros, E. Baeyens, e J. R. Perán, *MRCDD: a genetic algorithm for multiobjective robust control design*, Engineering Applications of Artificial Intelligence **15** (2002), no. 3, 285–301.
- [69] J. H. Holland, *Outline for a logical theory of adaptative systems*, Journal of the Association for Computer Machinery **9** (1962), no. 3, 297–314.
- [70] ———, *Adaptation in natural and artificial systems*, The University of Michigan Press, 1975.
- [71] J.C. Hung e B.C. Chen, *Genetic algorithm approach to fixed-order mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal deconvolution filter designs*, IEEE Transactions on Signal Processing **48** (2000), no. 12, 3451–3461.
- [72] K. J. Hunt, M. Sebek, e V. Kucera, *Polynomial solution of the standard multivariable \mathcal{H}_2 -optimal control problem*, IEEE Trans. on Automatic Control **39** (1993), 1502–1507.
- [73] Y. Jing, W. Pan, e G. M. Dimirovski, *Reduced-order \mathcal{H}_∞ controllers based on improved genetic algorithm*, XVI Triennial World Congress of the International Federation of Automatic Control (2005).
- [74] A.H. Jones, N. Ajilouni, Y.C. Lin, S.B. Kenway, e M. Uzam, *Genetic design of robust PID plus feedforward controllers*, IEEE Conference on Emerging Technologies and Factory Automation **1** (1996), no. 1, 267–271.
- [75] R. E. Kalman e J. E. Bertram, *Control system analysis and design via the second method of Lyapunov, I: continuous time systems*, IEEE Transactions on Automatic Control **4** (1959), no. 3, 112.
- [76] S. Kanev, C. Scherer, M. Verhaegen, e B. de Schutter, *Robust output-feedback controller design via local bmi optimization*, Automatica **40** (2004), no. 7, 1115–1127.
- [77] P. P. Khargonekar, I. R. Petersen, e K. Zhou, *Robust stabilization of uncertain linear systems: quadratic stabilizability and \mathcal{H}_∞ control theory*, IEEE Transactions on Automatic Control **35** (1990), no. 3, 356–361.

- [78] J. R. Koza, *Genetic programming - on the programming of computers by means of natural selection*, The MIT Press, Cambridge, MA, 1992.
- [79] R.A. Krohling, L.S. Coelho, e A.A.R. Coelho, *Evolution strategies for synthesis of mixed $\mathcal{H}_2/\mathcal{H}_\infty$ fixed structure controllers*, 14th IFAC World Congress (1999), 471–476.
- [80] H. Kwakernaak, *\mathcal{H}_2 -optimization - theory and applications to robust control design*, *Annual Reviews in Control* **26** (2002), 45–56.
- [81] J. Lam e H. K. Tam, *Robust output feedback pole assignment using genetic algorithm*, *Proceedings of the Institution of Mechanical Engineers* **214** (2000), 327–334.
- [82] F. Leibfritz e W. Lipinski, *Description of the benchmark examples in COM-Plib 1.0*, Technical Report (2003), Obtido em maio de 2005 no endereço <http://www.complib.de/>.
- [83] G. Leitman, *Guaranteed ultimate boundedness for a class of uncertain linear dynamical systems*, *IEEE Transactions on Automatic Control* **23** (1978), 1109–1110.
- [84] D. G. Luenberger, *Linear and nonlinear programming*, Addison-Wesley, 1984.
- [85] K.F. Man, K.S. Tang, e S. Kwong, *Genetic algorithms: Concepts and applications*, *IEEE Transactions on Industrial Electronics* **43** (1996), no. 5, 519–534.
- [86] J. M. Martínez, *Métodos computacionais de otimização*, XIX Congresso Nacional de Matemática Aplicada e Computacional, 1996.
- [87] F. Le Mauff e G. Duc, *Design of a controller for a steam generator of a power plant using robust control and genetic algorithm*, *IEEE Conference on Control Applications* **2** (2002), 1050–1055.
- [88] ———, *Design of fixed structure controllers with respect to various specifications using robust control and genetic algorithm*, *Proceedings of the American Control Conference* **6** (2002), 4994–4999.
- [89] G. Meinsma, *Frequency domain methods in \mathcal{H}_∞ control*, Ph.D. thesis, University of Twente, The Netherlands, 1993.
- [90] ———, *On the standard \mathcal{H}_2 problem*, *Proceedings of the 3rd. IFAC Symposium on Robust Control Design* (2000).
- [91] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, Springer Verlag, Berlin, 1996.
- [92] Z. Michalewicz e D. B. Fogel, *How to solve it: modern heuristics*, Springer Verlag, Berlin, 2000.
- [93] Z. Michalewicz, T. D. Logan, e S. Swaminathan, *Evolutionary operators for continuous convex parameter spaces*, *Proceedings of the 3rd Annual Conference on Evolutionary Programming* (1994), 84–97.
- [94] S. O. R. Moheimani, *Guaranteed cost control and robust filtering for uncertain systems*, Ph.D. thesis, The University of New South Wales, January 1996.

- [95] A. Nemirovskii e P. Gahinet, *The projective method for solving linear matrix inequalities*, Proceedings of American Control Conference **1** (1994), 840–844.
- [96] I. Nesterov e A. S. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, SIAM, Philadelphia, 1994.
- [97] D. Neumann e H. X. de Araújo, *Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for uncertain systems under pole placement constraints using genetic algorithms and LMIs*, Proceedings of the 2004 IEEE International Symposium on Intelligent Control (2004), 460–465.
- [98] N. Ohse e K. Ono, *Guaranteed cost controllers under pole placement constraints for uncertain linear systems: An LMI approach*, International Conference on Control Applications **1** (1998), 123–128.
- [99] A. Packard e J. C. Doyle, *Quadratic stability with real and complex perturbations*, IEEE Transactions on Automatic Control **35** (1990), no. 2, 198–201.
- [100] G.K.M. Pedersen, A.S. Langballe, e R. Wisniewski, *Synthesizing multi-objective $\mathcal{H}_2/\mathcal{H}_\infty$ dynamic controller using evolutionary algorithms*, Proceedings of the 15th IFAC World Congress (2002).
- [101] G.J. Pereira e H.X. de Araújo, *Robust output feedback controller design via genetic algorithms and LMIs: The mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problem*, Proceedings of the American Control Conference 2004 **4** (2004), no. 30, 3309–3314.
- [102] I. R. Petersen, *A stabilization algorithm for a class of uncertain systems*, Systems and Control Letters **8** (1987), no. 4, 351–357.
- [103] ———, *Guaranteed cost LQG control of uncertain linear systems*, IEE Proceedings in Control Theory and Applications, Part D **142** (1995), no. 2, 95–102.
- [104] I. R. Petersen e C. V. Hollot, *A Riccati equation approach to the stabilization of uncertain linear systems*, Automatica **22** (1986), no. 4, 397–412.
- [105] I. R. Petersen e D. C. McFarlane, *Robust state estimation for uncertain systems*, Proceedings of the 30th Conference on Decision and Control **3** (1991), 2630–2631.
- [106] ———, *Optimal guaranteed cost control of uncertain linear systems*, Proceedings of the 1992 American Control Conference (1992), 2929–2930.
- [107] ———, *Optimizing the guaranteed cost in the control of uncertain linear systems*, Robustness of Dynamic Systems with Parameter Uncertainties: Proceedings of the International Workshop on Robust Control (M. Mansour, S. Balemi, e W. Truöl, eds.), Birkhäuser, 1992, pp. 241–250.
- [108] ———, *Optimal guaranteed cost control and filtering for uncertain linear systems*, IEEE Transactions on Automatic Control **39** (1994), no. 9, 1971–1977.
- [109] W. H. Press, S. A. Teukolsky, W. T. Vetterling, e B. P. Flannery, *Numerical recipes in pascal: The art of scientific computing*, Cambridge University Press, 1994.
- [110] I. Rechenberg, *Evolution strategy*, Computational Intelligence: Imitating Life (J. Zurada, R. Marks, e C. Robinson, eds.), IEEE Press, 1994, pp. 147–159.

- [111] J. Renegar, *A mathematical view of interior-point methods in convex optimization*, MPS/SIAM Publications Series on Optimization, Philadelphia, 2001.
- [112] M. A. Rotea e P. P. Khargonekar, *Stabilization of uncertain systems with norm bounded uncertainty: a control Lyapunov approach*, SIAM Journal of Control and Optimization **27** (1989), no. 6, 1462–1476.
- [113] A. Saberi, P. Sannuti, e B. M. Chen, *\mathcal{H}_2 optimal control*, Prentice Hall International, London, 1995.
- [114] R. Salomon, *Evolutionary algorithms and gradient search: Similarities and differences*, IEEE Transactions on Evolutionary Computation **2** (1998), no. 2, 45–55.
- [115] C. Scherer, *Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control*, Volume of the Special Contributions to the ECC'95 (A. Isidori, ed.), Springer Verlag, 1995, pp. 173–216.
- [116] C. Scherer, P. Gahinet, e M. Chilali, *Multiobjective output-feedback control via LMI optimization*, IEEE Transactions on Automatic Control **42** (1997), no. 7, 896–911.
- [117] C. Scherer e S. Weiland, *Lecture notes DISC course on linear matrix inequalities in control*, obtido em <http://www.dsc.tudelft.nl/cscherer/2416/lmi.pdf> em abril de 2005, 1999.
- [118] H.-P. Schwefel, *Evolution and optimum seeking*, John Wiley and Sons Inc., New York, NY, 1995.
- [119] ———, *Evolutionary computation - a study on collective learning*, Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (N. Callaos, C. M. Khoong, e E. Cohen, eds.), vol. 2, International Institute of Informatics and Systems, Orlando, FL, 1997, pp. 198–205.
- [120] M. Srinivas e L.M. Patnaik, *Genetic algorithms: A survey*, IEEE Computer **27** (1994), no. 6, 17–26.
- [121] A. Stoorvogel, *The robust \mathcal{H}_2 problem: a worst case design*, IEEE Transactions on Automatic Control **38** (1993), no. 9, 1358–1370.
- [122] R. Storn e K. Price, *Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report - TR-95-012 - International Computer Science Institute (1995).
- [123] ———, *Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces.*, Journal of Global Optimization **11** (1997), no. 4, 341–359.
- [124] K. Takaba e T. Katayama, *\mathcal{H}_2 output feedback control for descriptor systems*, Automatica **34** (1998), no. 7, 841–850.
- [125] R. H. C. Takahashi, D. C. W. Ramos, e P. L. D. Peres, *Robust control synthesis via a genetic algorithm and LMIs*, XV Triennial World Congress of the IFAC (2002).
- [126] R. H. C. Takahashi, J. A. Vasconcelos, J. A. Ramírez, e L. Krahenbuhl, *A multiobjective methodology for evaluating genetic operators*, IEEE Transactions on Magnetics, Part 1 **34** (2003), no. 3, 1321–1324.

-
- [127] D. Thierens e D. Goldberg, *Convergence models of genetic algorithm selection schemes*, Proceedings of Parallel Problem Solving from Nature III **3** (1994), 119–129.
- [128] TQ Education and Training Ltd, <http://www.tq.com/>, *Ce108 coupled drives apparatus*.
- [129] J. Vesterstrom e R. Thomsen, *A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems*, IEEE Proceedings of the 2004 Congress on Evolutionary Computation **2** (2004), 1980–1987.
- [130] D. Wolpert, *No free lunch theorems for optimization*, IEEE Transactions on Evolutionary Computation **1** (1997), no. 1, 67–82.
- [131] S. J. Wright, *Primal-dual interior-point methods*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.
- [132] L. Xie, *Control and filtering of systems with parameter uncertainty*, Ph.D. thesis, University of New Castle, 1991.
- [133] L. Xie e C. E. de Souza, *Robust stabilization of uncertain systems and \mathcal{H}_∞ optimal control*, IEE Proceedings Part D **138** (1991), no. 5, 479–483.
- [134] L. Xie, C. E. de Souza, e M. Fu, *\mathcal{H}_∞ estimation for discrete-time linear uncertain systems*, International Journal of Robust and Nonlinear Control **1** (1991), 111–123.
- [135] G. Zames, *Feedback and optimal sensitivity: model reference transformation, multiplicative seminorms and approximate inverses*, IEEE Transactions on Automatic Control **26** (1981), no. 2, 301–320.
- [136] K. Zhou e K. Glover, *Robust and optimal control*, Prentice Hall, New Jersey, 1996.