

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Engenharia de
Eletricidade

*Convergência de Algoritmo Genético Hierárquico
para Recuperação da Malha LQR por
Controladores LQG/LTR*

Patrícia Helena Moraes Rêgo

São Luís
2007

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Engenharia de
Eletricidade

*Convergência de Algoritmo Genético Hierárquico
para Recuperação da Malha LQR por
Controladores LQG/LTR*

Patrícia Helena Moraes Rêgo

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da UFMA como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

**São Luís
2007**

Rêgo, Patrícia Helena Moraes

Convergência de Algoritmo Genético Hierárquico para Recuperação da Malha LQR por Controladores LQG/LTR / Patrícia Helena Moraes Rêgo. - São Luís, 2007.

140f.:il.

Dissertação (Mestrado em Engenharia de Eletricidade) - Centro de Ciências Exatas e Tecnologia, Universidade Federal do Maranhão, 2007.

1. Hierarquia de Algoritmos Genéticos 2. Análise de Convergência 3. Regulador Linear Quadrático 4. Recuperação da Malha de Transferência. I.Título.

CDU: 681.511.2

**Convergência de Algoritmo Genético Hierárquico
para Recuperação da Malha LQR por
Controladores LQG/LTR**

Patrícia Helena Moraes Rêgo

Submetida em 03/08/2007

BANCA EXAMINADORA

Prof. João Viana da Fonseca Neto

Dr. em Engenharia Elétrica

Orientador

Prof. Sebastian Yuri C. Catunda

Dr. em Engenharia Elétrica

Examinador Interno

Prof. Raimundo Carlos S. Freire

Dr. em Engenharia Elétrica

Examinador Externo

"A ciência vive de sucessivas soluções dadas a porquês cada vez mais sutis, cada vez mais próximos à essência dos fenômenos". **Pasteur**

Dedicatória

À minha família, em especial à minha mãe, Enilde Rodrigues Moraes Rêgo,
companheira em todas as minhas lutas;

Aos meus amigos, em especial ao prof. Zalmir Paixão Mendes.

Agradecimentos

À Deus pela minha existência, inspiração e por me conceder a graça de trilhar este caminho;

Ao Prof^o. orientador João Viana da Fonseca Neto, pelo aprendizado, incentivo, companheirismo e pela sua forma peculiar de orientar;

Aos professores Sebastian Yuri Catunda e Allan Kardec pela atenção e por suas profícuas críticas;

Ao PPGEE pela oportunidade, receptividade, atendimento;

Aos colegas do LCP Ivanildo, Fábio, João Inácio, Lucilene, Marlon, Joaquim, Márcio, Pedro Turibe, Jorge Henrique, Antônio José, Samy, Aline, pela convivência e pelos saudáveis e saudosos momentos de aprendizagem, reflexão e descontração vivenciados ao longo desta árdua, porém, gratificante jornada;

Aos professores que aceitaram a fazer parte da banca examinadora;

Aos professores do DEMATI da UEMA, em especial Francisco Francineide Grangeiro, Axel Winterhalder, Cristovam Dervalmar e Jackson Martins Reis pela colaboração que me prestaram;

À FAPEMA pelo apoio financeiro;

À todas as pessoas que contribuíram de forma direta ou indiretamente para a realização deste trabalho.

RESUMO

Propõe-se neste trabalho os modelos e a análise de convergência de um algoritmo genético hierárquico para recuperação da malha de projeto do regulador linear quadrático por controladores LQG/LTR (Linear Quadratic Gaussian/Loop Transfer Recovery). Os modelos dedicam-se à busca das matrizes de ponderações e covariâncias dos índices de desempenho dos projetos de controladores LQR (Linear Quadratic Regulator) e LQG (Linear Quadratic Gaussian), respectivamente, e à seleção de matrizes de ajuste para o ganho de recuperação da malha do projeto LQR. O objetivo da análise de convergência é promover melhorias no desempenho do algoritmo no sentido de gerar soluções satisfatórias e acelerar o tempo de convergência. O desempenho do algoritmo é avaliado em relação aos efeitos de uma estratégia elitista incorporada ao algoritmo e à variações nos valores de determinados parâmetros do algoritmo. A metodologia proposta é avaliada em um sistema dinâmico multivariável que representa uma aeronave.

Palavras-Chave: Hierarquia de Algoritmos Genéticos, Análise de Convergência, Regulador Linear Quadrático, Recuperação da Malha de Transferência.

ABSTRACT

In this work are proposed models and a convergence analysis of a hierarchical genetic algorithm for the linear quadratic regulator design loop recovery through LQG/LTR controllers. Models are oriented to the weighting and covariance matrices searching of the performance indices of the LQR and LQG design, respectively, and to the selection of the matrices for the LQR design loop recovery gain. The convergence analysis aims at promoting the enhancement of the algorithm performance, as well as to generate satisfactory solutions and speed up the convergence time. The algorithm performance is evaluated with respect to the effects of an elitist strategy embodied into the algorithm and to variations in the values of some given parameters of the algorithm. The proposed methodology is evaluated in a multi-variable dynamical system representing an aircraft.

Keyword: Hierarchy of Genetic Algorithms, Convergence Analysis, Linear Quadratic Regulator, Loop Transfer Recovery.

Lista de Tabelas

4.1	Ganhos do Controlador LQR.	41
4.2	Ganhos do Estimador de Espaço de Estado.	42
4.3	Parâmetros de ajuste v e matrizes de covariâncias para a recuperação da malha de transferência	45
4.4	Valores Singulares Máximos, Planta, Projetos LQR, LQG e LTR.	46
4.5	Valores Singulares Mínimos, Planta, Projetos LQR, LQG e LTR .	47
6.1	Parâmetros utilizados no algoritmo genético	59
6.2	Parâmetros da População Inicial-Matriz Q	59
6.3	Parâmetros da População Inicial-Matriz R	59
6.4	Alelos que seguem a estratégia de mutação proposta no algoritmo	59
6.5	Fatores de mutação dos alelos selecionados	60
6.6	Matrizes Q e R e seus respectivos traços geradas pelo AG-LQR $_{Q,R}$ nos estágios 1-5	73
6.7	Matrizes Q e R e seus respectivos traços geradas pelo AG-LQR $_{Q,R}$ nos estágios 6-10	74
6.8	Matrizes Q e R e seus respectivos traços geradas pelo AG-LQR $_{Q,R}$ nos estágios 11-14	75
6.9	Autovalores das Matrizes Q e R geradas pelo AG-LQR $_{Q,R}$ nos quatorze estágios	76
6.10	Autovalores, sensibilidades e ganho do controlador obtido pelo AG-LQR $_{Q,R}$ sem estratégia elitista	79
6.11	Autovalores, sensibilidades e ganho do controlador obtido pelo AG-LQR $_{Q,R}$ com estratégia elitista	79
6.12	Semente-Melhor indivíduo-Desempenho.	81
6.13	Estatísticas da função de <i>fitness</i> F_{max}^{desemp} -Condição inicial (Ξ_o^1, Θ_o^1) .	84

6.14	Estatísticas da função de <i>fitness</i> F_{min}^{desemp} -Condição inicial (Ξ_o^1, Θ_o^1) .	85
6.15	Estatísticas da função de <i>fitness</i> F_{max}^{desemp} -Condição inicial (Ξ_o^2, Θ_o^2) .	86
6.16	Estatísticas da função de <i>fitness</i> F_{min}^{desemp} -Condição inicial (Ξ_o^2, Θ_o^2) .	86
6.17	Estatísticas da função de <i>fitness</i> F_{max}^{desemp} -Condição inicial (Ξ_o^3, Θ_o^3) .	87
6.18	Estatísticas da função de <i>fitness</i> F_{min}^{desemp} -Condição inicial (Ξ_o^3, Θ_o^3) .	87
6.19	Estatísticas da função de <i>fitness</i> F_{max}^{desemp} -Condição inicial (Ξ_o^4, Θ_o^4) .	88
6.20	Estatísticas da função de <i>fitness</i> F_{min}^{desemp} -Condição inicial (Ξ_o^4, Θ_o^4) .	89
6.21	Número de indivíduos mutados-Diversidade genética- $p_m = 0.01$.	90
6.22	Número de indivíduos mutados-Diversidade genética- $p_m = 0.05$.	90
6.23	Número de indivíduos mutados-Diversidade genética- $p_m = 0.1$.	91
6.24	Número de indivíduos mutados-Diversidade genética- $p_m = 0.2$.	91
6.25	Função de desempenho-Diversidade genética- $n_{indiv} = 4$.	92
6.26	Função de desempenho-Diversidade genética- $n_{indiv} = 8$.	93
6.27	Função de desempenho-Diversidade genética- $n_{indiv} = 9$.	93
6.28	Função de desempenho-Diversidade genética- $n_{indiv} = 14$.	94
A.1	Parâmetros duais do LQR e do FK	100
C.1	Autovalores do sistema teste	115
C.2	Especificações de Projeto de Autoestrutura e Ganhos do Controlador base.	116

Lista de Figuras

2.1	Diagrama em blocos reduzido LQG/LTR para recuperação da malha LQR	25
2.2	Estrutura hierárquica de algoritmos para recuperação da malha LQR.	28
3.1	Modelo genético artificial de ordem 2 para as matrizes Q e R . . .	30
3.2	Ciclo de Busca Genética.	31
4.1	Diagrama de Valores Singulares - Projeto LQR - Escolha das matrizes de ponderação.	40
4.2	Diagrama de Valores Singulares - Projetos LQG e LQR	43
4.3	Recuperação LQG/LTR das propriedades de robustez.	46
4.4	Projeto LQG/LTR - Erros de Valores Singulares.	47
6.1	Função de Fitness Objetivo Normalizada - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	61
6.2	Restrições da Função de Fitness - Sensibilidades e Autovalores 1 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	62
6.3	Restrições da Função de Fitness - Sensibilidades e Autovalores 2 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	63
6.4	Restrições da Função de Fitness - Sensibilidades e Autovalores 3 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	64
6.5	Restrições da Função de Fitness - Sensibilidades e Autovalores 4 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	65

6.6	Restrições da Função de Fitness - Sensibilidades e Autovalores 5 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	66
6.7	Restrições da Função de Fitness - Sensibilidades e Autovalores 6 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	67
6.8	Função de Fitness Objetivo Normalizada - Super Indivíduo ao longo de 200 gerações - Sistema de Ordem 6.	69
6.9	Restrições da Função de Fitness - Sensibilidades e Autovalores 1 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	69
6.10	Restrições da Função de Fitness - Sensibilidades e Autovalores 2 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	70
6.11	Restrições da Função de Fitness - Sensibilidades e Autovalores 3 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	70
6.12	Restrições da Função de Fitness - Sensibilidades e Autovalores 4 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	71
6.13	Restrições da Função de Fitness - Sensibilidades e Autovalores 5 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	71
6.14	Restrições da Função de Fitness - Sensibilidades e Autovalores 6 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.	72
6.15	Curvas de Resposta ao impulso para as saídas 1 e 2.	77
6.16	Curvas de Resposta ao impulso para as saídas 3 e 4.	77
6.17	Melhor indivíduo da população inicial-Semente 5	82
6.18	Melhor indivíduo da população inicial-Semente 7	82
6.19	Melhor indivíduo da população inicial-Semente 8	83
A.1	Diagrama em blocos do sistema dinâmico estocástico	98
A.2	Diagrama em blocos do Filtro de Kalman	99

B.1	Diagrama em blocos do controlador com o Filtro de Kalman. . . .	102
B.2	Esquema Geral LQG/LTR para recuperação da Malha LQR. . . .	104

Lista de Abreviaturas e Siglas

AG	Algoritmo Genético
AE	Autoestrutura
AAE	Alocação de Autoestrutura
EAR	Equação Algébrica de <i>Riccati</i>
FK	Filtro de <i>Kalman</i>
LQ	<i>Linear Quadratic</i> (Linear Quadrático)
LQG	<i>Linear Quadratic Gaussian</i> (Gaussiano Linear Quadrático)
LQR	<i>Linear Quadratic Regulator</i> (Regulador Linear Quadrático)
LTR	<i>Loop Transfer Recovery</i> (Recuperação da Malha de Transferência)
LQG/LTR	<i>Linear Quadratic Gaussian / Loop Transfer Recovery</i>
MIMO	<i>Multiple-Input and Multiple-Output</i> (Múltiplas-Entradas e Múltiplas-Saídas)
SISO	<i>Single-Input and Single-Output</i> (Simples-Entrada e Simples-Saída)

Sumário

1	Introdução	12
1.1	Alocação de Autoestruturas	14
1.2	Computação Evolutiva	15
1.3	Motivação	16
1.4	Organização do Trabalho	16
1.5	Artigo Aceito	17
2	Modelo Hierárquico para Recuperação da Malha LQR	18
2.1	Modelo de Sintonia do Ganho LQR	20
2.1.1	Modelo de Busca $LQR_{Q,R}$	20
2.1.2	O Problema LQR	21
2.1.3	A Solução do LQR	21
2.2	Modelo de Sintonia do Ganho LQG	21
2.2.1	Modelo de Busca $LQG_{\Xi,\Theta}$	21
2.2.2	O Problema LQG	22
2.2.3	A Solução do LQG	22
2.3	Modelo de Sintonia do Ganho LQG/LTR	23
2.3.1	Modelo de Busca LTR	23
2.3.2	O Problema da Recuperação LTR	25
2.3.3	O Problema LTR - Recuperação na entrada	25
2.3.4	A Solução LTR - Recuperação na entrada	26
2.3.5	O Problema LTR - Recuperação na Saída	26
2.3.6	A Solução LTR - Recuperação na Saída	27
2.4	Conclusão	27

3	Modelo do AG-LQR_{Q,R}	29
3.1	Conceitos Básicos e Definições	29
3.2	Modelos Genético Artificial	31
3.2.1	Modelo das matrizes Q e R	32
3.2.2	Modelo da População QR	32
3.2.3	Modelo da População Inicial	33
3.2.4	Modelo de Avaliação dos Cromossomos	34
3.2.5	Modelos de Operações Genéticas	34
3.3	Conclusão	37
4	Projeto LQG/LTR	38
4.1	Ajuste do Ganho LQG/LTR	39
4.2	Projeto LQR/LQG	40
4.2.1	Projeto LQR	40
4.2.2	Projeto LQG	41
4.3	Projeto de Recuperação da Malha LQR	43
4.3.1	Mapeamento $F_{\Xi\Theta}$	44
4.3.2	Mapeamento F_{LTR}	44
4.3.3	Mapeamento $F_{LTR/LQG}$	44
4.3.4	Mapeamento $F_{LTR/LQR}$	44
4.4	Conclusão	48
5	Modelo do AG-LQG/LTR	49
5.1	Modelo Cromossômico	50
5.1.1	Modelagem Escalar v_i	50
5.2	Geração da População Inicial	51
5.3	Estrutura de <i>Fitness</i>	51
5.3.1	Cálculo Pré- <i>Fitness</i>	53
5.3.2	Função de Desempenho	53
5.4	Métodos de Seleção	54
5.5	Operações Genéticas	54
5.5.1	Operação de Seleção	55
5.5.2	Operação de <i>Crossover</i>	55
5.5.3	Operação de Mutação	55

5.6	Conclusão	56
6	Análise de Convergência do AG Hierárquico	57
6.1	Desempenho do AG-LQR _{Q,R}	57
6.1.1	Parâmetros Utilizados na Implementação do AG-LQR _{Q,R}	58
6.1.2	Evolução do AG-LQR _{Q,R} sem Estratégia Elitista	60
6.1.3	Evolução do AG-LQR _{Q,R} com Estratégia Elitista	67
6.1.4	Busca das Matrizes Q e R	78
6.2	Desempenho do AG-LQG/LTR	80
6.2.1	Análise do AG-LQG/LTR com variações na semente	80
6.2.2	Análise do AG-LQG/LTR com variações nas condições iniciais	83
6.2.3	Análise do AG-LQG/LTR com variações na probabilidade de mutação	89
6.2.4	Análise do AG-LQG/LTR com variações no tamanho da população	91
6.3	Conclusão	94
7	Conclusão	95
7.1	Trabalhos Futuros	96
A	Projeto do Filtro de Kalman	97
B	Recuperação dos Ganhos de Malha na Entrada e Saída	101
B.1	Recuperação do Ganho de Malha na Entrada	105
B.1.1	Recuperação da malha	106
B.1.2	Mostrando a Recuperação	107
B.2	Recuperação do Ganho de Malha na Saída	109
B.2.1	Recuperação da malha	110
B.2.2	Mostrando a Recuperação	111
C	O Sistema Dinâmico Multivariável Teste	114
D	Definição de um AG Canônico	117
D.1	Definição geral de um AG canônico	118
D.1.1	Operadores genéticos	119

D.1.2	Variantes de AGs Canônicos	120
E	Modelo do AG-LQG$_{\Xi, \Theta}$	121
E.1	Modelo das Matrizes Ξ e Θ	122
F	Algoritmos AG-LQR$_{Q,R}$	123
F.1	Programa Principal do AG para o LQR	123
F.1.1	Inicializações	123
F.1.2	População Inicial	124
F.1.3	Ciclo de Busca	124
F.2	Algoritmo da População Inicial	124
F.2.1	Função Semente	124
F.2.2	Geração Aleatória das Matrizes Q e R	125
F.2.3	Conversão das Matrizes Q e R em Simétricas	126
F.2.4	Construção da Matriz QR que contém os indivíduos	126
F.3	Algoritmo de Avaliação dos Cromossomos	127
F.3.1	Recuperação da Matriz Simétrica Q	127
F.3.2	Recuperação da Matriz Simétrica R	127
F.3.3	Parâmetros da Função de Fitness	128
F.4	Algoritmo de Operações Genéticas	128
F.4.1	Algoritmo de Seleção	129
F.4.2	Algoritmo de <i>Crossover</i>	129
F.4.3	Algoritmo de Mutação	130
F.5	Armazenamento do Melhor Indivíduo de cada Geração	131
G	Algoritmos AG-LQG/LTR	134
G.1	Programa Principal do AG para o LQG/LTR	134
G.1.1	Parâmetros	134
G.1.2	População Inicial	134
G.1.3	Ciclo de Busca	134
G.2	Algoritmo da População Inicial	134
G.2.1	Geração da Semente	135
G.2.2	Geração da População Inicial	135
G.3	Cálculo da função de fitness	135

G.4	Algoritmo de Seleção	135
G.5	Algoritmo de <i>Crossover</i>	135
G.6	Algoritmo de Mutação	136
Referências Bibliográficas		137

CAPÍTULO 1

INTRODUÇÃO

O controle linear quadrático (LQ) é um método de controle moderno que surgiu aproximadamente em 1960. Este método envolve alguns casos especiais, tais como: Projeto Regulador Linear Quadrático (LQR); Projeto Gaussiano Linear Quadrático (LQG) e Projeto Recuperação de Malha de Transferência (LTR), que podem ser considerados como metodologias para sintonia de controladores e observadores de estado.

O regulador linear quadrático é uma técnica de controle ótimo no domínio do tempo que tem qualidade de estabilidade robusta, margem de fase de 60 graus e margem de ganho infinito, (M.Maciejowski 1989). A lei de controle estabelecida minimiza um índice de desempenho que tem como parâmetros matrizes de ponderação do estado e de controle. A seleção das matrizes de ponderação constitui uma dificuldade para determinação de uma lei de controle, (Stein 1979), (Medanic *et al.* 1988), (Kawasaki and Shimemura 1983), (Graupe 1972) e (Harvey and Stein 1978), que satisfaz às restrições de projeto, tal como: alocação de autoestrutura. As referências (Fonseca Neto 2000) e (Teixeira Ferreira 2004) apresentam uma alternativa para superar esta dificuldade que são modelos de busca baseados em algoritmos genéticos.

O LQR fornece a vantagem de promover robustez garantida, mas os estados não são completamente disponíveis para realimentação. Por outro lado, o LQG é um sistema de controle estocástico resultante da combinação do LQR e o filtro de Kalman (FK) que contribui para viabilizar as implementações em tempo real de compensadores ótimos. A força do LQG consiste na disponibilização de todos os

estados para a realimentação ótima, e a desvantagem é que as propriedades de robustez do LQR são perdidas pela inserção do observador de estado. Este problema é contornado com o projeto de sistema de controle LQG/LTR, ou simplesmente LTR, que recupera a robustez garantida do LQR.

A metodologia de projeto LTR compõe-se de dois passos: no primeiro, no que se refere às especificações de estabilidade e desempenho, define-se a chamada malha objetivo; no segundo, recupera-se as características de resposta em frequência dessa malha ajustando de forma empírica um determinado parâmetro. Existem dois procedimentos de recuperação: no primeiro, a malha é aberta na saída fixando-se o ganho do filtro de Kalman e variando-se o ganho de realimentação do regulador; no segundo, a malha é aberta na entrada fixando-se o ganho do regulador linear quadrático e variando-se o ganho do filtro.

Apesar da metodologia LTR promover a recuperação de robustez do LQR, não mostra como sintonizar de forma automática os ganhos das malhas de controle LQR, LQG e LTR. Esta metodologia, por meio de um procedimento assintótico, recupera as características de resposta em frequência da malha LQR ajustando o parâmetro normalmente por tentativa e erro. Propomos uma estrutura de otimização baseada em técnicas de computação evolutiva, como algoritmos genéticos, para o ajuste eficiente desse parâmetro.

O número de aplicações de projeto de sistemas de controle LQG/LTR frente às características de estabilidade e desempenho robustas garantidas por este tipo de controle justifica o interesse em desenvolver pesquisas voltadas para metodologias de controle baseadas no projeto LQR. Por exemplo, no setor industrial, (Weerasooriya 1995), utilidade pública geração de energia, (Lee 1996), (Rab-Alibeik 2003), atenuação de ruídos auditivos, devido a perturbações externas dos automóveis e outros, (José Paulo F. *et al.* 2004), controle de máquinas de corrente contínua e controle de processos, (Gasem 1995). As referências (Lahdhiri 1993), (Wu 1995), (Paschall 1994), (Bailey 1990), (Jafar *et al.* 2006), (Kishor *et al.* 2004) e (Keller 2005) e as citadas mostram que a metodologia LQG/LTR está sendo incorporada nos diversos segmentos produtivos e estratégicos das sociedades industrializadas. Em face do exposto acima e constatação pelas referências citadas, conclui-se que o controle LQG/LTR encontra-se na fase de realização tecnológica.

O principal objetivo deste trabalho é apresentar os modelos e a análise de

convergência de um algoritmo genético hierárquico dedicado à recuperação da malha do projeto LQR por meio de controladores LQG/LTR. A análise de convergência é realizada em relação aos efeitos de uma estratégia elitista incorporada ao algoritmo e à influência de determinados parâmetros do algoritmo no comportamento da função de fitness, na diversidade genética, quantidade de soluções factíveis, tempo de convergência, e outros. Pesquisa recente sobre convergência de AG-LQR_{Q,R} fundamentada em estatísticas descritivas de métricas de fitness, encontra-se em (Wolff 2006). O algoritmo genético hierárquico é composto por modelos para a sintonia das malhas de controle dos projetos LQR, LQG e LQG/LTR. A proposta de um Modelo Hierárquico de algoritmos para o projeto de controladores LQR, LQG e LTR fundamentados em otimização combinatória e algoritmos genéticos é apresentada em (Brito Filho 2006).

As próximas seções abordam brevemente o problema de Alocação de Autoestruturas, enfocando sua relevância no contexto da teoria de controle e os métodos propostos utilizados para sua solução: o projeto LQR e computação evolutiva. Segue-se então, com a motivação que nos conduziu para desenvolver esta pesquisa e, finalmente, a organização deste trabalho.

1.1 Alocação de Autoestruturas

A alocação de autovalores e autovetores, chamada alocação de autoestrutura (AAE), faz-se necessária devido à influência que os autovalores e autovetores exercem na resposta temporal de sistemas dinâmicos multivariáveis: os autovalores são responsáveis pela estabilidade e taxa de crescimento ou decrescimento da resposta temporal dos sistemas, enquanto que os autovetores estão relacionados com a sua forma.

É importante ressaltar que somente a alocação dos autovalores não é suficiente para assegurar a robustez de malha fechada e os requisitos de desempenho para sistemas MIMO, ou seja, somente a alocação de autovalores não é suficiente para definir uma estrutura única de realimentação para o sistema. Isto acontece devido à liberdade promovida pela realimentação de estado ou saída em relação à seleção dos autovetores associados nos sistemas MIMO. Já para o caso de sistemas monovariáveis (SISO), os pólos de malha fechada definem uma única estrutura de

realimentação, principalmente quando estes têm liberdade para deslocar-se livremente dentro de uma certa área do plano complexo, (Liu 1998).

A AAE é oriunda de estudos de fenômenos observados em cordas oscilantes e da condução do som pelo ar, em que se verificou a necessidade de lidar-se com vários modos e frequências que ocorrem nestes sistemas. A viabilização da compreensão dos citados fenômenos ocorreu com a aplicação de métodos baseados em técnicas Lagrangeanas e Hamiltonianas, que produziram o conceito de coordenadas generalizadas, com suas formas e frequências modais associadas, (White 1991), o que conduziu ao entendimento de que a resposta de qualquer sistema oscilante pode ser decomposta em um conjunto de formas e frequências modais.

1.2 Computação Evolutiva

A Computação Evolutiva nasceu de algumas idéias introduzidas nos trabalhos de (Friedberg 1958), (Box 1957) e outros. Porém, somente a partir da década de 70 com (Holland 1962), (Rechengerg 1965) e (Fogel 1962) que se verifica um crescimento do número de publicações e conferências sobre este campo de pesquisa. A própria denominação do termo foi criado em 1991, representando um conjunto de técnicas inspiradas na teoria evolutiva de Darwin: algoritmos genéticos, programação evolutiva e estratégias evolutivas.

A capacidade que os métodos de computação evolutiva possuem para resolver problemas de busca e otimização, principalmente aqueles de natureza combinatória, tornam estas abordagens bastante atrativas. Esses métodos não são limitados por hipótese restritivas acerca do espaço de busca, como continuidade, existência de derivadas, unimodalidade, etc., o que geralmente ocorre nos métodos clássicos de otimização (geralmente baseados em gradientes).

No campo da engenharia de controle, as técnicas de computação evolutiva podem ser utilizadas em várias metodologias, especialmente naquelas que exigem a otimização de determinados parâmetros a fim de se obter um desempenho desejado. Por exemplo, os algoritmos genéticos têm sido utilizados nos projetos de sistemas de controle LQR, LQG e LQG/LTR para determinação de parâmetros destes controladores a fim de alocar autoestruturas de sistemas dinâmicos multivariáveis, (Fonseca Neto 2000) e (Teixeira Ferreira 2004).

1.3 Motivação

O desenvolvimento deste trabalho foi impulsionado por dois motivos. Primeiro, no que se refere ao objetivo de controle, a alocação de Autoestrutura via o projeto LQR é viável para projetar sistemas MIMO estáveis com boas características de desempenho, pois permite especificações de autovalores e autovetores, os quais influenciam nas respostas de sistemas de malha fechada e o LQR garante robustez à estabilidade e desempenho. Por outro lado, as metodologias de controle LQG e LQG/LTR promovem a realimentação dos estados e recuperação de robustez. Segundo, no que diz respeito à aplicação de técnicas de computação evolutiva, os algoritmos genéticos realizam de forma automática e inteligente a sintonia dos ganhos de controladores LQR, LQG e LTR, superando assim os métodos por tentativas e erros.

1.4 Organização do Trabalho

O conteúdo deste trabalho está organizado da seguinte forma: O Capítulo 2 descreve os métodos desenvolvidos para a sintonia dos ganhos de controladores LQR, LQG e LQG/LTR por meio de uma hierarquia de modelos baseados em otimização combinatória e algoritmos genéticos. O Capítulo 3 apresenta o modelo de um algoritmo genético para realizar a busca das matrizes de ponderações do índice de desempenho quadrático do projeto LQR que são consideradas como parâmetros para sintonizar os ganhos do controlador LQR, tendo em vista a alocação de autoestrutura desejada. Inicia-se o Capítulo com conceitos básicos e definições de elementos que constituem um algoritmo genético dentro do contexto da modelagem genética das matrizes de ponderações. Os modelos que compõem o referido algoritmo são descritos desde as representações genético artificiais das matrizes de ponderação, passando pelos modelos da população, avaliação dos cromossomos até chegar nos modelos das Operações Genéticas. O Capítulo 4 é direcionado para o desenvolvimento do projeto LQG/LTR e tem por objetivo mostrar a necessidade de um terceiro algoritmo genético para concretizar a implementação do modelo hierárquico. O Capítulo 5 apresenta o modelo de um algoritmo genético para o ajuste do ganho de recuperação da malha LQR pela entrada. Inicialmente são discutidos os conceitos de indivíduo, alelo, geração e população da genética

artificial no contexto da modelagem do parâmetro de ajuste do projeto LTR. São apresentados os métodos para geração da população inicial, estrutura de *fitness*, métodos de seleção e o tipo de operações genéticas. Já no capítulo 6 é realizada uma análise do desempenho do algoritmo genético hierárquico em relação aos efeitos de uma estratégia elitista incorporada ao algoritmo e à variações nos valores de determinados parâmetros do algoritmo. O Capítulo 7 apresenta as conclusões do trabalho e propostas de trabalhos futuros. Por fim, os Apêndices são voltados para o desenvolvimento do projeto filtro de *Kalman*, recuperação dos ganhos de malha na entrada e saída, o sistema dinâmico teste, definição de um algoritmo genético canônico, modelo do AG-LQG $_{\Xi, \Theta}$ e rotinas do AG-LQR $_{Q, R}$ e AG-LQG/LTR.

1.5 Artigo Aceito

Fonseca Neto, J. V.; Abreu, Ivanildo Silva; Rêgo, Patrícia H. M.; Wolf, Marlon P. M. e Silva, Orlando Fonseca. Modelos e Convergência de um Algoritmo Genético para Alocação de Autoestrutura via LQR. *Revista IEEE-AL*, Abril 2007.

CAPÍTULO 2

MODELO HIERÁRQUICO PARA RECUPERAÇÃO DA MALHA LQR

A complexidade dos sistemas dinâmicos e a crescente demanda de requisitos de desempenho de um sistema de malha fechada exige o uso de controladores mais sofisticados e complexos. Por exemplo, os sistemas altamente não lineares normalmente requerem o uso de controladores mais complexos que aqueles lineares de baixa ordem quando devem ser satisfeitos objetivos que vão além de estabilidade. Porém, somente durante as últimas décadas têm desenvolvimentos significativos na área de computação habilitando novas abordagens para projeto de controladores para sistemas dinâmicos complexos: abordagens que tem por base a simulação de processos biológicos.

O controle inteligente é baseado em simulação de processos biológicos, (Passino 2004). As áreas de biologia, neurociência cognitiva, psicologia, evolução, e outras fornecem conceitos que podem ser usados para estabelecer a funcionalidade de sistemas de tomada de decisão sofisticados para automação de alta tecnologia. Além disso, existem situações onde humanos, ou grupos de humanos, tem conhecimento significativo em como resolver um problema de automação e este conhecimento pode ser automatizado em algoritmos computacionais devido a tarefa ser extremamente árdua.

Segundo (Meystel 1985), o controle Inteligente é um procedimento computacionalmente eficiente para atingir metas de um sistema dinâmico complexo de controle baseado em certos princípios de Inteligência Computacional. As metas

podem ser entendidas, por exemplo, como requisitos desejados de desempenho e estabilidade. A inteligência computacional busca por meio de técnicas inspiradas na Natureza (redes neurais, computação evolucionária, lógica fuzzy, sistemas especialistas) o desenvolvimento de sistemas inteligentes que imitam aspectos do comportamento humano.

No contexto deste trabalho, usa-se controle inteligente para estabelecer um conjunto de metodologias para o ajuste "inteligente" de ganhos de controladores LQR, LQG e LTR baseadas em modelos de otimização combinatória e técnicas de computação evolutiva, como algoritmos genéticos. Os modelos de otimização dedicam-se às seleções dos pares de matrizes de ponderações do estado e controle e matrizes de covariâncias da perturbação no estado e ruído de medida dos projetos LQR e LQG, respectivamente, que são consideradas como parâmetros para determinação das leis de controle do LQR e LQG, e à seleção de matrizes de ajuste para o ganho de recuperação da malha do projeto LQR. Normalmente, os ajustes desses parâmetros são feitos por métodos de tentativas e erros ou por métodos clássicos que funcionam bem para sistemas lineares de uma entrada e uma saída (SISO) e de baixa ordem; entretanto, eles não são apropriados para sistemas de entradas e saídas múltiplas (MIMO) ou sistemas de alta ordem que são característicos de muitos problemas de controle na vida real.

Sob o ponto de vista de objetivo de controle, as metodologias citadas acima são utilizadas para alocar Autoestruturas em Sistemas MIMO. Dessa forma, os ganhos das malhas dos controladores LQR, LQG e LQG/LTR devem ser determinados de modo a satisfazer as restrições de alocação de autoestruturas. A importância da AAE é justificada, pois os autovalores refletem a estabilidade e taxa de crescimento ou decrescimento da resposta temporal dos sistemas, enquanto os autovetores estão relacionados com a sua forma, (Lewis and Syrmos 1995).

O objetivo deste capítulo é descrever os métodos desenvolvidos para a sintonia dos ganhos de controladores LQR, LQG e LTR por meio de uma hierarquia de modelos baseados em otimização combinatória e algoritmos genéticos. Os modelos de busca que são utilizados para sintonizar os ganhos das malhas LQR, LQG e LTR são apresentados nas Seções 2.1, 2.2 e 2.3, respectivamente.

2.1 Modelo de Sintonia do Ganho LQR

O Modelo para Sintonia dos Ganhos LQR enfoca a seleção das matrizes de ponderação (Q, R) no sentido de alocar a autoestrutura do sistema dinâmico. O problema é modelado como um mapeamento do espaço de matrizes $((Q, R), \mathbb{R})$ em um controlador $K_{Q,R} \in \mathbb{R}^{m \times n}$ que aloca uma determinada autoestrutura. Um modelo de otimização combinatória é construído por Fonseca, (Fonseca Neto 2000), para resolver o problema de Alocação de Autoestruturas em Sistemas Dinâmicos Multivariáveis. O método proposto por Fonseca Neto é uma alternativa para alocar a Autoestrutura através do LQR, superando as buscas por tentativa e erro do projetista para determinar as matrizes de ponderação.

2.1.1 Modelo de Busca LQR $_{Q,R}$

O problema LQR é formulado para alocar a Autoestrutura em Sistemas Dinâmicos Multivariáveis. Esta formulação permite a utilização de técnicas de computação evolutiva para determinar o ganho ótimo $K = R^{-1}B^T S_{LQR}(Q, R)$ do controlador que realize a alocação. A estrutura de otimização combinatória para busca das matrizes de ponderação é:

$$\min_{Q,R} \sum_{i=1}^n p_i s_i(Q, R) \quad (2.1)$$

s.a

$$\begin{aligned} s_i(Q, R) &\leq 1, \quad i = 1, \dots, n \\ \lambda_{ei} &\leq \lambda_{ci}(Q, R) \leq \lambda_{di}, \quad i = 1, \dots, n, \end{aligned}$$

sendo p_i a i -ésima ponderação das sensibilidades e $s_i = S_i/\epsilon_i$ a i -ésima sensibilidade normalizada em relação à i -ésima especificação de projeto, com $\epsilon_i > 0$. Q e R são matrizes simétricas semi-definidas e definidas positivas, respectivamente. O índice de desempenho menor ou igual a n , significa que a maior parte das sensibilidades foram satisfeitas, servindo como indicador global de convergência do processo de busca. Formulações alternativas são apresentadas por Liu (Liu 1998) e Bottura (Bottura and Fonseca Neto 1999). Cada par (λ_i, v_i) representa um objetivo para o sistema com n -estados.

2.1.2 O Problema LQR

O Problema LQR é formulado como uma estrutura de otimização com objetivo de determinar uma lei de controle $u(t) = -K_{LQR}x(t)$. Esta lei minimiza um índice de desempenho quadrático e tem como restrição a equação de estado linear invariante no tempo,

$$\min_u \frac{1}{2} \int_{t_0}^T [x^T Q(t)x + u^T R(t)u] dt \quad (2.2)$$

s.a

$$\dot{x} = Ax + Bu,$$

sendo $x \in \mathbb{R}^n$ e $u \in \mathbb{R}^m$, o par (A, B) é controlável, o par (A, C) é observável, $P(T) \geq 0$, $Q(T) \geq 0$ e $R > 0$ são matrizes simétricas, com $Q \in \mathbb{R}^{n \times n}$ e $R \in \mathbb{R}^{m \times m}$ semi-definidas e definidas positivas, respectivamente.

2.1.3 A Solução do LQR

As matrizes de ponderação Q e R , determinadas pelo Modelo de Busca $LQR_{Q,R}$, são parâmetros da equação de *Riccati*, $0 = A^T P + PA - PBR^{-1}B^T P + Q$, $t \leq T$. A solução P da Equação Algébrica de *Riccati* fornece os ganhos do controlador, $K_{LQR} = R^{-1}B^T P$, para calcular $u_{LQR} = -K_{LQR}x$.

2.2 Modelo de Sintonia do Ganho LQG

O Modelo para Sintonia dos Ganhos LQG enfoca a seleção das matrizes de covariâncias Ξ e Θ . O projeto do filtro de *Kalman* leva em consideração a alocação da Autoestrutura no problema de estimação de estado estocástico. Esta formulação permite a utilização de técnicas de Computação Evolutiva para determinar um observador estocástico. Um método para o problema de AAE de Estimador de Estado Estocástico via FK e AG foi publicado por (Ferreira 2003).

2.2.1 Modelo de Busca $LQG_{\Xi, \Theta}$

Utiliza-se as restrições de AE (autovalores e autovetores) para formular o problema de estimação de estado estocástica, com o objetivo de encontrar $L =$

$\Sigma_{\Xi, \Theta} C^T \Theta^{-1}$ do FK, de forma a satisfazer estas restrições. O problema é modelado como um problema de otimização combinatória, permitindo a determinação de um estimador $L_{\Xi, \Theta}$ por meio de Algoritmo genético,

$$\min_{\Xi, \Theta} \sum_{i=1}^n p_i s_i(\Xi, \Theta) \quad (2.3)$$

s.a

$$\begin{aligned} s_i(\Xi, \Theta) &\leq 1, \quad i = 1, \dots, n \\ \lambda_{ei} &\leq \lambda_{ci}(\Xi, \Theta) \leq \lambda_{di}, \quad i = 1, \dots, n, \end{aligned}$$

sendo $s_i = S_i/\epsilon_i$ é a i -ésima sensibilidade normalizada em relação à i -ésima especificação de projeto, com $\epsilon_i > 0$. $\Xi = \Xi^T \geq 0$ e $\Theta = \Theta^T > 0$ são as matrizes de covariâncias de ruído do processo e ruído de medida, respectivamente.

2.2.2 O Problema LQG

No projeto Linear Quadrático Gaussiano um observador de estado estocástico é adicionado ao Projeto LQR para estimação do estado por meio da saída medida. Temos um problema de controle ótimo estocástico,

$$\min_u E \left[\int_0^\infty (x^T Q(t)x + u^T R(t)u) dt \right] \quad (2.4)$$

s.a

$$\dot{x} = Ax(t) + Bu(t) + G\xi(t),$$

sendo $x \in \mathbb{R}^n$ e $u \in \mathbb{R}^m$, o par (A, B) é controlável, o par (A, C) é observável, $Q(t) \geq 0$ e $R(t) > 0$ são matrizes semi-definidas e definidas positivas, respectivamente, com $Q \in \mathbb{R}^{n \times n}$ e $R \in \mathbb{R}^{m \times m}$.

2.2.3 A Solução do LQG

As matrizes de covariâncias (Ξ, Θ) determinadas pelo modelo de busca $LQG_{\Xi, \Theta}$ são os parâmetros da EAR $A\Sigma + \Sigma A^T + G\Xi G^T - \Sigma C^T \Theta^{-1} C \Sigma = 0$, $t \leq T$. A solução

Σ da EAR fornece o ganho do observador, $L = \Sigma C^T \Theta^{-1}$, que é utilizado para calcular uma estimativa $\hat{x}(t)$ do estado $x(t)$ a partir da observação da saída $\{y(\tau), \tau \leq t\}$, sendo $\Theta(t) > 0$ e $\Xi(t) \geq 0$, matrizes simétricas definidas e semi-definidas positivas, respectivamente, com $\Xi \in \mathbb{R}^{n \times n}$ e $\Theta \in \mathbb{R}^{p \times p}$. Aplica-se a matriz de ganhos do LQR à estimativa gerada pelo observador. A solução deste problema é dada por $u(t) = -K\hat{x}(t)$.

2.3 Modelo de Sintonia do Ganho LQG/LTR

O problema LQG/LTR com recuperação de malha de realimentação é formulado como um problema de otimização que pode ser aplicado tanto para as recuperações das malhas de saída ou de entrada. Os objetivos são: a) encontrar as matrizes de ponderação Q e R do problema LQR, determinando o controlador $K_{LQR/LTR}$ com recuperação pela saída, b) encontrar as matrizes de covariância do problema LQG, determinando o ganho do observador $L_{LQG/LTR}$ com recuperação pela entrada.

A utilização de procedimentos por tentativa e erro para o projeto LTR justifica o desenvolvimento de um algoritmo genético que fornece o ganho da recuperação de malha de forma automática e inteligente. O ganho do controlador K , fixado no procedimento, é determinado por meio da metodologia desenvolvida por (Fonseca Neto 2000) e (Fonseca Neto 2003). O procedimento para recuperação do ganho de malha de realimentação LTR na entrada, (Ferreira 2003), mostra devidamente a escolha das matrizes de covariâncias. Esta escolha é função de variações de um parâmetro v_i que aproxima a malha de controle LTR para a malha de controle LQR. Porém, a metodologia aplicada faz uso do método de tentativa e erro para determinação do parâmetro v_i .

2.3.1 Modelo de Busca LTR

A seleção das matrizes de covariâncias (Ξ, Θ) ou ponderações (Q, R) são as duas alternativas para recuperação da malha do LQR. Na recuperação pela saída, a busca é direcionada para (Q, R) , as variações nas matrizes de ponderação tendo como referência (Q_0, R_0) e a lei para suas variações $(Q_i = v_i^2 Q_0 + CC^T, R_i = v_i^2 R_0)$ em função de um escalar v_i . Na recuperação pela entrada, a busca é

direcionada para (Ξ, Θ) , as variações nas matrizes de covariância tendo como referência (Ξ_o, Θ_o) e a lei que estabelece as suas variações $(\Xi_i = v_i^2 \Xi_o + BB^T, \Theta = v_i^2 \Theta_o)$ em função de um escalar v_i . O problema é modelado como um problema de otimização combinatória, permitindo a determinação dos ganhos do controlador $K = R^{-1}B^T P_{LQR}(Q, R)$ ou do ganho do observador $L_{\Xi, \Theta}$ que recuperem a malha L_{LQR} por meio de $L_{LQG/LTR}$ como representado na Equação (2.5), (Brito Filho 2006)

$$\min_v \sum_{l=1}^{n_{freq}} \left[\left(\sigma_l^{L_{LQR}-M} - \sigma_l^{L_{LTR}-M} \right) + \left(\sigma_l^{L_{LQR}-m} - \sigma_l^{L_{LTR}-m} \right) \right] \quad (2.5)$$

sujeito a

$$\begin{aligned} \sigma_{lesq}^{max-M} &\leq \sigma_l^{L_{LTR}-M} \leq \sigma_{ldir}^{max-M} \\ \sigma_{lesq}^{min-m} &\leq \sigma_l^{L_{LTR}-m} \leq \sigma_{ldir}^{min-m} \\ \lambda_{iesq} &\leq \lambda_{ic} \leq \lambda_{idir} \\ s_i &\leq \epsilon, \end{aligned} \quad (2.6)$$

sendo L_{LQR} a malha do controle LQR, $L_{LQG/LTR}$ a malha recuperada pelo projeto LQG/LTR. $\sigma_l^{L_{LTR}-M}$, σ_{lesq}^{max-M} e σ_{ldir}^{max-M} são os maiores valores singulares e seus limites à esquerda e à direita, respectivamente. $\sigma_l^{L_{LTR}-m}$, σ_{lesq}^{min-m} e σ_{ldir}^{min-m} são os menores valores singulares e seus limites à esquerda e a direita, respectivamente. As restrições $\lambda_{iesq} \leq \lambda_{ic} \leq \lambda_{idir}$ e $s_i \leq \epsilon$ são as restrições de alocação de autoestrutura.

A estratégia de recuperação da malha do controle LQR por meio do projeto LQG/LTR ocorre pelo ajuste do ganho K_{QR} do controlador LQR ou pelo ajuste do ganho $L_{\Xi\Theta}$ do observador de estado. Na Figura 2.1 apresenta-se o diagrama de blocos para o sistema, controle e observador para fins de projeto de recuperação da malha.

2.3.2 O Problema da Recuperação LTR

A recuperação da malha do LQR é conduzida no domínio da frequência. O modelo Planta, Controle e Observador do projeto LQG é reduzido ao diagrama planta $G(s)$ e compensador $F(s)$ para fins de projeto da malha de recuperação. O projeto desta tem por base os pontos de abertura, o ponto 1 fornece o ganho de malha aberta na entrada e o ponto 2 o ganho na saída, Figura 2.1.

A redução do diagrama Planta, Controle e Observador estabelece as seguintes relações entre os sinais de saída e entrada com os parâmetros do modelo da planta do sistema de controle. O sinal de entrada, $U(s) = -H_u(s)U(s) - H_y(s)Y(s)$, sendo $H_u(s) = -K(sI - A + LC)^{-1}B = K\Phi_o B$ e $H_y(s) = -K(sI - A + LC)^{-1}L = K\Phi_o L$, $\Phi_o(s) = (sI - A + LC)^{-1}$. A função de retorno de Y para U , $\frac{U(s)}{Y(s)} = F(s) = K\Phi_r L$, sendo $\Phi_r = [sI - (A - BK - LC)]^{-1}$.

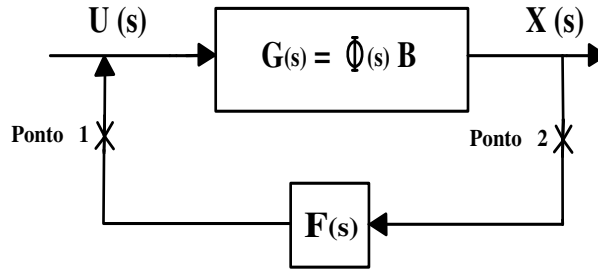


Figura 2.1: Diagrama em blocos reduzido LQG/LTR para recuperação da malha LQR .

2.3.3 O Problema LTR - Recuperação na entrada

Abrindo o modelo do sistema no Ponto 1 obtém-se o *Ganho de Malha Aberta na Entrada*, $L_r^e(s) = F(s)G(s) = K\Phi_r LC\Phi B$. O problema é formulado para determinar o ganho $L_{\Xi\Theta}$ do observador que recupere a malha do L_{LQR} , ou seja,

$$\min_{L \in \Theta} L_{LQR} - K\Phi_r LC\Phi B \quad (2.7)$$

sujeito a

$$|L| \leq \varepsilon \quad (2.8)$$

$$\lambda_{esq} \leq \lambda_c \leq \lambda_{dir}$$

$$s_i \leq \epsilon.$$

$$(2.9)$$

As restrições $|L| \leq \varepsilon$ do problema são operacionais, limitam-se aos ajustes do ganho L do observador.

2.3.4 A Solução LTR - Recuperação na entrada

As matrizes de covariância (Ξ, Θ) , determinada pelo Modelo de Busca $LTR_{\Xi\Theta}$, são parâmetros da equação de *Riccati*, $A\Sigma + \Sigma A^T + G(v^2\Xi_o + BB^T)G^T - \Sigma C^T(v^2\Theta_o)^{-1}C\Sigma = 0$, $t \leq T$. A solução Σ da Equação Algébrica de *Riccati* fornece o ganho do observador, $L = \Sigma C^T(v^2\Theta_o)^{-1}$.

2.3.5 O Problema LTR - Recuperação na Saída

Abrindo o modelo do sistema no Ponto 2 obtém-se o *Ganho de Malha Aberta na Saída*, $L_r^o(s) = G(s)F(s) = C\Phi BK\Phi_r L$. O problema é formulado para determinar o ganho K_{QR} do controlador que recupere a malha do L_{LQR} , ou seja,

$$\min_{K_{QR}} L_{LQR} - C\Phi BK\Phi_r L \quad (2.10)$$

sujeito a

$$|K| \leq \varepsilon \quad (2.11)$$

$$\lambda_{esq} \leq \lambda_c \leq \lambda_{dir}$$

$$s_i \leq \epsilon.$$

$$(2.12)$$

As restrições $|K| \leq \varepsilon$ do problema são operacionais, limitam-se aos ajustes do ganho K do controlador.

2.3.6 A Solução LTR - Recuperação na Saída

As matrizes de ponderação (Q, R) , determinada pelo Modelo de Busca LTR_{QR} , são parâmetros da equação de *Riccati*, $0 = A^T P + P A - P B (v_i^2 R_o)^{-1} B^T P + (v_i^2 Q_o + C C^T)$, $t \leq T$. A solução P da Equação Algébrica de *Riccati* fornece os ganhos do controlador, $K_{LTR} = (v_i^2 R_o)^{-1} B^T P$, para o cálculo da lei de controle, $U_{LTR} = -K_{LTR}x$.

2.4 Conclusão

Apresentou-se o desenvolvimento do projeto de controladores LQR, LQG e LQG/LTR utilizando uma hierarquia de algoritmos baseados em otimização combinatória e algoritmos genéticos, enfocando-se os modelos para sintonizar os ganhos das malhas de controle. O projeto LQG/LTR com recuperação na entrada foi implementado por tentativa e erro, (Teixeira Ferreira 2004), e suas características foram utilizadas para justificar uma modelagem inteligente para o ajuste dos ganhos. A solução inteligente é o desenvolvimento do modelo genético-artificial para o ajuste do ganho da malha de recuperação.

A Figura 2.2 ilustra as etapas de construção do algoritmo hierárquico para recuperação da malha do LQR. Na primeira etapa, o algoritmo determina os ganhos do controlador LQR que realize a alocação de autoestrutura desejada. A segunda etapa consiste do cálculo dos ganhos do observador de estado estocástico que satisfaz as restrições de AAE do filtro de *Kalman*. O desenvolvimento desta etapa é justificado uma vez que o estado não é diretamente disponível para realimentação ótima, mas necessita ser estimado a partir de saídas medidas. Entretanto, com a inserção do observador de estado no projeto LQR ocorre uma perda das propriedades de robustez do LQR, e é necessário recuperar a malha do LQR na etapa seguinte.

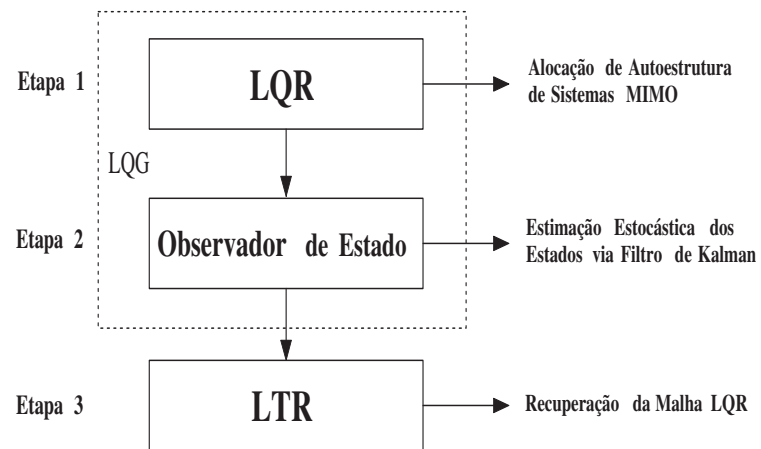


Figura 2.2: Estrutura hierárquica de algoritmos para recuperação da malha LQR.

CAPÍTULO 3

MODELO DO AG-LQR _{Q,R}

O objetivo deste capítulo é apresentar o modelo do AG que realiza a busca das matrizes de ponderação Q do estado e R do projeto de controle LQR de forma a determinar um controlador que satisfaça a restrição de alocação de autoestrutura, conduzindo a um sistema dinâmico com bom desempenho e boa estabilidade. Este modelo está alicerçado em uma estrutura de otimização para busca das matrizes de ponderações. A estrutura mostra-se adequada para utilização de métodos de computação evolutiva.

Apresenta-se na Seção 3.1 conceitos básicos e definições de termos da genética artificial dentro do contexto da modelagem genética das matrizes de ponderação Q e R . Os conceitos de cromossomos, genótipos, fenótipos, genes e outros termos que foram criados para sintetizar idéias que fazem parte do algoritmo proposto são abordados. Os modelos que compõem o algoritmo genético para a busca das matrizes de ponderação são descritos na Seção 3.2. Os modelos abordam desde as representações genético artificial das matrizes, passando pelas operações cromossômicas até chegar nas avaliações de *fitness*.

3.1 Conceitos Básicos e Definições

A analogia entre algoritmos genéticos e o sistema natural é apresentada nesta Seção para explicar termos básicos da computação evolutiva usados nesta dissertação. A analogia é feita por meio das matrizes de ponderação Q e R .

Um *cromossomo* artificial é uma estrutura de dados que representa uma das

possíveis soluções do espaço de busca do problema dadas pelo par de matrizes $Q_{n \times n}$ e $R_{m \times m}$. Um elemento do tipo Q unido a um do tipo R é representado no formato de um código genético, chamado *genótipo*, geralmente dado como *strings* de números, cadeias de caracteres. Devido à característica simétrica dessas matrizes pode-se considerar apenas os elementos diagonais e todas as componentes da parte superior em relação às entradas diagonais.

Os elementos que constituem as matrizes simétricas Q e R conjugadas correspondem aos *genes* da genética natural. A quantidade de genes de um cromossomo é dado por $n_Q + n_R$, sendo $n_Q = n(n+1)/2$ e $n_R = m(m+1)/2$. Cada gene possui um valor que é chamado de *alelo* e uma posição ou *locus*. Na genética artificial, para o caso específico das matrizes, o valor de cada elemento representa um alelo e seus índices (i, j) representam a posição. Os alelos podem ser representados nas bases numéricas, decimal, hexadecimal, binária, etc; este tipo de representação é genericamente chamado de *alfabeto* do modelo cromossômico.

A Figura 3.1 ilustra um caso particular das matrizes Q e R sendo transformadas em string, com $n = m = 2$.

$$Q \cup R = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \cup \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \xrightarrow{\text{Transformação}} \boxed{\begin{matrix} \text{cromossomo-}QR \\ q_{11} \mid q_{12} \mid q_{22} \mid r_{11} \mid r_{12} \mid r_{22} \end{matrix}}$$

Figura 3.1: Modelo genético artificial de ordem 2 para as matrizes Q e R .

Uma função de avaliação positiva, relacionada à função objetivo e suas restrições, é construída para avaliar a qualidade do cromossomo (solução), chamada *função de fitness* ou *função de adequabilidade*.

Um conjunto de cromossomos- QR constitui uma população de indivíduos- QR , cada indivíduo consistindo apenas de um único cromossomo. Nesse contexto considera-se indivíduo e cromossomo como estruturas idênticas. A combinação de Q e R forma a prescrição genética total para a construção de algum organismo.

O termo *ciclo de busca* consiste em uma série de passos sequenciais que envolve as operações cromossômicas, solução da equação algébrica de *Riccati*, cálculo dos ganhos de realimentação, cálculo de autovalores e autovetores, avaliação da

função de *fitness* e a verificação do critério de parada. Cada indivíduo recebe uma pontuação com valores escalares determinados na fase de *pré-fitness* ou de cálculos intermediários. A *pré-fitness* corresponde a determinação dos ganhos do controlador por meio da solução da EAR, montagem do sistema de malha fechada e computação dos autovalores e autovetores. A função de *fitness* verifica as restrições, ordenando os indivíduos de acordo com o grau de satisfabilidade das autoestruturas. O fluxograma da Figura 3.2 representa o caminho do ciclo de busca genética.

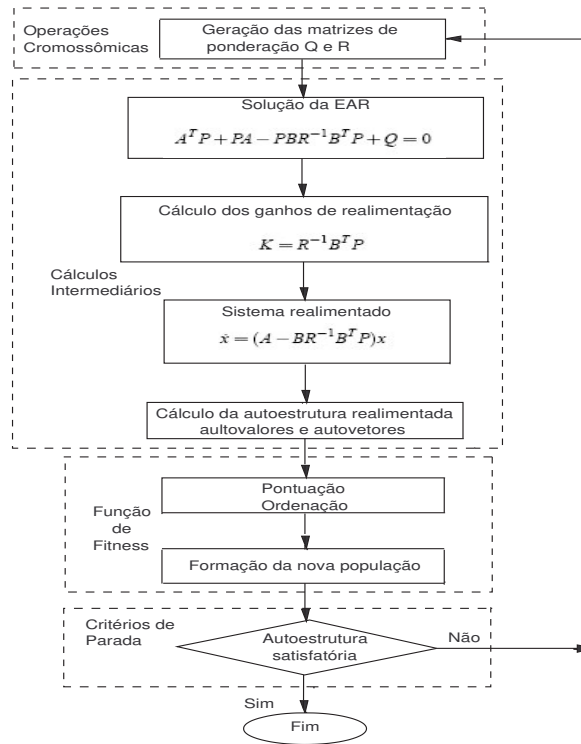


Figura 3.2: Ciclo de Busca Genética.

3.2 Modelos Genético Artificial

Nesta Seção apresenta-se os modelos que compõem o algoritmo genético para a busca das matrizes de ponderação. Os modelos abordam desde as representa-

ções genético artificial das matrizes, passando pelas operações cromossômicas até chegar nas avaliações de *fitness*. Estes modelos representam uma modelagem da estrutura de otimização, Equação (2.1), para busca das matrizes de ponderações em termos de algoritmos e estruturas de dados de computação evolutiva.

3.2.1 Modelo das matrizes Q e R

Inicialmente, considere a ordem das matrizes $A_{n \times n}$, $B_{n \times m}$ e $C_{p \times n}$ do sistema dinâmico. As dimensões das matrizes do modelo e as especificações do projeto LQR nos levam a agrupar as matrizes dos pesos $Q_{n \times n}$ e $R_{m \times m}$. Estas matrizes devem obedecer as restrições do problema de otimização do LQR, as matrizes $Q_{n \times n}$ e $R_{m \times m}$ devem ser simétricas, semi-definida positiva e definida positiva, respectivamente.

A característica simétrica dessas matrizes nos conduzem a um modelo com matrizes triangulares superiores ou inferiores. O modelo do cromossomo que representa as matrizes Q e R ,

$$\begin{aligned} QR_z &= \bigcup_{j,i=1}^n q_{ij} \wedge \bigcup_{j,i=1}^m r_{ij}, & i \leq j \\ z &= 1, \dots, n_{indiv}, \end{aligned} \quad (3.1)$$

sendo n o número de linhas ou colunas da matriz A , m o número de colunas da matriz B . Os genes representam os elementos q_{ij} e r_{ij} da matriz QR_z ; e n_{indiv} é o número de cromossomos individuais de uma certa população.

3.2.2 Modelo da População QR

Considerando que o cromossomo QR_z , Equação (E.1), é um elemento composto por g genes, o qual representa as matrizes $Q_{n \times n}$ e $R_{m \times m}$, a quantidade de genes de um cromossomo depende da dimensão n da matriz dinâmica do sistema e de m que representa o número de entradas do sistema. A quantidade g de genes de uma solução cromossômica é dada por $g = \frac{n(n+1)+m(m+1)}{2}$.

A população cromossômica é representada da seguinte forma

$$QR_{n_{indiv} \times g} = \left[QR_1; QR_2; QR_3; \dots; QR_{n_{indiv}} \right]. \quad (3.2)$$

A notação $n_{indiv} \times g$ indica a quantidade individual do cromossomo de uma população e a quantidade de genes em cada solução cromossômica ou indivíduo, respectivamente. A população tem n_{indiv} indivíduos e cada indivíduo tem $n_Q + n_R$ genes, sendo $n_Q = n(n+1)/2$ e $n_R = m(m+1)/2$. Os indivíduos QR de uma população são modelados como,

$$qr_z^w = \begin{cases} q_{i,j} & , w < n_Q + 1, \quad i, j = 1, \dots, n_Q \\ r_{i,j} & , w > n_Q, \quad i, j = 1, \dots, n_R \end{cases} \quad z = 1, \dots, n_{indiv}, \quad (3.3)$$

sendo n_Q e n_R as quantidades dos elementos das matrizes simétricas Q e R , respectivamente e n_{indiv} é o número de indivíduos.

3.2.3 Modelo da População Inicial

O modelo de geração aleatório das matrizes iniciais Q e R ,

$$q_{i,j} = \begin{cases} p_{Q\alpha} + p_{Q\beta}\kappa_{Qii} & i = j \\ p_{Q\gamma}\kappa_{Qij} & i \neq j \end{cases} \quad i, j = 1, \dots, n, \quad (3.4)$$

sendo $p_{Q\alpha}$ e $p_{Q\beta}$ parâmetros fixos e variáveis dos elementos da diagonal $q_{i,j}$, respectivamente. Esses parâmetros são os mesmos para a população inteira. O parâmetro κ_{Qii} é uma pseudo variável aleatória que pondera o parâmetro variável $p_{Q\beta}$ para cada gene isoladamente. O parâmetro $p_{Q\gamma}$ é o mesmo para cada elemento não diagonais e o parâmetro κ_{Qij} assume valores aleatórios para cada elemento da matriz Q . O modelo de geração do indivíduo para a matriz R é similar ao modelo de geração do indivíduo da matriz Q e assume a seguinte forma,

$$r_{i,j} = \begin{cases} p_{R\alpha} + p_{R\beta}\kappa_{Rii} & i = j \\ p_{R\gamma}\kappa_{Rij} & i \neq j \end{cases} \quad i, j = 1, \dots, n. \quad (3.5)$$

Esses parâmetros de controle permitem manter a população em uma certa região do espaço de solução que garantem as restrições de positividade para as matrizes Q e R .

3.2.4 Modelo de Avaliação dos Cromossomos

A avaliação cromossômica é baseada em uma função *Fitness*. Esta função é uma estrutura de otimização muito complexa, que envolve a solução do LQR e cálculo da autoestrutura de malha fechada. O desempenho de cada cromossomo do z -ésimo indivíduo da população $QR_{n_{indiv} \times g}$ é avaliado e cada indivíduo QR_z é pontuado. O modelo da função *Fitness* requer a solução do LQR, o espectro de malha fechada e seus autovetores:

$$K_z = LQR_z(A, B, Q_z, R_z) \quad (3.6)$$

$$A_z = (A - BK_z) \quad (3.7)$$

$$(\lambda_z, V_z, W_z) = \text{eigen}(A_z) \quad (3.8)$$

$$S_z = \frac{\|V_z\|^2 \|W_z\|^2}{\langle V_z, W_z \rangle} \quad (3.9)$$

$z = 1, \dots, n_{indiv},$

sendo K_z o ganho produzido por um indivíduo z . A_z é a matriz de malha fechada para o ganho K_z . λ_z , V_z e W_z são os autovalores e autovetores à direita e esquerda, respectivamente. S_z é a sensibilidade associada.

3.2.5 Modelos de Operações Genéticas

As operações genéticas são representadas por quatro modelos. O primeiro modelo está direcionado para a seleção dos indivíduos para reprodução, o segundo modelo armazena o melhor indivíduo da geração atual para geração seguinte, e os outros com os operadores genéticos crossover e mutação.

Modelo de Seleção

A operação de seleção baseada nos valores da função de *fitness*, é um dos principais elementos de um algoritmo genético. A seleção dos indivíduos é representada pelo modelo seguinte:

$$\kappa_z = Fit_{pop} \kappa_{rand} \quad (3.10)$$

$$QR_{select} = QR_j \Leftrightarrow \max_j \kappa_z < \sum_{j=1}^{n_{indiv}} Fit_j, \quad (3.11)$$

$$z = 1, \dots, n_{indiv},$$

sendo κ_z a escolha aleatória para o z -ésimo indivíduo baseado no número pseudo-aleatório κ_{rand} , e Fit_{pop} é o desempenho populacional total. QR_{select} é o j -ésimo indivíduo selecionado.

Modelo de Seleção Elitista

O modelo de seleção elitista normalmente é acoplado a outros métodos de seleção, na tentativa de se aumentar a velocidade de convergência do algoritmo e garantir a convergência caso o ótimo global seja descoberto durante o processo de busca. Este processo simplesmente armazena os n_e ($n_e = 1$) melhores indivíduos da população atual para a próxima geração, garantindo que estes cromossomos não sejam destruídos nas etapas de recombinação e mutação. A seleção elitista é representada pelo seguinte modelo:

$$QR_{G+1,super} = \begin{cases} QR_{G,melhor} \Leftrightarrow (\sum s_i(QR_{G,melhor}) < \sum s_i(QR_{G,super}) \text{ e} \\ QR_{G,melhor} \in \Omega_f) \text{ ou } (QR_{G,melhor} \in \Omega_f \\ \text{e } QR_{G,super} \notin \Omega_f \\ QR_{G,super} \Leftrightarrow (\sum s_i(QR_{G,melhor}) \geq \sum s_i(QR_{G,super}) \text{ e} \\ QR_{G,super} \in \Omega_f) \text{ ou } (QR_{G,melhor} \notin \Omega_f), \end{cases} \quad (3.12)$$

sendo $\Omega_f = \{QR_z; \lambda_{ei} \leq \lambda_i(QR_z) \leq \lambda_{di} \text{ e } s_i(QR_z) \leq 1\}$ a região de factibilidade, $\sum s_i(QR_z)$ a soma das sensibilidades de todos os autovalores correspondente ao indivíduo QR_z , $QR_{G,super}$ o super indivíduo da população da geração G (mantido reservado e não participa do processo evolucionário) e $QR_{G,melhor}$ o melhor indivíduo da população da geração G excluindo o super indivíduo.

Modelo de Cruzamento

A recombinação ou *crossover* é o operador AG responsável pela combinação de dois indivíduos. Neste estágio os indivíduos QR_{l_1} e QR_{l_2} da população $QR_{crom \times g}$ trocam informação genética. O modelo para esta operação considera os l_1 -ésimo e l_2 -ésimo indivíduos de uma dada população G , $l_1 \neq l_2$,

$$QR_{G+1,l_1} = \alpha(QR_{G,l_1}) + (1 - \alpha)(QR_{G,l_2}) \quad (3.13)$$

$$QR_{G+1,l_2} = \alpha(QR_{G,l_2}) + (1 - \alpha)(QR_{G,l_1}). \quad (3.14)$$

O índice G representa a G -ésima geração da população e o índice $G + 1$ representa a próxima geração. O indivíduo QR_{G,l_1} representa l_1 -ésimo indivíduo e QR_{G,l_2} representa o l_2 -ésimo indivíduo da população G respectivamente os quais fazem parte do cruzamento. O parâmetro α é um número aleatório gerado entre 0 e 1.

Modelo de Mutação

A mutação é um elemento essencial neste método de busca, esse operador modifica os genes nos cromossomos para gerar um novo cromossomo. A principal característica desta operação é evitar uma convergência prematura e gerar um novo material genético. Este modelo considera a seleção de um indivíduo somente para desempenhar a operação de mutação,

$$q_{ij}^{novo} = q_{ij}^l b^{x_{local}} \quad (3.15)$$

$$i = 1, \dots, (n^2 + n)/2 \text{ e}$$

$$j = 1, \dots, (n^2 + n)/2,$$

sendo b a base determinística do multiplicador exponencial, $b > 1$, x_{local} é o expoente aleatório, $0 < x_{local} < 1$. Este modelo genético desempenha uma mutação decimal. Cada elemento QR_z da população $QR_{crom \times g}$ tem probabilidade de $P = 0.05$. Em outras palavras, 5% de chance de ser mutado. Se um certo elemento é escolhido, um novo gene é aleatoriamente gerado para substituir o gene de um cromossomo antigo selecionado. Depois, o algoritmo de mutação implementa seu modelo, Equação (5.17).

3.3 Conclusão

Apresentou-se o modelo de um algoritmo genético destinado à seleção das matrizes de ponderação do índice de desempenho quadrático do projeto LQR para AAE em sistemas MIMO. Os modelos que compõem o referido algoritmo foram descritos desde as representações genético artificiais das matrizes de ponderação, passando pelos modelos da população, avaliação dos cromossomos até chegar nos modelos das operações genéticas.

CAPÍTULO 4

PROJETO LQG/LTR

As metodologias para projeto LQG/LTR apresentadas em (Cruz 1996), (Doyle and Stein 1979), (Kwakernaak and Sivan 1972) e (Ferreira e Fonseca *et al.* 2003) mostram a necessidade do desenvolvimento de um algoritmo para a recuperação inteligente da malha. Desta forma, obtém-se uma hierarquia de métodos baseados em algoritmos genéticos para projeto e síntese de controladores ótimos do tipo LQG/LTR.

Neste capítulo apresenta-se um procedimento para o ajuste do ganho L do observador que recupera pela entrada a malha do LQR. O ajuste do ganho baseia-se nas modificações automáticas do parâmetro v_i , tendo como referências as matrizes de ponderação, afim de determinar o ganho ótimo do observador que recupere a robustez da malha LQR.

No Ajuste do Ganho LQG/LTR, Seção 4.1, apresenta-se o mapeamento de um procedimento para sintonia da malha LTR. O Projeto LQR/LQG, Seção 4.2, enfoca a aplicação de modelos para Sintonia dos Ganhos LQR e LQG. Na Seção 4.3 apresenta-se o procedimento para Recuperação da Malha do LQR; este ajuste é uma aplicação do mapeamento proposto na Seção 4.1. A análise e comentários sobre o projeto são relatados ao longo da Seção. O modelo do sistema teste é apresentado no Apêndice C.

4.1 Ajuste do Ganho LQG/LTR

O procedimento de ajuste do ganho do observador para recuperação da malha do LQR pela entrada consiste da escolha das matrizes de covariâncias, solução da equação algébrica de *Riccati*, cálculo do Ganho Ótimo e uma métrica para avaliação do processo de recuperação. O mapeamento central do procedimento é o operador $F_{\Xi\Theta}$ que seleciona as matrizes de covariâncias por meio de variações no parâmetro v_i . A seguir apresenta-se os mapeamentos para o ajuste do Ganho LQG/LTR. O mapeamento do espaço das matrizes de covariâncias é dado por

$$F_{\Xi\Theta} : (\Xi_0, \Theta_0, v_i) \rightarrow (\Xi_i, \Theta_i), \quad (4.1)$$

sendo $F_{\Xi\Theta}$ uma função de matrizes de covariâncias que define as regras para o ajuste destas matrizes de acordo com o parâmetro v_i . A solução da equação de *Riccati* no domínio de parâmetro estocástico é descrito por

$$F_{LTR} : (\Xi_i, \Theta_i, A, C) \rightarrow \Sigma_{LTR}, \quad (4.2)$$

sendo Σ_{LTR} o conjunto de soluções de *Riccati* que são obtidos com as variações de v_i . A solução da equação de *Riccati* é mapeada no ganho ótimo do projeto LQG,

$$F_{LTR/LQG} : (\Sigma_{LTR}, C) \rightarrow L_{LTR-entrada}, \quad (4.3)$$

sendo $L_{LTR-entrada}$ do LQG que recupera a malha do LQR. O mapeamento final deve avaliar a qualidade da recuperação, se o ganho L_{LTR} se aproxima da malha do projeto LQR,

$$F_{LTR/LQR} : (L_{LTR-entrada}, L_{LQR}) \rightarrow \mathbb{R}. \quad (4.4)$$

A função $F_{LTR/LQR}$ representa um funcional que deve ser estabelecido afim de quantificar a qualidade da recuperação. Normalmente, os valores singulares máximos e mínimos dos projetos LTR e LQR são comparados ponto a ponto, (Ferreira e Fonseca *et al.* 2003) e (Lewis and Syrmos 1995).

4.2 Projeto LQR/LQG

O objetivo principal desta seção é enfatizar a importância das matrizes de ponderações e covariâncias no projeto de sistema de controle ótimo. Os projetos LQR e LQG são realizados em termos da busca dessas matrizes.

4.2.1 Projeto LQR

A determinação do ganho do controlador de estado envolve um esforço computacional considerável, considerando a solução da EAR e cálculo da lei de controle u . A Figura 4.1 mostra os gráficos dos valores singulares para variações do ganho K_{LQR} .

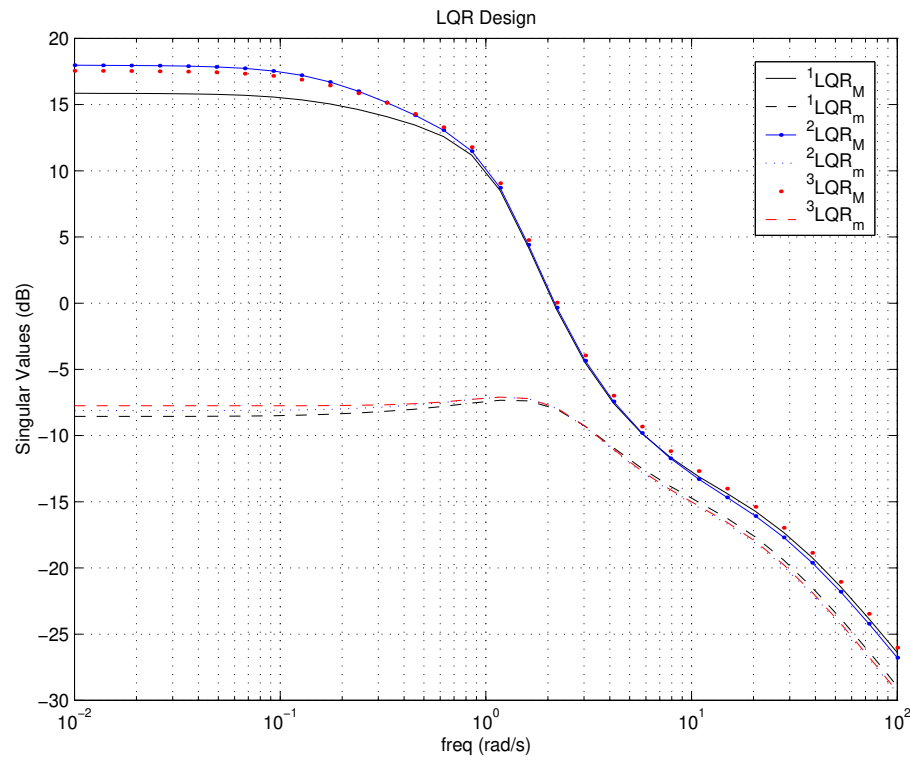


Figura 4.1: Diagrama de Valores Singulares - Projeto LQR - Escolha das matrizes de ponderação.

Na Figura 4.1, pode-se destacar as melhores soluções para os critérios de

desempenho e de estabilidade. Das três soluções obtidas, a de nº 3 apresentou melhor desempenho, pois obteve o maior dos valores singulares mínimos, em baixas frequências, dentre as soluções. Já para a estabilidade, a melhor solução foi a de nº 2, pois obteve o menor dos valores singulares máximos, em altas frequências. Dessa forma, e devido às proximidades das curvas serem mais acentuadas em altas frequências, resolveu-se fixar o controlador obtido na solução de nº 3 produzida pelo AG. Porém, qualquer uma das outras duas soluções poderia trazer resultado satisfatório, pois, como já ressaltado anteriormente, seus valores singulares estão bem próximos. As matrizes de ponderação para o caso 3 são dadas por

$$Q_{LQR} = \begin{bmatrix} 10.1 & 1.9 & 2.0 & 1.7 & 0.4 & 1.2 \\ & 8.2 & 2.6 & 0.7 & 0.6 & 2.5 \\ & & 7.3 & 1.7 & 2.2 & 1.6 \\ & & & 38.1 & 2.0 & 1.2 \\ & & & & 78.0 & 0.3 \\ & & & & & 29.2 \end{bmatrix}, \quad (4.5)$$

$$R_{LQR} = \begin{bmatrix} 31.9 & 4.6 \\ & 25.9 \end{bmatrix}. \quad (4.6)$$

A Tabela 4.1, mostra os ganhos do controlador para o caso 3, obtidos utilizando-se as matrizes de ponderação Q_{LQR} e R_{LQR} ,

Tabela 4.1: Ganhos do Controlador LQR.

K	Ganhos do Controlador
	0.1851 0.0061 -0.0325 -1.1791 -0.0106 1.5988
	0.0198 0.1943 -0.4226 -0.7849 -1.2051 2.6245

4.2.2 Projeto LQG

A alocação da autoestrutura é realizada por meio de um algoritmo genético, o qual é uma implementação do Modelo de Busca $LQG_{\Xi, \Theta}$, Seção 2.2.1. Este algoritmo seleciona as matrizes de covariâncias que satisfazem as restrições de projeto, tabela 4.2. Para 10 (dez) gerações, o AG obteve 10 soluções, ou seja,

encontrou 10 matrizes de covariâncias Ξ da perturbação no estado e Θ do ruído de medida. Para a solução 3, temos

$$\Xi_o = \begin{bmatrix} 0.0271 & 0.0077 & 0.0019 & 0.0016 & 0.0026 & 0.0020 \\ & 0.0270 & 0.0050 & 0.0074 & 0.0053 & 0.0057 \\ & & 0.0255 & 0.0061 & 0.0073 & 0.0046 \\ & & & 0.0230 & 0.0054 & 0.0084 \\ & & & & 0.0269 & 0.0070 \\ & & & & & 0.0255 \end{bmatrix} \quad (4.7)$$

$$\Theta_o = \begin{bmatrix} 0.0309 & 0.0024 & 0.0009 & 0.0070 \\ & 0.0301 & 0.0033 & 0.0027 \\ & & 0.0306 & 0.0016 \\ & & & 0.0303 \end{bmatrix}. \quad (4.8)$$

Na Tabela 4.2 é apresentado os ganhos do estimador, obtidos da aplicação das matrizes de covariâncias Ξ e Θ , geradas pelo AG na solução do LQG.

Tabela 4.2: Ganhos do Estimador de Espaço de Estado.

L_{LQG}	Ganhos do Estimador
	0.0022 0.0011 0.0028 0.0025
	0.0045 0.0082 0.0033 0.0051
	1.2452 -0.0767 0.3881 -0.2937
	-0.0143 0.3616 -0.3195 0.1551
	0.5273 -0.4254 1.4531 -0.6137
	-0.0886 0.1919 -0.4891 0.4201

A Figura 4.2 ilustra o comportamento dos valores singulares para os projetos de controle LQR e LQG. O projeto LQG é justificado uma vez que uma estimativa precisa ser realizada sempre que existe falta de sensores para medir os estados ou estes estados são variáveis sem associação a fenômenos físicos. Entretanto, com o uso do LQG ocorre uma perda das propriedades de robuztez do LQR, isto é, o sistema torna-se instável, e é necessário recuperar a malha do LQR.

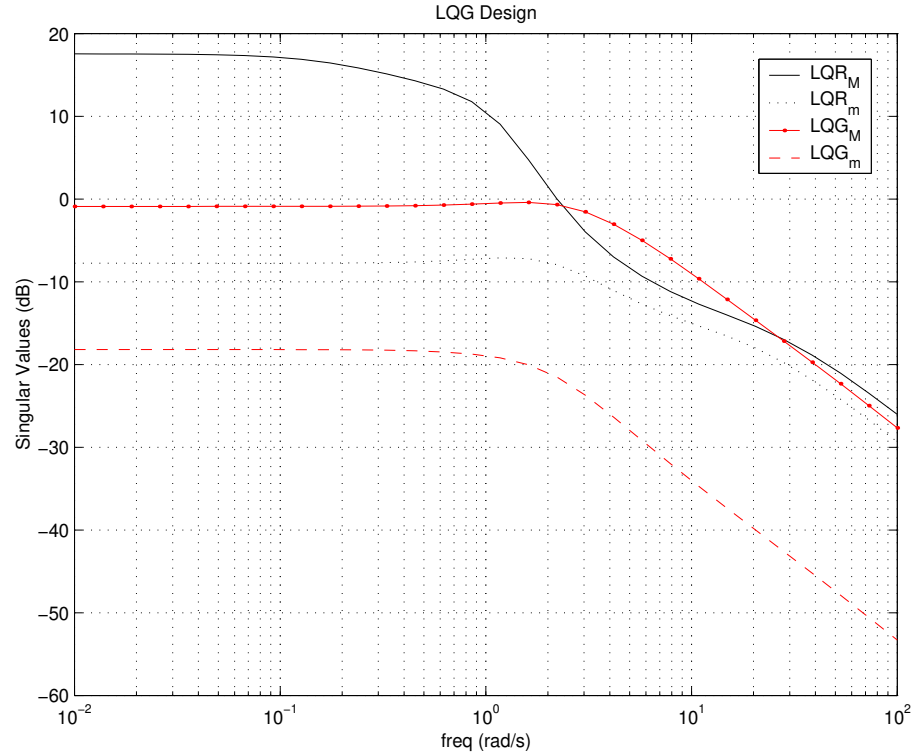


Figura 4.2: Diagrama de Valores Singulares - Projetos LQG e LQR

4.3 Projeto de Recuperação da Malha LQR

Nesta Seção apresenta-se o procedimento de ajuste do ganho do observador para recuperação da malha do LQR pela entrada. Os operadores $F_{\Xi\Theta}$, F_{LTR} , $F_{LTR/LQG}$ e $F_{LTR/LQR}$ representam os mapeamentos para a escolha das matrizes de covariâncias, a determinação da solução da equação de *Riccati*, cálculos dos ganhos e métricas de verificação, respectivamente. A partir de variações no parâmetro v_i recupera-se a robustez, ou seja, pode-se aproximar a malha do LQG da Malha LQR, (Doyle and Stein 1979). Cada mapeamento é descrito a seguir.

4.3.1 Mapeamento $F_{\Xi\Theta}$

O ajuste das matrizes de covariâncias da perturbação no estado e do ruído de medida é realizado por meio de variações no escalar $v_i \in \mathbb{R}$, como segue

$$\Xi_i = v_i^2 \Xi_0 + BB^T \quad (4.9)$$

$$\Theta_i = v_i^2 \Theta_0. \quad (4.10)$$

4.3.2 Mapeamento F_{LTR}

As matrizes de covariâncias Ξ_i e Θ_i ajustadas são as entradas da EAR

$$0 = A\Sigma_{LTR} + \Sigma_{LTR}A^T + G\Xi_iG^T - \Sigma_{LTR}C^T\Theta_i^{-1}C\Sigma_{LTR}. \quad (4.11)$$

4.3.3 Mapeamento $F_{LTR/LQG}$

O cálculo do ganho do estimador é descrito por

$$L_{LTR-entrada} = \Sigma_{LTR}C^T(v_i^2\Theta_0)^{-1}. \quad (4.12)$$

4.3.4 Mapeamento $F_{LTR/LQR}$

A avaliação do mapeamento $F_{LTR/LQR}$ permite comparar a evolução do processo de determinação da malha de recuperação. O processo de recuperação é ilustrado para quatro situações de variação do parâmetro v_i , como descrito abaixo.

Inicialmente o ganho K do controlador é mantido constante e o ganho de recuperação de malha é determinado como uma função do parâmetro v_i . A etapa seguinte é determinar o ganho $L_{LTR-entrada}$ que recupera a robustez do projeto LQR. As matrizes são selecionadas de acordo com as relações (4.9) e (4.10). Na Tabela 4.1 são mostrados os ganhos do controlador fixados no projeto LQG/LTR e na Tabela 4.3 são apresentadas as matrizes de covariâncias da perturbação no estado e do ruído de medida para realizar os ajustes de acordo com as Equações (4.9) e (4.10). Também na Tabela 4.3 são apresentados os valores de ajuste (v_1, v_2, v_3, v_4).

Tabela 4.3: Parâmetros de ajuste v e matrizes de covariâncias para a recuperação da malha de transferência

Parâmetros de Ajuste v_i	Matrizes de Covariâncias	
	Ξ_i	Θ_i
$v_1 = 1$	800 0 0 0 0 0	1 0 0 0
	1250 0 0 0 0	1 0 0
	0 0 0 0	1 0
	0 0 0	1
	0 0	
	0	
$v_2 = 0.1$	404.00 0 0 0 0 0	0.01 0 0 0
	631.25 0 0 0 0	0.01 0 0
	0 0 0 0	0.01 0
	0 0 0	0.01
	0 0	
	0	
$v_3 = 0.01$	400.0400 0 0 0 0 0	$(1.0e^{-003})*$
	625.0625 0 0 0 0	0.1 0 0 0
	0 0 0 0	0.1 0 0
	0 0 0	0.1 0
	0 0	0.1
	0	
$v_4 = 0.001$	400.0004 0 0 0 0 0	$(1.0e^{-005})*$
	625.0006 0 0 0 0	0.1 0 0 0
	0 0 0 0	0.1 0 0
	0 0 0	0.1 0
	0 0	0.1
	0	

Na evolução do processo de recuperação, Figura 4.3, verifica-se que quando $v_i \rightarrow 0$, as propriedades de robustez perdidas com a estimação tendem a ser recuperadas.

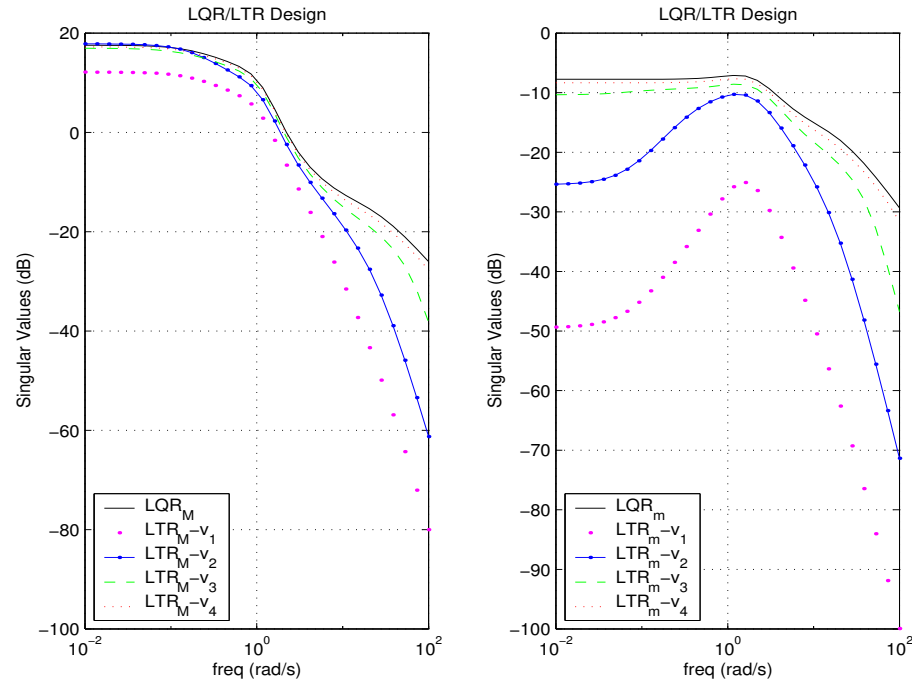


Figura 4.3: Recuperação LQG/LTR das propriedades de robustez.

Os valores singulares máximos e mínimos, Tabelas 4.4 e 4.5, definidos para uma dada faixa de frequência, são úteis para estabelecer os limites de robustez, bem como realizar uma análise dos métodos para controle de sistema. Como pode ser visto, a associação da Figura 4.3 com essas tabelas é usada para avaliar a sintonia do parâmetro v_i .

Tabela 4.4: Valores Singulares Máximos, Planta, Projetos LQR, LQG e LTR.

Projeto	SVD_{max}						
	Baixa freq BW-Hz					Alta freq BW-Hz	
	0.01	0.1	1	10	100	10^3	10^4
Planta	22.7	21.9	8.6	-19.1	-51.2	-90.8	-131
LQR	17.5	17.1	10.6	-12.3	-26.0	-45.8	-65.8
LQG	-0.9	-0.9	-0.5	-9.0	-27.7	-47.6	-67.6
LTR - v_1	12.1	11.7	4.5	-30.1	-80.0	-139.6	-199.6
LTR - v_2	17.8	17.1	8.2	-18.8	-61.3	-120.8	-180.8
LTR - v_3	17.0	16.3	9.7	-15.0	-38.2	-96.9	-156.9
LTR - v_4	17.2	16.8	10.3	-13.3	-27.7	-74.9	-134.8

Tabela 4.5: Valores Singulares Mínimos, Planta, Projetos LQR, LQG e LTR

Projeto	SVD_{min}						
	Baixa freq BW-Hz					Alta freq BW-Hz	
	0.01	0.1	1	10	100	10^3	10^4
Planta	-36.3	-31.7	-14.2	-24	-57	-96.9	-137
LQR	-7.8	-7.7	-7.2	-15.0	-29.3	-49.2	-69.2
LQG	-18.2	-18.2	-18.9	-34.0	-53.3	-73.2	-93.2
$LTR - v_1$	-49.3	-44.7	-26.7	-49.0	-99.9	-159.7	-219.7
$LTR - v_2$	-25.4	-21.0	-10.4	-24.8	-71.4	-131.0	-191.0
$LTR - v_3$	-10.3	-9.7	-8.6	-18.3	-46.8	-106.2	-166.2
$LTR - v_4$	-8.3	-8.3	-7.7	-16.2	-31.6	-83.9	-143.9

Na Figura 4.4 ilustra-se os erros de valores singulares em relação aos valores singulares do projeto LQR. Esta forma de comportamento de convergência descreve um crescimento na taxa de convergência quando $v_i \rightarrow 0$.

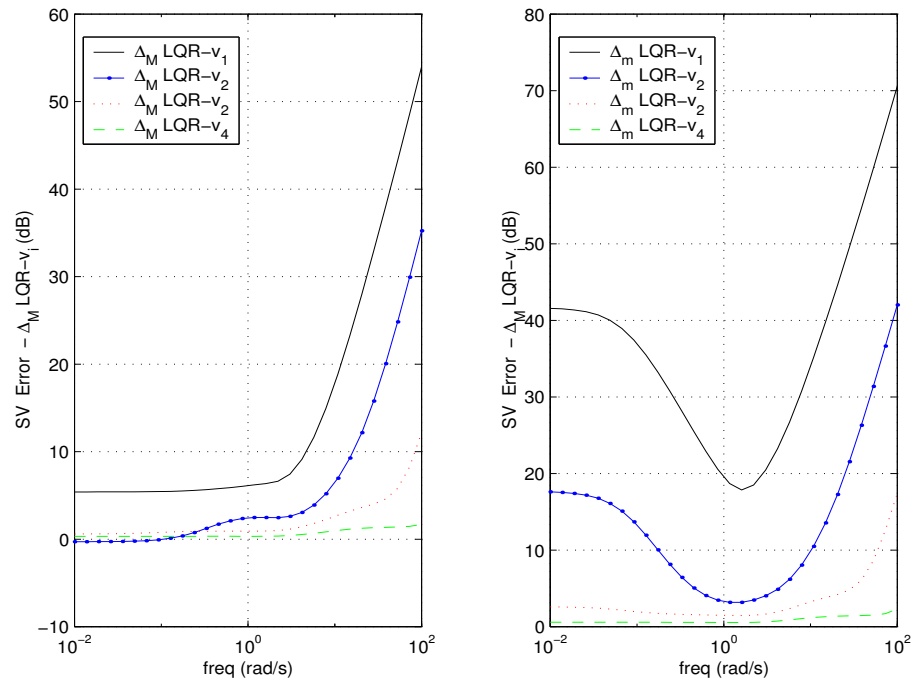


Figura 4.4: Projeto LQG/LTR - Erros de Valores Singulares.

4.4 Conclusão

O desenvolvimento do projeto LQG/LTR mostra a necessidade de um terceiro algoritmo genético para concretizar a implementação do modelo hierárquico que recupera a malha de realimentação. Os projetos LQR e LQG foram desenvolvidos utilizando algoritmos genéticos.

CAPÍTULO 5

MODELO DO AG-LQG/LTR

O principal enfoque deste Capítulo é apresentar o modelo de um algoritmo genético para o ajuste do ganho de recuperação da malha, que constitui uma contribuição para melhoria do método de projeto LQG/LTR. O objetivo do modelo de AG é representar uma busca inteligente do ganho de recuperação da malha, pois os métodos por tentativa e erro não garantem uma solução ótima. Este método complementa a pesquisa desenvolvida por (Fonseca Neto 2000) e (Ferreira 2003) para seleção das matrizes de ponderações do projeto LQR e das matrizes de covariâncias do projeto LQG, respectivamente.

O Modelo Cromossômico, Seção 5.1, apresenta o modelo genético artificial do mapeamento de indivíduos de uma população em termos da recuperação da malha pela entrada no projeto LQG/LTR, discutindo-se os conceitos de indivíduo, alelo, geração e população da genética artificial no contexto da modelagem do parâmetro de ajuste do projeto LTR. Os métodos para Geração da População Inicial são discutidos na Seção 5.2. A Estrutura de *Fitness*, Seção 5.3, é uma estrutura funcional que estabelece os critérios de identificação, suficiência e necessidade para a sobrevivência de indivíduos ou aceitação de soluções e envolve o cálculo Pré-*Fitness*, Função de Desempenho, Pontuação e Ordenação de populações. Em Métodos de Seleção, Seção 5.4, apresenta-se os métodos para seleção de indivíduos e o tipo de operações cromossômicas. Na Seção 5.5 as Operações Genéticas são modeladas como mapeamentos dos conjuntos de indivíduos de uma população em conjuntos de indivíduos que são gerados pelos operadores de seleção, *crossover* e mutação.

5.1 Modelo Cromossômico

O escalar v_i é o único parâmetro do processo de ajuste que é codificado como sendo um cromossomo artificial. A busca deste parâmetro gera conjuntos de matrizes de covariâncias que são modeladas como indivíduos de uma população. O mapeamento de indivíduos i de uma população j é dado por

$$F_{\Xi\Theta}^j : (v_i^j, \Xi_0, \Theta_0) \rightarrow (\Xi_i, \Theta_i) \quad i = 1, \dots, n_{ind} \quad \text{e} \quad j = 1, \dots, n_{pop}, \quad (5.1)$$

sendo n_{ind} o número de indivíduos e n_{pop} a quantidade de populações constituída por n_{ind} indivíduos. O mapeamento representado pela relação (5.1) em termos de operadores é

$$\Xi_i^j = (v_i^j)^2 \Xi_0 + BB^T \quad (5.2)$$

e

$$\Theta_i^j = (v_i^j)^2 \Theta_0. \quad (5.3)$$

5.1.1 Modelagem Escalar v_i

Os escalares v_i são modelados como cromossomos, a base numérica decimal é utilizada para representar os seus alelos. Cada escalar é chamado de indivíduo v_i e cada conjunto de escalares da forma $v_{i,k}^j$ caracteriza o i -indivíduo com n_{alelo} alelos da população j ,

$$v_i^j = v_{i,1}^j \cup v_{i,2}^j \dots \cup v_{i,k}^j \dots \cup v_{i,n_{alelo}}^j, \quad i = 1, \dots, n_{ind} \quad \text{e} \quad j = 1, \dots, n_{pop}, \quad (5.4)$$

sendo n_{ind} o número de indivíduos, n_{pop} a quantidade de populações com n_{ind} indivíduos. Cada individuo possui uma quantidade n_{alelo} de alelos artificiais. Estes alelos representam propriedades dos escalares v_i e possuem significado análogo ao da genética natural.

define um critério seletivo por espécie e o tipo do meio ambiente artificial, a fim de que indivíduo seja considerado apto a participar do processo de busca. Todos os indivíduos v_i^j de uma população j são mapeados em uma estrutura que representa as características da recuperação da malha, em relação ao atendimento dos critérios do projeto,

$$F_{LTR/LQR}^{fitness} : (\Upsilon_{id}, \Upsilon_{suf}, \Upsilon_{nec}) \rightarrow \Upsilon_{LTR/LQR}, \quad (5.7)$$

sendo Υ_{id} , Υ_{suf} e Υ_{nec} os conjuntos de identificação, suficiência e necessidade, respectivamente. O conjunto $\Upsilon_{LTR/LQR}$ representa os atributos de um indivíduo,

$$\Upsilon_{LTR/LQR} = \{\Upsilon_{id}^{ij}, \Upsilon_{suf}^{ij}, \Upsilon_{nec}^{ij}\}. \quad (5.8)$$

O mapeamento dos elementos do conjunto $\Upsilon_{LTR/LQR}$ é objetivo da estrutura de *fitness*. Este objetivo que representa o conjunto de identificação é constituído pelo tamanho da palavra, índice de adequabilidade e origem.

O conjunto de suficiência Υ_{suf}^{ij} é a garantia de que todos os indivíduos de uma população satisfazem os limites de minimização da função de desempenho. No caso das especificações de projeto do LQR é dado por

$$\Upsilon_{suf}^{LQR} = \{\sigma_{LQR}^{max}, \sigma_{LQR}^{min}\}, \quad (5.9)$$

sendo σ_{LQR}^{max} e σ_{LQR}^{min} os valores singulares máximos e mínimos da malha LQR. No caso de um indivíduo v_i^j ,

$$\Upsilon_{suf}^{ij} = \{\sigma_{ij}^{max}, \sigma_{ij}^{min}\}, \quad (5.10)$$

sendo σ_{ij}^{max} e σ_{ij}^{min} os valores singulares máximos e mínimos da malha LTR.

O conjunto de suficiência, Υ_{suf}^{ij} , relaciona-se com as condições globais. As condições que estão relacionadas com áreas específicas da busca que representam as restrições do problema de controle e qualidade do desempenho do modelo, tal

como complexidade do algoritmo e outras condições impostas pelo modelo de busca genética. O conjunto Υ_{nec}^{LQR} de restrições para o projeto LQR é dado por

$$\Upsilon_{nec}^{LQR} = \{\lambda_i, \quad \nu_i, \quad \delta_i, \}, \quad (5.11)$$

sendo λ_i , ν_i e δ_i , os autovalores, autovetores à direita e as sensibilidades dos autovalores, respectivamente. De forma similar ao conjunto de restrições de necessidade tem-se o conjunto Υ_{nec}^{ij} .

O conjunto de identificação e restrições LQR assume um valor padrão de referência para espécie, alelos, origem do cromossomo e informação sobre o processo de busca,

$$\Upsilon_{id}^{LQR} = \{limit_{\sigma_i}, \quad orig_{ij}, \quad alelos_{ij}\}, \quad (5.12)$$

sendo $limit_{\sigma_i}$, $orig_{ij}$ e $alelos_{ij}$ o limite de variação de cada indivíduo v_i^j em torno dos valores singulares de referências

A estrutura de *fitness* é decomposta em blocos funcionais que representam o Cálculo Pré-*Fitness* e a Função de Desempenho.

5.3.1 Cálculo Pré-*Fitness*

A solução da Equação Algébrica de *Riccati* e os Cálculos do Ganho de Malha LTR e dos Valores Singulares são as três etapas para determinação do índice de desempenho de cada indivíduo de uma população j .

5.3.2 Função de Desempenho

A função de desempenho é um mapeamento do erro quadrático entre os valores singulares máximos e mínimos dos projetos LQR e LQG/LTR,

$$F_{ij}^{desemp} = \sum_{l=1}^{n_{freq}} \left[SVD_{ij}^{LTR} - SVD_{ij}^{LQR} \right]^2, \quad (5.13)$$

sendo SVD_{ij}^{LTR} e SVD_{ij}^{LQR} os valores singulares da recuperação e do projeto LQR, respectivamente, correspondente a um conjunto de n_{freq} frequências. As

soluções são submetidas a este teste que permite avaliar a recuperação da malha e o mapeamento $F_{LTR/LQR}$.

5.4 Métodos de Seleção

No contexto deste trabalho, os métodos de seleção são classificados de acordo com a finalidade do processo de escolha. Existem três processos de seleção que são nitidamente distintos durante a busca do parâmetro v_i , e todos eles são dependentes de um gerador de números pseudo-aleatórios.

O primeiro método consiste na escolha das operações genéticas; é classificado pela escolha do tipo de operação genética a ser realizada em cada passo do ciclo de busca e pela manipulação dos alelos cromossômicos em função da forma de atuação dos operadores genéticos. O segundo método relaciona-se com o procedimento para escolha de indivíduos que sofrem ações dos operadores genéticos. O terceiro método relaciona-se com a escolha da quantidade de indivíduos para compor a população permanente, após o término de cada ciclo de busca, e constitui a fase final do algoritmo de *fitness*; a quantidade de indivíduos a ser incluída varia entre 1 e o número máximo de indivíduos que compõem a população permanente.

5.5 Operações Genéticas

As operações genéticas são mapeamentos dos conjuntos de indivíduos de uma população j em conjuntos de indivíduos de uma população em $j + 1$. Estes mapeamentos são realizados pelos operadores F_{gen}^κ genéticos de seleção, *crossover* e mutação. De forma geral, estes mapeamentos são dados por,

$$\begin{aligned} F_{gen}^\kappa : (v_{i1}^j, \quad v_{i2}^j) &\rightarrow v_i^{j+1} \\ i &= 1, \dots, n_{ind} \\ \text{e } j &= 1, \dots, n_{pop}. \end{aligned} \tag{5.14}$$

O conjunto κ dos operadores genéticos é definido por três operadores,

$$\kappa : \{X_S, X_C, X_M\}. \tag{5.15}$$

Os operadores X_S , X_C , X_M são estabelecidos nos itens operação de seleção, *crossover* e mutação.

5.5.1 Operação de Seleção

A operação de seleção baseada nos valores da função de *fitness* é um dos principais elementos de um algoritmo genético. A seleção é tipicamente implementada por uma roleta sendo que cada indivíduo é representado por uma fatia proporcional a sua aptidão relativa. Neste trabalho, a roleta é baseada no produto do cálculo do desempenho total de todos os indivíduos da população por um gerador de número pseudo-aleatório. Em seguida, determina-se a *fitness* parcial e compara-se com a *fitness* gerada pela roleta e na fase final ocorre o armazenamento do valor selecionado.

5.5.2 Operação de *Crossover*

A operação de *crossover* gera cromossomos para a próxima população a partir da combinação aleatória de dois indivíduos i_1 e i_2 de uma mesma população j ,

$$\begin{aligned} v_{i_1}^{j+1} &= \alpha_1 v_{i_1}^j + \alpha_2 v_{i_2}^j \\ v_{i_2}^{j+1} &= \alpha_2 v_{i_1}^j + \alpha_1 v_{i_2}^j \end{aligned} \tag{5.16}$$

$$i = 1, \dots, n_{ind},$$

sendo $\alpha_1 = 0.1$ e $\alpha_2 = 0.05$ parâmetros que ponderam os valores de alelos dos cromossomos. Neste estágio os indivíduos $v_{i_1}^j$ e $v_{i_2}^j$ trocam material genético entre si.

5.5.3 Operação de Mutação

Os cromossomos gerados a partir do operador de *crossover* são submetidos a operação de mutação com uma probabilidade de mutação $p_m = 5\%$. O operador de Mutação é utilizado para garantir uma maior varredura do espaço de busca e evitar convergência prematura. Esse operador troca o conteúdo de uma posição do cromossomo (alelo, símbolo, valor do gene) para gerar um novo cromossomo.

Os procedimentos para operação de mutação decimal são implementados da seguinte maneira: a mutação consiste de uma modificação nos alelos de indivíduos escolhidos aleatoriamente de acordo com o princípio de seleção natural de Darwin. A mutação para um cromossomo- v_i de uma população j e o valor de cada alelo é modificado de maneira multiplicativa,

$$v_i^{j+1} = \alpha v_i^j \quad (5.17)$$

$$i = 1, \dots, n_{ind},$$

sendo α um número pseudo-aleatório, $0 < \alpha < 1$.

5.6 Conclusão

Apresentou-se o modelo de um algoritmo genético para o ajuste do ganho de recuperação da malha LQR pela entrada. Discutiu-se os conceitos de indivíduo, alelo, geração e população no contexto da modelagem do parâmetro de ajuste do projeto LTR. Os métodos para geração da população inicial e os métodos de seleção foram descritos. O modelo do AG-LQG/LTR compõe-se do modelo cromossômico do parâmetro de ajuste, modelo da estrutura de *fitness* e os modelos de operações genéticas.

ANÁLISE DE CONVERGÊNCIA DO AG HIERÁRQUICO

O AG Hierárquico consiste de uma estratégia de busca das matrizes de ponderações e covariâncias a fim de alocar autoestruturas em sistemas dinâmicos multivariáveis. A análise de convergência do AG Hierárquico para recuperação da malha do projeto LQR pela entrada é realizada em duas etapas. Na primeira etapa avalia-se o desempenho do AG-LQR_{Q,R} em relação aos efeitos de uma estratégia elitista incorporada ao algoritmo para o melhoramento de suas propriedades de convergência no sentido de gerar soluções mais eficientes para o problema de controle que consiste na determinação da malha LQR. A segunda etapa consiste da análise do desempenho do AG-LQG/LTR em relação às variações nos valores de determinados parâmetros do algoritmo, tais como: semente do gerador pseudo-aleatório, condições iniciais, tamanho da população e probabilidade de mutação. A abordagem desta análise está direcionada para promover melhorias no desempenho do AG-LQG/LTR no sentido de gerar soluções satisfatórias para o problema da recuperação da malha LQR e acelerar o tempo de convergência do algoritmo.

6.1 Desempenho do AG-LQR_{Q,R}

Nesta Seção são apresentadas comparações de desempenho entre um AG-LQR_{Q,R} sem estratégia elitista e um AG-LQR_{Q,R} com estratégia elitista. Ini-

cialmente verifica-se o desempenho do algoritmo sem estratégia elitista por meio do comportamento da função de *fitness* objetivo normalizada e suas restrições para o melhor indivíduo ao longo de ciclos de busca. Em seguida, verifica-se o desempenho do algoritmo com estratégia elitista, salientando-se a evolução do super indivíduo ao longo de ciclos de busca. Além disso, são apresentadas as matrizes de ponderações Q e R geradas pelo AG-LQR $_{Q,R}$ para o melhor indivíduo da população final para cada processo, sendo examinados os autovalores, as sensibilidades dos autovalores de malha fechada e os ganhos ótimos do controlador obtido, a fim de comparar os desempenhos dos dois mecanismos de busca.

Para implementação do AG-LQR $_{Q,R}$, utilizou-se o ambiente computacional *MATLAB*, o sistema teste apresentado no Apêndice C, as restrições de projeto mostradas na Tabela C.2, os modelos genéticos-artificiais do Capítulo 3 e os parâmetros apresentados na Subseção 6.1.1

6.1.1 Parâmetros Utilizados na Implementação do AG-LQR $_{Q,R}$

Esta Subseção apresenta os parâmetros utilizados na implementação do AG-LQR $_{Q,R}$, os quais envolvem o tamanho da população, semente do gerador pseudo-aleatório utilizado para gerar a população inicial e para realizar as operações genéticas do processo, assim como para a ocorrência de qualquer evento aleatório desse processo, parâmetros constantes da população inicial para garantir as restrições de positividade das matrizes Q e R , quantidade de pares de indivíduos selecionados pela roleta para participar da operação *crossover*, probabilidade de *crossover*, alelos que seguem a estratégia de mutação proposta no algoritmo, fatores de mutação dos alelos selecionados, probabilidade de mutação, tamanho da elite e o número de iterações do ciclo de busca. Esses parâmetros são apresentados nas tabelas 6.1-6.5.

Tabela 6.1: Parâmetros utilizados no algoritmo genético

Tamanho da população	$n_{indiv} = 50$
Semente do gerador pseudo-aleatório	$s = 8$
Nº de pares de indivíduos para participarem da operação <i>crossover</i>	$n_{cross} = 25$
Probabilidade de <i>crossover</i>	$p_c = 1$
Probabilidade de mutação	$p_m = 0.05$
Tamanho da elite	$n_e = 1$
Nº de iterações	$n_{ger} = 200$

Tabela 6.2: Parâmetros da População Inicial-Matriz Q .

	Parâmetro			Parâmetro	
	$p_{Q\alpha}$	$p_{Q\beta}$		$p_{Q\alpha}$	$p_{Q\beta}$
	Fixo	Variável		Fixo	Variável
q_{11}	3.50	10.000	q_{22}	3.50	10.00
q_{33}	5.00	15.000	q_{44}	20.00	25.00
q_{55}	65.00	25.000	q_{66}	12.00	25.00
q_{ij}		$p_{Q\gamma}$			
		3.270			

Tabela 6.3: Parâmetros da População Inicial-Matriz R .

	Parâmetro			Parâmetro	
	$p_{R\alpha}$	$p_{R\beta}$		$p_{R\alpha}$	$p_{R\beta}$
	Fixo	Variável		Fixo	Variável
r_{11}	20.00	25.00	r_{22}	20.00	15.00
r_{ij}		$p_{R\gamma}$			
		13.00			

Tabela 6.4: Alelos que seguem a estratégia de mutação proposta no algoritmo

$j_alelo_1 = 22$	$j_alelo_2 = 24$	$j_alelo_3 = 23$
$j_alelo_4 = 1$	$j_alelo_5 = 7$	$j_alelo_6 = 12$
$j_alelo_7 = 16$	$j_alelo_8 = 19$	$j_alelo_9 = 21$

Tabela 6.5: Fatores de mutação dos alelos selecionados

$mut_1_2_alelo_k = 15$	$mut_1_2_alelo_v = 40$
$mut_3_alelo_k = 2$	$mut_3_alelo_v = 13$
$mut_F_1_alelo_k = 4$	$mut_F_1_alelo_v = 10$
$mut_A_1_alelo_k = 3.27$	$mut_A_1_alelo_v = 0$

6.1.2 Evolução do AG-LQR_{Q,R} sem Estratégia Elitista

O desempenho do AG-LQR é verificado por meio do comportamento da função de *fitness* objetivo normalizada e de suas restrições. Esta função é dada pela soma das sensibilidades normalizadas correspondente à função objetivo da estrutura de otimização (2.1). Já o comportamento das restrições da função de *fitness* é avaliado por meio das sensibilidades normalizadas s_i e pelos autovalores normalizados em relação aos limites especificados à esquerda e à direita, ou simplesmente autovalores normalizados à esquerda e à direita.

A fim de verificar que a autoestrutura de malha fechada atende às especificações de projeto (AAE), Tabela C.2, utiliza-se a seguinte convenção: 1) A parte real dos autovalores normalizados à esquerda de malha fechada deverá ser menor que 1 e maior que 1 em relação ao limite especificado à direita; 2) Se a parte imaginária do limite especificado à esquerda for não nula, então a parte imaginária do autovalor normalizado de malha fechada correspondente deverá ser menor que 1, caso contrário deverá ser igual a 1; 3) A sensibilidade normalizada s_i de cada autovalor λ_i deverá ser menor que 1.

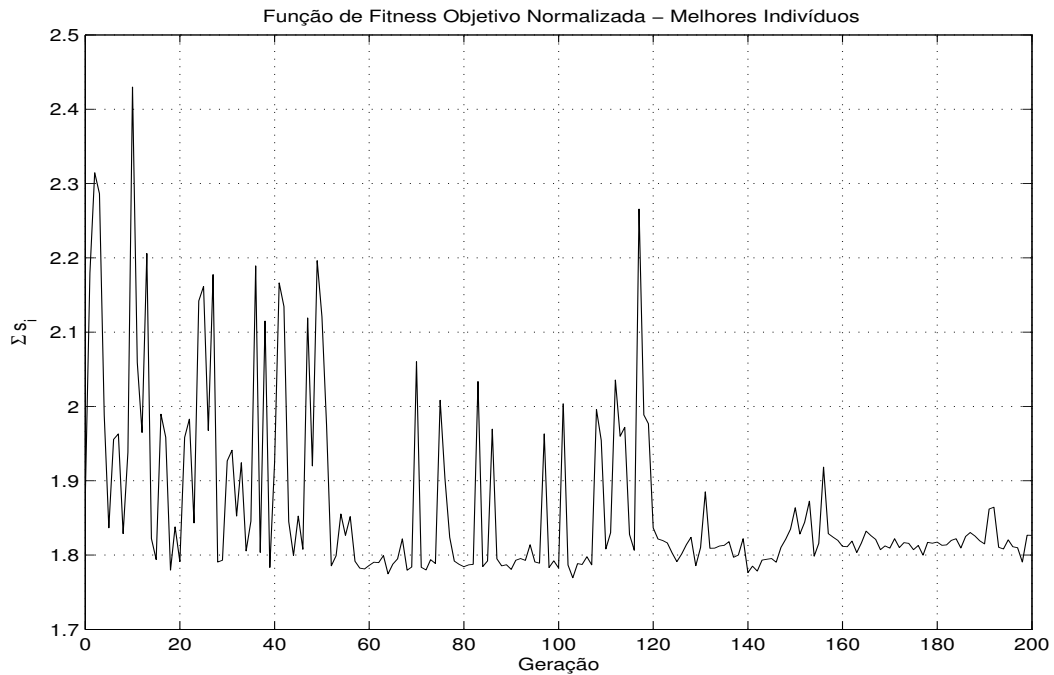


Figura 6.1: Função de Fitness Objetiva Normalizada - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

Na Figura 6.1 ilustra-se o comportamento da função de *fitness* objetivo normalizada para o melhor indivíduo da população de cada geração ao longo de 200 ciclos de busca. Observa-se frequências de oscilação significativas até aproximadamente a geração 120, e embora as frequências de oscilação após a geração 120 sejam menos acentuadas devido a uma menor diversidade de material genético e com menores amplitudes, não se verifica um comportamento decrescente da função de *fitness*.

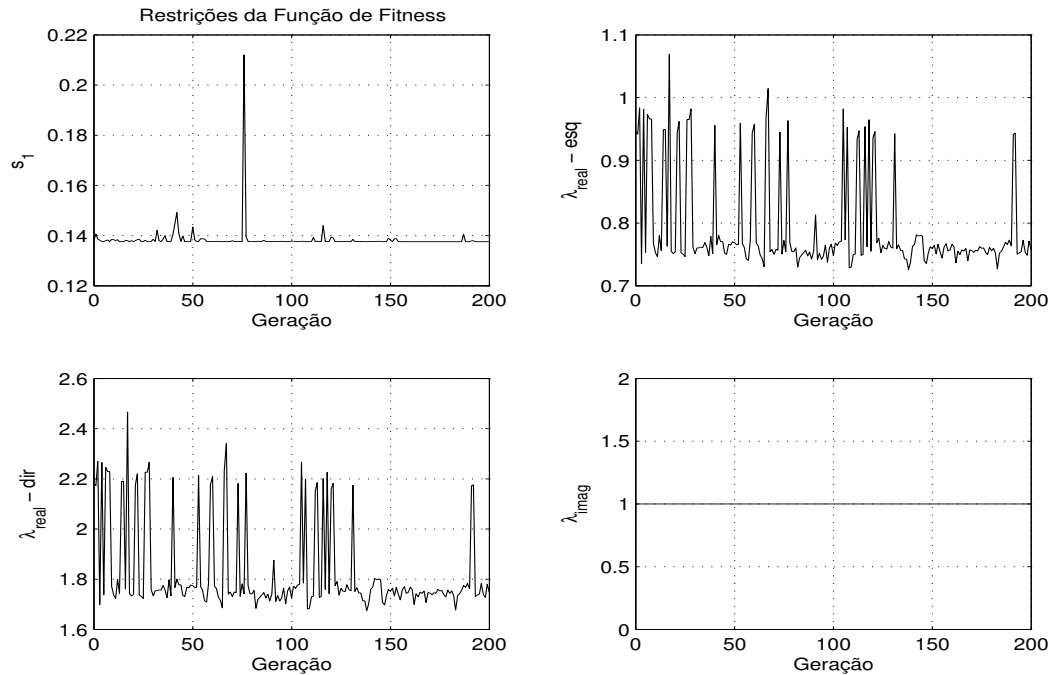


Figura 6.2: Restrições da Função de Fitness - Sensibilidades e Autovalores 1 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

Na Figura 6.2 ilustra-se o comportamento das sensibilidades e autovalores normalizados 1 de malha fechada para o melhor indivíduo da população de cada geração ao longo de 200 ciclos de busca. Observa-se que as partes reais dos autovalores normalizados à esquerda são menores que 1 para as 200 gerações, com exceção das gerações 17 e 67. As partes reais dos autovalores normalizados à direita são maiores que 1 e as partes imaginárias dos autovalores normalizados são iguais a 1 para as 200 gerações. Além disso, como a parte imaginária do limite especificado à esquerda correspondente é zero, então todos os autovalores normalizados 1 atendem às especificações de autovalores, excetuando-se as gerações 17 e 67. Verifica-se também que todas as sensibilidades normalizadas são menores que 1 para as 200 gerações, e portanto, todas atendem às especificações de sensibilidades, além de apresentar um comportamento de equilíbrio aproximadamente após a geração 70.

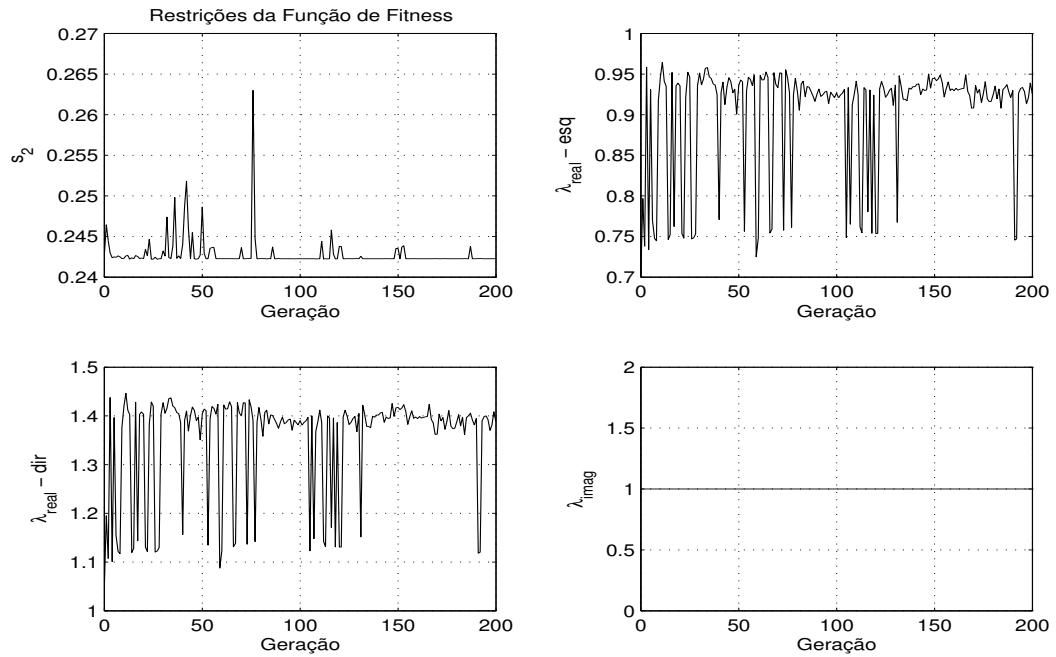


Figura 6.3: Restrições da Função de Fitness - Sensibilidades e Autovalores 2 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

Na Figura 6.3 ilustra-se o comportamento das sensibilidades e autovalores normalizados 2 de malha fechada para o melhor indivíduo da população de cada geração ao longo de 200 ciclos de busca. Observa-se um comportamento similar com relação ao atendimento das restrições de projeto quando comparado com o dos autovalores e sensibilidades normalizados 1 de malha fechada, sendo que todos os autovalores e sensibilidades normalizados 2 satisfazem às especificações de autoestrutura para as 200 gerações.

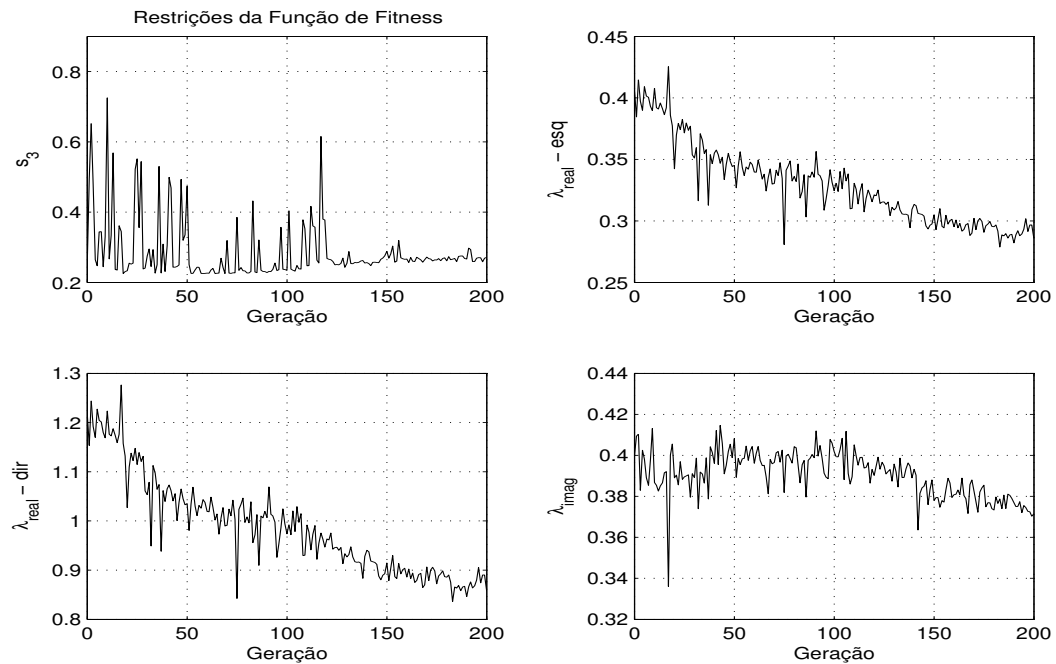


Figura 6.4: Restrições da Função de Fitness - Sensibilidades e Autovalores 3 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

Na Figura 6.4 ilustra-se o comportamento das sensibilidades e autovalores normalizados 3 de malha fechada para o melhor indivíduo da população de cada geração ao longo de 200 ciclos de busca. Observa-se que as partes reais de todos os autovalores normalizados à esquerda atendem às especificações de projeto, pois são menores que 1 para as 200 gerações e que também as partes imaginárias dos autovalores normalizados de malha fechada atendem as especificações, pois são menores que 1 para as 200 gerações e o valor especificado correspondente é diferente de zero. Já as partes reais dos autovalores normalizados à direita nas 31 primeiras gerações atendem às especificações de projeto, pois são maiores que 1, mas durante as gerações 32 a 107 oscilam entre valores maiores que 1 e menores que 1, apresentando em seguida uma tendência a serem menores que 1, o que caracteriza um comportamento não satisfatório dos autovalores normalizados. Observa-se também que as sensibilidades normalizadas apresentam frequências de oscilação significativas até aproximadamente a geração 120, sendo

que nas gerações posteriores as oscilações são pequenas. Contudo, todas atendem às especificações de sensibilidades, pois são menores que 1 para as 200 gerações.

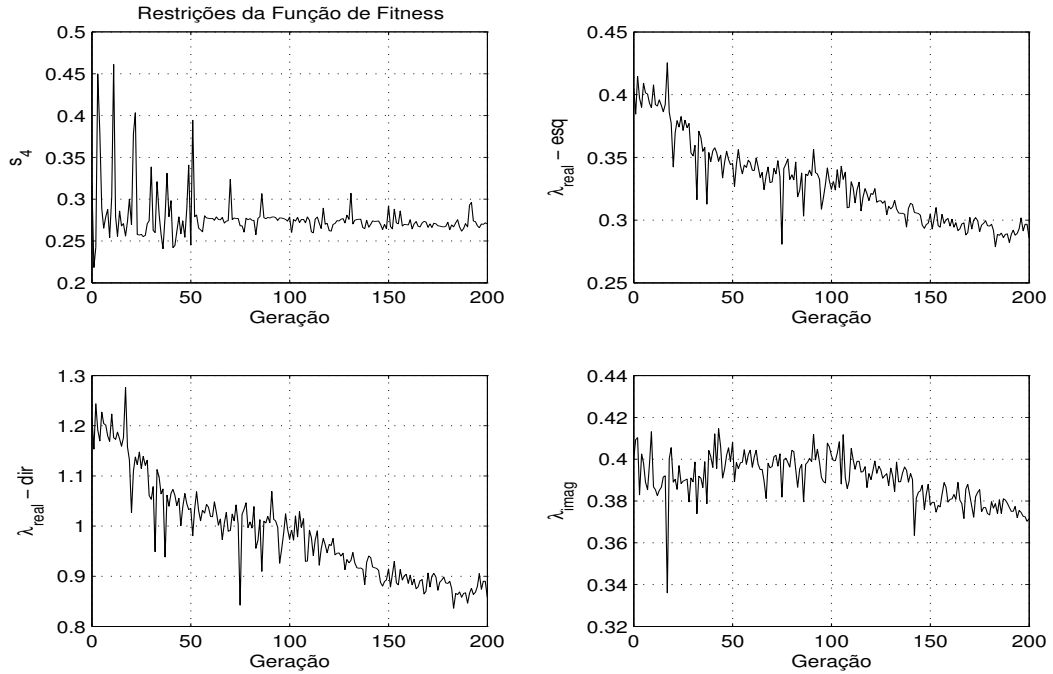


Figura 6.5: Restrições da Função de Fitness - Sensibilidades e Autovalores 4 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

Na Figura 6.5 ilustra-se o comportamento das sensibilidades e autovalores normalizados 4 de malha fechada para o melhor indivíduo da população de cada geração ao longo de 200 ciclos de busca. Observa-se um comportamento similar dos autovalores normalizados 4 quando comparado com o dos autovalores normalizados 3, e portanto, mostram-se com uma tendência não satisfatória. Já as sensibilidades normalizadas mostram-se todas satisfatórias para as 200 gerações e verifica-se desde aproximadamente a geração 50 frequências de oscilação pequenas.

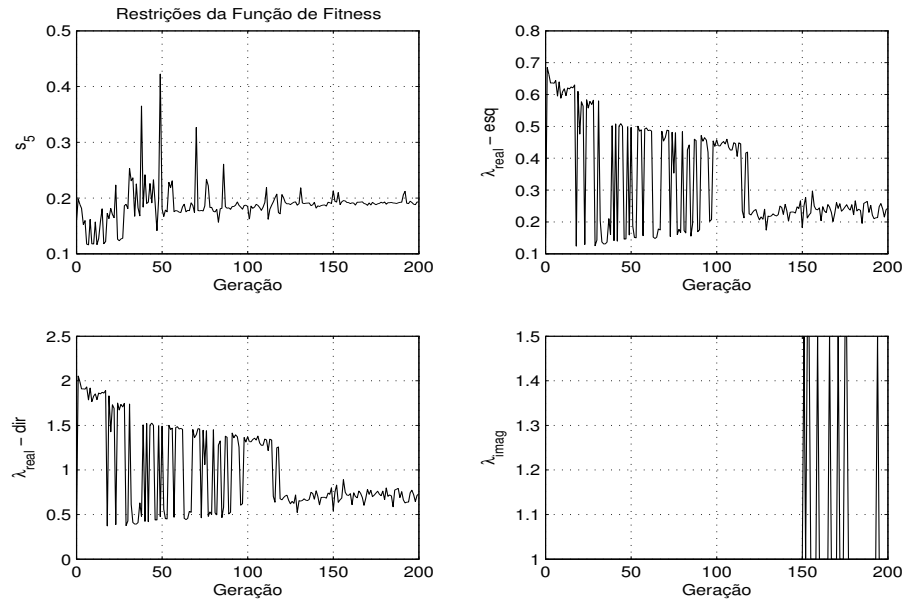


Figura 6.6: Restrições da Função de Fitness - Sensibilidades e Autovalores 5 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

Na Figura 6.6 ilustra-se o comportamento das sensibilidades e autovalores normalizados 5 de malha fechada para o melhor indivíduo da população de cada geração ao longo de 200 ciclos de busca. Observa-se que as partes reais de todos os autovalores normalizados à esquerda atendem às especificações de projeto para as 200 gerações, pois são menores que 1. Já as partes reais dos autovalores normalizados à direita atendem às especificações de projeto nas 17 primeiras gerações e durante as gerações 98 a 114. A partir da geração 18 até aproximadamente a geração 97 elas oscilam entre valores maiores que 1 e menores que 1 e em seguida se estacionam a valores menores que 1, apresentando dessa forma uma tendência não satisfatória. Verifica-se também que as partes imaginárias dos autovalores normalizados atendem às especificações até aproximadamente a geração 151, e nas gerações posteriores oscilam entre 1 e 1,5, logo atendem às especificações para o valor 1 e não atendem para 1,5, pois o valor especificado é zero. As sensibilidades normalizadas mostram-se todas satisfatórias para as 200 gerações.

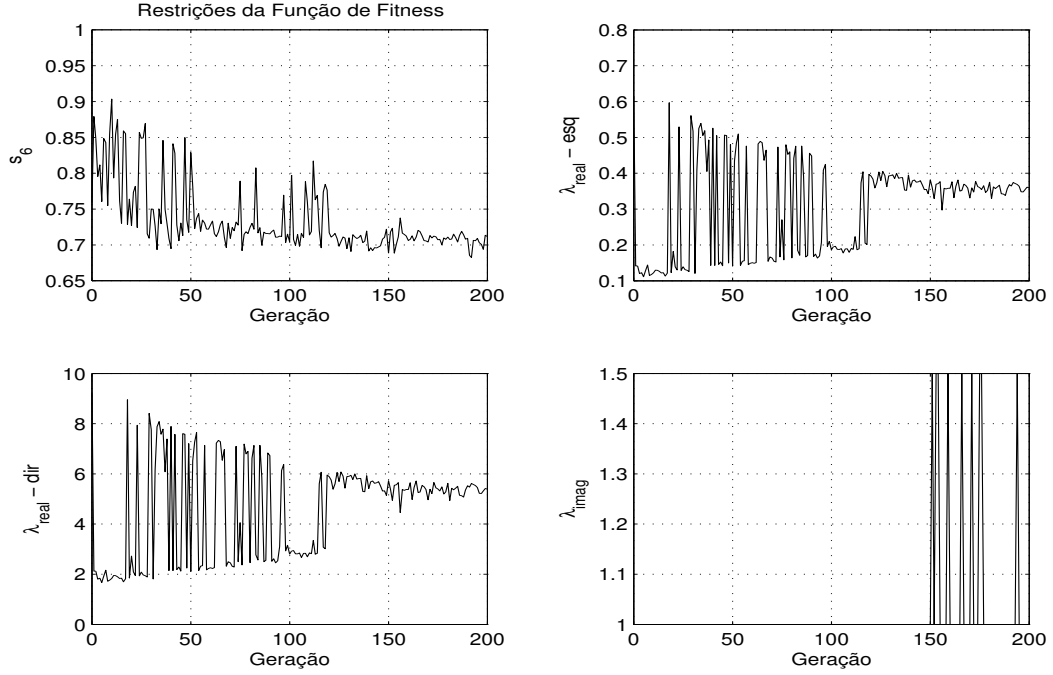


Figura 6.7: Restrições da Função de Fitness - Sensibilidades e Autovalores 6 Normalizados - Melhor Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

Na Figura 6.7 ilustra-se o comportamento das sensibilidades e autovalores normalizados 6 de malha fechada para o melhor indivíduo da população de cada geração ao longo de 200 ciclos de busca. Observa-se que as partes reais de todos os autovalores normalizados à esquerda e à direita atendem às especificações de projeto para as 200 gerações, pois são menores que 1 e maiores que 1, respectivamente. Já as partes imaginárias dos autovalores normalizados de malha fechada atendem às especificações de projeto até aproximadamente a geração 150, pois são iguais a 1 e o valor especificado é zero, e em seguida oscilam entre 1 e 1.5. As sensibilidades normalizadas mostram-se todas satisfatórias para as 200 gerações.

6.1.3 Evolução do AG-LQR_{Q,R} com Estratégia Elitista

Um estudo similar pode ser realizado para se verificar o desempenho do AG-LQR_{Q,R} com estratégia elitista por meio do comportamento da função de *fitness*

objetivo normalizada e de suas restrições para o super indivíduo da população de cada geração ao longo de 200 ciclos de busca.

Observa-se um desempenho satisfatório do AG-LQR_{Q,R} com estratégia elitista quando comparado com o desempenho do AG-LQR_{Q,R} sem estratégia elitista, pois todos os autovalores e sensibildades normalizados satisfazem às especificações de autoestrutura para as 200 gerações, Figuras 6.9-6.14, com exceção do autovalor normalizado 5 que não atende à especificação de autovalor apenas na geração 1. Além disso a função de *fitness* objetivo normalizada possui um comportamento decrescente a partir da geração 2, Figura 6.8, indicando dessa forma uma evolução do super indivíduo ao longo de ciclos de busca. Apesar da função de *fitness* objetivo normalizada assumir um valor na geração 1 menor do que na geração imediatamente posterior, o super indivíduo da população da geração 2 é considerado melhor do que o da população da geração 1, levando em consideração que este último não corresponde a uma solução factível, enquanto que a autoestrutura de malha fechada correspondente ao primeiro satisfaz às especificações de projeto. Isso explica o aparecimento de um pico no gráfico da função de *fitness* objetivo normalizada da geração 1 para a geração 2.

Verifica-se também que o melhor indivíduo obtido pelo primeiro mecanismo de busca ao longo de 200 ciclos de busca foi encontrado na geração 103 com valor de *fitness* igual a 1,7694, entretanto esse indivíduo foi perdido na geração seguinte, enquanto que no segundo mecanismo o mesmo se manteve durante o processo.

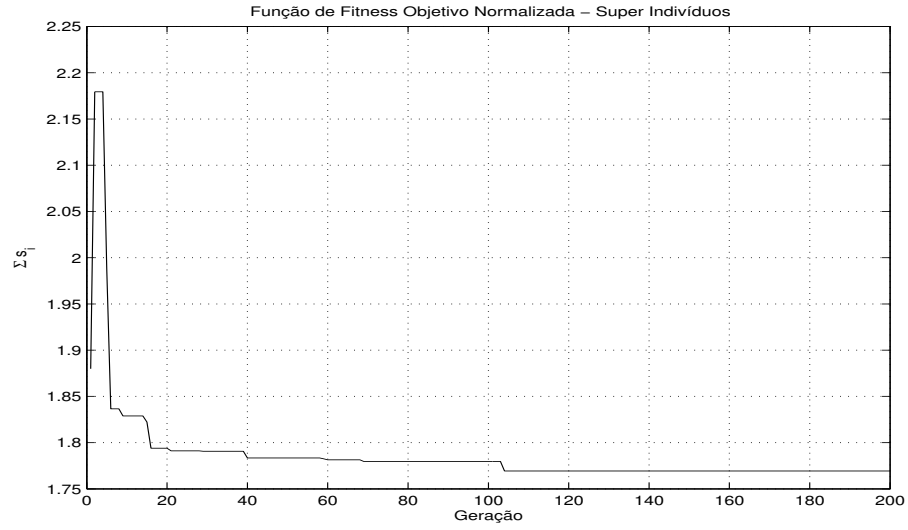


Figura 6.8: Função de Fitness Objetiva Normalizada - Super Indivíduo ao longo de 200 gerações - Sistema de Ordem 6.

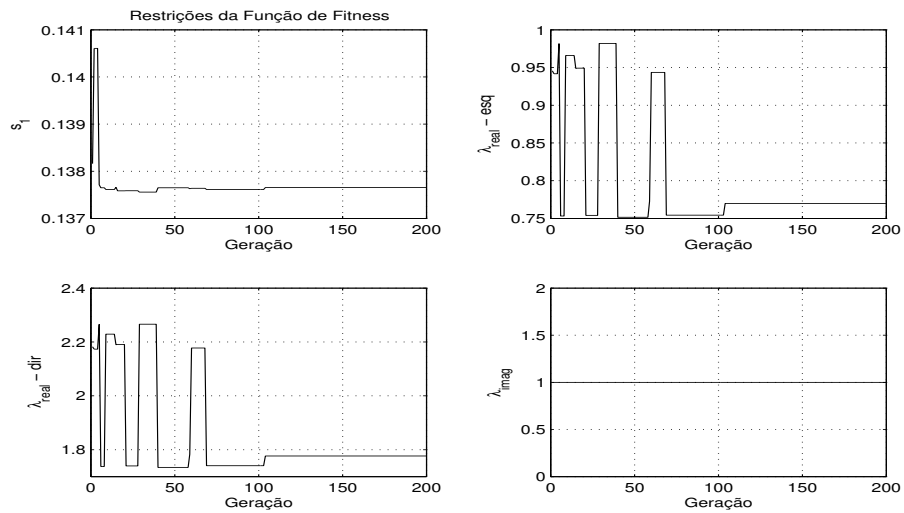


Figura 6.9: Restrições da Função de Fitness - Sensibilidades e Autovalores 1 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

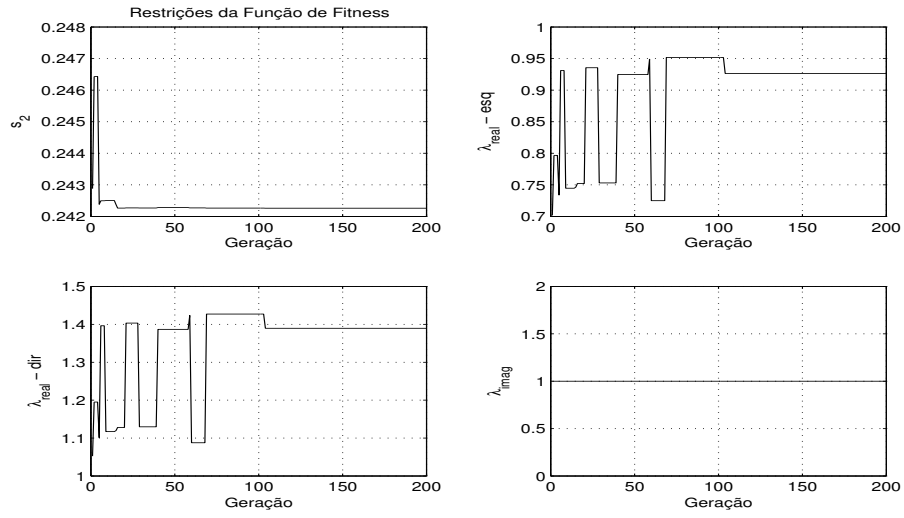


Figura 6.10: Restrições da Função de Fitness - Sensibilidades e Autovalores 2 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

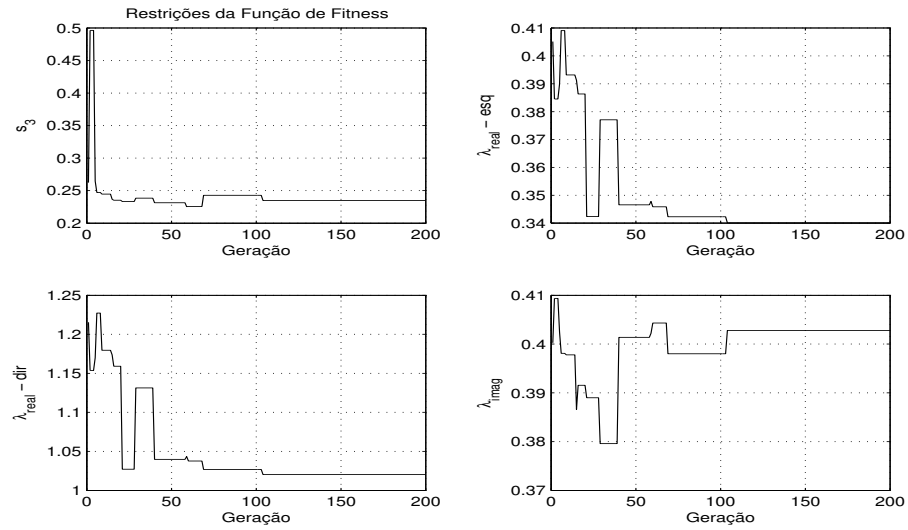


Figura 6.11: Restrições da Função de Fitness - Sensibilidades e Autovalores 3 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

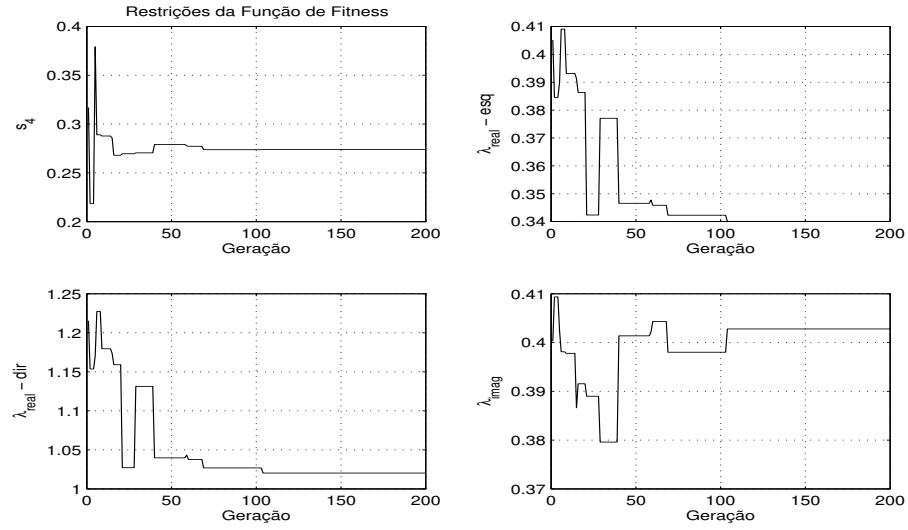


Figura 6.12: Restrições da Função de Fitness - Sensibilidades e Autovalores 4 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

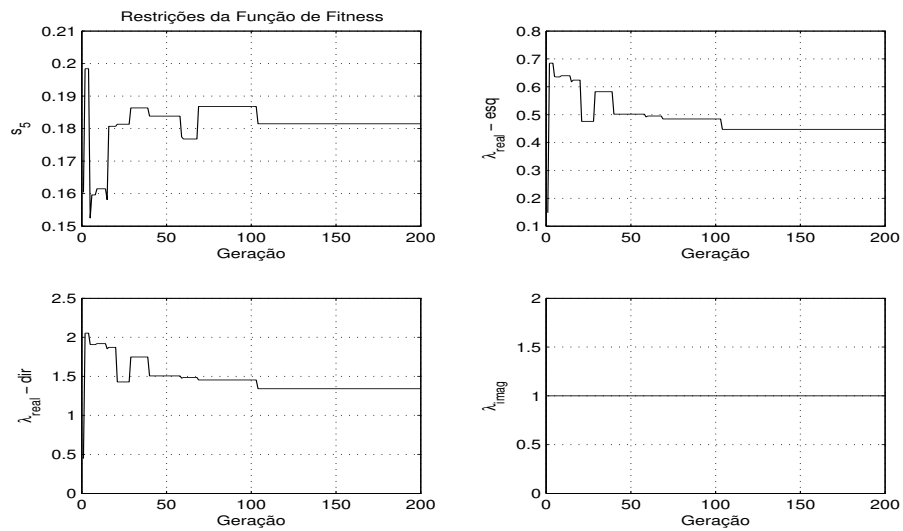


Figura 6.13: Restrições da Função de Fitness - Sensibilidades e Autovalores 5 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

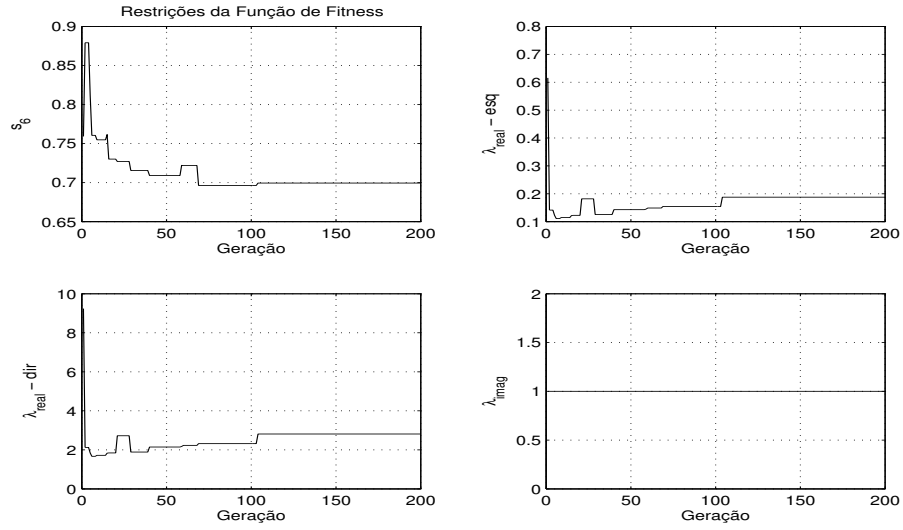


Figura 6.14: Restrições da Função de Fitness - Sensibilidades e Autovalores 6 Normalizados - Super Indivíduo ao longo de 200 gerações - Sistema de ordem 6.

A seguir mostram-se as características das matrizes de ponderação Q do estado e R do controle geradas pelo AG-LQR $_{Q,R}$ nos quatorze estágios sequencias definidos pela variação dos valores da função de *fitness* objetivo normalizada do super indivíduo. São apresentados os respectivos autovalores e traços dessas matrizes os quais podem ser úteis para associar a positividade dessas matrizes com as respostas ao impulso dos controladores obtidos pelo AG a fim de verificar o desempenho do AG.

Estágio	Q	Traço de Q	R	Traço de R
1	7.3934 1.0056 1.4834 0.4096 1.6940 2.7018 9.4349 1.4086 0.7323 0.7115 2.7081 8.0870 1.6732 0.1060 3.1430 23.9318 1.9290 2.9940 80.8417 0.4056 15.3840	145.0728	30.1961 5.4575 23.8023	53.9983
2	12.0971 2.1619 0.9302 1.8850 0.4616 0.6686 7.1653 1.3168 1.5047 2.2040 2.8812 15.2290 1.8400 1.0506 2.4905 26.3808 1.4962 0.1540 81.0336 1.8373 19.2103	161.1162	27.2713 1.7889 25.6399	52.9111
3	7.6996 1.9532 0.9640 1.9038 1.6591 1.4388 10.4223 1.4187 1.1331 1.5739 2.2438 11.0487 1.7102 1.3604 2.5182 17.5722 1.5431 0.6728 77.2836 0.5297 30.9813	155.0076	35.2559 5.3590 26.7142	61.9702
4	8.5630 1.3471 1.2307 1.7404 1.4504 1.9523 7.8138 0.9605 1.8894 1.8285 2.4550 9.4525 2.7357 1.2477 2.1188 31.9540 1.2177 0.7850 81.9897 0.6049 20.8869	160.6599	31.1013 6.7700 31.3687	62.4700
5	10.1908 2.0302 1.2000 2.5488 1.2687 1.4603 8.9135 1.0579 1.4730 1.6643 2.2239 10.6894 1.5435 1.5914 1.9817 30.0589 1.5840 1.0132 69.6899 0.9616 20.7532	150.2957	31.4725 6.2923 27.2500	58.7225

Tabela 6.6: Matrizes Q e R e seus respectivos traços geradas pelo AG-LQR $_{Q,R}$ nos estágios 1-5

Estágio	Q	Traço de Q	R	Traço de R
6	9.8430 1.6449 1.3345 1.7023 1.8234 1.4946 9.1783 1.2395 1.7378 1.5813 2.1348 10.5134 1.7263 1.3151 2.1598 22.7868 0.9770 1.1018 71.1056 0.9020 21.6328	145.0599	35.8078 6.7619 27.7489	63.5566
7	9.9927 1.5557 1.3184 1.5352 1.8213 1.4548 8.8816 1.2542 1.8269 0.2137 2.0027 10.2395 1.4832 1.3936 1.9998 22.2936 2.8051 1.0934 72.0810 0.9229 4.7331	128.2215	32.9698 6.3951 28.1016	61.0714
8	9.4943 1.7732 1.4447 1.8857 1.7461 1.3576 9.0182 1.0465 2.0732 1.4819 1.9842 10.2488 1.8685 1.3689 1.9162 19.9296 1.3472 0.8128 53.0082 1.0150 16.5422	118.2414	36.9532 10.5169 29.0339	65.9871
9	9.1548 1.6465 1.5010 1.6106 1.3795 1.2978 8.4296 1.0258 1.8064 1.1550 1.8151 8.2787 1.6968 1.4296 2.4740 14.4594 1.4492 1.7348 52.6863 0.9176 17.4651	110.4738	33.4933 8.3697 22.7025	56.1958
10	9.7240 1.3332 1.4661 1.3162 1.4396 1.4351 8.5606 1.1204 1.9206 1.6046 1.5233 8.3367 3.0207 1.6413 1.6668 12.6894 1.3024 1.6516 38.7966 1.4887 10.8887	88.9960	40.2869 8.3222 41.3523	81.6392

Tabela 6.7: Matrizes Q e R e seus respectivos traços geradas pelo AG-LQR $_{Q,R}$ nos estágios 6-10

Estágio	Q	Traço de Q	R	Traço de R
11	10.5360 1.5094 1.7089 1.1917 1.5217 1.5586 10.8100 1.7514 2.0460 1.7275 1.7122 8.2642 1.5398 2.0140 1.7164 8.1603 1.3091 1.1351 37.8088 1.2765 11.1871	86.7665	31.8320 6.5605 36.8159	68.6478
12	9.5752 1.2911 1.7799 1.6102 1.4152 1.6620 10.7864 1.6249 2.0173 1.7208 1.8587 9.1224 2.8006 1.9972 2.8079 8.2811 1.2407 1.6013 37.8381 1.2112 11.3983	87.0015	37.6015 8.4910 39.0694	76.6709
13	8.2072 1.7261 1.3901 1.3844 1.4514 1.7095 9.7974 1.5855 1.6173 1.5503 1.7598 9.8065 1.5283 1.9625 2.1742 9.4847 1.5525 1.2983 33.5114 1.3643 9.4323	80.2395	38.8937 8.5694 37.3672	76.2609
14	7.7561 2.2859 1.7319 1.0369 1.2137 1.4484 9.4241 1.9528 1.5697 1.6029 1.9600 6.5049 1.9559 1.7276 2.0373 10.4069 1.8256 1.6337 24.8798 1.5325 12.5080	71.4797	23.7160 9.3856 39.3227	63.0387

Tabela 6.8: Matrizes Q e R e seus respectivos traços geradas pelo AG-LQR $_{Q,R}$ nos estágios 11-14

Tabela 6.9: Autovalores das Matrizes Q e R geradas pelo AG-LQR $_{Q,R}$ nos quatorze estágios

Estágio	Autovalores de Q						Autovalores de R	
	1	2	3	4	5	6	1	2
1	6.1431	7.0920	8.3926	16.9322	25.5520	80.9610	20.6743	33.3240
2	5.8005	12.1253	13.8752	20.8948	27.1894	81.2309	24.4895	28.4217
3	6.5383	9.5336	11.3181	18.4025	31.7663	77.4489	24.1324	37.8377
4	6.7118	7.9147	9.3740	21.8056	32.7199	82.1340	24.4637	38.0063
5	7.3410	9.5062	11.2656	21.4669	30.8025	69.9135	22.7242	35.9983
6	7.7819	8.9998	11.0105	21.0837	24.9002	71.2838	23.9069	39.6497
7	3.5437	7.7884	8.8599	12.6694	23.0061	72.3540	23.6931	37.3783
8	7.3729	8.8200	10.1487	16.8805	21.7196	53.2998	21.7560	44.2311
9	6.8218	7.2914	9.3657	13.9581	20.1159	52.9210	18.1399	38.0559
10	6.4983	7.4920	8.7724	10.1696	16.7728	39.2909	32.4803	49.1589
11	6.5607	7.3040	9.2153	9.4822	15.8280	38.3762	27.3061	41.3417
12	5.6887	7.7616	8.6940	9.3700	17.0834	38.4038	29.8128	46.8581
13	6.8274	7.4352	8.1036	8.4579	15.2173	34.1981	29.5271	46.7338
14	4.9408	6.2189	8.7541	9.7718	15.5962	26.1979	19.3135	43.7252

Para efeito de análise do desempenho do AG-LQR $_{Q,R}$, mostra-se a seguir as curvas de resposta ao impulso, Figuras 6.15-6.16, para os controladores obtidos pelo AG-LQR $_{Q,R}$ nos estágios 2, 7 e 14. Percebe-se que com a diminuição dos traços das matrizes de ponderação do estado, algumas características de resposta ao impulso são melhoradas de um estágio sequencial para outro. Para a saída y_1 o controlador gerado no estágio 14 atingiu melhor desempenho, conforme a especificação de valor máximo M_p de ultrapassagem, além disso apresenta instante de pico t_p e tempo de acomodação t_s menor em relação aos dos outros dois controladores. Para a saída y_2 , verifica-se que o valor de t_p do controlador gerado no estágio 2 é menor em relação aos dos controladores obtidos nos estágios 7 e 14, entretanto estes mostram melhor desempenho em termos de especificações t_s e M_p . Já para a saída y_3 , os valores de t_p e t_s dos controladores obtidos nos estágios 7 e 14 são

um pouco menores em relação ao controlador gerado no estágio 2, além disso os valores de M_p são menores. Finalmente, para a saída y_4 os controladores gerados nos estágios 2 e 14 apresentam valores máximos de ultrapassagem e instantes de pico menores, ao passo que os controladores gerados nos estágios 7 e 14 atingiram o tempo de acomodação relativamente melhor.

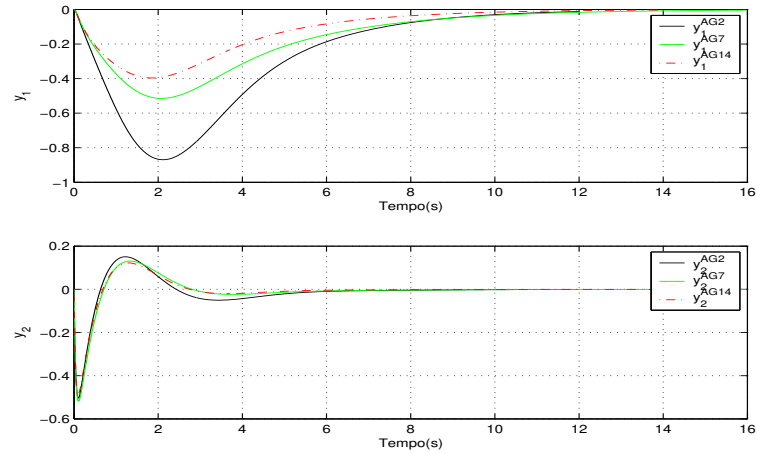


Figura 6.15: Curvas de Resposta ao impulso para as saídas 1 e 2.

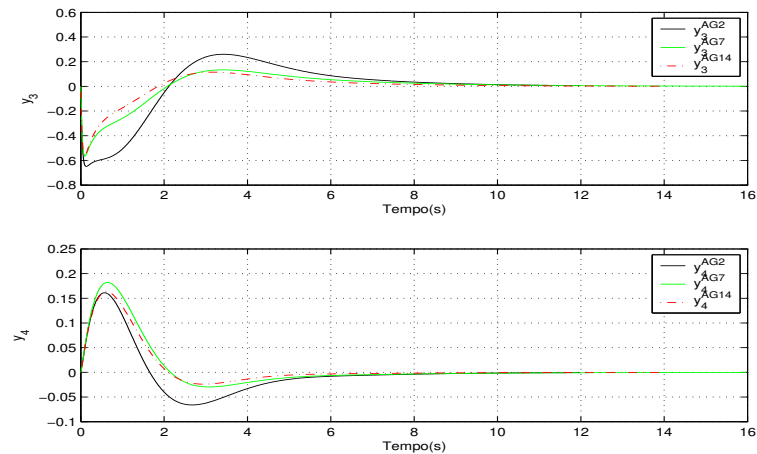


Figura 6.16: Curvas de Resposta ao impulso para as saídas 3 e 4.

6.1.4 Busca das Matrizes Q e R

As matrizes de ponderações Q e R produzidas pelo AG-LQR $_{Q,R}$ para o melhor indivíduo da população final de cada estratégia são mostradas a seguir.

Estratégia não Elitista

$$Q_{AG-LQR} = \begin{bmatrix} 9.5091 & 1.9662 & 1.9252 & 2.1556 & 1.3219 & 1.1376 \\ & 9.3097 & 1.9154 & 1.3142 & 1.7343 & 1.3089 \\ & & 8.0836 & 1.7970 & 1.2999 & 1.7525 \\ & & & 8.7745 & 2.1845 & 1.5802 \\ & & & & 10.3997 & 1.7206 \\ & & & & & 9.2775 \end{bmatrix} \quad (6.1)$$

$$R_{AG-LQR} = \begin{bmatrix} 33.2162 & 8.8878 \\ & 40.2191 \end{bmatrix} \quad (6.2)$$

Estratégia Elitista

$$Q_{AG-LQR} = \begin{bmatrix} 7.7561 & 2.2859 & 1.7319 & 1.0369 & 1.2137 & 1.4484 \\ & 9.4241 & 1.9528 & 1.5697 & 1.6029 & 1.9600 \\ & & 6.5049 & 1.9559 & 1.7276 & 2.0373 \\ & & & 10.4069 & 1.8256 & 1.6337 \\ & & & & 24.8798 & 1.5325 \\ & & & & & 12.5080 \end{bmatrix} \quad (6.3)$$

$$R_{AG-LQR} = \begin{bmatrix} 23.7160 & 9.3856 \\ & 39.3227 \end{bmatrix} \quad (6.4)$$

A Tabela 6.10, mostra os autovalores de malha fechada, as sensibilidades dos autovalores e os ganhos do controlador, obtidos utilizando as matrizes de ponderação representadas por (6.1)-(6.2).

Tabela 6.10: Autovalores, sensibilidades e ganho do controlador obtido pelo AG-LQR_{Q,R} sem estratégia elitista

No	Autovalores	S_i	Ganho
1	-22.7065	1.0015	0.1626 0.0064 -0.1706 -0.8984 -0.2256 1.4863 0.0047 0.1234 -0.2716 -0.4165 -0.3085 1.0164
2	-27.7614	1.0006	
3	-0.8573 + 1.1321i	1.4221	
4	-0.8573 - 1.1321i	1.4172	
5	-1.0782	1.9166	
6	-0.7322	1.5514	

A Tabela 6.11, mostra os autovalores de malha fechada, as sensibilidades dos autovalores e os ganhos do controlador, obtidos utilizando as matrizes de ponderação representadas por (6.3)-(6.4).

Tabela 6.11: Autovalores, sensibilidades e ganho do controlador obtido pelo AG-LQR_{Q,R} com estratégia elitista

No	Autovalores	S_i	Ganho
1	-23.0854	1.0059	0.1950 0.0066 -0.1522 -1.3063 -0.2171 2.0791 -0.0023 0.1316 -0.2483 -0.3240 -0.4259 1.0774
2	-27.8484	1.0023	
3	-1.0411 + 1.2107i	3.6814	
4	-1.0411 - 1.2107i	1.4563	
5	-1.3680	2.2364	
6	-0.4616	1.7297	

6.2 Desempenho do AG-LQG/LTR

Nesta Seção analisa-se os efeitos de variações de parâmetros, tais como semente do gerador pseudo-aleatório, condições iniciais, probabilidade de mutação e tamanho da população no desempenho do AG-LQG/LTR. A Subseção 6.2.1 mostra o efeito de variações na semente do gerador pseudo-aleatório no comportamento da população inicial. Na Subseção 6.2.2 avalia-se o desempenho da população de cada geração em relação à variações nas matrizes de covariâncias iniciais. A Subseção 6.2.3 mostra a influência da probabilidade de mutação na diversidade genética da população e quantidade de indivíduos mutados. A subseção 6.2.4 analisa os efeitos de variações do tamanho da população na quantidade de soluções factíveis, diversidade genética e convergência prematura.

6.2.1 Análise do AG-LQG/LTR com variações na semente

A semente no AG desempenha um papel importante, pois ela determina a sequência de números pseudo-aleatórios para a geração da população inicial e para realizar as operações genéticas do processo, assim como para a ocorrência de qualquer evento aleatório desse processo. Esta subseção mostra o comportamento do melhor indivíduo da população inicial em relação à variação nos valores decimais da função semente. Para esta análise considerou-se as seguintes sementes: 2, 3, 4, 5, 7, 8, 9, 10, 11 e 12. A Tabela 6.12 relaciona o valor da semente, o melhor indivíduo da população inicial com seus respectivos valores de *fitness* F_{max}^{desemp} e F_{min}^{desemp} .

A semente 8 destaca-se por produzir melhores resultados, já que proporciona um menor valor de *fitness* para o melhor indivíduo. As sementes 10 e 12 apresentam resultados próximos ao da semente 8. Já a semente 5 produz resultados menos satisfatórios, pois ocasiona um maior valor de *fitness* para o melhor indivíduo. A semente 9 apresenta resultado bem próximo ao da semente 5.

Tabela 6.12: Semente-Melhor indivíduo-Desempenho.

Semente	Indivíduo	F_{max}^{desemp}	F_{min}^{desemp}
2	0.0712	2.2603	0.2811
3	0.0295	1.4031	0.1427
4	0.0122	0.7688	0.0707
5	0.1338	3.2997	0.4445
7	0.0338	1.6306	0.1765
8	0.0011	0.0921	0.0085
9	0.1334	3.2945	0.4436
10	0.0015	0.1234	0.0113
11	0.0075	0.5210	0.0473
12	0.0014	0.1177	0.0108

As Figuras 6.17, 6.18 e 6.19 ilustram o comportamento da população inicial em termos de comparação dos diagramas de valores singulares das malhas LQR e LTR para o melhor indivíduo considerando-se as sementes 5, 7 e 8, respectivamente. Claramente, pode ser observado que o comportamento das curvas mostra um desempenho melhor para a semente 8.

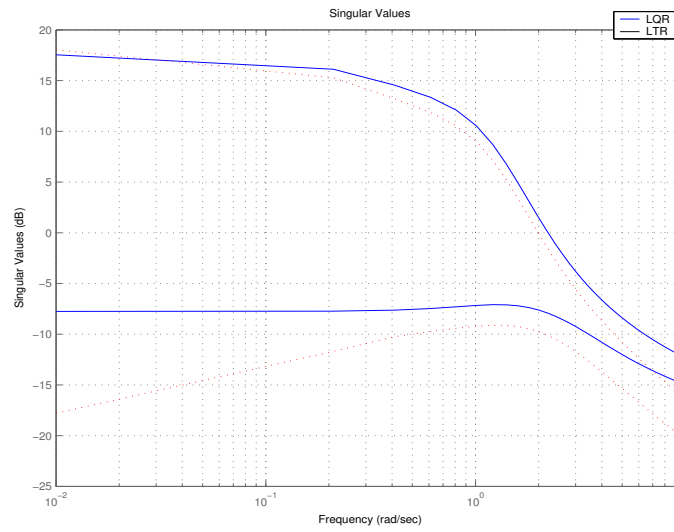


Figura 6.17: Melhor indivíduo da população inicial-Semente 5

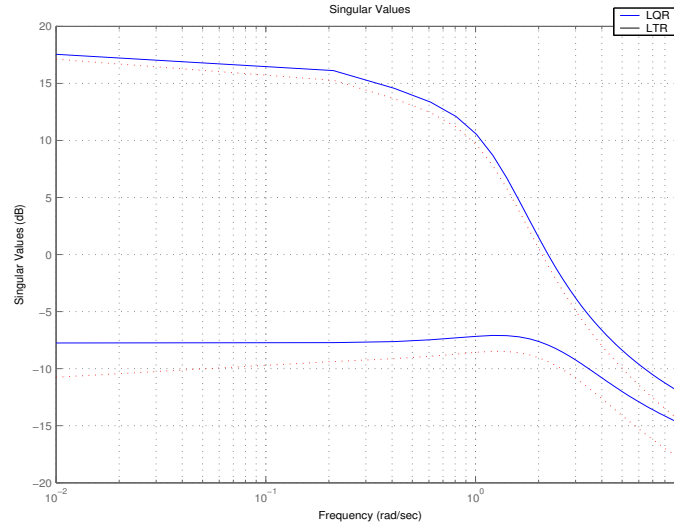


Figura 6.18: Melhor indivíduo da população inicial-Semente 7

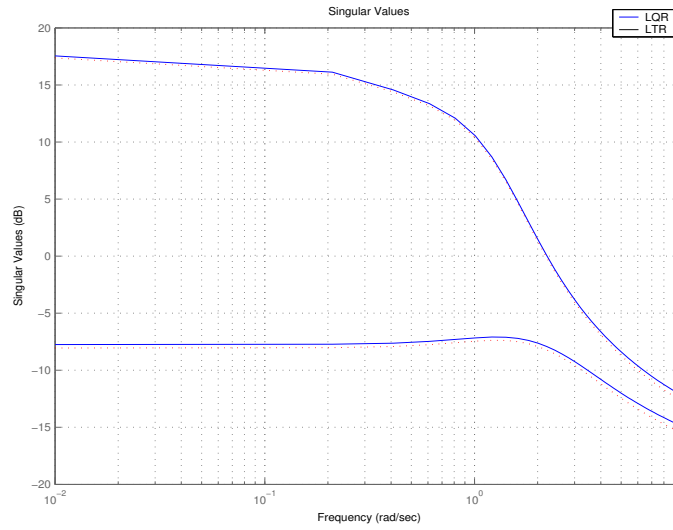


Figura 6.19: Melhor indivíduo da população inicial-Semente 8

6.2.2 Análise do AG-LQG/LTR com variações nas condições iniciais

Nesta subseção avalia-se a convergência do AG-LQG/LTR em relação à variações das matrizes de covariâncias iniciais dadas no domínio do mapeamento $F_{\Xi\Theta}$, Equações (4.9)-(4.10). Para esta análise considerou-se quatro pares de matrizes de covariâncias (Ξ_o, Θ_o) diagonais, representadas por (6.5)-(6.12). Estatísticas da função de *fitness* que fornecem o valor mínimo, a média, o valor máximo e o desvio padrão da soma dos erros quadráticos entre os valores singulares das malhas de projeto LQR e LTR correspondente a um conjunto de 50 faixas de frequências, para cada geração, são apresentadas nas Tabelas 6.13-6.20 a fim de verificar o desempenho do AG para cada caso.

Nesta análise verifica-se *fitness* com menores valores à medida que se diminui o traço das matrizes de covariâncias iniciais. O resultado mais satisfatório para o desempenho do AG é obtido para o caso 4, do qual se observa menores valores de *fitness* para cada geração quando comparado com os outros casos.

$$\Xi_o^1 = \begin{bmatrix} 625 & 0 & 0 & 0 & 0 & 0 \\ & 400 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \quad (6.5)$$

$$\Theta_o^1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ & 10 & 0 & 0 \\ & & 10 & 0 \\ & & & 10 \end{bmatrix} \quad (6.6)$$

Tabela 6.13: Estatísticas da função de *fitness* F_{max}^{desemp} -Condição inicial (Ξ_o^1, Θ_o^1) .

Geração	Max	Média	Min	Std
1	36.1122	29.3092	15.8473	5.4066
2	5.4063	5.1217	4.4902	0.2597
3	2.1446	2.0550	1.8247	0.0993
4	0.5695	0.5147	0.1368	0.1150
5	0.1026	0.0971	0.0867	0.0060
6	0.0165	0.0157	0.0146	0.0006
7	0.0025	0.0024	0.0021	0.0001

Tabela 6.14: Estatísticas da função de *fitness* F_{min}^{desemp} -Condição inicial (Ξ_o^1, Θ_o^1) .

Geração	Max	Média	Min	Std
1	3.5582	3.2735	2.5050	0.2752
2	0.9321	0.8376	0.6521	0.0807
3	0.2352	0.2215	0.1878	0.0147
4	0.0532	0.0481	0.0131	0.0106
5	0.0098	0.0093	0.0083	0.0006
6	0.0016	0.0015	0.0014	0.0001
7	0.0002	0.0002	0.0002	0.0000

$$\Xi_o^2 = \begin{bmatrix} 400 & 0 & 0 & 0 & 0 & 0 \\ & 250 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \quad (6.7)$$

$$\Theta_o^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & 0 \\ & & & 1 \end{bmatrix} \quad (6.8)$$

Tabela 6.15: Estatísticas da função de *fitness* F_{max}^{desemp} -Condição inicial (Ξ_o^2, Θ_o^2) .

Geração	Max	Média	Min	Std
1	7.8394	7.0931	5.4450	0.7111
2	3.0268	2.8212	2.5096	0.1521
3	0.9399	0.8631	0.7874	0.0530
4	0.1801	0.1612	0.0515	0.0322
5	0.0297	0.0284	0.0265	0.0010
6	0.0046	0.0044	0.0042	0.0001
7	0.0007	0.0007	0.0006	0.0000

Tabela 6.16: Estatísticas da função de *fitness* F_{min}^{desemp} -Condição inicial (Ξ_o^2, Θ_o^2) .

Geração	Max	Média	Min	Std
1	1.5836	1.4146	0.9457	0.1801
2	0.3807	0.3470	0.2953	0.0252
3	0.0881	0.0807	0.0735	0.0051
4	0.0171	0.0154	0.0050	0.0030
5	0.0029	0.0028	0.0026	0.0001
6	0.0004	0.0004	0.0004	0.0000
7	0.0001	0.0001	0.0001	0.0000

$$\Xi_o^3 = \begin{bmatrix} 120 & 0 & 0 & 0 & 0 & 0 \\ & 45 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \quad (6.9)$$

$$\Theta_o^3 = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ & 0.01 & 0 & 0 \\ & & 0.01 & 0 \\ & & & 0.01 \end{bmatrix} \quad (6.10)$$

Tabela 6.17: Estatísticas da função de *fitness* F_{max}^{desemp} -Condição inicial (Ξ_o^3, Θ_o^3) .

Geração	Max	Média	Min	Std
1	2.5749	2.3481	1.8474	0.1908
2	0.7764	0.6906	0.5169	0.0747
3	0.1434	0.1331	0.1080	0.0110
4	0.0231	0.0207	0.0048	0.0049
5	0.0036	0.0034	0.0030	0.0002
6	0.0005	0.0005	0.0005	0.0000
7	0.0001	0.0001	0.0001	0.0000

Tabela 6.18: Estatísticas da função de *fitness* F_{min}^{desemp} -Condição inicial (Ξ_o^3, Θ_o^3) .

Geração	Max	Média	Min	Std
1	0.3059	0.2686	0.1909	0.0304
2	0.0725	0.0644	0.0483	0.0070
3	0.0137	0.0127	0.0104	0.0010
4	0.0022	0.0020	0.0005	0.0005
5	0.0004	0.0003	0.0003	0.0000
6	0.0001	0.0001	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000

$$\Xi_o^4 = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \quad (6.11)$$

$$\Theta_o^4 = \begin{bmatrix} 0.0001 & 0 & 0 & 0 \\ & 0.0001 & 0 & 0 \\ & & 0.0001 & 0 \\ & & & 0.0001 \end{bmatrix} \quad (6.12)$$

Tabela 6.19: Estatísticas da função de *fitness* F_{max}^{desemp} -Condição inicial (Ξ_o^4, Θ_o^4) .

Geração	Max	Média	Min	Std
1	1.3517	1.1928	0.8437	0.1341
2	0.2894	0.2494	0.1852	0.0332
3	0.0482	0.0439	0.0360	0.0042
4	0.0075	0.0066	0.0015	0.0016
5	0.0011	0.0011	0.0010	0.0001
6	0.0002	0.0002	0.0002	0.0000
7	0.0000	0.0000	0.0000	0.0000

Tabela 6.20: Estatísticas da função de *fitness* F_{min}^{desemp} -Condição inicial (Ξ_o^4, Θ_o^4) .

Geração	Max	Média	Min	Std
1	0.1302	0.1135	0.0788	0.0136
2	0.0273	0.0236	0.0176	0.0031
3	0.0047	0.0043	0.0035	0.0004
4	0.0007	0.0006	0.0001	0.0002
5	0.0001	0.0001	0.0001	0.0000
6	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000

6.2.3 Análise do AG-LQG/LTR com variações na probabilidade de mutação

Nesta subseção analisa-se o efeito da probabilidade de mutação p_m na diversidade genética do AG-LQG/LTR. A diversidade genética, d_g , pode ser medida tomando-se a razão entre o valor mínimo e a média da função de *fitness* em cada geração, isto é,

$$d_g^j = \frac{\min_i \{F_{ij}^{desemp}\}}{\sum_{i=1}^{n_{ind}} F_{ij}^{desemp} / n_{ind}}, \quad i = 1, \dots, n_{ind} \quad \text{e} \quad j = 1, \dots, n_{pop}, \quad (6.13)$$

em tal caso $0 < d_g \leq 1$. Quando $d_g = 1$, isto indica que todos os indivíduos tem o mesmo código genético, ou seja, eles representam o mesmo ponto no espaço de otimização. Neste caso a diversidade é pequena. Para evitar convergência prematura, p_m precisa ser aumentada de tal maneira a introduzir novo material genético. Por outro lado, se $d_g \ll 1$ isto significa que existe uma grande diversidade genética na população. Para evitar uma busca essencialmente aleatória e diminuir as chances de uma boa solução ser destruída, p_m precisa ser reduzida.

As tabelas 6.21-6.24 relacionam a quantidade de indivíduos que participaram da operação de mutação e a diversidade genética considerando-se $p_m = 0.01$,

$p_m = 0.05$, $p_m = 0.10$ e $p_m = 0.2$, respectivamente, para cada geração. Percebe-se que a geração 7 do caso 1 ($p_m = 0.01$) se destaca por apresentar uma menor diversidade genética, enquanto que a geração 7 do caso 4 ($p_m = 0.2$) apresenta uma maior diversidade genética. Além disso, a quantidade de indivíduos mutados no caso 1 é zero.

Tabela 6.21: Número de indivíduos mutados-Diversidade genética- $p_m = 0.01$.

Geração	n_{mut}	d_g^{max}	d_g^{min}
1	0	0.7565	0.7140
2	0	0.7920	0.7709
3	0	0.8284	0.8304
4	0	0.8151	0.8164
5	0	0.8793	0.8797
6	0	0.9187	0.9188
7	0	0.9522	0.9523

Tabela 6.22: Número de indivíduos mutados-Diversidade genética- $p_m = 0.05$.

Geração	n_{mut}	d_g^{max}	d_g^{min}
1	0	0.7565	0.7140
2	0	0.7920	0.7709
3	0	0.8284	0.8304
4	1	0.2327	0.2346
5	0	0.8835	0.8838
6	0	0.9383	0.9383
7	2	0.8674	0.8676

Tabela 6.23: Número de indivíduos mutados-Diversidade genética- $p_m = 0.1$.

Geração	n_{mut}	d_g^{max}	d_g^{min}
1	0	0.7565	0.7140
2	3	0.4024	0.3876
3	0	0.8093	0.8115
4	0	0.9111	0.9118
5	0	0.9341	0.9343
6	1	0.5856	0.5860
7	2	0.4297	0.4298

Tabela 6.24: Número de indivíduos mutados-Diversidade genética- $p_m = 0.2$.

Geração	n_{mut}	d_g^{max}	d_g^{min}
1	1	0.6312	0.5547
2	3	0.4024	0.3878
3	1	0.6049	0.6089
4	1	0.5991	0.6012
5	2	0.4518	0.4527
6	2	0.6594	0.6597
7	2	0.0002	0.0005

6.2.4 Análise do AG-LQG/LTR com variações no tamanho da população

Um outro parâmetro que influencia no desempenho do algoritmo genético é o tamanho da população. Uma população pequena geralmente fornece uma pequena cobertura do espaço de busca do problema, enquanto que uma grande população fornece uma cobertura representativa do espaço de otimização, além de prevenir convergência prematura para pontos de ótimos locais. Entretanto, para se trabalhar com grandes populações, pode ser necessário um maior número de iterações do algoritmo.

As tabelas 6.25, 6.26, 6.27 e 6.28 relacionam a função de desempenho e a diversidade genética considerando-se $n_{indiv} = 4$, $n_{indiv} = 8$, $n_{indiv} = 9$ e $n_{indiv} = 14$, respectivamente, para cada geração. Como pode-se verificar, o primeiro caso mostra melhores resultados em termos de *fitness*, entretanto a diversidade genética é pequena, sendo esta característica mais acentuada nas gerações 4, 5, 6 e 7. Já o terceiro caso mostra uma diversidade genética mais abrangente, porém com maiores valores de *fitness*. Podemos realçar outros resultados em termos de número de soluções factíveis, para tal considera-se como indivíduo factível aquele que apresenta erro quadrático entre os valores singulares das malhas LQR e LTR menor que 0.0001 por cada valor de frequência. No primeiro caso todos os indivíduos das gerações 6 e 7 são factíveis, e no quarto caso todos os indivíduos da geração 7 são factíveis. O segundo e o terceiro caso não apresentam indivíduos factíveis, o que necessitaria de mais iterações.

Tabela 6.25: Função de desempenho-Diversidade genética- $n_{indiv} = 4$.

Geração	F_{max}^{desemp}	F_{min}^{desemp}	d_g^{max}	d_g^{min}
1	1.3291	0.1328	0.7837	0.7017
2	0.4304	0.0391	0.9457	0.9456
3	0.0793	0.0073	0.9512	0.9517
4	0.0137	0.0013	1.0000	1.0000
5	0.0021	0.0002	1.0000	1.0000
6	0.0003	0.0000	1.0000	1.0000
7	0.0000	0.0000	1.0000	1.0000

Tabela 6.26: Função de desempenho-Diversidade genética- $n_{indiv} = 8$.

Geração	F_{max}^{desemp}	F_{min}^{desemp}	d_g^{max}	d_g^{min}
1	1.7184	0.1907	0.7108	0.6274
2	0.5533	0.0503	0.6990	0.6877
3	0.1695	0.0155	0.9641	0.9645
4	0.0073	0.0007	0.2781	0.2800
5	0.0046	0.0004	0.9973	0.9973
6	0.0007	0.0001	0.9982	0.9982
7	0.0000	0.0000	0.2926	0.2926

Tabela 6.27: Função de desempenho-Diversidade genética- $n_{indiv} = 9$.

Geração	F_{max}^{desemp}	F_{min}^{desemp}	d_g^{max}	d_g^{min}
1	1.8940	0.2202	0.5585	0.3917
2	1.1253	0.1080	0.3854	0.2209
3	0.8969	0.0834	0.3437	0.1889
4	0.6952	0.0636	0.2481	0.1339
5	0.9999	0.0942	0.3513	0.1959
6	0.2207	0.0201	0.1236	0.0605
7	1.8452	0.2119	0.5444	0.3770

Tabela 6.28: Função de desempenho-Diversidade genética- $n_{indiv} = 14$.

Geração	F_{max}^{desemp}	F_{min}^{desemp}	d_g^{max}	d_g^{min}
1	2.3333	0.2927	0.7565	0.7140
2	0.8131	0.0750	0.7920	0.7709
3	0.1845	0.0169	0.8284	0.8304
4	0.0083	0.0008	0.2327	0.2346
5	0.0056	0.0005	0.8835	0.8838
6	0.0009	0.0001	0.9383	0.9383
7	0.0001	0.0000	0.8674	0.8676

6.3 Conclusão

Este capítulo mostrou como é possível melhorar o desempenho do AG hierárquico. Os resultados revelam que a semente do gerador pseudo-aleatório, condições iniciais, probabilidade de mutação e tamanho da população são parâmetros que influenciam no comportamento do algoritmo e que o ajuste adequado desses parâmetros conduz a melhorias no desempenho do AG. Além disso, a estratégia elitista incrementou melhorias substanciais no desempenho do AG.

CAPÍTULO 7

CONCLUSÃO

Neste trabalho apresentou-se um modelo hierárquico de algoritmos genéticos destinado à recuperação da malha de transferência pela entrada como uma alternativa atrativa imbuída de inteligência para sintonizar controladores e observadores de estados. Os projetos de controladores LQR, LQG e LQG/LTR foram desenvolvidos utilizando uma hierarquia de algoritmos baseada em otimização combinatoria e algoritmos genéticos, enfocando os modelos para sintonizar os ganhos das malhas de controle.

Na análise de desempenho do AG Hierárquico percebeu-se a influência de determinados parâmetros do AG-LQG/LTR no comportamento da função de *fitness*, na diversidade genética da população, quantidade de indivíduos mutados, convergência prematura, quantidade de soluções factíveis e, que o ajuste adequado destes parâmetros conduz a melhorias no desempenho do referido algoritmo. Além disso, a análise de convergência realizada em relação aos efeitos de uma estratégia elitista incorporada ao AG-LQR_{Q,R} evidenciou por meio da evolução do super indivíduo ao longo do processo melhorias substanciais no comportamento do AG-LQR_{Q,R}.

O procedimento de desempenho do método foi avaliado em um modelo de sistema dinâmico multivariável que representa uma aeronave. Baseado na implementação do modelo e análise de desempenho de controladores e observadores, este método de sintonia é visto como uma boa alternativa para aplicações do mundo real. O desenvolvimento dos modelos culminou em um procedimento para o projeto inteligente de sintonia de controladores LQR, LQG e LQG/LTR que se

baseia em um hierarquia composta de três algoritmos genéticos.

7.1 Trabalhos Futuros

Para continuação e direcionamento de futuros trabalhos, convém destacar os seguintes tópicos:

- Desenvolver uma estrutura unificada que coordene a operação de busca da hierarquia;
- Desenvolver métricas específicas que associam as restrições de autoestrutura com a resposta temporal do sistema.
- Investigar a robustez em relação a sensibilidade dos autovalores, após a recuperação da malha LQR pela entrada;
- Associação da positividade das matrizes de ponderações e a solução da equação de Riccati com a recuperação LTR.

APÊNDICE A

PROJETO DO FILTRO DE KALMAN

No projeto do FK são considerados ruídos do processo e da medida, ambos gaussianos de média zero, variância conhecida e não correlacionados entre si, os quais serão utilizados para se obter a estimativa do estado.

Considere o sistema dinâmico representado por

$$\dot{x}(t) = Ax(t) + G\xi(t), \quad (\text{A.1})$$

em que $x(t) \in R^n$, $\xi(t) \in R^m$ é um processo estocástico chamado de ruído no estado (ou no processo), que se admite ser branco, gaussiano e tal que

$$E[\xi(t)] = 0, \quad (\text{A.2})$$

$$E[\xi(t)\xi(t+\tau)^T] = \Xi\delta(t-\tau). \quad (\text{A.3})$$

$\Xi = \Xi^T > 0$ é a matriz de covariância do ruído no estado ou densidade espectral do ruído de processo, que indica a potência do ruído e $\delta(t-\tau)$, *Delta de Dirac*, o qual traduz que o valor do ξ no instante t não está relacionado com seu valor em outro instante $\tau \neq t$. Admitindo-se que os sensores não meçam o estado diretamente, mas apenas variáveis de saída sujeitas à ação de um ruído de medida aditivo, isto é,

$$y(t) = Cx(t) + \nu(t) \quad y(t) \in R^p, \quad (\text{A.4})$$

$\nu(t)$ é suposto branco, gaussiano, independente de $\xi(t)$ e tal que

$$E[\nu(t)] = 0; \quad (\text{A.5})$$

$$E[\nu(t)\nu(t+\tau)^T] = \Theta\delta(t-\tau); \quad (\text{A.6})$$

$$E[\xi(t)\nu(t+\tau)^T] = 0, \quad \text{para todo } t \text{ e } \tau, \quad (\text{A.7})$$

em que $\Theta = \Theta^T > 0$ é a matriz de covariância do ruído de medida ou densidade espectral do ruído de medida, que indica a potência do ruído. O par (A, C) é suposto observável. O sistema dinâmico acima pode ser representado na forma de diagrama de blocos, conforme mostra a Figura A.1.

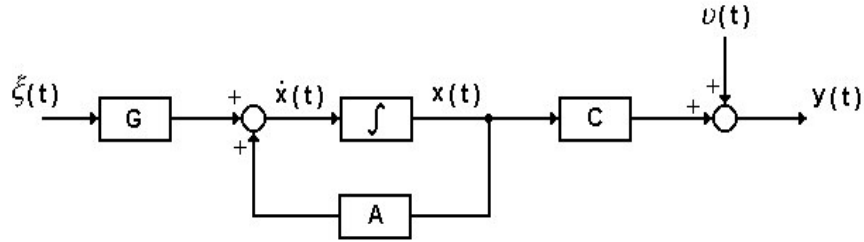


Figura A.1: Diagrama em blocos do sistema dinâmico estocástico

O problema a ser resolvido consiste em obter-se uma estimativa $\hat{x}(t)$ do estado $x(t)$ a partir da observação da saída $\{y(\tau), \tau \leq t\}$. O Filtro de Kalman FK é um sistema dinâmico, com a estrutura representada na Figura A.2, em que a matriz de ganho L é dada por

$$L = \Sigma C^T \Theta^{-1}, \quad (\text{A.8})$$

em que Σ é a única solução simétrica definida positiva da Equação Algébrica de Riccati-EAR (Cruz 1996), apresentada abaixo

$$A\Sigma + \Sigma A^T + G\Xi G^T - \Sigma C^T \Theta^{-1} C \Sigma = 0. \quad (\text{A.9})$$

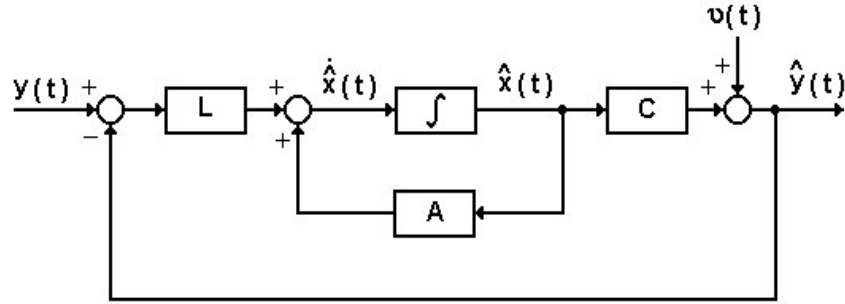


Figura A.2: Diagrama em blocos do Filtro de Kalman

A estimativa gerada pelo filtro é ótima no sentido de que a variância do erro de estimação, Equação A.10, é mínima, ou seja

$$\hat{x}(t) = x(t) - \hat{x}(t), \quad (\text{A.10})$$

isto é,

$$\min \sum_{i=1}^n E\{[x_i(t) - \hat{x}_i(t)]^2\}. \quad (\text{A.11})$$

Sendo assim, a dinâmica do Filtro de Kalman é dada por

$$\dot{\hat{x}}(t) = A\hat{x}(t) + L[y(t) - C\hat{x}(t)]. \quad (\text{A.12})$$

O Filtro de Kalman é tal que

$$\text{Re}[\lambda_i(A - LC)] < 0, \quad (i = 1, 2, \dots, n). \quad (\text{A.13})$$

Dualidade entre a Estimação e o Controle

O Filtro de Kalman e o LQR são duais (Blackmore and Bitmead 1995), o que significa que é possível obter relações análogas nos dois casos simplesmente fazendo-se a correspondência entre os parâmetros indicada na Tabela A.1.

Tabela A.1: Parâmetros duais do LQR e do FK

LQR	FK
A	A^T
B	C^T
Q	$G\Xi G^T$
R	Θ
P	Σ
K	L^T

APÊNDICE B

RECUPERAÇÃO DOS GANHOS DE MALHA NA ENTRADA E SAÍDA

O objetivo do projeto LQG/LTR é fazer o sistema completo (Regulador com observador de estados) tender ao LQR, que é robusto. Considera-se a planta,

$$\dot{x}(t) = Ax(t) + Bu(t) + G\xi(t), \quad (\text{B.1})$$

$$y(t) = Cx(t) + \nu(t), \quad (\text{B.2})$$

sendo $x(t) \in R^n$, $u(t) \in R^m$, $\xi(t)$ um ruído branco e gaussiano no processo, $\nu(t)$ um ruído branco e gaussiano de medida e considera-se ainda o sistema com o observador (Filtro de Kalman) da Figura B.1.

A variação do estado estimado é dada por

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\tilde{y}(t), \quad (\text{B.3})$$

sendo

$$\tilde{y}(t) = y(t) - C\hat{x}(t). \quad (\text{B.4})$$

Substituindo-se a Equação (B.4) em (B.3)

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)), \\ &= (A - LC)\hat{x}(t) + Bu(t) + Ly(t). \end{aligned} \quad (\text{B.5})$$

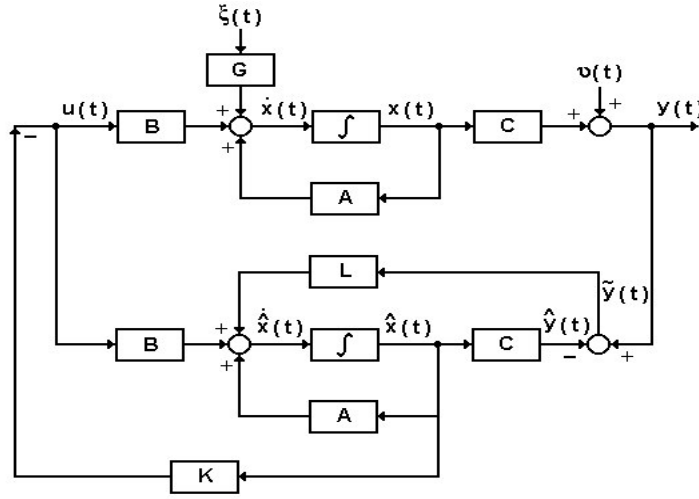


Figura B.1: Diagrama em blocos do controlador com o Filtro de Kalman.

Aplicando-se a Transformada de Laplace na Equação (B.5),

$$\begin{aligned} s\hat{X}(s) &= (A - LC)\hat{X}(s) + BU(s) + LY(s), \\ \hat{X}(s) &= (sI - A + LC)^{-1}[BU(s) + LY(s)]. \end{aligned} \quad (\text{B.6})$$

Considere a lei de controle ótimo,

$$U(s) = -K\hat{X}(s). \quad (\text{B.7})$$

Substituindo-se a Equação (B.6) em (B.7),

$$U(s) = -K(sI - A + LC)^{-1}[BU(s) + LY(s)]. \quad (\text{B.8})$$

Expandindo,

$$U(s) = -H_u(s)U(s) - H_y(s)Y(s), \quad (\text{B.9})$$

sendo

$$H_u(s) = -K(sI - A + LC)^{-1}B, \quad (\text{B.10})$$

e

$$H_y(s) = -K(sI - A + LC)^{-1}L. \quad (\text{B.11})$$

A matriz do estimador é definida por

$$\Phi_o(s) = (sI - A + LC)^{-1}. \quad (\text{B.12})$$

Logo

$$H_u(s) = K\Phi_o B, \quad (\text{B.13})$$

e

$$H_y(s) = K\Phi_o L. \quad (\text{B.14})$$

Manipulando com a Equação (B.9) a fim de obter uma função de retorno de Y para U ,

$$U(s)(I - H_u) = -H_y Y(s). \quad (\text{B.15})$$

Reorganizando os termos da Equação (B.15), obtém-se

$$U(s) = -(I + H_u)^{-1}H_y Y(s). \quad (\text{B.16})$$

A função de transferência da realimentação,

$$F(s) = \frac{U(s)}{Y(s)}, \quad (\text{B.17})$$

sendo

$$F(s) = -(I + H_u)^{-1}H_y, \quad (\text{B.18})$$

a função de transferência em termos das matrizes do Sistema e ganhos dos controle e do observador,

$$F(s) = \{I + K[sI - (A - LC)]^{-1}B\}^{-1}K\Phi_o L. \quad (\text{B.19})$$

Usando-se o lema da inversa, a Equação (B.17) pode ser escrita da seguinte maneira: $(A_1 + B_1 C_1 D_1)^{-1} = A_1^{-1} - A_1^{-1} B_1 (D_1 A_1^{-1} B_1 + C_1^{-1})^{-1} D_1 A_1^{-1}$. Assim, fazendo: $A_1 = I$, $B_1 = K$, $C_1 = (sI - A + LC)^{-1}$ e $D_1 = B$, tem-se

$$\begin{aligned} F(s) &= \{I - K[sI - (A - BK - LC)]^{-1} B\} K \Phi_o L, \\ &= \{K - K[sI - (A - BK - LC)]^{-1} BK\} \Phi_o L, \\ &= K\{I - [sI - (A - BK - LC)]^{-1} BK\} \Phi_o L. \end{aligned}$$

Fatorando $[sI - (A - BK - LC)]^{-1}$, tem-se

$$\begin{aligned} F(s) &= K[sI - (A - BK - LC)]^{-1} \\ &\quad \{[sI - (A - BK - LC)] - BK\} \Phi_o L, \\ &= K[sI - (A - BK - LC)]^{-1} \Phi_o^{-1} \Phi_o L, \\ &= K[sI - (A - BK - LC)]^{-1} L. \end{aligned} \tag{B.20}$$

Definindo,

$$\Phi_r = [sI - (A - BK - LC)]^{-1}. \tag{B.21}$$

A Figura B.2 apresenta o diagrama de blocos para o sistema, controle e observador para fins de projeto de recuperação da malha LQR.

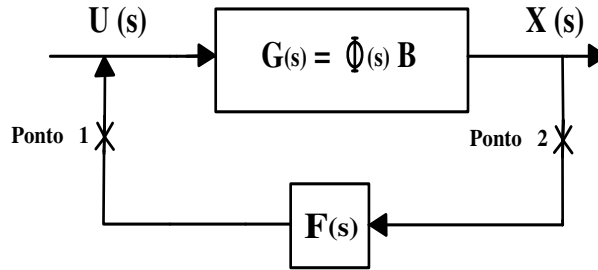


Figura B.2: Esquema Geral LQG/LTR para recuperação da Malha LQR.

A estratégia de recuperação da malha LQR por meio do projeto LQG/LTR ocorre pelo ajuste do ganho K_{QR} do controlador LQR ou pelo ajuste do ganho $L_{\Xi\Theta}$ do observador de estado.

Abrindo-se a malha na entrada da Figura B.2 no Ponto 1, o ganho do regulador referido à entrada,

$$L_r^e(s) = F(s)G(s) = K\Phi_r LC\Phi B, \quad (\text{B.22})$$

sendo $L_r^e(s)$ o *Ganho de Malha Aberta na Entrada*.

Abrindo-se a malha na saída da Figura B.2 no Ponto 2, o ganho do regulador referido à saída,

$$L_r^o(s) = G(s)F(s) = C\Phi BK\Phi_r L, \quad (\text{B.23})$$

sendo $L_r^o(s)$ o *Ganho de Malha Aberta na Saída*.

B.1 Recuperação do Ganho de Malha na Entrada

No processo de recuperação da malha LQR pela entrada, o ganho K do controlador é mantido constante e recupera-se a robustez do LQR por meio de ajustes no ganho L do filtro de *Kalman*,

$$L_r^e(s) \rightarrow L_{LQR}(s), \quad (\text{B.24})$$

sendo $L_r^e(s)$ o ganho da recuperação e o ganho de malha do $L_{LQR}(s)$. Se a função de transferência do sistema for quadrada e de fase mínima, então L poderá ser determinado de modo que $L_r^e(s)$ torne-se arbitrariamente próximo de $L_{LQR}(s)$.

Considerando-se o modelo dinâmico estocástico em variáveis de estado dado pelas Equações (B.1) e (B.2) e considerando que $\xi(t)$ e $\nu(t)$ são ruídos brancos, gaussianos e independentes, com média zero e covariâncias,

$$E \left\{ \begin{bmatrix} \xi(t) \\ \nu(t) \end{bmatrix} \begin{bmatrix} \xi(t) & \nu(t) \end{bmatrix} \right\} = \begin{bmatrix} \Xi & 0 \\ 0 & \Theta \end{bmatrix}. \quad (\text{B.25})$$

No projeto LQG, a solução da Equação Algébrica de Riccati, Σ , Equação (B.26), fornece o ganho L do Filtro de Kalman, Equação (B.27)

$$0 = A\Sigma + \Sigma A^T + G\Xi G^T - \Sigma C^T \Theta^{-1} C \Sigma, \quad (\text{B.26})$$

$$L = \Sigma C^T \Theta^{-1}. \quad (\text{B.27})$$

B.1.1 Recuperação da malha

A recuperação da malha LQR pela entrada é realizada por meio das variações nas matrizes de covariâncias. Fazendo-se as escolhas convenientes para as matrizes de covariâncias

$$\Xi = v^2 \Xi_o + BB^T \quad (\text{B.28})$$

e

$$\Theta = v^2 \Theta_o. \quad (\text{B.29})$$

Considerando,

$$G = I. \quad (\text{B.30})$$

Substituindo-se as Equações (B.28)-(B.30), tem-se para as Equações (B.26) e (B.27),

$$0 = A\Sigma + \Sigma A^T + (v^2 \Xi_o + BB^T) - \Sigma C^T (v^2 \Theta_o)^{-1} C \Sigma, \quad (\text{B.31})$$

$$L = \Sigma C^T (v^2 \Theta_o)^{-1}. \quad (\text{B.32})$$

Segundo Lewis (Lewis and Syrmos 1995), Kwakernaak (Kwakernaak and Sivan 1972) mostra que sob as hipóteses: $v \rightarrow 0$ e $\Sigma \rightarrow 0$, obtém-se para as Equações (B.31) e (B.32)

$$\Sigma C^T (v^2 \Theta_o)^{-1} C \Sigma \rightarrow BB^T \quad (\text{B.33})$$

e

$$L \rightarrow \Sigma C^T (v^2 \Theta_o)^{-1}, \quad (\text{B.34})$$

mas

$$\begin{aligned} L(v^2 \Theta_o)L^T &= \Sigma C^T (v^2 \Theta_o)^{-1} (v^2 \Theta_o) [\Sigma C^T (v^2 \Theta_o)^{-1}]^T, \\ &= \Sigma C^T (v^2 \Theta_o)^{-1} (v^2 \Theta_o) (v^2 \Theta_o)^{-1} C \Sigma, \\ &= \Sigma C^T (v^2 \Theta_o)^{-1} C \Sigma. \end{aligned} \quad (\text{B.35})$$

Logo, das Equações (B.33) e (B.35)

$$L(v^2\Theta_o)L^T \rightarrow BB^T. \quad (\text{B.36})$$

A solução,

$$L \rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}, \quad (\text{B.37})$$

sendo U uma matriz unitária.

Definindo-se a matriz de realimentação do compensador por

$$\Phi_c(s) = [sI - (A - BK)]^{-1}, \quad (\text{B.38})$$

o ganho de malha na entrada é

$$\begin{aligned} L_r(s) &= F(s)G(s) = K[sI - (A - BK - LC)]^{-1}LC\Phi B, \\ &= K[sI - (A - BK) + LC]^{-1}LC\Phi B, \\ &= K[\Phi_c^{-1} + LC]LC\Phi B. \end{aligned} \quad (\text{B.39})$$

B.1.2 Mostrando a Recuperação

Usando o lema da inversa, como feito anteriormente, Equação (B.17), na Equação (B.39), tem-se

$$\begin{aligned} L_r(s) &= K[\Phi_c - \Phi_c L(I + C\Phi_c L)^{-1}C\Phi_c]LC\Phi B, \\ &= K\Phi_c[I - L(I + C\Phi_c L)^{-1}C\Phi_c]LC\Phi B, \\ &= K\Phi_c[L - L(I + C\Phi_c L)^{-1}C\Phi_c L]C\Phi B. \end{aligned}$$

Fatorando $[I + C\Phi_c L]^{-1}$, tem-se

$$\begin{aligned} L_r(s) &= K\Phi_c L[I - (I + C\Phi_c L)^{-1}C\Phi_c L]C\Phi B, \\ &= K\Phi_c L(I + C\Phi_c L)^{-1}[(I + C\Phi_c L) - C\Phi_c L]C\Phi B, \\ &= K\Phi_c L(I + C\Phi_c L)^{-1}C\Phi B. \end{aligned} \quad (\text{B.40})$$

Calculando-se $L(I + C\Phi_c L)^{-1}$, quando $L \rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}$

$$\begin{aligned}
 L(I + C\Phi_c L)^{-1} &\rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}(I + C\Phi_c \frac{1}{v}BU\Theta_o^{-\frac{1}{2}})^{-1}, \\
 &\rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}(vI + C\Phi_c BU\Theta_o^{-\frac{1}{2}})^{-1}v, \\
 &\rightarrow \underbrace{BU\Theta_o^{-\frac{1}{2}}(vI + C\Phi_c BU\Theta_o^{-\frac{1}{2}})^{-1}}_{v \rightarrow 0}, \\
 &\rightarrow BU\Theta_o^{-\frac{1}{2}}(C\Phi_c BU\Theta_o^{-\frac{1}{2}})^{-1}, \\
 &\rightarrow BU\Theta_o^{-\frac{1}{2}}(U\Theta_o^{-\frac{1}{2}})^{-1}(C\Phi_c B)^{-1}, \\
 L(I + C\Phi_c L)^{-1} &\rightarrow B(C\Phi_c B)^{-1}.
 \end{aligned} \tag{B.41}$$

Assim, das Equações (B.40) e (B.41),

$$L_r(s) \rightarrow K\Phi_c B(C\Phi_c B)^{-1}C\Phi B. \tag{B.42}$$

Como

$$\Phi_c = (\Phi^{-1} + BK)^{-1}, \tag{B.43}$$

usando-se novamente o lema da inversa, na Equação (B.43),

$$\begin{aligned}
 \Phi_c &= \Phi - \Phi B(K\Phi B + I)^{-1}K\Phi, \\
 &= \Phi[I - B(K\Phi B + I)^{-1}K\Phi].
 \end{aligned}$$

Então

$$\Phi_c B = \Phi B[I - (K\Phi B + I)^{-1}K\Phi B], \tag{B.44}$$

$$C\Phi_c B = C\Phi B[I - (K\Phi B + I)^{-1}K\Phi B],$$

a matriz inversa de $C\Phi_c B$,

$$(C\Phi_c B)^{-1} = [I - (K\Phi B + I)^{-1}K\Phi B]^{-1}(C\Phi B)^{-1}. \tag{B.45}$$

Substituindo as Equações (B.44) e (B.45) na Equação (B.42),

$$\begin{aligned} L_r(s) \rightarrow & K\Phi B[I - (K\Phi B + I)^{-1}K\Phi B] \\ & [I - (K\Phi B + I)^{-1}K\Phi B]^{-1}(C\Phi B)^{-1}C\Phi B. \end{aligned} \quad (\text{B.46})$$

Assim,

$$L_r(s) \rightarrow K\Phi B \quad \text{ou} \quad L_r(s) \rightarrow L_{LQR}(s). \quad (\text{B.47})$$

De (B.47) tem-se a prova de que escolhendo-se devidamente as matrizes de covariâncias e variando-se v pode-se obter a robustez desejada.

B.2 Recuperação do Ganho de Malha na Saída

No processo de recuperação da malha LQR pela saída, o ganho L do filtro *Kalman* é mantido constante e recupera-se a robustez do LQR por meio de ajustes no ganho K do controlador LQR,

$$L_r^o(s) \rightarrow L_{LQR}(s). \quad (\text{B.48})$$

Esta relação é considerada a primeira abstração da solução proposta por (Kwakernaak and Sivan 1972) e que são amplamente discutidas em (Doyle and Stein 1979) e (Lewis and Syrmos 1995). Esta equação deixa margem para propostas, como atingir este objetivo. A nossa proposta consiste no desenvolvimento de um algoritmo genético que atua em níveis de recuperação de malha.

Considerando-se o modelo dinâmico estocástico em variáveis de estado,

$$\dot{x}(t) = Ax(t) + Bu(t) + G\xi(t), \quad (\text{B.49})$$

$$y(t) = Cx(t) + \nu(t), \quad (\text{B.50})$$

sendo $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $\xi(t)$ um ruído branco e gaussiano no processo, $\nu(t)$ um ruído branco e gaussiano de medida e considerando-se ainda o sistema com o observador (Filtro de *Kalman*) da Figura B.1.

Da mesma forma, abrindo-se a malha na saída da Figura B.2 (*Ponto 2*), o ganho do regulador,

$$L_r^o(s) = G(s)F(s) = C\Phi BK\Phi_r L \quad (\text{B.51})$$

B.2.1 Recuperação da malha

A recuperação da malha LQR é realizada por meio das variações nas matrizes de ponderação, fazendo-se as escolhas convenientes para as covariâncias

$$Q_i = v^2 Q_0 + BB^T \quad (\text{B.52})$$

e

$$R_i = v^2 R_0. \quad (\text{B.53})$$

Substituindo-se as Equações (B.52)-(B.53), tem-se para as Equações (B.26) e (B.27),

$$0 = AP_i + \Sigma A^T + (v^2 P_i + BB^T) - PB^T(v^2 P_i)^{-1}BP \quad (\text{B.54})$$

$$K = R^{-1}B^T P. \quad (\text{B.55})$$

Segundo Lewis (Lewis and Syrmos 1995), Kwakernaak (Kwakernaak and Sivan 1972) mostra que sob as hipóteses: $v \rightarrow 0$ e $\Sigma \rightarrow 0$, obtém-se para as Equações (B.54) e (B.55)

$$PC^T(v^2\Theta_o)^{-1}C\Sigma \rightarrow BB^T \quad (\text{B.56})$$

e

$$L \rightarrow \Sigma C^T(v^2\Theta_o)^{-1}, \quad (\text{B.57})$$

mas

$$\begin{aligned} L(v^2\Theta_o)L^T &= \Sigma C^T(v^2\Theta_o)^{-1}(v^2\Theta_o)[\Sigma C^T(v^2\Theta_o)^{-1}]^T, \\ &= \Sigma C^T(v^2\Theta_o)^{-1}(v^2\Theta_o)(v^2\Theta_o)^{-1}C\Sigma, \\ &= \Sigma C^T(v^2\Theta_o)^{-1}C\Sigma. \end{aligned} \quad (\text{B.58})$$

Logo, das Equações (B.56) e (B.58)

$$L(v^2\Theta_o)L^T \rightarrow BB^T. \quad (\text{B.59})$$

A solução,

$$L \rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}, \quad (\text{B.60})$$

sendo U uma matriz unitária.

B.2.2 Mostrando a Recuperação

Usando o Lema da Inversa, Equação (B.17) na Equação (B.39),

$$\begin{aligned} L_r^o(s) &= C\Phi BK[\Phi_c - \Phi_c L(I + C\Phi_c L)^{-1}C\Phi_c]L, \\ &= C\Phi BK\Phi_c[I - L(I + C\Phi_c L)^{-1}C\Phi_c]L, \\ &= C\Phi BK\Phi_c[L - L(I + C\Phi_c L)^{-1}C\Phi_c L]. \end{aligned}$$

Fatorando $[I + C\Phi_c L]^{-1}$,

$$\begin{aligned} L_r^o(s) &= C\Phi BK\Phi_c L[I - (I + C\Phi_c L)^{-1}C\Phi_c L], \\ &= C\Phi BK\Phi_c L(I + C\Phi_c L)^{-1}[(I + C\Phi_c L) - C\Phi_c L], \\ &= C\Phi BK\Phi_c L(I + C\Phi_c L)^{-1}. \end{aligned} \quad (\text{B.61})$$

Considerando que

$$L \rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}. \quad (\text{B.62})$$

Calcula-se $L(I + C\Phi_c L)^{-1}$,

$$\begin{aligned}
 L(I + C\Phi_c L)^{-1} &\rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}(I + C\Phi_c \frac{1}{v}BU\Theta_o^{-\frac{1}{2}})^{-1}, \\
 &\rightarrow \frac{1}{v}BU\Theta_o^{-\frac{1}{2}}(vI + C\Phi_c BU\Theta_o^{-\frac{1}{2}})^{-1}v, \\
 &\rightarrow \underbrace{BU\Theta_o^{-\frac{1}{2}}(vI + C\Phi_c BU\Theta_o^{-\frac{1}{2}})^{-1}}_{v \rightarrow 0}, \\
 &\rightarrow BU\Theta_o^{-\frac{1}{2}}(C\Phi_c BU\Theta_o^{-\frac{1}{2}})^{-1}, \\
 &\rightarrow BU\Theta_o^{-\frac{1}{2}}(U\Theta_o^{-\frac{1}{2}})^{-1}(C\Phi_c B)^{-1}, \\
 L(I + C\Phi_c L)^{-1} &\rightarrow B(C\Phi_c B)^{-1}.
 \end{aligned} \tag{B.63}$$

Assim, das Equações (B.61) e (B.63),

$$L_r^o(s) \rightarrow C\Phi BK\Phi_c B(C\Phi_c B)^{-1}. \tag{B.64}$$

Como

$$\Phi_c = (\Phi^{-1} + BK)^{-1}, \tag{B.65}$$

usando-se novamente o Lema da Inversa, na Equação (B.65),

$$\begin{aligned}
 \Phi_c &= \Phi - \Phi B(K\Phi B + I)^{-1}K\Phi, \\
 &= \Phi[I - B(K\Phi B + I)^{-1}K\Phi].
 \end{aligned}$$

Então,

$$\Phi_c B = \Phi B[I - (K\Phi B + I)^{-1}K\Phi B], \tag{B.66}$$

$$C\Phi_c B = C\Phi B[I - (K\Phi B + I)^{-1}K\Phi B].$$

A matriz inversa de $C\Phi_c B$,

$$(C\Phi_c B)^{-1} = [I - (K\Phi B + I)^{-1}K\Phi B]^{-1}(C\Phi B)^{-1}. \tag{B.67}$$

Substituindo as Equações (B.66) e (B.67) na Equação (B.64),

$$\begin{aligned} L_r^o(s) \rightarrow & C\Phi B K\Phi B [I - (K\Phi B + I)^{-1} K\Phi B] \\ & [I - (K\Phi B + I)^{-1} K\Phi B]^{-1} (C\Phi B)^{-1}. \end{aligned}$$

Assim,

$$L_r^o(s) \rightarrow K\Phi B \quad \text{ou} \quad L_r(s) \rightarrow L_{LQR}(s). \quad (\text{B.68})$$

Da Equação (B.68) e do mapeamento (B.62) tem-se a prova de que escolhendo-se devidamente as matrizes de covariâncias e variando-se v pode-se obter a robustez desejada.

APÊNDICE C

O SISTEMA DINÂMICO MULTIVARIÁVEL TESTE

Para testar a metodologia proposta nesta dissertação escolheu-se como sistema teste o modelo de uma aeronave da *Lockheed*, tipo *L1011 Tristar*. Este modelo foi utilizado por (Davis and Clarke 1995) para realizar uma alocação de autoestrutura, ou seja, alocar os autovalores numa determinada faixa e alocar os autovetores para o desacoplamento de modos. Tais autores utilizaram o projeto *LQR* para determinar os ganhos do controlador por realimentação de estados e um Algoritmo Genético para determinar as matrizes de ponderação que devem satisfazer as especificações de projeto. O modelo da aeronave é linearizado em uma condição de cruzeiro, (Sobel and Shapiro 1985). As matrizes A de estado, B de controle e C de saída para a representação no espaço de estado são:

$$A = \begin{bmatrix} -20.00 & 0.0000 & 0.0000 & 0.0000 & 0.000000 & 0.0000 \\ 0.0000 & -25.00 & 0.0000 & 0.0000 & 0.000000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.000000 & 0.0000 \\ -0.744 & -0.032 & 0.0000 & -1.540 & -0.00420 & 1.5400 \\ 0.3370 & -1.120 & 0.0000 & 0.2490 & -1.00000 & -5.200 \\ 0.0200 & 0.0000 & 0.0386 & -0.996 & -0.00029 & -0.117 \end{bmatrix} \quad (C.1)$$

$$B = \begin{bmatrix} 20.00 & 00.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 00.00 & 25.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}^T,$$

$$C = \begin{bmatrix} 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (\text{C.2})$$

Os autovalores do sistema teste são mostrados na Tabela C.1. O sistema é estável e tem um pólo de variação de curto tempo, três pólos de variações mais longas e um pólo oscilante.

Tabela C.1: Autovalores do sistema teste

Nº	Autovalores
1	-0.2276
2	-0.8955
3	$-0.7670 - j0.9251$
4	$-0.7670 + j0.9251$
5	-0.8955
6	-25.0

O AG proposto realizou a seleção das matrizes de ponderações, e os ganhos do controlador ótimo foram obtidos pelo método de *Schur*.

As especificações de projeto, faixas de autovalores e suas sensibilidades associadas, são mostradas na Tabela C.2. Estas restrições são obtidas de uma autoestrutura imposta por um controle ótimo, chamado o controlador base, implementado no modelo linear do sistema dinâmico.

Tabela C.2: Especificações de Projeto de Autoestrutura e Ganhos do Controlador base.

Nº.	Autovalores	S_i	Ganhos
1	$-30.00 \leq Re \leq -13.00$	7.28	$\begin{matrix} 0.00 & 0.00 & -0.11 & -3.29 & -0.45 & 2.82 \\ 0.00 & 0.00 & -0.29 & -0.96 & -0.68 & 3.89 \end{matrix}$
2	$-30.00 \leq Re \leq -20.00$	4.13	
3	$-3.00 \leq Re \leq -1.00$ $-3.000 \leq Imag \leq 3.00$	5.24	
4	$-3.00 \leq Re \leq -1.00$ $-3.00 \leq Imag \leq 3.00$	5.24	
5	$-3.00 \leq Re \leq -1.00$	9.92	
6	$-3.00 \leq Re \leq -0.20$	2.18	

DEFINIÇÃO DE UM AG CANÔNICO

Os algoritmos genéticos (AGs) foram desenvolvidos por J.H. Holland e sua equipe na Universidade de Michigan durante a década de 70. Eles tinham dois objetivos fundamentais: primeiro, explicar os processos adaptativos dos sistemas naturais; segundo, conceber sistemas artificiais de acordo com as propriedades de tais sistemas naturais. Desses trabalhos surgiu o livro "Adaption in Natural and Artificial Systems", publicado em 1975.

Pode-se afirmar, de uma maneira geral, que os algoritmos genéticos são métodos de busca aleatória polarizada e otimização, baseados na teoria evolutiva de Darwin. São mais eficientes que as técnicas de busca e otimização locais baseadas no gradiente da função objetivo do problema, e aquelas baseadas em buscas puramente aleatórias; não são limitados por hipótese restritivas acerca do espaço de busca, como continuidade, existência de derivadas, unimodalidade, etc., o que geralmente ocorre nos métodos clássicos de otimização (geralmente baseados em gradientes).

A implementação de um algoritmo genético requer que as possíveis soluções do problema sejam representadas no formato de código genético (representação genotípica), geralmente dado como *strings* de números, chamados cromossomos, e uma função de avaliação positiva, relacionada à função objetivo, deve ser construída para avaliar a qualidade (*fitness*) do cromossomo (solução). O algoritmo genético opera em uma população (conjunto de cromossomos ou soluções) de forma iterativa, e em cada iteração (geração) são aplicados operadores genéticos, tais como: seleção (indivíduos são selecionados para formar a geração seguinte com

uma probabilidade proporcional ao seu valor de fitness na população), *crossover* ou recombinação (troca aleatória do material genético entre dois cromossomos produzindo novas soluções) e mutação (alteração aleatória, com pequena probabilidade, no valor do gene do *string*). Estes novos indivíduos são avaliados por meio de uma função de adequabilidade (*fitness*) e se apresentarem uma melhor solução que a anterior são inseridos na população.

D.1 Definição geral de um AG canônico

Podemos definir formalmente um algoritmo genético canônico como um n-ú-
plo

$$AG = (\Sigma, \Omega, m, P, f, X_S, X, \Psi, \tau), \quad (D.1)$$

sendo Σ o espaço de busca (fenótipos); Ω o espaço de representação (genótipos, incluindo alfabeto e tamanho de *string*); P uma população de m indivíduos, $P = (b_1, b_2, \dots, b_m)$, b_i representa cada indivíduo, ou seja, b_i é uma possível solução para o problema, com $i = 1, \dots, m$; f representa a função objetivo (*fitness*), retornando um valor positivo e real na avaliação de cada indivíduo, $f : b_i \rightarrow \mathbb{R}^+$, $i = 1, \dots, m$; X_S o operador de seleção de r pais para o cruzamento, $X_S : P \rightarrow \{p_1, p_2, \dots, p_r\}$; X o conjunto de operadores X_C (*crossover*) e X_M (mutação) que produzem s filhos à partir de r pais, $X = \{X_C, X_M\} : \{p_1, p_2, \dots, p_r\} \rightarrow \{f_1, f_2, \dots, f_s\}$; Ψ o operador de substituição que substitui s indivíduos selecionados da geração P_t pelos s indivíduos obtidos da aplicação dos operadores X_C e X_M na nova geração P_{t+1} , $P_{t+1} = P_t - \Psi(P_t) + \{f_1, f_2, \dots, f_s\}$; τ o critério de parada (número predefinido de etapas) .

Um algoritmo genético canônico pode ser esquematizado como segue

INICIALIZAÇÃO

Escolher uma população inicial de indivíduos;

Determinar a *fitness* de cada indivíduo;

REPETE

Aplicar o operador de seleção X_S ;

Aplicar o operador *Crossover* X_C ;

Aplicar o operador Mutaç o X_M ;

Determinar a *fitness* f de cada indivíduo para a atual população;

Determinar a nova geração P_t por meio de Ψ ;

AT  que se aplique algum crit rio de parada.

D.1.1 Operadores gen ticos

Sele o Proporcional: Ao usar a sele o proporcional a popula o da gera o seguinte   determinada por m experimentos aleat rios independentes. A probabilidade de um indiv duo b_i ser selecionado da m -upla $(b_1, b_2, \dots, b_m)$ para ser um membro da gera o seguinte em cada experimento  

$$P\{b_i\} = \frac{f(b_i)}{\sum_{j=1}^m f(b_j)} > 0. \quad (\text{D.2})$$

Operador Muta o: Opera independentemente sobre cada indiv duo perturbando probabilisticamente cada *string* de bit. O evento de alterar o j - simo bit do i - simo indiv duo   estocasticamente independente e ocorre com probabilidade $\mu \in (0, 1)$. A probabilidade de que o *string* b_i mude para b'_i ap s a muta o  

$$P\{b_i \rightarrow b'_i\} = \mu^{H(b_i, b'_i)} (1 - \mu)^{l - H(b_i, b'_i)} > 0, \quad (\text{D.3})$$

sendo $H(b_i, b'_i)$ a dist ncia Hamming entre os *strings* b_i e b'_i , definida como o n mero de posi es de bits que n o coincidem zeros e uns em forma de bits em cada posi o correspondente de b_i e b'_i .

Operador *Crossover*:   aplicado com alguma probabilidade $\chi \in [0, 1]$ para construir um *string* de bit a partir de pelo menos dois outros *strings* escolhidos aleatoriamente.

D.1.2 Variantes de AGs Canônicos

A estratégia elitista refere-se ao algoritmo genético que guarda o indivíduo que possui maior valor de fitness. Ela foi primeiramente introduzida por Kenneth De Jong em 1975. Este método força os AGs a reterem um certo número de melhores indivíduos da geração atual para a geração seguinte. Tais indivíduos podem ser destruídos se eles não forem selecionados para reprodução ou se eles forem destruídos pelos operadores *crossover* e mutação. Resultados em (Rudolph 1994) e (Suzuki 1995) mostram que o elitismo pode melhorar o desempenho do AG significativamente. A principal vantagem deste método é o fato dele garantir a convergência, ou seja, caso o ótimo global seja descoberto durante o processo de busca, o algoritmo genético deve convergir para tal solução.

APÊNDICE E

MODELO DO AG-LQG $_{\Xi, \Theta}$

O modelo do AG-LQG $_{\Xi, \Theta}$ dedica-se à busca das matrizes de covariâncias Ξ da perturbação no estado e Θ do ruído de medida de forma a determinar um estimador de estado estocástico que satisfaça a restrição de alocação de autoestrutura. Este modelo representa uma modelagem da estrutura de otimização (2.3) em termos de algoritmos e estruturas de dados de computação evolutiva. Assim, o AG deve encontrar um ganho $L_{\Xi, \Theta}$ do estimador de forma a alocar os autovalores e autovetores de acordo com critérios especificados pelo projetista.

Devido à dualidade entre o LQR e o Filtro de Kalman, Apêndice A, utiliza-se para o desenvolvimento deste algoritmo, basicamente, a mesma estrutura utilizada no Capítulo 3. Também, as mesmas preocupações apresentadas durante a execução do Projeto LQR, agora serão levadas em consideração para o Projeto do Filtro de Kalman, ou seja, determinar-se-á um estimador que leve o sistema a um bom desempenho com estabilidade, de acordo com as especificações de projeto.

A Seção E.1, mostra o modelo genético artificial para as matrizes de covariâncias Ξ e Θ , utilizadas para resolver o problema de alocação de autoestrutura do Estimador de Estado. A criação deste modelo contribui para a redução da carga computacional e contém as informações necessárias para a reconstrução das matrizes de covariâncias na forma matricial de origem.

E.1 Modelo das Matrizes Ξ e Θ

Da mesma forma como feito para o LQR, o primeiro passo no desenvolvimento do AG é criar um modelo para as matrizes de covariâncias Ξ e Θ , ou seja, gerar um cromossomo que represente bem tais matrizes e que diminua a carga computacional. Inicialmente, considere a ordem das matrizes $A_{n \times n}$ e $C_{p \times n}$ do sistema dinâmico estocástico representado pelas equações (A.1)-(A.4), com $G = I$.

As dimensões das matrizes apresentadas e as especificações do Projeto do estimador nos levam a agrupar as matrizes de covariâncias $\Xi_{n \times n}$ e $\Theta_{p \times p}$. Essas matrizes devem satisfazer as restrições de positividade no problema de estimação de estado estocástica, ambas simétricas e definidas positivas.

Guardando apenas os elementos superiores e diagonais das matrizes, devido à simetria apresentada por elas, tem-se como resultado o cromossomo que irá representar as matrizes de covariâncias Ξ e Θ , respectivamente

$$\begin{aligned} \Xi\Theta_k &= \bigcup_{i,j=1}^n \xi_{i,j} \wedge \bigcup_{\iota,l=1}^p \nu_{\iota,l}, \quad i \leq j, \quad \iota \leq l \\ k &= 1, \dots, n_{indiv}, \end{aligned} \quad (E.1)$$

sendo n o número de linhas ou colunas da matriz A , p o número de linhas da matriz C . Os genes representam os elementos $\xi_{i,j}$ e $\nu_{\iota,l}$ da matriz $\Xi\Theta_k$ e n_{indiv} é o número de cromossomos;

Os próximos passos a serem executados para o desenvolvimento do AG na busca das matrizes de covariâncias Ξ da perturbação no estado e Θ do ruído de medida para o problema do Filtro de Kalman são, basicamente, os mesmos utilizados no desenvolvimento do AG, apresentado no Capítulo 3. Este algoritmo pode ser encontrado em (Ferreira 2003).

APÊNDICE F

ALGORITMOS AG-LQR_{Q,R}

F.1 Programa Principal do AG para o LQR

F.1.1 Inicializações

Sistema Dinâmico

$$\dot{x} = A_{n \times n}x + B_{n \times m}u$$

$$y = C_{p \times n}x$$

$A \leftarrow$ matriz de estado
 $B \leftarrow$ matriz de controle
 $C \leftarrow$ matriz de saída

Especificações de Projeto

Sens $\leftarrow$ Parâmetros de sensibilidade
Ydir $\leftarrow$ Limite direito dos autovalores em malha fechada
Yesq $\leftarrow$ Limite esquerdo dos autovalores em malha fechada

Parâmetros

$n \leftarrow$ dimensão do vetor de estado
 $m \leftarrow$ número de entradas do sistema
 $p \leftarrow$ número de saídas do sistema
 $n_Q \leftarrow n * (n + 1) / 2$
 $n_R \leftarrow m * (m + 1) / 2$
 $n_{indiv} \leftarrow$ tamanho da população

Indicação do Número de Indivíduos Classificados

```
cont      ←  0
contger   ←  0
```

F.1.2 População Inicial

<i>GA_semente</i>	Função que busca a semente QR gravada
<i>GA_pop_inic</i>	Função que gera uma população inicial a partir da semente
<i>GA_calc_pinic</i>	Função que executa os cálculos necessários para escolha de indivíduos

F.1.3 Ciclo de Busca

```
ndigit ← input('\n Digite o número de gerações para o LQR \n')
For nger ← 1 até 1000
  If nger <= ndigit
    contger ← contger+1
    GA_reprod      % Reproduz os melhores indivíduos
    GA_crossover   % Faz o cruzamento entre os indivíduos
    GA_calc_xover  % Cálculo teste da nova população
    GA_muta        % Executa a mutação probabilística dos indivíduos
    GA_calc        % Novo cálculo teste da nova geração gerada pelos programas anteriores
    GA_LQR_melhor_si % Acumula o melhor indivíduo de cada geração
  End
End
```

F.2 Algoritmo da População Inicial

Na geração aleatória das matrizes Q e R , Subseção F.2.2, a população inicial é gerada via uma seqüência pseudo-aleatória distribuída uniformemente. A **Transformação de Q e R em Matrizes Simétricas**, Subseção F.2.3, é um procedimento projetado para garantir as especificações de projeto **LQR** e a **Construção da matriz QR** , Subseção F.2.4, é um procedimento que agrupa o modelo genético das soluções ou indivíduos QR_z .

F.2.1 Função Semente

O algoritmo proposto possui a opção de alterar a **semente** (número inteiro inicial), dos números aleatórios que são utilizados ao longo do AG. A consequência imediata é a mudança da população inicial, o que modifica todas as gerações.

```

a ← número decimal
S ← a
rand('state',S)

```

F.2.2 Geração Aleatória das Matrizes Q e R

Os parâmetros $p_{Q_{ii}\alpha}$, $p_{Q_{ii}\beta}$, $p_{Q\gamma}$, $p_{R_{ii}\alpha}$, $p_{R_{ii}\beta}$ e $p_{R\gamma}$ permitem controlar a população em uma certa região do espaço solução para garantir a restrição de positividade definida.

```

For z ← 1 : nindiv
  For i ← 1 : n
    For j ← 1 : n
      If i == j
        If i == 1 & j == 1
          Q(i,j) ← pQ11α + pQ11β * rand
        End
        If i == 2 & j == 2
          Q(i,j) ← pQ22α + pQ22β * rand
        End
        If i == 3 & j == 3
          Q(i,j) ← pQ33α + pQ33β * rand
        End
        If i == 4 & j == 4
          Q(i,j) ← pQ44α + pQ44β * rand
        End
        If i == 5 & j == 5
          Q(i,j) ← pQ55α + pQ55β * rand
        End
        If i == 6 & j == 6
          Q(i,j) ← pQ66α + pQ66β * rand
        End
      Else
        Q(i,j) ← pQγ * rand
      End
    End
  End
  For i ← 1 : m
    For j ← 1 : m
      If i == j
        If i == 1 & j == 1
          R(i,j) ← pR11α + pR11β * rand
        End
        If i == 2 & j == 2
          R(i,j) ← pR22α + pR22β * rand
        End
      Else
        R(i,j) ← pRγ * rand
      End
    End
  End
End

```

F.2.3 Conversão das Matrizes Q e R em Simétricas

O armazenamento das matrizes Q e R transformadas, em sua forma simétrica algébrica, é definida em uma forma triangular inferior.

```

For i ← 1 : n
  For j ← 1 : n
    If i > j
      Q(i,j) ← Q(j,i)
    End
  End
End
For i ← 1 : m
  For j ← 1 : m
    If i > j
      R(i,j) ← R(j,i)
    End
  End
End
End

```

F.2.4 Construção da Matriz QR que contém os indivíduos

A inserção dos indivíduos QR_z na população é realizada pelo modelo genético QR . A população tem n_{indiv} indivíduos e cada indivíduo tem $n_Q + n_R$ genes, sendo $n_Q = n(n+1)/2$ e $n_R = m(m+1)/2$. O pseudo-código

```

w ← 1
If w < n_Q + 1
  For i ← 1 : n
    c ← i
    For j ← c : n
      QR(z, w) ← Q(i, j)
      w ← w + 1
    End
  End
End
If w > n_Q
  For i ← 1 : m
    c ← i
    For j ← c : m
      QR(z, w) ← R(i, j)
      w ← w + 1
    End
  End
End
End
End

```

F.3 Algoritmo de Avaliação dos Cromossomos

Para cada autovalor dentro da margem especificada que satisfaz as especificações de projeto, é atribuído 1 ponto ao indivíduo QR_z . Para as sensibilidades dos autovalores, é atribuído 0.01 ponto para cada autovetor que satisfaz as especificações de sensibilidade $s_i < 1$. Os cálculos acima se repetem para todos os indivíduos da população $QR_{crom \times G}$.

Os dois primeiros procedimentos, Subseções F.3.1 e F.3.2, desta função realiza a recuperação das matrizes Q e R de um dado cromossomo QR . As matrizes recuperadas são processadas na solução do procedimento para cálculo de Parâmetros de Fitness, Subseção F.3.3.

F.3.1 Recuperação da Matriz Simétrica Q

Cada indivíduo QR_z da população é transformado na matriz simétrica Q .

```

For  $z \leftarrow 1 : n_{indiv}$ 
  If  $w < n_Q + 1$ 
    For  $i \leftarrow n$ 
       $c \leftarrow i$ 
      For  $j \leftarrow c : n$ 
         $Q(i, j) \leftarrow QR(z, w)$ 
         $w \leftarrow w + 1$ 
      End
    End
  End
  For  $i \leftarrow 1 : n$ 
    For  $j \leftarrow 1 : n$ 
      If  $i < j$ 
         $Q(j, i) \leftarrow Q(i, j)$ 
      End
    End
  End
End

```

F.3.2 Recuperação da Matriz Simétrica R

Cada indivíduo QR_z da população é transformado na matriz simétrica R .

```

    If  $w > n_Q$ 
      For  $i \leftarrow 1 : m$ 
         $c \leftarrow i$ 
        For  $j \leftarrow c : m$ 
           $R(i, j) \leftarrow QR(z, w)$ 
           $w \leftarrow w + 1$ 
        End
      End
    End
    For  $i \leftarrow 1 : m$ 
      For  $j \leftarrow 1 : m$ 
        If  $i < j$ 
           $R(j, i) \leftarrow R(i, j)$ 
        End
      End
    End
  End

```

F.3.3 Parâmetros da Função de Fitness

O cálculo dos parâmetros atuais da função de fitness para habilitar a avaliação da população é o propósito principal deste procedimento. Esta função requer a solução do LQR, espectro da malha fechada e seus autovetores,

```

 $[K, S] \leftarrow lqr2(A, B, Q, R)$ 
 $Z \leftarrow (A - B * K)$ 
 $[V, D] \leftarrow eig(Z)$ 
 $[W, D] \leftarrow eig(Z')$ 
 $W \leftarrow conj(W')$ 
  For  $j \leftarrow 1 : n$ 
    For  $i \leftarrow 1 : n$ 
       $Vi(i) \leftarrow V(i, j)$ 
    End
    For  $i \leftarrow 1 : n$ 
       $Wi(i) \leftarrow W(i, j)$ 
    End
     $Yi(z, j) \leftarrow D(j, j)$ 
     $Si(z, j) \leftarrow (norm(Vi) * norm(Wi)) / abs(dot(Vi, Wi))$ 
     $si(z, j) \leftarrow Si(z, j) / Sensqr(j)$ 
  End
End

```

F.4 Algoritmo de Operações Genéticas

As operações genéticas podem ser classificadas em dois tipos ou ações para realizar o paradigma do Algoritmo Genético. O primeiro tipo é a seleção aleatória que diz respeito a seleção dos indivíduos para reprodução, Subseção F.4.1, e

consiste da seleção aleatória de indivíduos para cruzamento. O segundo tipo são os operadores genéticos: *crossover*, Subseção F.4.2, e mutação, Subseção F.4.3.

F.4.1 Algoritmo de Seleção

Este algoritmo executa a seleção dos indivíduos que serão manipulados pelos operadores genéticos,

```

For    z ← 1 : nindiv
      roulette ← Fittotal * rand
      rol ← roulette
      Fitp ← 0
      elem ← 0
      For i ← 1 : nindiv
        Fitp ← Fitp + Fitness(i)
        If rol ≤ Fitp
          elem ← i
          rol ← rol - Fitness(i)
        End
      End
      For i ← 1 : nQ + nR
        QR1(z, i) ← QR(elem, i)
      End
End

```

F.4.2 Algoritmo de *Crossover*

Os genes q_{ij} e r_{ij} do indivíduo QR_1 são combinados com os genes do indivíduo QR_2 para gerar dois descendentes no acasalamento. O algoritmo realiza a seleção dos indivíduos a serem manipulados pelos operadores genéticos,

```

i ← 1
j ← 1
For    z ← 1 : nindiv/2
      For i ← 1 : nQ + nR
        alfa ← rand
        QR(j, i) ← alfa * QR1(j, i) + (1 - alfa) * QR1(j + 1, i)
        QR(j + 1, i) ← (1 - alfa) * QR1(j, i) + (alfa) * QR1(j + 1, i)
      End
      j ← j + 2
End

```

F.4.3 Algoritmo de Mutação

Os genes q_{ij} e r_{ij} do elemento QR_1 são transformados ou mutáveis considerando auto variações de seus genes,

```

For    z ← 1 : nindiv
      jmut ← 0
      For j ← 1 : nQ + nR
        probmut ← rand
        prob ← 100 * probmut
        If prob < probtmut
          jmut ← j
          If j == jalelo_1 | j == jalelo_2
            probmut ← rand
            genQR ← QR(z, j)
            QR(z, j) ← mut1_2_alelo_k + mut1_2_alelo_v * probmut
            genmut ← j
            gentaxa ← QR(z, j)
          End
          If j == jalelo_3
            probmut ← rand
            genQR ← QR(z, j)
            QR(z, j) ← mut3_alelo_k + mut3_alelo_v * probmut
            genmut ← j
            gentaxa ← QR(z, j)
          End
          If j < jalelo_1
            probmut ← rand
            genQR ← QR(z, j)
            If j == jalelo_4 | j == jalelo_5 | j == jalelo_6 | j == jalelo_7 |
              j == jalelo_8 | j == jalelo_9
                QR(z, j) ← mutF_1_alelo_k + mutF_1_alelo_v * probmut
                genmut ← j
                gentaxa ← QR(z, j)
              Else
                QR(z, j) ← mutA_1_alelo_k * probmut
                genmut ← j
                gentaxa ← QR(z, j)
              Else
            End
          End
        End
      End
    End
  End
End

```

F.5 Armazenamento do Melhor Indivíduo de cada Geração

A função *GA_LQR_melhor_si* armazena na variável *melhor_indiv_si_01* o melhor indivíduo de cada geração, com suas respectivas sensibilidades.

```

    cont_si_xover_B ← 0
    si_xover_chav ← 1
  For z ← 1 : ndiv
    For i ← 1 : n
      If si_xover(z,i) < 1
        cont_si_xover_B ← cont_si_xover_B + 1
      End
    End
    If cont_si_xover_B == n
      If si_xover_chav == 1
        si_xover_B ← [ z  si_xover(z,1)  si_xover(z,2)  si_xover(z,3)  si_xover(z,4)  si_xover(z,5) ...
                                                                si_xover(z,6) ]
        si_xover_chav ← 2
      Else
        si_xover_B ← [ si_xover_B; z  si_xover(z,1)  si_xover(z,2)  si_xover(z,3)  si_xover(z,4) ...
                                                                si_xover(z,5)  si_xover(z,6) ]
      End
    End
  End
End
If si_xover_B == 2
  si_xover_chav ← 3
  [ m_si, n_si ] ← size(si_xover_B)
  For z ← 1 : m_si
    For k ← 1 : n
      s_i_norm_e(k) ← abs(real(Yi_xover(z,k))/real(Yesq(k)))
      If s_i_norm_e(k) < 1
        lab_si_e(k) ← 1
      Else
        lab_si_e(k) ← 0
      End
      s_i_norm_d(k) ← abs(real(Yi_xover(z,k))/real(Ydir(k)))
      If s_i_norm_d(k) > 1
        lab_si_d(k) ← 1
      Else
        lab_si_d(k) ← 0
      End
    End
  End
  rank_y_R_esq ← sum(lab_si_e)
  rank_y_R_dir ← sum(lab_si_d)

```

```

For  $k \leftarrow 1 : n$ 
  If  $\text{imag}(Y_{esq}(k)) \sim 0$ 
     $lab\_si\_I_{esq}(k) \leftarrow \text{abs}(\text{imag}(Yi\_xover(z, k)) / \text{imag}(Y_{esq}(k)))$ 
    If  $lab\_si\_I_{esk}(k) < 1$ 
       $lab\_si\_ie(k) \leftarrow 1$ 
    Else
       $lab\_si\_ie(k) \leftarrow 0$ 
    End
  Else
    If  $\text{imag}(Yi\_xover(z, k)) \sim 0$ 
       $lab\_si\_I_{esk}(k) \leftarrow 1.5$ 
       $lab\_si\_ie(k) \leftarrow 0$ 
    Else
       $lab\_si\_I_{esk}(k) \leftarrow 1$ 
       $lab\_si\_ie(k) \leftarrow 1$ 
    End
  End
End
 $rank\_y\_I\_esq \leftarrow \text{sum}(lab\_si\_ie)$ 
 $sum\_rank\_T(z) \leftarrow rank\_y\_esq + rank\_y\_dir + rank\_y\_I\_esq$ 
End
End
If  $si\_xover\_chav == 3$ 
   $zz \leftarrow 1000$ 
   $[m\_si, n\_si] \leftarrow \text{size}(si\_xover\_B)$ 
   $sub\_chav\_01 \leftarrow 1$ 
   $si\_min\_melhor \leftarrow 20000$ 
  For  $z \leftarrow 1 : m\_si$ 
    If  $sum\_rank\_T(z) == 18$ 
      If  $si\_min\_melhor > \text{sum}(si\_xover\_B(z, 2 : 7))$ 
         $zz \leftarrow z$ 
         $si\_min\_melhor \leftarrow \text{sum}(si\_xover\_B(z, 2 : 7))$ 
      End
    End
  End
End
If  $si\_min\_melhor \sim 20000$ 
   $melhor\_indiv\_si\_A1 \leftarrow [si\_xover\_B(zz, 1) \quad si\_xover\_B(zz, 2) \quad si\_xover\_B(zz, 3) \quad si\_xover\_B(zz, 4) \quad \dots$ 
     $si\_xover\_B(zz, 5) \quad si\_xover\_B(zz, 6) \quad si\_xover\_B(zz, 7) \quad \text{sum}(si\_xover\_B(zz, 2 : 7))]$ 
  If  $contger == 0$ 
     $melhor\_indiv\_si\_01 \leftarrow [melhor\_indiv\_si\_A1]$ 
  Else
     $melhor\_indiv\_si\_01 \leftarrow [melhor\_indiv\_si\_01 ; melhor\_indiv\_si\_A1]$ 
  End
End
End
End

```

```

If   $si\_xover\_chav < 3 \mid zz == 1000$ 
   $si\_min\_melhor \leftarrow 2000$ 
  For  $z \leftarrow 1 : n_{indiv}$ 
     $si\_teto \leftarrow sum(si\_xover(z, 1 : 6))$ 
    If  $si\_min\_melhor > si\_teto$ 
       $zz \leftarrow z$ 
       $si\_min\_melhor > si\_teto$ 
    End
  End
End
For  $k \leftarrow 1 : n$ 
   $s\_i\_norm\_e(k) \leftarrow abs(real(Yi\_xover(z, k))/real(Yesq(k)))$ 
   $s\_i\_norm\_d(k) \leftarrow abs(real(Yi\_xover(z, k))/real(Ydir(k)))$ 
End

```

ALGORITMOS AG-LQG/LTR

G.1 Programa Principal do AG para o LQG/LTR

G.1.1 Parâmetros

n_{indiv} $\leftarrow$ tamanho da população
 n_{ger} $\leftarrow$ número de gerações

G.1.2 População Inicial

Pop_inicial Função que busca a semente e gera uma população inicial a partir da semente
Fitnessfun Função que executa os cálculos do fitness da população inicial

G.1.3 Ciclo de Busca

```
For il  $\leftarrow$  1 :  $n_{ger}$ ,
    Seleção      % Reproduz os melhores indivíduos
    Crossover    % Faz o cruzamento entre os indivíduos
    Mutação      % Executa a mutação probabilística dos indivíduos
    Fitnessfun    % Novo cálculo do fitness da nova geração gerada pelos programas anteriores
End
```

G.2 Algoritmo da População Inicial

O algoritmo proposto possui a opção de alterar a semente (número inteiro inicial), dos números aleatórios que são utilizados ao longo do AG. A consequência imediata é a mudança da população inicial, o que modifica todas as gerações.

G.2.1 Geração da Semente

```
S ← 8
rand('state',S)
```

G.2.2 Geração da População Inicial

```
For z ← 1 :  $n_{indiv}$ 
    q1(i) ← rand
End
pop_inicial ← q1
```

G.3 Cálculo da função de fitness

```
For z ← 1 :  $n_{indiv}$ 
    q0 ← q1(z)
    LQG_M_Variav_LTR_entrada
    Fitness_par{z} ←  $sum((abs(SV\_lqr - SV\_ltr\_safonov)^2, 2)$ 
    If i1 > 0
        erro{z,i1} ←  $abs(SV\_lqr - SV\_ltr\_safonov)^2$ 
    End
End
Fitotal z ←  $zeros(2,1)$ 
For z ← 1 :  $n_{indiv}$ 
    Fitotal ← Fitotal+Fitness_par{z}
End
```

G.4 Algoritmo de Seleção

```
For z ← 1 :  $n_{indiv}$ 
    roulette ← Fitotal*rand
    rol ← roulette
    Fitp ←  $zeros(2,1)$ 
    elem ← 0
    For i ← 1 :  $n_{indiv}$ 
        Fitp ← Fitp+Fitness_par{i}
        If rol ≤ Fitp
            elem ← i
            rol ← rol - Fitp
        End
    End
    q01(z) ← q1(elem)
End
```

G.5 Algoritmo de *Crossover*

```
i ← 1
```

```

For     $z \leftarrow 1 : n_{indiv}/2$ 
       $\text{alfa\_1} \leftarrow 0.1$ 
       $\text{alfa\_2} \leftarrow 0.05$ 
       $q1(i) \leftarrow \text{alfa\_1} * q01(i) + \text{alfa\_2} * q01(i+1)$ 
       $q1(i+1) \leftarrow \text{alfa\_2} * q01(i) + \text{alfa\_1} * q01(i+1)$ 
       $i \leftarrow i + 2$ 
End

```

G.6 Algoritmo de Mutação

```

prob_mut  $\leftarrow 0.05$ 
j_mut  $\leftarrow 0$ 
For  $z \leftarrow 1 : n_{indiv}$ 
  prob  $\leftarrow rand$ 
  If  $\text{prob} < \text{prob\_mut}$ 
    j_mut  $\leftarrow j$ 
     $q1(j) \leftarrow q1(j) * rand$ 
  End
End

```

Referências Bibliográficas

- Bailey, G.V.; J.M. (1990). LQG/LTR robust control system design for low-pressure feedwater heater train Murphy. *Southeastcon '90, Proceedings, IEEE Vol.3*, 853 – 856.
- Blackmore, Perry A. and Robert R. Bitmead (1995). Duality Between the Discrete-Time Kalman and LQ Control Law. *IEE Transaction on Automatic Control*.
- Bottura, C. P. and J.V. da Fonseca Neto (1999). Parallel genetic algorithm fitness function team for eigenstructure assignment via lqr designs. *Congress on Evolutionary Computation - CEC99. Washington, DC, USA. Vol 2*, 1035–1042.
- Brito Filho, Joaquim Gomes (2006). Alocação de auto-estrutura utilizando controle robusto lqg/ltr e computação evolutiva. Master's thesis. UFMA.
- Cruz, José Jaime da (1996). Controle Robusto Multivariável. *EDUSP*.
- Davis, R. and T. Clarke (1995). Parallel Implementation of a Genetic Alghorithm. *Control Eng. Practice* **3**(1), 11–19.
- Doyle, J. C. and G. Stein (1979). Robustness with Observers. *IEEE Transactions on Automatic Control* **24**(4), 607–611.
- Ferreira, Carlos C. T. e Fonseca Neto, J. V. (2003). Projeto do Filtro de Kalman via Alocação de Autoestrutura e Algoritmo Genético. *VI Simpósio Brasileiro de Automação Inteligente*.

- Ferreira e Fonseca, Carlos C. T., J. V. Fonseca Neto and F. A. Torrico (2003). Alocação de Autoestrutura via Controle LQG/LTR e Computação Evolutiva. *VI Simpósio Brasileiro de Automação Inteligente*.
- Fonseca Neto, João Viana da (2000). Alocação Computacional Inteligente de Autoestruturas para Controle Multivariável. PhD thesis. UNICAMP.
- Fonseca Neto, João Viana da (2003). *An Inequalities Method for Multivariable System's Eigenstructure Assignment via Genetic Multi-Objective Optimization*. 2003 American Control Conference. Denver-Colorado.
- Gasem, C. Zhou; J.R. Whiteley; E.A. Misawa; K.A.M. (1995). Application of enhanced LQG/LTR for distillation control. *Control Systems Magazine, IEEE* **15**(4), 56 – 63.
- Graupe, D. (1972). Derivation of Weighting Matrices towards satisfying Eigenvalue requirements. *Int. J. Control* **16**(5), 881–888.
- Harvey, Charles A. and Gunter Stein (1978). Quadratic Weights for Asymptotic Regulator. *IEEE Transactions on Automatic Control* **23**(3), 378–387.
- Jafar, Zarei, Montazeri Allahyar, Motlagh Mohammad Reza Jahed and Javad Posh-tan (2006). Design and comparison of LQG/LTR and Hinf controllers for a VSTOL Flight control System. *IEEE* pp. 1–18.
- José Paulo F., Garcia, Bortolo Edson S., Ribeiro Jean Marcos S. and Garcia Lizete Maria C. F. (2004). Active Noise attenuation Using LQG/LTR Control. *Eletrônica de Potência* **9**(2), 23–26.
- Kawasaki, Naoya and Etsujiro Shimemura (1983). Determining Quadratic Weighing Matrices to Locate Poles in a Specified Region. *Automatica* **19**(5), 555–560.
- Keller, J.P. (2005). Interactive Control System Design. *Control Engineering and Practice* **14**, 177–184.
- Kishor, Nand, Saini R.P. and Dr.S.P.Singh (2004). LQG/LTR Controller for Speed Governing of Hydro-Turbine. *IEEE*. pp. 1125–1128.

- Kwakernaak, H. and R. Sivan (1972). *Linear Optimal Control Systems*. Wiley-Interscience. New York.
- Lahdhiri, T.; Alouani, A.T. (1993). LQG/LTR pitch attitude control of an earth-orbiting spacecraft. *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on* **vol.1**, Pages:445 – 446.
- Lee, Myoen-Song Choi; Jeong-Woo Lee; K.Y. (1996). An auxiliary LQG/LTR robust controller design for cogeneration plants Young-Moon Park. *Energy Conversion, IEEE Transactions on* **1**(2), 407 – 413.
- Lewis, Frank L. and Vassilis L. Syrmos (1995). *Optimal Control*. John Wiley and Sons, Inc.. USA.
- Liu, G.P. e Patton, R.J (1998). *Eigenstructure Assignment for Control System Design*. John Willey & Sons.
- Medanic, J., H.S. Tharp and W.R. Perkins (1988). Pole Placement by Performance Criterion Modification. *IEEE Transactions on Automatic Control* **33**(5), 469–472.
- Meystel, A. (1985). Intelligent control: Issues and perspectives. *IEEE Workshop on Intelligent Control*.
- M.Maciejowski, J. (1989). *Multivariable Feedback Design*. Addison - Wiley.
- Paschall, J.M.; D.B. Ridgely; R.N. (1994). Autopilot Design for a Tail-Controlled Missile Using LQG/LTR With Eigenstructure Reassignment Brown. *American Control Conference* **Vol.3**, 3278 – 3282.
- Passino, Kevin M. (2004). *Biomimicry for Optimization, Control, and Automation*. first ed.. Springer.
- Rab-Alibeik, H.; Setayeshi, S. (2003). Improved temperature control of a PWR nuclear reactor using an LQG/LTR based controller. *Nuclear Science, IEEE Transactions on* **50**(1), 211 – 218.
- Rudolph, Günter (1994). Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks, and Cybernetics*.

- Sobel, K. M. and E. Y. Shapiro (1985). Eigenstructure Assignment: A Tutorial Part I Theory. In: *Proceedings of American Control Conference 5th* . Vol. 1. Saint-Nazaire, USA. pp. 456–460.
- Stein, G. (1979). Generalized Quadratic Weights for Asymptotic Regulator Properties. *IEEE Transactions on Automatic Control* **24**(4), 559–566.
- Suzuki, Joe (1995). A Markov Chain Analysis on Simple Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*.
- Teixeira Ferreira, Carlos Cesar (2004). Alocação de autoestrutura Utilizando Controle Robusto LQG/LTR e Computação Evolutiva. PhD thesis. UFMA.
- Weerasooriya, S.; Phan, D.T. (1995). Discrete-time LQG/LTR design and modeling of a disk drive actuator tracking servo system. *Industrial Electronics, IEEE Transactions on* **Volume: 42**(Issue: 3), Pages:240 – 247.
- White, B.A. (1991). Assignment of Eigenstructure by use of Polynomial Matrices . *Journal of Systems and Control Engineering - Proc Instn Mech Engers*. **205**, 207–214.
- Wolff, Marlon Paulo Melo (2006). Análise de convergência de algoritmo genético dedicado a alocação de autoestrutura via controle Ótimo. Master's thesis. UFMA.
- Wu, Ying-Yu Tzou. Hsiang-Jui (1995). LQG/LTR control of an AC induction servo drive. *IEEE Transactions on* **10**, 214 – 221.