

Universidade Federal do Pará  
Instituto de Tecnologia  
Programa de Pós-Graduação em Engenharia Elétrica

---

*Controle Inteligente LQR Neuro-Genético para  
Alocação de Autoestrutura em Sistemas  
Dinâmicos Multivariáveis*

---

Ivanildo Silva Abreu

Belém - Pará  
30/10/2008

Universidade Federal do Pará  
Instituto de Tecnologia  
Programa de Pós-Graduação em Engenharia Elétrica

---

*Controle Inteligente LQR Neuro-Genético para  
Alocação de Autoestrutura em Sistemas  
Dinâmicos Multivariáveis*

---

**Ivanildo Silva Abreu**

Tese de Doutorado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará, como parte dos requisitos necessários para obtenção do grau de Doutor em Engenharia Elétrica.

**Belém - Pará  
30/10/2008**

Abreu, Ivanildo Silva

Controle Inteligente LQR Neuro-Genético para Alocação de Autoestrutura em Sistemas Dinâmicos Multivariáveis. / Ivanildo Silva Abreu.  
- Belém - Pará, 2008.

\*\*f.:il.

Tese de Doutorado em Engenharia Elétrica.  
- Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará.  
- Instituto de Tecnologia , Universidade Federal do Pará, 2008.

CDU 62-52

**Controle Inteligente LQR Neuro-Genético para  
Alocação de Autoestrutura em Sistemas  
Dinâmicos Multivariáveis**

**Ivanildo Silva Abreu**

Aprovado em 30/10/2008

**BANCA EXAMINADORA**

---

**Prof. João Viana da Fonseca Neto (UFMA)**

Doutor. em Engenharia Elétrica-UNICAMP

Orientador

---

**Prof. Orlando Fonseca Silva (UFPA)**

Doutor. em Engenharia de Produção-(UFSC)

Co-orientador

---

**Prof. Roberto Célio Limão de Oliveira (UFPA)**

Doutor. em Engenharia Elétrica (UFSC)

Examinador Interno

---

**Prof. Walter Barra Junior (UFPA)**

Doutor. em Engenharia Elétrica (UFPA)

Examinador Interno

---

**Prof. André Maurício Damasceno Ferreira (CEFET-PA)**

Doutor. em Engenharia Elétrica (UFPA)

Examinador Externo

---

**Prof. José Carlos Teles Campos (UFC)**

Doutor. em Engenharia Elétrica (USP)

Examinador Externo

"A **Deus** por me conduzir e fortalecer de forma tão presente em todos os momentos da minha vida."

# Dedicatória

Aos meus queridos e amáveis pais, Elder Abreu e Maria Sinhorinha Silva Abreu, por toda a eternidade e por tudo que me proporcionaram na vida, mostrando-me sempre o exemplo de força, justiça, bom-senso, responsabilidade e fé, indispensáveis nessa trajetória de vida.

A minha esposa, Maria Aparecida Costa, pelo apoio, incentivo, paciência, compreensão e carinho que foram fundamentais e decisivos para a concretização deste trabalho de pesquisa;

Ao meu querido irmão, César Roberto Silva Abreu, que não mais está entre nós, sinto muito a sua falta, minha eterna gratidão pelos incentivos e apoio moral nos momentos árduos da vida.

A Dona Elza Ferreira Costa, minha sogra, que desempenha um papel de mãe;

Aos meus cunhados e amigos, José de Jesus, Elza Maria, Ferreira, Elizabeth, Costa, Rogener, Roberto Carlos, José Roberto, Beth, João Carlos, Fernando, Raquel, Vinícius, Bruno, Jonas e Jonathan pelo grande incentivo e momentos agradáveis de descontração;

A minha grande família, Cesar Roberto (*in memorian*), Cida, Júlio César, Wagner, Goreth, Ivar, Júnior, Vinícius, Serginho, Gerson, Arícia, Juliana, César Roberto ..., pelos momentos em que estive ausente, mas que souberam compreender, muito obrigado.

Ao meu sogro, José Calazans Pereira Costa (*in memorian*), minha sincera admiração pelas suas virtudes e exemplo de um bom homem.

Ao Prof<sup>o</sup>. Dr. Raimundo Nazareno Cunha Alves (*in memorian*), por ter me dado esta oportunidade de expandir meus conhecimentos, minha gratidão e meu sincero respeito. Que Deus lhe proporcione o descanso eterno.

# Agradecimentos

---

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

Ao meu orientador, Prof. Dr. João Viana da Fonseca Neto, pela credibilidade em mim depositada, pela disponibilidade em compartilhar seus conhecimentos, objetividade nas orientações, amizade, paciência e incentivo que fortaleceram a minha formação intelectual. Agradeço também a ele por ter me repassado os fundamentos iniciais que gerou este trabalho.

Aos professores do DEMATI da UEMA, pelo incentivo, amizade e presteza;

Ao Prof. Orlando Fonseca Silva (UFPA) que inicialmente não mediu esforços nesta luta, o meu muito obrigado;

Ao Programa de Pós-Graduação da UFMA, por ter-me recebido no LCP, para dar continuidade a pesquisa;

À Universidade Estadual do Maranhão, pela liberação e oportunidade de realização desta pesquisa;

À CAPES, pelo apoio financeiro;

Ao Programa de pós-graduação da UFPA pela oportunidade, receptividade e atendimento;

Aos colegas do LCP (UFMA), em particular ao Fábio Nogueira da Silva que contribuiu com comentários críticos e revisão do material referente à rede neural artificial, Prof.<sup>a</sup> Patrícia Helena, João Inácio, Pedro Turibe, Lucilene, Marlon, Joaquim, Márcio Siqueira, Jorge Farid, Vitor, Hugo, Jorge Henrique, Renan, Samy, Aline Coelho, Antônio, Paula Renata, Anderson e Carolina, pela convivência e pelos saudáveis e saudosos momentos de aprendizagem, reflexão e descontração vivenciados ao longo desta árdua, porém, gratificante jornada;

Ao Prof. Carlos César Teixeira, (CEFET-MA), por esclarecimentos e discussões sobre sua dissertação de mestrado;



Aos professores que aceitaram fazer parte da banca examinadora;

Aos professores do curso de Pós-Graduação em Engenharia de Eletricidade da UFMA;

À coordenação da pós-graduação, a secretária Odila, João Victor e Alcides Martins, por paciência e colaboração;

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

## RESUMO

Nesta tese é apresentado um modelo neuro-genético, orientado a síntese de controladores no espaço de estado baseado no projeto do Regulador Linear Quadrático, para alocação de autoestrutura em sistemas dinâmicos multivariáveis. O modelo neuro-genético representa uma fusão de um algoritmo genético e uma rede neural recorrente para realizar a seleção das matrizes de ponderação e resolver a equação algébrica de *Riccati*, respectivamente. Um modelo de 6<sup>a</sup> ordem de uma aeronave, um modelo de 6<sup>a</sup> ordem de um gerador de indução duplamente alimentado de uma planta eólica e um modelo de 4<sup>a</sup> ordem de um circuito elétrico, são usados para avaliar a fusão dos paradigmas de inteligência computacional e o desempenho da metodologia do projeto de controle. O desempenho dos modelos neuro-genéticos são avaliados por momentos estatísticos de primeira e segunda ordem para o algoritmo genético, enquanto que a rede neural é avaliada por superfícies da função energia e da norma do infinito da equação algébrica de *Riccati*. São feitas comparações com o método de *Schur*.

**Palavras-Chave:** Teoria de Controle, Regulador Linear Quadrático, Equação Algébrica de *Riccati*, Otimização, Rede Neural Recorrente, Algoritmo Genético, Controle Inteligente

## ABSTRACT

In this thesis is presented a neural-genetic model, oriented to state space controllers synthesis, based on the Linear Quadratic Regulator design, for eigenstructure assignment of multivariable dynamic systems. The neural-genetic model represents a fusion of a genetic algorithm and a recurrent neural network to perform the weighting matrices selection and the algebraic *Riccati* equation solution, respectively. In order to assess the LQR design, the procedure was applied in a 6<sup>th</sup> order aircraft model, 6<sup>th</sup> order doubly fed induction generator model of a wind plant and a 4<sup>th</sup> order electric circuit model which were used to evaluate the fusion of the computational intelligence paradigms and the control design method performance. The performance of the neural-genetic models are evaluated by the first and second statistics moments for the genetic algorithm, whereas the neural network is evaluated by surfaces of the energy function and of the norm of the infinity of the algebraic equation of Riccati and the results compared to the results obtained by using Schur's Method.

**Keywords:** Control Theory, Linear Quadratic Regulator, Algebraic *Riccati* Equation, Optimization, Recurrent Neural Network, Genetic Algorithm, Computational Intelligence, Eigenstructure assignment, Multivariate Control, Intelligent Control.

# Lista de Tabelas

5.1	Restrições de projeto para o $LQR$ e sensibilidades . . . . .	104
5.2	Circuito Elétrico: Autovalores e sensibilidades. . . . .	104
5.3	Parâmetros da Matriz $Q$ - População Inicial. . . . .	107
5.4	Parâmetros da Matriz $R$ - População Inicial. . . . .	108
5.5	População inicial-LQR - estatísticas da sensibilidade normalizada. . . . .	109
5.6	População final - função objetivo e traços de matrizes. . . . .	110
5.7	População final AG - sensibilidades e função objetivo dos indivíduos viáveis . . . . .	111
5.8	População Final - Indivíduos Viáveis - Parte Real dos Autovalores. . . . .	112
5.9	População final - indivíduos viáveis - parte imaginária dos autovalores. . . . .	112
5.10	População final $LQR$ - estatística da sensibilidade normalizada. . . . .	112
5.11	População final do melhor indivíduo - autovalores e traços das matrizes de ponderação. . . . .	114
5.12	Método de <i>Bryson</i> : Autovalores e traços das matrizes de Ponderação. . . . .	116
5.13	Autovalores do circuito elétrico usando $Q_{AG}$ e $R_{AG}$ . . . . .	120
5.14	Autovalores do circuito elétrico usando $Q_{Bryson}$ e $R_{Bryson}$ . . . . .	120
E.1	Autovalores do sistema dinâmico sem controlador $DFIG$ . . . . .	186
E.2	Autovalores do Sistema Dinâmico Compensado $DFIG$ . . . . .	186
E.3	Restrições do Projeto $LQR$ para $DFIG$ . . . . .	187
E.4	Ganhos do Controlador $LQR$ . . . . .	187
E.5	Autovalores, Sensibilidades e Ganhos do Controlador- $DFIG$ . . . . .	188
E.6	<i>Fitness</i> da População Inicial . . . . .	191
E.7	Autovalores do Sistema de Malha Fechada $DFIG$ . . . . .	206
F.1	Autovalores do Sistema Dinâmico <i>Aeronave</i> . . . . .	211

F.2	Restrições do Projeto <i>LQR</i> . . . . .	212
F.3	Ganhos do Controlador LQR. . . . .	212
F.4	Autovalores, Sensibilidades e Ganhos do Controlador-Aeronave. . . . .	213
F.5	<i>Fitness</i> da População Inicial . . . . .	215
F.6	<i>Fitness</i> da População Geração 10 . . . . .	219
F.7	<i>Fitness</i> da População Final . . . . .	222

# Lista de Figuras

1.1	Paradigmas de busca das matrizes de ponderação e solução da EAR do projeto de controle ótimo LQR . . . . .	14
3.1	Sistema de realimentação de estado. . . . .	33
3.2	Seleção das matrizes de ponderação do projeto LQR. . . . .	35
3.3	Algoritmo do Método de <i>Bryson</i> . . . . .	37
3.4	Região de estabilidade em forma de cone. . . . .	38
3.5	Algoritmo do regulador com condições de estabilidade. . . . .	39
3.6	Algoritmo de controle ótimo modal. . . . .	41
3.7	Solução da EAR - paradigmas de controle ótimo do projeto LQR. . . . .	42
4.1	Modelo de fusão neuro-genético. . . . .	62
4.2	Modelo do controlador ótimo neuro-genético. . . . .	64
4.3	Estrutura algébrica de arquitetura da RNR . . . . .	65
4.4	Método genético artificial de ordem 2 para as matrizes $Q$ e $R$ . . . . .	71
4.5	Ciclo de Busca Genética. . . . .	73
4.6	Função de ativação linear simétrica . . . . .	89
4.7	Diagrama de blocos do processo dinâmico do projeto LQR . . . . .	92
5.1	Diagrama de um circuito elétrico de 4ª ordem. . . . .	102
5.2	Modelo do controlador ótimo neuro-genético. . . . .	105
5.3	Sensibilidade total de cada indivíduo da população inicial . . . . .	108
5.4	Processo de evolução-média da função objetivo e melhor função objetivo de gerações . . . . .	111
5.5	Comparação da média das sensibilidades com a sensibilidade de cada autovalor. . . . .	113
5.6	Resposta ao impulso de $x_1$ e $x_2$ - circuito elétrico de 4ª ordem. . . . .	117

5.7	Resposta ao impulso de $x_3$ e $x_4$ - circuito elétrico de 4ª ordem. . . . .	118
5.8	Resposta ao impulso - Circuito Elétrico de 4ª Ordem: Bryson e Controladores. . . . .	119
5.9	Valores Singulares: Circuito elétrico de 4ª Ordem. . . . .	121
5.10	Superfície da norma do infinito para variação dos parâmetros $\eta_v = 1000 - 10000$ e $\eta_z = 1 - 10$ . . . . .	123
5.11	Superfície da energia para variação dos parâmetros $\eta_v = 1000 - 10000$ e $\eta_z = 1 - 10$ . . . . .	124
5.12	Superfície da norma do infinito para variação dos parâmetros $\eta_v$ e $\eta_z$ .125	
5.13	Superfície da energia para os parâmetros $\eta_v = 20 - 200$ e $\eta_z = 3 - 10$ .126	
5.14	Energia em função do número de iterações. . . . .	127
5.15	Norma do infinito da solução da EAR . . . . .	127
5.16	Norma do infinito do Fator de Cholesky . . . . .	128
5.17	Norma do infinito das camadas de entrada, ocultas e saída para $\eta_v = 100$ e $\eta_z = 8$ . . . . .	130
5.18	Solução associada à norma do infinito das camadas de entrada, oculta e saída para $\eta_v = 1000$ e $\eta_z = 8$ . . . . .	131
E.1	Gerador de Indução duplamente alimentado. . . . .	181
E.2	Esquema do gerador de indução com o controlador <i>LQR</i> . . . . .	184
E.3	Sensibilidade total normalizada da população inicial. . . . .	189
E.4	Função de <i>fitness</i> total população inicial. . . . .	190
E.5	Função de <i>fitness</i> população Inicial- <i>DFIG</i> . . . . .	192
E.6	Sensibilidade total normalizada população geração 10- <i>DFIG</i> . . . . .	193
E.7	Função de <i>fitness</i> população geração 10- <i>DFIG</i> . . . . .	194
E.8	Função de <i>Fitness</i> Mapeada População Geração 10- <i>DFIG</i> . . . . .	195
E.9	Sensibilidade Total Normalizada da População Final- <i>DFIG</i> . . . . .	196
E.10	Função de <i>Fitness</i> da População Final- <i>DFIG</i> . . . . .	197
E.11	Função de <i>Fitness</i> Mapeada da População Final- <i>DFIG</i> . . . . .	198
E.12	Decomposição em Valores Singulares- <i>DFIG</i> . . . . .	199
E.13	Superfície da energia para variação dos parâmetros- $\eta_v=1450-1600$ e $\eta_z=100000-1600000$ . . . . .	200
E.14	Norma do infinito da solução da EAR para $\eta_v=1000-2500$ e $\eta_z=10000-2000000$ . . . . .	201

E.15	Energia- <i>DFIG</i> em função do número de iterações. . . . .	202
E.16	Norma do infinito da solução da EAR- <i>DFIG</i> versus número de iterações. . . . .	203
E.17	Norma do infinito da solução do Fator de Cholesky - <i>DFIG</i> . . . . .	204
E.18	Resposta ao impulso-planta eólica, método Bryson e LQR-AG . . . . .	205
F.1	Sensibilidade Total Normalizada. . . . .	214
F.2	Função de <i>Fitness</i> Mapeada - População Inicial - Sistema de Ordem 6. . . . .	216
F.3	Frequência da Função de <i>Fitness</i> - População Inicial - Sistema de ordem 6. . . . .	217
F.4	Função de <i>Fitness</i> Mapeada - População Geração 10-Sistema de Ordem 6. . . . .	220
F.5	Frequência da Função de <i>Fitness</i> - População Geração 10-Sistema de Ordem 6. . . . .	221
F.6	Função de <i>Fitness</i> Mapeada - População Final-Sistema de Ordem 6. . . . .	222
F.7	Frequência da Função de <i>Fitness</i> -População Final-Sistema de Ordem 6. . . . .	223
F.8	Curvas de resposta ao impulso para as saídas 1 e 2. . . . .	225
F.9	Curvas de resposta ao impulso do controlador base e controlador usando <i>AG</i> para as saídas 3 e 4. . . . .	226
F.10	Gráfico da magnitude do valor singular em função da frequência do diagrama de Bode para sistemas MIMO. . . . .	227
F.11	$n_{\max}$ da solução <i>P</i> para a variação $\eta_v$ : 1-600 e $\eta_z$ :500-2500 para um sistema de 6 <sup>a</sup> ordem. . . . .	231
F.12	$n_{\max}$ da solução <i>P</i> para a variação $\eta_v$ : 1-120 e $\eta_z$ :500-2500 para um sistema de 6 <sup>a</sup> ordem. . . . .	232
F.13	$n_{\max}$ da solução <i>P</i> para a variação $\eta_v$ : 1-120 e $\eta_z$ :500-2500 para um sistema de 6 <sup>a</sup> ordem. . . . .	233
F.14	Superfície da função energia para a variação $\eta_v$ : 10-110 e $\eta_z$ : 14000-32000 para aeronave. . . . .	234



# Lista de Abreviaturas e Siglas

AG	Algoritmo Genético
AE	Autoestrutura
AAE	Alocação de Autoestrutura
EAR	Equação Algébrica de <i>Riccati</i>
EMQ	Erro Médio Quadrático
$\xi$	Função de Energia da Rede Neural Recorrente
H	Hamiltoniano do Processo
IC	Inteligência Computacional
J	Índice de Desempenho Quadrático
K	Matriz de Ganho de Realimentação de Estado
LQ	<i>Linear Quadratic</i> (Linear Quadrático)
LQG	<i>Linear Quadratic Gaussian</i> (Linear Quadrático Gaussiano)
<i>LQR</i>	<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>
LF	<i>Lógica Fuzzy</i>
L, Y	<i>Camadas Ocultas da Rede Neural Recorrente e Fator de Cholesky</i>
MIMO	<i>Multiple-Input and Multiple-Output</i> (Múltiplas-Entradas e Múltiplas-Saídas)
$\eta_v, \eta_z$	Parâmetros de Aprendizagem da Rede neural Recorrente
P	Camada de Saída da Rede neural: Solução da Equação de Riccati
Q, R	Matrizes de Ponderação Associada ao Vetor de Estado e Controle
PI	Problema Irrestrito
PR	Problemas Restritos
PVI	Problema de Valor Inicial
PPNL	Problema de Programação Não Linear
RNR	Rede Neural Recorrente
RNA	Rede Neural Artificial
SISO	<i>Single-Input and Single-Output</i> (Simples-Entrada e Simples-Saída)
SVD	Singular Value Decomposition (Decomposição em Valores Singulares)
U	Camada de Entrada da Rede Neural Recorrente

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	O Projeto LQR e Inteligência Computacional . . . . .	13
1.2	Objetivos . . . . .	16
1.3	Motivação . . . . .	16
1.4	Justificativa . . . . .	17
1.5	Contribuição . . . . .	17
1.6	Organização da Tese de Doutorado . . . . .	18
1.7	Artigos e Trabalhos Publicados, Aceitos e Submetidos . . . . .	20
1.7.1	Trabalhos Publicados . . . . .	20
1.7.2	Trabalhos Aceitos . . . . .	21
1.7.3	Trabalho Submetido . . . . .	21
<b>2</b>	<b>Estado da Arte: Controle Inteligente</b>	<b>22</b>
2.1	Introdução . . . . .	23
2.2	Alocação de Autoestrutura . . . . .	24
2.3	Controle Multivariável . . . . .	25
2.4	Controle Ótimo . . . . .	27
2.5	Controle Neuro-Genético . . . . .	28
2.6	Conclusão . . . . .	30
<b>3</b>	<b>Seleções e Soluções dos Parâmetros da EAR</b>	<b>31</b>
3.1	Fundamentos do LQR Inteligente . . . . .	32
3.2	Seleções das Matrizes Q e R . . . . .	34
3.2.1	Método de <i>Bryson</i> . . . . .	36
3.2.2	Projeto do Regulador com Condições de Estabilidade . . . . .	37
3.2.3	Controle Ótimo Modal . . . . .	40

3.3	Resoluções da Equação Algébrica de Riccati . . . . .	42
3.3.1	Método de <i>Newton</i> . . . . .	43
3.3.2	Método da Função Sinal Matricial . . . . .	44
3.3.3	Método de Autovetores . . . . .	47
3.3.4	Abordagem do Método de <i>Schur</i> . . . . .	52
3.3.5	Método Simplético . . . . .	55
3.4	Conclusão . . . . .	59
<b>4</b>	<b>Metodologia Neuro - Genética para Solução do LQR</b>	<b>61</b>
4.1	Metodologia Neuro-Genética . . . . .	64
4.2	Metodologia Genética para Busca das Matrizes QR . . . . .	66
4.2.1	Definição Geral de um AG Canônico . . . . .	68
4.3	Método de Busca da Estrutura de Otimização Genética $LQR_{Q,R}$ . . . . .	69
4.4	Conceitos Básicos e Definições . . . . .	70
4.5	Modelos Genéticos Artificiais . . . . .	73
4.5.1	Modelo das matrizes $Q$ e $R$ . . . . .	75
4.5.2	Modelo da População $QR$ . . . . .	76
4.5.3	Modelo da População Inicial . . . . .	76
4.5.4	Modelo de Avaliação dos Cromossomos . . . . .	77
4.5.5	Modelos de Operações Genéticas . . . . .	78
4.6	Metodologia Neuronal para Solução EAR-LQR . . . . .	80
4.7	Caracterização da Equação Algébrica de Riccati . . . . .	85
4.8	Formulação da Solução da Equação de <i>Riccati</i> . . . . .	86
4.9	Arquitetura da Rede Neuronal Recorrente . . . . .	87
4.10	Treinamento da Rede Neural Recorrente . . . . .	90
4.11	Estabilidade e Solvabilidade . . . . .	93
4.11.1	Análise de Solvabilidade . . . . .	96
4.12	Limitações de Vantagens da Metodologia Neuro-Genética . . . . .	98
4.13	Conclusão . . . . .	99
<b>5</b>	<b>Projeto LQR Neuro - Genético: Experimentos Computacionais e Validações</b>	<b>100</b>
5.1	Projeto $LQR$ Neuro-Genético: Circuito Elétrico . . . . .	101
5.1.1	Descrição no Espaço de estados . . . . .	103

5.1.2	Análise Modal . . . . .	103
5.1.3	Análise no Domínio do Tempo . . . . .	104
5.1.4	Análise de Valores Singulares . . . . .	105
5.1.5	Sintonia de Ganho do Projeto <i>LQR</i> . . . . .	106
5.1.6	População Final - AG . . . . .	109
5.1.7	Avaliação do Controlador <i>LQR</i> . . . . .	114
5.1.8	Rede Neural Recorrente para Solução de <i>Riccati</i> . . . . .	122
5.2	Projetos LQR Neuro-Genético: <i>DFIG</i> e Aeronave . . . . .	131
5.3	Discussão do Modelo de Fusão . . . . .	132
5.4	Conclusão . . . . .	132
<b>6</b>	<b>Conclusão</b>	<b>133</b>
6.1	Propostas Futuras . . . . .	134
<b>A</b>	<b>Projeto de Sistema Linear com Critério Quadrático</b>	<b>136</b>
A.1	Formulação do Problema . . . . .	137
A.2	O Problema do Regulador de Estado . . . . .	141
<b>B</b>	<b>Formulação Clássica do <i>LQR</i></b>	<b>153</b>
B.1	A Solução Variacional do <i>LQR</i> . . . . .	156
B.2	Abordagem Clássica para EAR . . . . .	161
<b>C</b>	<b>Computação Evolutiva</b>	<b>163</b>
C.1	Definição de um AG Canônico . . . . .	164
C.2	Conceitos Básicos . . . . .	165
<b>D</b>	<b>Algoritmos AG-LQR<sub>Q,R</sub></b>	<b>167</b>
D.1	Programa Principal do AG para o LQR . . . . .	167
D.1.1	Inicializações . . . . .	167
D.1.2	População Inicial . . . . .	168
D.1.3	Ciclo de Busca . . . . .	168
D.2	Algoritmo da População Inicial . . . . .	168
D.2.1	Função Semente . . . . .	169
D.2.2	Geração Aleatória das Matrizes <i>Q</i> e <i>R</i> . . . . .	169
D.2.3	Conversão das Matrizes <i>Q</i> e <i>R</i> em Simétricas . . . . .	170

D.2.4	Construção da Matriz $QR$ que contém os indivíduos . . . . .	171
D.3	Algoritmo de Avaliação dos Cromossomos . . . . .	171
D.3.1	Recuperação da Matriz Simétrica $Q$ . . . . .	172
D.3.2	Recuperação da Matriz Simétrica $R$ . . . . .	172
D.3.3	Parâmetros da Função de <i>Fitness</i> . . . . .	173
D.4	Algoritmo de Operações Genéticas . . . . .	173
D.4.1	Algoritmo de Seleção . . . . .	174
D.4.2	Algoritmo de <i>Crossover</i> . . . . .	174
D.4.3	Algoritmo de Mutação . . . . .	175
D.5	Armazenamento do Melhor Indivíduo de cada Geração . . . . .	176
<b>E</b>	<b>Projeto LQR Neuro-Genético: <i>DFIG</i></b>	<b>179</b>
E.1	Geradores Eólicos <i>DFIG</i> . . . . .	180
E.2	Modelo Gerador-Turbina . . . . .	182
E.3	Modelo Completo . . . . .	185
E.4	Análise de Convergência- <i>DFIG</i> . . . . .	189
E.5	Processo de Busca Geração 10- <i>DFIG</i> . . . . .	192
E.6	População Final- <i>DFIG</i> . . . . .	195
E.7	Solução Schur-EAR . . . . .	205
E.8	Solução Neuronal-EAR . . . . .	206
<b>F</b>	<b>Projeto <i>LQR</i> Neuro-Genético: Aeronave</b>	<b>209</b>
F.1	Modelo da Aeronave . . . . .	210
F.2	Análise de Convergência . . . . .	213
F.2.1	População Inicial . . . . .	213
F.2.2	<i>Fitness</i> da População Inicial . . . . .	214
F.3	Processo de Busca . . . . .	217
F.4	População Final . . . . .	221
F.5	Resposta ao Impulso . . . . .	223
F.6	Decomposição em Valores Singulares- <i>SVD</i> . . . . .	226
F.7	Análise de Convergência da RNA . . . . .	227
F.8	Análise de Estabilidade da RNA . . . . .	229
F.9	Simulação Computacional da RNA . . . . .	229
F.10	Variações nos Parâmetros $\eta_z$ e $\eta_v$ . . . . .	230

---

F.11 Discussão do Modelo de Fusão . . . . .	234
F.12 Conclusão . . . . .	235
<b>Referências Bibliográficas</b>	<b>236</b>

## CAPÍTULO 1

# INTRODUÇÃO

---

Nesta tese, apresenta-se uma metodologia para resolver o problema alocação da autoestrutura via controle inteligente *LQR* neuro-genético em sistemas dinâmicos multivariáveis. Utiliza-se o projeto do regulador linear quadrático (*LQR*) e técnicas de inteligência computacional (IC) que imitam o processo de evolução natural de organismos biológicos para determinar a solução do problema.

Com o surgimento das abordagens das redes neurais artificiais (RNAs) e algoritmo genético (AG), promoveu-se uma melhoria relevante nas metodologias de controle moderno, em face a sintonia de ganho e ajuste de parâmetros que eram realizados por tentativa e erro (Johnson e Grimble ,1987). Considerando as limitações associadas a ajustes realizados por tentativa e erro, a primeira está relacionada com a imposição de alvo do controle em sistemas dinâmicos multivariáveis e a outra, está relacionada com síntese a em tempo real de controladores no espaço de estado que é baseado no projeto do Regulador Linear Quadrático.

No contexto de controle inteligente, o alvo é apresentar o desenvolvimento de um modelo neuro-genético que permite o projeto do regulador linear quadrático robusto e confiável, bem como implementações em tempo real de controladores ótimos, (Mehdi Amirijoo e Son ,2008) e (Mehdi Amirijoo e Gunnarsson ,2007).

As próximas Seções enfocam o problema do controle multivariável sob o ponto de vista de alocação de autoestrutura AAE e os métodos de inteligência computacional como parte integrante nas soluções dos problemas de controle. A Seção 1.1 introduz o problema de controle multivariável no contexto de alocação de autoestrutura (AAE) no problema do controlador *LQR*, bem como a discussão e

evolução do problema de controle proposto nesta abordagem. A Seção 1.2 descreve os objetivos desta tese. A Seção 1.3 exibe a elaboração desta pesquisa por meio de técnicas de Inteligência Computacional (IC). A Seção 1.4 justifica o uso desses algoritmos evolucionários que utilizam modelos computacionais dos processos naturais de evolução como uma ferramenta para resolução de problemas, a Seção 1.5 focaliza a contribuição de maior importância desta metodologia, que é a viabilização de integração da técnica de computação evolutiva AG e as técnicas de computação conexionista RNA. Na Seção 1.6 apresenta-se uma descrição breve do conteúdo de cada capítulo e apêndices. Finalmente na Seção 1.7 apresenta-se os artigos publicados em congressos e revistas, aceitos e submetidos.

## 1.1 O Projeto LQR e Inteligência Computacional

O fluxograma da Fig. 1.1 ilustra a evolução como também a fundamentação da metodologia proposta. Primeiramente, inicia-se uma abordagem sobre o problema de controle ótimo via projeto do Regulador Linear Quadrático (*LQR*), sendo que a seleção das matrizes de ponderação é o foco principal. Vários procedimentos podem ser utilizados para esta seleção. Entre eles, destacam-se os métodos heurísticos ou empíricos, (Lewis e Syrmos, 1995), métodos determinísticos (Johnson e Grimble, 1987) ou métodos de computação evolutiva. Para a solução da equação algébrica de *Riccati*, (EAR) existem diversos métodos clássicos para sua resolução. Entre eles destacam-se os métodos de *Newton*, função sinal matricial, autosistemas e simpléticos. Entretanto, nosso enfoque está direcionado para as técnicas de inteligência computacional, dando destaque para os Algoritmos Genéticos (AGs), responsável pela busca das matrizes de ponderação e as Redes Neurais Artificiais (RNA) para resolver a equação algébrica de *Riccati*.



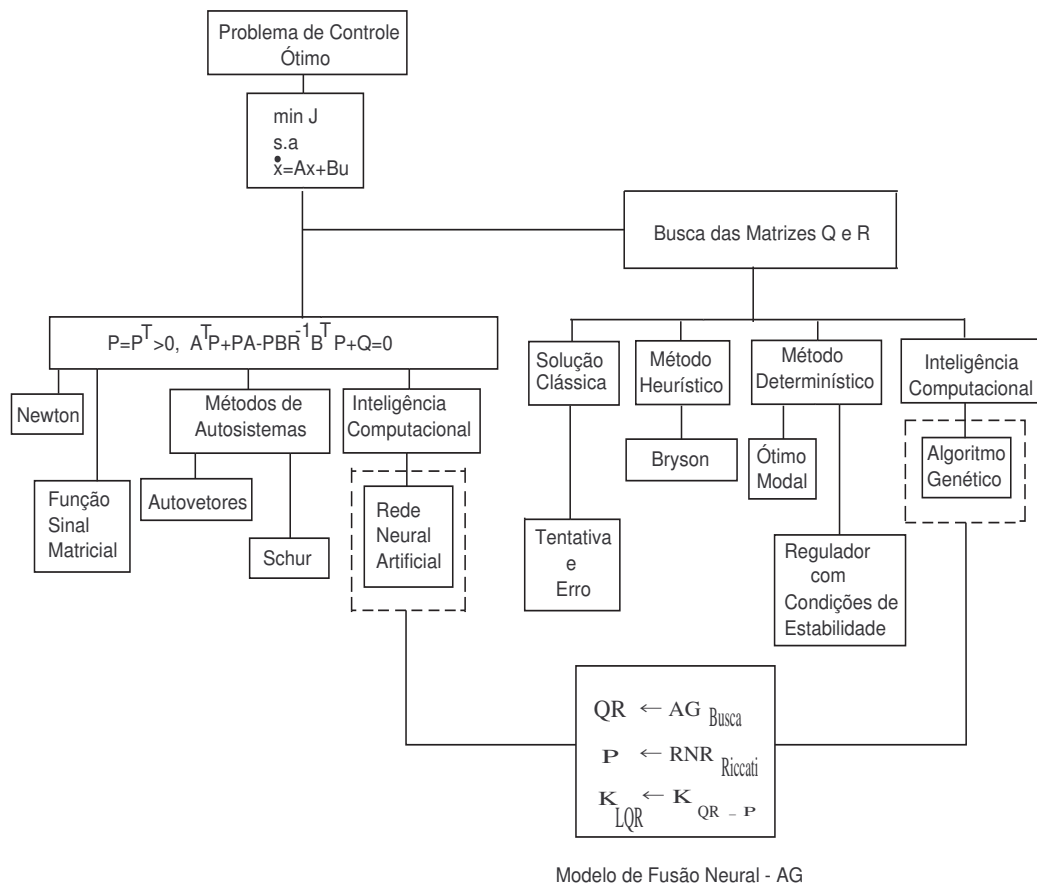


Figura 1.1: Paradigmas de busca das matrizes de ponderação e solução da EAR do projeto de controle ótimo LQR

O diagrama da Fig. 1.1, possui duas ramificações que estão diretamente envolvidas com a síntese de projeto do controle  $LQR$ . Uma está relacionada com a seleção das matrizes de ponderação enquanto que a outra está associada com a solução da equação algébrica de *Riccati*. Estas vertentes têm por base os dois paradigmas de inteligência computacional (Pedrycz, 1998), para resolver os seguintes problemas: seleção das matrizes  $QR$  (Fonseca Neto, 2003), e a equação algébrica de *Riccati* (EAR). A razão principal para investigar as soluções neuronais da EAR não é apenas devido a alta habilidade de capacidade em recuperação de falhas, mas também superar problemas numéricos tais como a inversão de matrizes na

solução da EAR. Esta característica adiciona uma implementação potencial para síntese de controladores  $LQR$  para aplicações em tempo real.

No diagrama da Fig. 1.1 representa-se, a evolução dos métodos sustentados para enfatizar o projeto  $LQR$ . Deste modo, os fundamentos dos modelos propostos estão baseados em uma fusão dos métodos de inteligência computacional (IC). O foco desta tese, estão nos métodos de IC, destacando-se o AG para realizar a seleção das matrizes de ponderação e a rede neural recorrente (RNR) para resolver a equação algébrica de *Riccati*, (Wuang e Wu ,1998), (Abreu ,2006).

Os métodos de seleção das matrizes  $Q$  e  $R$  da Fig.1.1, são orientados para impor um especificado alvo de controle. Uma classificação desses métodos é baseada em suas características de busca, tais como: heurística, empírica (Johnson e Grimble ,1987) e inteligência computacional. As soluções da EAR apresentadas na Fig. 1.1 estão fundamentadas no método de *Newton*, função sinal matricial, auto-sistemas, métodos simpléticos (Petkov e Konstantinov ,1991), bem como redes neurais artificiais (Wuang e Wu ,1998).

A habilidade que os métodos de inteligência computacional possuem para realizar a busca da solução ótima no espaço de soluções com maior probabilidade de encontrar um ótimo global em relação aos métodos tradicionais, faz com que esta abordagem se torne mais atrativa na resolução dos problemas de controle, em particular os AGs, (Neto ,2000). O fato de não exigir necessariamente informações sobre a derivada, estimativa inicial formal da região de busca e a natureza estocástica do mecanismo de busca, justificam a aplicação destes métodos quando certos aspectos do problema são desconhecidos ou de difícil avaliação pelo projetista. Nesta tese, as técnicas de computação evolutiva são utilizadas no projeto  $LQR$ , em particular naquela que requerem a minimização de parâmetros afim de se obter um desempenho desejado, (Ferreira ,2003).

O outro paradigma de inteligência computacional que merece destaque, são as Redes Neurais Recorrentes (RNR). Este paradigma é um modelo computacional não linear, inspirado na estrutura e operação do cérebro humano, que procura reproduzir características humanas, tais como: aprendizado, associação, generalização e abstração. As redes neurais são efetivas no aprendizado de padrões a partir de dados não lineares, com ruído ou compostos de exemplos contraditórios. Neste trabalho, utiliza-se redes neurais para a resolução da Equação Algébrica de

*Riccati*, (Wuang e Wu ,1998) e (Abreu ,2006).

## 1.2 Objetivos

Especificamente, o objetivo deste trabalho é apresentar modelos de AG, RNA e análises de convergências direcionadas a seleção das matrizes de ponderação e solução da equação algébrica de *Riccati*. Pesquisas sobre convergência de  $AG - LQR$  fundamentada em estatística de *fitness*, encontra-se na referência (Wolff ,2006). O AG é formado por modelos para sintonia da malha de controle do projeto  $LQR$ , enquanto que a rede neural recorrente está associada a resolução da EAR. Em resumo, podemos dizer que esta tese tem como foco, desenvolver modelos baseados em paradigmas de Inteligência Computacional para o projeto  $LQR$  e modelos de Algoritmo Genético e Rede Neural Artificial.

## 1.3 Motivação

A motivação para o desenvolvimento desta pesquisa está alicerçada nos paradigmas de inteligência computacional (Pedrycz ,1998), para solucionar o problema de alocação de autoestruturas de sistemas dinâmicos via projeto do regulador linear quadrático, tais como algoritmos genéticos e redes neuronais recorrentes. Em outras palavras, a abordagem de controle inteligente, proposta nesta tese, consiste em uma metodologia para a sintonia dos ganhos do controlador que está fundamentada em técnicas de IC. Os algoritmos genéticos podem ser utilizados para se obter, de forma automatizada, as matrizes  $Q$  de ponderação do vetor de estado, e  $R$  de ponderação do sinal de controle, superando dessa forma os métodos por tentativa e erro. As redes neuronais, por sua vez, são utilizadas para resolver a equação algébrica de *Riccati* por meio de uma solução simétrica e positiva definida. Quanto à questão do controle, a alocação de autoestrutura via projeto  $LQR$  é viável para sistemas MIMO estáveis com boas características de desempenho, visto que permitem especificações dos autovalores e autovetores, os quais influenciam nas respostas de sistemas de malha fechada, e o controlador  $LQR$  garante robustez no desempenho e estabilidade, (Liu e Patton ,1998).

## 1.4 Justificativa

O porque de utilizar o AG: A seleção das matrizes de ponderação constitui uma dificuldade para determinação da lei de controle, (Doyle e Stein ,1979), (Medanic e Perkins ,1988), (Kawasaki e Shimemura ,1983), (Graupe ,1972), (Harvey e Stein ,1978), que satisfaz às restrições de projeto, tais como, alocação de autoestrutura. Os trabalhos de pesquisas desenvolvidos por (Neto ,2000) e (Ferreira e Neto ,2003) apresentam uma alternativa para superar esta dificuldade, que são justamente os modelos de busca baseados em técnicas de computação evolutiva, dentre as quais, a baseada em algoritmo genético (AG).

Dessa maneira, os resultados dos métodos evolutivos, aparecem com forte tendência nos últimos anos, principalmente devido ao avanço computacional e ao fato dos problemas tornarem-se cada vez mais complexos. Dá-se destaque a denominada otimização natural, caracterizada pela forte semelhança dada entre os procedimentos das rotinas computacionais e o comportamento de animais, teoria da evolução e cérebro humano. Os algoritmos genéticos são compreendidos como métodos de otimização natural, determinísticos e aprendizagem de máquina, manipulando espaços de soluções potenciais, utilizando mecanismos inspirados na teoria da seleção natural de *Charles Darwin* e na genética de *G. Mendel*, (Goldberg ,1985).

O porque de utilizar a RNR: O problema de alocação de autoestrutura em sistemas dinâmicos multivariáveis, consiste em resolver a solução da equação algébrica de *Riccati*. Diversas abordagens já foram utilizadas para resolvê-la, dentre elas destaca-se o método de *Schur*. O método de Schur apresenta um grande esforço computacional que consiste em inversão de matrizes. Para contornar tal dificuldade, utiliza-se a rede neural recorrente, visto que a solução da EAR está associada a um problema de otimização irrestrito.

## 1.5 Contribuição

- A principal contribuição desta tese de doutorado, consiste em mostrar a viabilidade de integrar as abordagens de inteligência computacional, tais como algoritmos genéticos e redes neurais recorrentes para síntese de sistemas de controle ótimo do tipo *LQR*. A seleção das matrizes de ponderação

é realizada por um algoritmo genético enquanto que a solução da equação algébrica de *Riccati* é fornecida por uma rede neural recorrente.

- Apresentar a estrutura e o modelo da rede neuronal recorrente, as estratégias usadas para a conversão do problema inicial de otimização restrito em um problema de otimização irrestrito.

## 1.6 Organização da Tese de Doutorado

O restante da tese está organizada em capítulos que descrevem o controle ótimo *LQR* clássico, os modelos baseados nos paradigmas de IC, experimentos computacionais, análise de desempenho e conclusões, bem como apêndices para dar suporte ao desenvolvimento da metodologia para o trabalho de pesquisa.

O Capítulo 2 apresenta as idéias e trabalhos principais da área até o presente momento. Nele, consideram-se as metodologias utilizadas para o desenvolvimento desta pesquisa, focalizada para o problema de controle (alocação de autoestrutura), ou seja, imposição de autovalores e autovetores associados a sistemas MIMO, formulação do projeto LQR por meio de uma estrutura de otimização, modelos computacionais para realizar a busca das matrizes de ponderação, resolução da equação algébrica de *Riccati* (solução do problema de controle) e fusão de metodologias de inteligência computacional.

O Capítulo 3 destaca a apresentação do problema de controle do projeto LQR, a seleção das matrizes de ponderação de estado e controle e a equação algébrica de *Riccati* (EAR). As matrizes constituem os parâmetros do índice de desempenho quadrático  $J$  da estrutura de otimização do problema de controle bem como da equação algébrica de *Riccati*. A solução da equação algébrica de *Riccati* é formulada por uma estrutura de otimização irrestrita cuja função objetivo (energia) é o erro médio quadrático. A solução deste problema irrestrito se resume em resolver a EAR, cuja solução é uma matriz simétrica e positiva definida que é obtida via rede neural recorrente.

Quando nos referimos a formulação clássica, estamos falando a respeito da dedução da lei de controle, considerando o índice de desempenho quadrático que minimiza o desvio dos estados e energia do sistema. Contudo, a solução proposta exige a seleção das matrizes de ponderação e a solução da equação de *Riccati*.

Diversas abordagens são apresentadas para realização da seleção dessas matrizes e solução da equação de *Riccati*.

Neste capítulo apresenta-se ainda a formulação do Controle Linear Quadrático sob o ponto de vista clássico (Athans e Falb, 1966), como também do ponto de vista de incorporar novas restrições ou utilizar o desenvolvimento clássico na minimização de novas funções objetivos.

Este capítulo mostra também o problema de controle ótimo, que é formulado por uma estrutura de otimização restrita. Para sua solução, utiliza-se as técnicas de cálculo variacional afim de obter suas equações canônicas, destacando-se a matriz *Hamiltoniana*. Apresenta-se também a equação algébrica de *Riccati*, pois o problema de se determinar esta lei de controle, está associado com sua resolução. Para solução da equação de *Riccati*, apresentam-se vários métodos, entre eles o método de *Newton*, função sinal matricial, métodos de auto-sistemas (autovetores e *Schur*) e método simplético. Destacam-se os métodos de busca das matrizes de ponderação deste projeto: método de *Bryson*, projeto do regulador com condições de estabilidade e controle ótimo modal.

No Capítulo 4 descreve-se a solução genética para realizar a busca das matrizes de ponderação  $Q$  e  $R$  com o objetivo de impor à autoestrutura (autovalores e autovetores) do sistema dinâmico, por meio de um problema de otimização restrito cuja função objetivo, a sensibilidade normalizada, é não linear. Apresenta-se alguns conceitos básicos e os modelos genéticos de: matrizes  $Q$  e  $R$ , população inicial, avaliação cromossômica e dos operadores genéticos (seleção, *crossover* e mutação).

Discute-se ainda neste capítulo a solução neuronal da equação algébrica de *Riccati*. Apresenta-se a sua caracterização por meio de um problema de otimização irrestrito, cuja função objetivo é a função energia que é dada pelo erro médio quadrático (EMQ), para avaliar o desempenho da rede neural que também é não linear. A arquitetura da rede neural é representada por equações dinâmicas não-lineares e resultados analíticos que dão sustentação matemática à abordagem: análise de estabilidade utilizando-se o método direto de *Lyapunov* e análise de solvabilidade que indica a transição de estado de ativação que forma um fluxo de gradiente que minimiza a função energia, Erro Médio Quadrático (EMQ).

O Capítulo 5 discute um exemplo ilustrativo para demonstrar o desempenho da abordagem proposta, ou seja, o projeto de controlador para o experimento computacional, objetivando o desenvolvimento das habilidades nas realizações do projeto de sistema de controle ótimo. O exemplo de projeto tomado é de um circuito elétrico de quarta ordem. Apresentamos os modelos de um gerador de indução duplamente alimentado para planta eólica (DFIG) e de uma aeronave, ambas de sexta ordem nos Apêndices E e F, respectivamente, para validação do modelo de fusão neuro-genético.

O Capítulo 6 apresenta as conclusões, comentários e propostas futuras sobre a metodologia. Por fim, os Apêndices são direcionados para complementar a fundamentação da abordagem considerada.

## 1.7 Artigos e Trabalhos Publicados, Aceitos e Submetidos

### 1.7.1 Trabalhos Publicados

1. Solução da Equação Algébrica de *Riccati* via Rede Neural Recorrente para Projeto do Regulador Linear Quadrático (Congresso Brasileiro de Automática - CBA, Salvador-Ba, Brasil, 2006) Autores: Ivanildo Silva Abreu, João Viana da Fonseca Neto, Fábio Nogueira Silva, Orlando Fonseca Silva e Allan Kardec Barros, Qualis IC, Vol. 6, N<sup>o</sup>1, March 2008.
2. Modelos e Convergência de um Algoritmo Genético para Alocação de Autoestrutura via *LQR* (Revista IEEE Latina Americana, 2007). Autores: João Viana da Fonseca Neto, Ivanildo Silva Abreu, Patricia Helena Morais Rego, Marlon Paulo de Mello Wolff e Orlando Fonseca Silva.
3. Controle *LQR* Inteligente para Geradores Eólicos *DFIG* (Simpósio Brasileiro de Automação Inteligente-SBAI, Florianópolis-SC, Brasil, 2007) Autores: Ivanildo Silva Abreu, João Viana da Fonseca Neto, Fábio Nogueira Silva, José Carlos Teles e Vandilberto Pereira Pinto.

### 1.7.2 Trabalhos Aceitos

- Neural-Genetic State Space Controllers Synthesis Based on Linear Quadratic Regulator Design for Eigenstructure Assignment (Transactions on Systems, Man, and Cybernetics, Part B), Qualis IA, Autores: João Viana da Fonseca Neto, Ivanildo Silva Abreu e Fábio Nogueira Silva.

### 1.7.3 Trabalho Submetido

- Hierarchical Model of Genetic Algorithms for LQR Loop Transfer Recovery in LQG Controllers (Asian Journal of Control - Submetido) Autores: João Viana da Fonseca Neto, Ivanildo Silva Abreu, Joaquim Gomes Filho, Patricia Helena Moraes Rêgo e Fábio Nogueira Silva.



## CAPÍTULO 2

# ESTADO DA ARTE: CONTROLE INTELIGENTE

---

Este Capítulo tem por objetivo apresentar o estado da arte ou estado do conhecimento sobre os assuntos da tese intitulada "Controle Inteligente  $LQR$  Neuro-Genético para Alocação de Autoestrutura em Sistemas dinâmicos Multivariáveis". Neste trabalho enfatiza-se quais assuntos foram estudados, e mais ainda, cita-se alguns trabalhos correlacionados. Faz-se necessário que sejam realizados mais estudos específicos neste assunto que envolve este trabalho, uma vez que a literatura tem reportado uma quantidade insuficiente de resultados que possam formular conclusões mais aprofundadas ou embasar com maior segurança resultados inéditos. Em seguida apresenta-se na Seção 2.1 uma introdução, sobre controle inteligente.

Posteriormente é introduzida a apresentação do atual estado da arte sobre "Controle Inteligente  $LQR$  Neuro-Genético para Alocação de Autoestrutura em Sistemas Dinâmicos Multivariáveis". Ela é dividida em quatro partes de acordo com as principais áreas da pesquisa. São elas: alocação de autoestrutura, controle multivariável, controle ótimo e controle neuro-genético.

Em cada seção são apresentadas as perspectivas adotadas pelos pesquisadores para a solução do problema (por exemplo, que tipo de ferramenta podem ser utilizadas para realizar a alocação de autoestrutura) e descritos alguns trabalhos mais conhecidos e relacionados com este tema. São também apresentados o *background*, indispensáveis para o melhor entendimento que envolve conceitos da área de in-

teligência computacional, e ainda apresenta algumas idéias de trabalhos inéditos sobre o tema.

na Seção 2.2 é introduzido o problema de alocação de autoestrutura direcionado aos sistemas dinâmicos MIMO e outros trabalhos mais recentes. A Seção 2.3 apresenta o controle multivariável sob o ponto de vista da alocação de autoestrutura direcionado a esta pesquisa e também resumos de outros trabalhos. A Seção 2.4 caracteriza o regulador linear quadrático como uma abordagem de controle que é formulado como um problema de programação não linear como também outras pesquisas em evidência. A Seção 2.5 apresenta as metodologias baseada em algoritmos genéticos, redes neurais e também exhibe algumas metodologias de inteligência computacional, sendo que a tendência é fazer a fusão de novos paradigmas a fim de que os deméritos de um paradigma possam ser compensados pelos méritos do outro paradigma. A Seção 2.6 conclui o capítulo com observações e comentários.

## 2.1 Introdução

Devido ao fato da abordagem desenvolvida de controle inteligente considerar o Controle Ótimo, Algoritmos Genéticos (AGs) e Redes Neurais Artificiais (RNAs), apresenta-se o estado da arte do problema de controle (alocação de autoestrutura) que está intrinsecamente associado com a solução do problema, resolução da EAR. Especificamente, o estado da arte desta tese trata da imposição de autovalores e autovetores que é realizada por um algoritmo evolucionário que está fundamentado em modelos computacionais (AG), caracterização do controlador LQR que é feita por meio de um problema de otimização não linear restrito, uma abordagem de *Soft Computing* que imita o mecanismo do processo de informação biológica (RNR) para resolução do problema de controle (EAR) e finalmente a fusão das abordagens.

Entre as diversas abordagens de inteligência computacional, utilizam-se os Algoritmos Genéticos (AGs) como mecanismos de busca estocástica, polarizado para resolver o problema de alocação de autoestrutura do projeto *LQR* e a rede neural artificial recorrente (RNR) para resolver a equação algébrica de *Riccati* (EAR). Esta abordagem tem por base as metodologias de IC, (Pedrycz ,1998),

para resolver o problema com a mínima interferência do projetista ou operador. Parte destes desenvolvimentos estão fundamentados nas pesquisas realizadas por (Neto ,2000), que são os AGs utilizados para determinação das matrizes de ponderação do projeto  $LQR$ , e a solução da equação algébrica de *Riccati* ( $EAR$ ) via RNR em (Wuang e Wu ,1998), (Abreu ,2006) e (Abreu ,2007).

O restante do capítulo está dividido em seções.

## 2.2 Alocação de Autoestrutura

Em geral, o problema de alocação de autoestrutura (AAE), consiste em impor autovalores e autovetores associados aos sistemas dinâmicos multivariáveis (MIMO). Os autovalores são responsáveis pela estabilidade e taxa de crescimento e decaimento da resposta temporal dos sistemas, enquanto que os autovetores estão relacionados com sua forma, ou seja, o efeito da autoestrutura na resposta do sistema. Isto significa, verificar os efeitos provocados pelas interações entre a autoestrutura, as condições iniciais e as matrizes de entrada e de saída do sistema.

É de suma importância enfatizar que somente a alocação de autovalores não é suficiente para garantir a robustez de malha fechada e os requisitos de desempenho para sistemas MIMO, ou seja, apenas a alocação de autovalores (pólos) não é suficiente para definir uma estrutura única de realimentação para o sistema. Isto acontece devido à liberdade promovida pela realimentação dos estados ou saída em relação a seleção dos autovetores associados nos sistemas de múltiplas entradas e múltiplas saídas.

Para o caso de uma única entrada e única saída (SISO), os autovalores de malha fechada definem uma estrutura de realimentação única, principalmente quando estes autovalores possuem liberdade para deslocar-se livremente no interior de certa área do plano complexo, (Liu e Patton ,1998).

Com relação a modelagem do AG, isto é, as representações genético-artificiais dos elementos que formam o algoritmo, tais como, os operadores genéticos de reprodução, *crossover* e mutação, a análise de convergência propriamente dita é baseada em métricas estatísticas da função de *fitness* que nos informam a respeito do desempenho do AG em relação a busca das matrizes de ponderação, necessárias ao processo de alocação de autoestrutura (AAE).

Esta abordagem apresenta uma rede neural recorrente para resolver a equação algébrica de *Riccati* contínua no tempo. Uma vez que a rede neural recorrente é um sistema dinâmico, ou seja, formado por um conjunto de equações diferenciais não-lineares, mostra-se que o problema de síntese do *LQR* é formulado pela resolução de um sistema de equações matriciais, cuja solução é simétrica e definida positiva para *EAR*.

Considerando as duas abordagens da nossa proposta, o problema é reformulado para sintetizar o controlador *LQR*, que é resumido por uma solução simétrica e positiva definida, cuja solução é exibida por uma RNR, de modo que a arquitetura da rede neural recorrente é composta de quatro camadas, sendo que a camada de saída representa a solução da *EAR*. Conforme esta abordagem, a RNR é um sistema dinâmico não-linear de modo que a estabilidade pode ser analisada usando o método direto de *Lyapunov*. Uma condição necessária e suficiente para o gradiente convergir para uma solução simétrica e definida positiva da equação algébrica de *Riccati*, a partir de qualquer estado inicial, está relacionada ao *rank* do sistema dinâmico.

Em ((Zhaoquan Cai ,2008)), investiga-se as aplicações de uma classe generalizada de equações matriciais de *Sylvester* de segunda-ordem . Usando-se algumas transformações matriciais elementares, uma solução analítica geral da equação matricial generalizada de *Sylvester* é obtida de forma iterativa. Baseado na solução geral analítica proposta da equação matricial de *Sylvester*, apresenta-se um método paramétrico de alocação de autoestrutura de realimentação de estado em sistemas de segunda ordem. Destaca-se ainda a proposta de uma abordagem baseada na projeção de autoestrutura e alocação de autoestrutura linear para projetar uma eficiente lei de controle em (C. Duval e Gorrecc ,2006).

## 2.3 Controle Multivariável

A alocação de autovalores e autovetores, também denominada de Alocação de Autoestrutura (AAE), faz-se necessária devido à influência que os autovalores e autovetores exercem na resposta temporal dos sistemas dinâmicos multivariáveis (MIMO). Os autovalores têm um papel fundamental na estabilidade e taxa de variação da resposta temporal de tais sistemas, enquanto que os autovetores estão

associados a sua forma, (Kailath ,1980).

Pode-se destacar os resultados apresentados por (Moore ,1976) como muito importantes no estudo da problemática completa da autoestrutura; ele propõe um reconhecimento da liberdade oferecida, além das especificações de projeto, pela realimentação de estado em sistemas multivariáveis quando os autovalores são distintos; esta liberdade refere-se a conjunto de autovalores.

Pode-se dizer que a AAE foi originada nos estudos de fenômenos observados nas cordas oscilantes e condução do som pelo ar, em que observou-se a necessidade de lidar com vários modos de frequências que surgem nestes sistemas. A viabilização da compreensão dos citados fenômenos ocorreram com a aplicação de métodos baseados em técnicas lagrangeanas e hamiltonianas (White ,1991).

Segundo (Kailath ,1980), a alocação de autovalores foi a primeira aplicação do método no espaço de estado em sistemas lineares. Segundo (Kalman ,1969), *J. Bertram* foi o primeiro pesquisador a observar o problema de alocação de pólos e sua importância. Podemos destacar os trabalhos de *Kalman* como sendo de suma importância e precursora para concepção e solução do problema de alocação de autovalores, tanto com otimização como sem otimização, seguidos de (Kalman ,1960a) e (Kalman ,1960b), que levaram ao desenvolvimento dos primeiros resultados analíticos por (Rissanen ,1960), que escreveu a primeira publicação sobre esse assunto.

O trabalho de (Wonham ,1967), estabelece que os autovalores de malha fechada de qualquer sistema controlável podem ser arbitrariamente alocados por meio de realimentação de estado. A partir deste estudo, uma série de métodos foram desenvolvidos visando a alocação dos autovalores, (Porter ,1969b), (Porter ,1969a), (Fletcher ,1981a), (Fletcher ,1981b) e (Nichols e Van Dooren ,1984). (Neto ,2000) descreve de forma abrangente a evolução da AAE. (Yuehjen E. Shao e Hsu ,2007) discute o controle multivariável para o controle de processos estatísticos que podem não ser apropriados para os processos de monitoração que possuem características múltiplas.

## 2.4 Controle Ótimo

O Regulador Linear Quadrático ( $LQR$ ) é uma metodologia de controle formulada como uma aplicação particularizada de um problema de otimização restrito, que é caracterizado pela busca do vetor de controle  $u$  que minimiza o índice de desempenho quadrático  $J$ .

Este índice de desempenho constitui um mapeamento dos vetores de estado  $x$  e controle  $u$ , sendo ponderado pelos parâmetros matriciais  $QR$  e possui, como restrições, o modelo de equações no espaço de estado linear e invariante no tempo.

O problema  $LQR$  é formulado no domínio do tempo e possui importantes propriedades, tais como qualidade de estabilidade robusta em termos da margem de fase de 60 graus e margem de ganho infinito, (Doyle e Stein ,1981). A lei de controle estabelecida tem como parâmetros, as matrizes de ponderação do estado  $Q$  simétrica e semi-definida positiva e de ponderação do controle  $R$ , simétrica e definida positiva, respectivamente.

O regulador linear quadrático é considerado a fundação da metodologia  $LQG/LTR$  de controle robusto. O projeto Linear Quadrático Gaussiano ( $LQG$ ), realiza a estimação das variáveis de estado para realimentação ótima, enquanto que o projeto de Recuperação de Malha de Transferência ( $LTR$ ), realiza a recuperação das características de robustez do projeto  $LQR$ , que são perdidas devido a inserção do observador de estado. A importância do método  $LQR$  pode ser mensurada pelos impactos técnicos e científicos que são reportados em (H. Ko e Kim ,2004), (Ksouri ,2002) e (Abreu ,2007). Poucos artigos tratam do estudo de convergência dos AGs em suas aplicações, entretanto, para dar um maior suporte ao estudo e convergência, faz-se necessário realizar um estudo maior em convergência de algoritmo genético, (Rudolph ,1994a) e (Suzuki ,1995). A importância tecnológica do controlador  $LQR$  pode ser verificada em (Gasem ,1995), (Lee ,1996) e (Weerasooriya ,1995).

O problema do  $LQR$  é formulado sob o ponto de vista das matrizes de ponderação  $Q$  e  $R$ . Essas matrizes constituem os parâmetros da  $EAR$ , cuja solução fornece a lei de controle que minimiza o referido índice de desempenho. Os elementos dessas matrizes são ajustados para sintonizar o ganho de controle ótimo  $K$  do controlador, conduzindo a desejada alocação de autoestrutura.

Do ponto de vista prático, porém, supor que todas as componentes do estado são conhecidas, isto é, podem ser mensuradas, não é uma hipótese realista, pois muitas vezes, isto pode significar altos custos associados a sensores; além disso, é possível que alguma componente do estado não tenha significado físico e, portanto, não pode ser medida. Estuda-se o problema de existência de controle ótimo para custo funcional tendo essencialmente a forma quadrática.

## 2.5 Controle Neuro-Genético

Poucos artigos tratam do estudo de convergência dos AGs,entretanto, para tornar realmente mais eficiente um algoritmo de computação evolutiva é necessário se determinar volumosos resultados de convergência, (Suzuki ,1995), (Rudolph ,1994b), (Wolff ,2006) e outros. A importância tecnológica da metodologia de controle  $LQR$  pode ser verificada em (Gasem ,1995), (Lee ,1996), (Weerasooriya ,1995) e outros.

As técnicas clássicas de otimização possuem ampla divulgação, sendo elas confiáveis e de ampla aplicação nos campos da engenharia e outras ciências. Entretanto, essas técnicas podem apresentar dificuldades numéricas e problemas de robustez relacionados com o fato da função a ser otimizada ou as restrições do problema não serem contínuas. Pode-se ter ainda funções não convexas, multimodalidade, existência de ruídos nas funções, necessidade de se trabalhar com valores discretos para as variáveis, existência de mínimos ou máximos locais (Rawlins ,1993).

Uma metodologia baseada em Algoritmo Genético (Ferreira ,2003), foi desenvolvida para selecionar as matrizes  $Q$  e  $R$  da lei de controle e uma rede neural recorrente artificial que é usada para resolver a equação algébrica de *Riccati*, *EAR*, (Abreu ,2006). O processo do projeto para controladores robustos baseados na metodologia  $LQR$  é executado em três estágios. Inicialmente ele realiza a seleção das matrizes de ponderação, a solução da equação algébrica de *Riccati* que demanda um grande esforço computacional e o cálculo do ganho do controlador.

Outra questão importante é a garantia da positividade e simetria da solução da equação algébrica de *Riccati*. Para garantir essa positividade será necessário fazer uma fatoração  $LU$ , mais especificamente a fatoração de *Cholesky*. Mostra-se

a representação de um sistema dinâmico no espaços de estados, para o problema do funcional quadrático do valor de saída do sistema de modo que, o índice de desempenho é minimizado. Apresenta-se ainda o índice de desempenho a ser minimizado e a estrutura da rede neuronal recorrente para a representação do sistema dinâmico.

A principal razão para investigar as soluções neuronais da Equação Algébrica de Riccati, *EAR*, está em sua alta capacidade de generalização e de aprendizagem, além de apresentar características intrínsecas de tolerância a falhas, também contorna o problema de inversão de matriz. Esta característica adiciona um potencial em implementação para síntese de controladores *LQR* em aplicações de tempo real.

De modo geral, a robustez do algoritmo genético (AG) tem contribuído para soluções mais precisas no alvo de controle e as redes neurais artificiais (RNA) têm contribuído para identificação de sistemas, filtragem e estruturas de controle. A maior parte das pesquisas relacionadas a abordagens neuro-genética, (Valdivieso ,2002), (Ilke Turkmena e Karabogac ,2006) e (Pedrycz ,2007), estão focalizadas no AG associado com o algoritmo de treinamento da RNA. Neste cenário da abordagem neuro-genético, a RNA é treinada pelo AG, (M. Nasser e Abedini ,2008), (Fraile-Ardanuy e Zufiria ,2007) e (V. R. Adineh e Jelvani ,2008).

Levando-se em conta nossa proposta, que é realizar a alocação de autoestrutura utilizando computação evolutiva, o AG é encarregado do problema de controle enquanto que a RNA é responsável pela solução da metodologia do controle *LQR*. Conseqüentemente estes dois argumentos são estabelecidos pela abordagem neuro-genética. O primeiro argumento está preocupado com as ações do AG e RNA que são realizadas por modelos desacoplados. O AG fornece um parâmetro da RNA. No segundo argumento, as ações do AG e da RNA são realizadas por modelos acoplados. O AG é parte do algoritmo de treinamento da RNA, (Ping Zhang e Kumar ,2004), (Preeti Bajaj ,2007) e (Jih-Gau Juanga e Chiena ,2008). A principal característica de cada argumento é que o AG fornece suporte às atividades de seleção dos parâmetros do alvo de controle (primeiro argumento) e também, apóia o algoritmo de treinamento da RNA (segundo argumento).

Uma fusão das metodologias de inteligência computacional é aplicada no controle de modo deslizante por (Kaynak ,2001). Fusões de uma rede neural artificial



(RNA), lógica *fuzzy* (LF) e algoritmos genéticos em aplicações industriais são relatados em (Jain e Martin ,1999). Em (Feng e Lozano ,1999), o controle inteligente é abordado como uma metodologia de projeto de sistema de controle. (Sarangapani ,2006) apresenta um controle de sistemas não-lineares de tempo discreto para sistemas inteligentes baseados em redes neurais

No *Handbook* de controle inteligente neural (White e Sofge ,1992) apresenta uma abordagem de conjuntos *fuzzy* e adaptativa de controle ótimo como uma plataforma de controle inteligente. Uma outra abordagem de inteligência computacional em controle é apresentado em (King ,1999).

(Wu Yongjian ,2007) descreve a aplicação de um sistema de controle inteligente híbrido para apresentar em tempo real o controle rotativo de produção de alumina. A estratégia de controle inteligente híbrido é explicada com ênfase especial no controle *fuzzy* adaptativo e controle especialista baseado em regras.

## 2.6 Conclusão

Neste capítulo foram apresentados alguns dos trabalhos que utilizam a fusão de um sistema híbrido que forma esta tese, principalmente na aplicação da abordagem neuro-genética, com ênfase nas pesquisas destacadas. Nesta pesquisa de controle inteligente, que utiliza um modelo de fusão neuro-genético, o alvo consiste em resolver o problema de controle utilizando o projeto *LQR*. O problema relacionado com a Equação Algébrica de *Riccati*, a escolha das matrizes de ponderação e suas relações com os métodos de seleção ótima, são discutidos nesta abordagem. Pode-se afirmar que é bastante significativo utilizar metodologias que estão fundamentados na fusão, em particular redes neurais e algoritmos genéticos.

# SELEÇÕES E SOLUÇÕES DOS PARÂMETROS DA EAR

---

A apresentação do projeto *LQR* sob o ponto de vista de sua formulação, juntamente com o problema da seleção das matrizes de ponderação e a solução da EAR formam os fundamentos do problema de controle via projeto LQR para alocação de autoestrututra desta tese. Justificamos a necessidade de cada desses componentes no contexto.

A formulação básica apresentada em (Athans e Falb ,2007), descreve a origem da lei de controle ótimo, sendo que a partir deste ponto, enfocam-se os dois problemas básicos que são resolvidos por computação evolutiva: seleção das matrizes de ponderação via AG e a solução da equação algébrica de *Riccati* via Redes Neurais Recorrentes (RNR).

Neste Capítulo apresenta-se a formulação do Controle Linear Quadrático sob o ponto de vista clássico, (Athans e Falb ,1966), como também do ponto de vista de incorporar novas restrições ou utilizar o desenvolvimento clássico na minimização de novas funções objetivos.

Apresenta-se na Seção 3.1 os fundamentos de controle ótimo para a construção do problema de AAE. Na Seção 3.2 apresenta-se os métodos heurístico e determinístico para escolha das matrizes de ponderação do projeto *LQR*. A Seção 3.3 exhibe os métodos numéricos usados na solução da EAR, sendo que alguns deles exploram a estrutura da matriz Hamiltoniana. Dentre esses métodos destaca-se o método de *Schur*. A Seção 3.4 conclui o capítulo com comentários e observações.

### 3.1 Fundamentos do LQR Inteligente

A teoria de controle moderno promoveu importantes avanços no desenvolvimento de métodos para projetos e análise de controladores. O projeto de controladores para sistemas dinâmicos multivariáveis que leva em consideração a minimização de um índice de desempenho é chamado de controle ótimo. A construção deste Capítulo é complementada pelo Apêndice B.

O problema de controle ótimo, consiste em determinar uma lei de controle que minimize um índice de desempenho quadrático, tendo como restrição o sistema dinâmico representado por  $\dot{x}$ , (sistemas de equações lineares diferenciais e invariantes no tempo) e  $[t_o, T]$  é o intervalo de tempo de interesse. Por questão de simplicidade o termo funcional  $t$  será suprimido, (Athans e Falb ,2007). Este problema pode ser formulado em uma estrutura de otimização restrita dada por

$$\min_u \frac{1}{2} \int_{t_o}^T [x^T Q x + u^T R u] dt \quad (3.1)$$

s.a

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

O projeto do regulador linear quadrático (*LQR*) é feito em três etapas:

1. A busca das matrizes de ponderação;
2. Solução da Equação Algébrica de *Riccati*;
3. Cálculo do ganho do controlador ótimo.

A lei de controle linear dada por  $u = -Kx$  é a lei de controle ótimo. Em consequência, se os elementos da matriz  $K$ , ganho de realimentação, forem determinados de modo a minimizar o índice de desempenho, então  $u = -Kx$  é ótimo para qualquer que seja o estado inicial  $x(0)$ . O problema de projetar um sistema de controle de realimentação linear que minimiza um índice de desempenho quadrático pode ser reduzido ao problema de obter uma solução simétrica e definida positiva da Equação Algébrica de *Riccati*, (Wuang e Wu ,1998),

$$A^T P + PA - PBR^{-1}B^T P + Q = 0, \quad P = P^T > 0. \quad (3.2)$$

Pode-se descrever de forma resumida o controle  $LQR$  segundo sua resolução clássica. O problema de alocação de autoestrutura via projeto  $LQR$ , consiste na determinação da matriz de ganho ótimo  $K(Q, R)$ , que impõe o sistema de malha fechada especificado por  $\dot{x} = [A - BK(Q, R)]x$ , ou na forma de autovalores e autovetores que satisfaça as restrições de autoestrutura, (Liu e Patton ,1998). A autoestrutura imposta ao sistema de malha fechada pela lei de controle, deve satisfazer as condições de projeto, que são as restrições de autovalores e autovetores, de modo que resulte na autoestrutura especificada pelo projetista. As restrições de autovalores são especificadas por uma região no semiplano esquerdo complexo , sendo esta região, limitada pelas desigualdades  $\lambda_{ei} \leq \lambda_{ci} \leq \lambda_{di}$ , para  $i = 1, \dots, n$   $\lambda_{ei}$  e  $\lambda_{di}$  são os extremos deste intervalo e são respectivamente os  $i$ -ésimos autovalores complexos conjugados representando as fronteiras especificadas à esquerda e à direita do autovalor, enquanto que o termo central designa o  $i$ -ésimo autovalor alocado que satisfaz as restrições.

O diagrama de blocos ilustra a configuração com realimentação de estado na Fig. 3.1 a seguir, sendo  $\dot{x} = (A - BK)x$  a lei de controle no sistema, de modo que esta lei de controle ótimo deve conduzir o sistema à estabilidade, isto é, todos os autovalores da matriz  $(A - BK)$  possuem parte real no semi-plano esquerdo do plano complexo- $s$  (Chen ,1999).

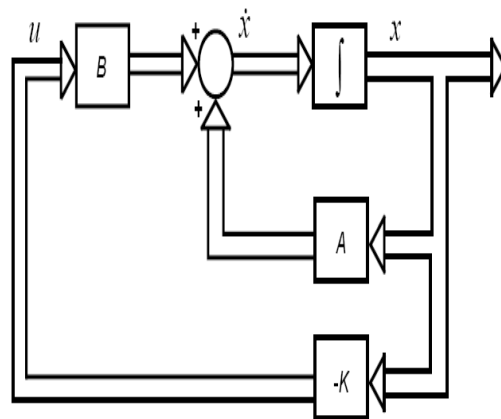


Figura 3.1: Sistema de realimentação de estado.

O problema de alocação de autoestrutura via projeto regulador linear quadrático, é contextualizado na determinação dos parâmetros matriciais de ganho ótimo  $K(Q, R) = R^{-1}B^T P$  que impõe o sistema de malha fechada, ou ainda na forma de autovalores e autovetores, autoestrutura, que satisfaça as especificações do projetista. As matrizes  $A$  e  $B$  da Eq.(3.2), são fornecidas pelo modelo linearizado da planta, as matrizes  $Q$ ,  $R$  e  $P$  são obtidos via abordagens clássicas quanto via Inteligência Computacional.

## 3.2 Seleções das Matrizes $Q$ e $R$

Um dos grandes desafios na teoria de controle usando o projeto do regulador linear quadrático, é a escolha das matrizes de ponderação  $Q$ , associada aos estados do sistema, e  $R$ , associada às variáveis de controle. A dificuldade de escolha se deve ao fato de não existir um método sistemático para tal seleção, sendo normalmente adotada a forma diagonal para essas matrizes e tendo sua escolha realizada por meio de várias simulações, tentativa e erro, de modo que os valores selecionados para as matrizes de ponderação satisfaçam os critérios estabelecidos pelo projetista, uma vez que a determinação dessas matrizes influencia na determinação do ganho do controlador.

A idéia da escolha das matrizes de ponderação é especificar o desempenho do sistema, principalmente em termos do tempo de estabilização e um nível de sobre-sinal (*overshoot*).

A Fig. 3.2 mostra a ramificação pertinente à busca das matrizes de ponderação do projeto  $LQR$  sendo algumas delas clássicas e outra baseadas em algoritmos evolucionários que utilizam metodologias computacionais para resolução de problemas.

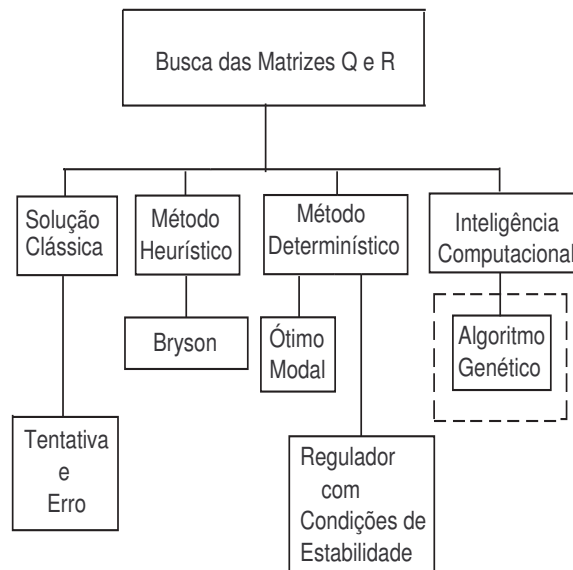


Figura 3.2: Seleção das matrizes de ponderação do projeto LQR.

Em alguns trabalhos científicos divulga-se as diversas metodologias para a escolha dos elementos das matrizes de ponderação. Dentre esses, destaca-se (Johnson e Grimble ,1987) devido a sua simplicidade de implementação e ainda por apresentar resultados satisfatórios em relação à obtenção das matrizes de ponderação.

As matrizes de ponderação do índice quadrático, são selecionadas para caracterizar o desempenho do desejado do sistema de controle em malha fechada. As abordagens disponíveis para escolha dessas matrizes são baseadas nas seguintes metodologias: métodos heurísticos, método de controle ótimo modal, método do lugar das raízes ótimo assintótico e o método da ponderação dinâmica. As abordagens heurísticas dividem-se em duas metodologias: o Método de *Bryson* e as relações entre as figuras de méritos do projeto convencional, que foi inicialmente proposto com uma maior variedade de detalhes por (Dazzo ,1984).

A primeira técnica proposta para seleção das matrizes de ponderação do custo funcional quadrático foram as heurísticas. O método do quadrado do inverso é ainda usado como um primeiro teste exploratório por diversos projetistas usando a metodologia de projeto de controle ótimo.

### 3.2.1 Método de *Bryson*

O Método de *Bryson*, também conhecido como quadrado do inverso, tem como idéia central normalizar as saídas e o termo de controle dentro da função índice de desempenho quadrático, sendo que as matrizes de ponderação  $Q$  e  $R$  são definidas por:

$$Q = \begin{bmatrix} q_1 & 0 & \dots & 0 \\ 0 & q_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & q_{ne} \end{bmatrix}, R = \begin{bmatrix} r_1 & 0 & \dots & 0 \\ 0 & r_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{na} \end{bmatrix} \quad (3.3)$$

sendo  $ne$  o número de estados e  $na$  o número de atuadores no sistema de controle. O desempenho desejado do sistema, obtido pelo ajuste das matrizes de ponderação, são escolhidos segundo (Bryson ,1969) por:

$$q_i = \frac{1}{(\Delta x_i)^2}, r_i = \frac{1}{(\Delta u_i)^2}. \quad (3.4)$$

Os valores de  $\Delta u_i$  são baseados no máximo esforço de controle ou valor máximo de operações dos atuadores, enquanto que os valores de  $\Delta x_i$  são baseados na faixa/intervalo de operações dos estados, (Kwakernaak e Sivan ,1972) e (Kwakernaak ,2000). A Fig. 3.3 ilustra o fluxograma do programa implementado para o método de *Bryson*, segundo (Johnson e Grimble ,1987), ( n.d.a), (Wolff ,2006) e ( n.d.b).

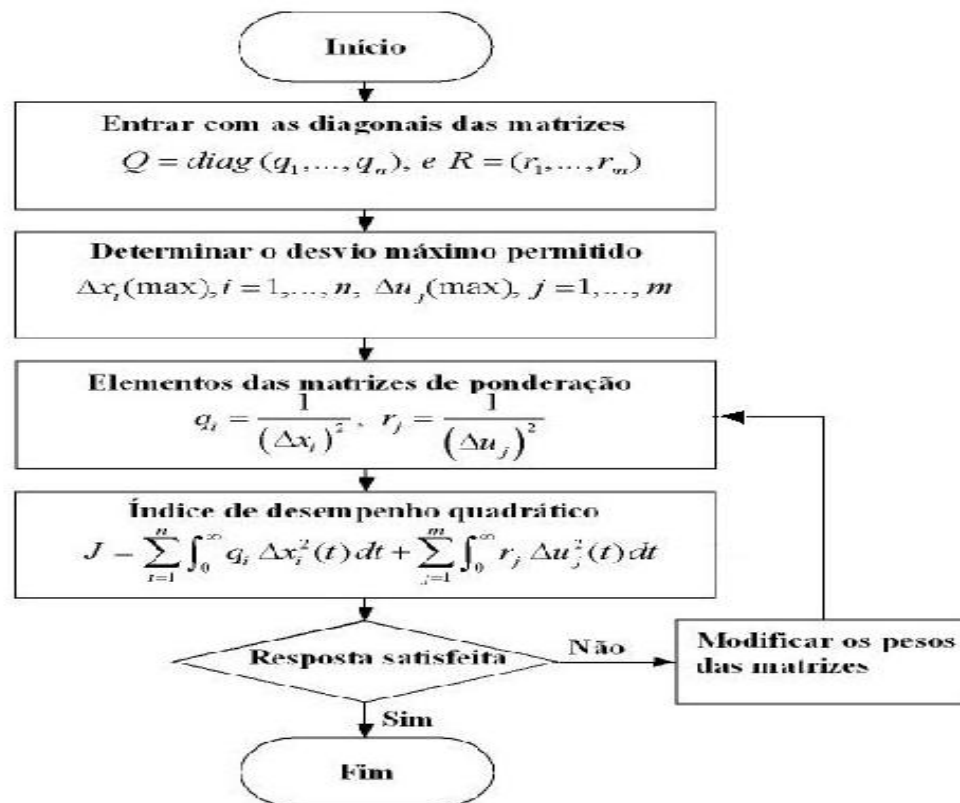


Figura 3.3: Algoritmo do Método de *Bryson*.

### 3.2.2 Projeto do Regulador com Condições de Estabilidade

Neste procedimento não é necessário indicar as posições exatas dos pólos de malha fechada do plano complexo ou plano- $s$ . É suficiente especificar uma região do semi-plano esquerdo do plano complexo ou plano- $s$  de maneira que os pólos de malha fechada devam se encontrar na região especificada. A região especificada para a posição dos pólos de malha fechada é delimitada pela região em forma de cone segundo a Fig. 3.4, (Johnson e Grimble ,1987).



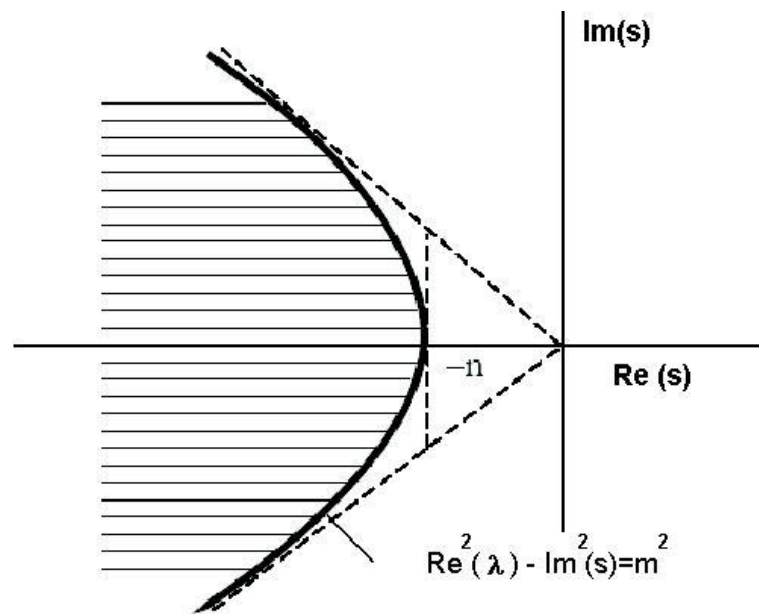


Figura 3.4: Região de estabilidade em forma de cone.

Na Fig. 3.5 é mostrado o fluxograma do programa implementado para o método de projeto do regulador com condições de estabilidade, conforme (Johnson e Grimble ,1987) e (Wolff ,2006).

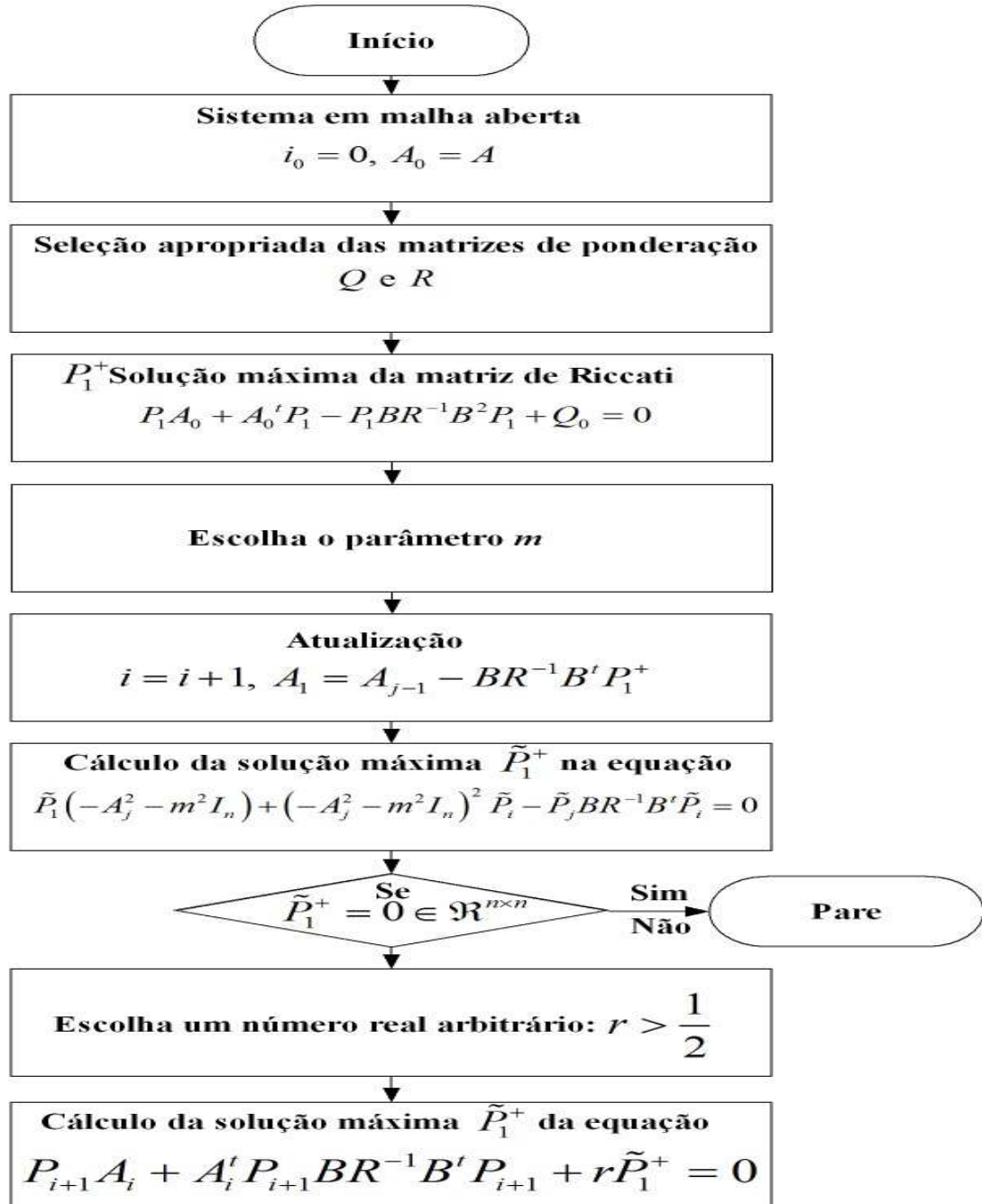


Figura 3.5: Algoritmo do regulador com condições de estabilidade.

### 3.2.3 Controle Ótimo Modal

O objetivo do controle ótimo modal é determinar uma lei de realimentação de estado  $u = -Kx$  de maneira que a matriz do sistema de malha fechada  $\dot{x} = (A - BK)x$  tenha os seus autovalores pré-estabelecidos. O controle ótimo modal é baseado na convencional alocação de pólos, que em vez de escolher o ganho de realimentação diretamente, os parâmetros de projeto da função quadrática são posicionados de acordo com as matrizes  $Q$  e  $R$  até atingir os objetivos do projeto de controle, (Johnson e Grimble ,1987) e (Wolff ,2006).

Na Fig. 3.6 é ilustrado o fluxograma do programa implementado para o algoritmo do controle ótimo modal.

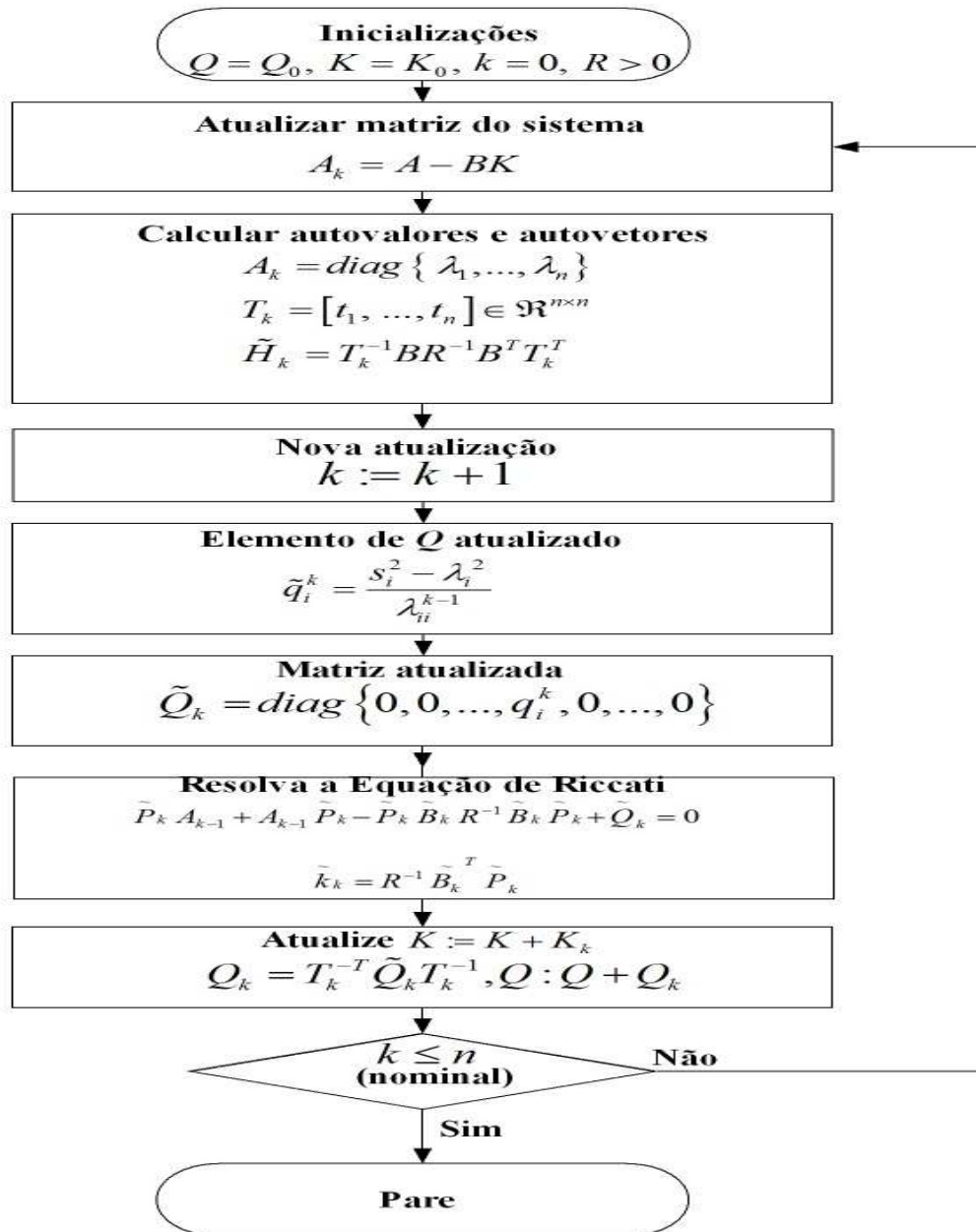


Figura 3.6: Algoritmo de controle ótimo modal.

### 3.3 Resoluções da Equação Algébrica de Riccati

Esta seção introduz de forma resumida um estudo sobre os métodos numéricos usados na solução da equação algébrica matricial de *Riccati*. A descrição dos métodos seqüenciais são baseados em (Petkov e Konstantinov ,1991) e os métodos paralelos em (Gardiner e Laub ,1991a) e (Kleinman ,1968).

O diagrama da Fig. 3.7, representa uma das ramificações das metodologias de suporte que realça o projeto *LQR*. Esta vertente é baseada em métodos determinísticos (Newton e Função Sinal Matricial), métodos de autosistemas que exploram a matriz Hamiltoniana (Métodos de Schur e Autovetores), métodos simpléticos que não exploram a estrutura da matriz Hamiltoniana e uma abordagem de inteligência computacional (rede neural artificial), cujo propósito consiste em resolver a equação de *Riccati* (Wuang e Wu ,1998) e (Abreu ,2006).

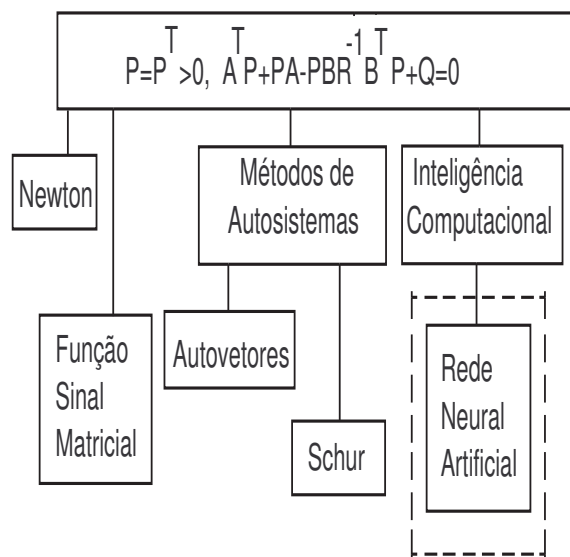


Figura 3.7: Solução da EAR - paradigmas de controle ótimo do projeto LQR.

Os métodos numéricos tomados para solução da *EAR*, são:

- Método de *Newton*;
- Método da Função Sinal;

- Abordagem de Auto-Sistemas que é formada por dois métodos para resolução da *EAR*: Métodos do Autovetor e *Schur* (baseados na matriz *Hamiltoniana*);
- Método Simplético.

A *EAR* é analisada em termo da seleção das melhores matrizes  $Q$  e  $R$ . Esta análise é tratada mais adiante por métodos que permitem reduzir as operações para essas matrizes.

A equação algébrica de Riccati, é amplamente utilizada na síntese de sistemas de controle, enquanto que as equações de *Lyapunov*, são usadas na análise de estabilidade e existência de soluções de controle para esses sistemas (Doyle e Stein ,1979). Apresenta-se a Equação Matricial Algébrica de *Riccati*, Eq.(3.5)

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (3.5)$$

cuja solução está intrinsicamente relacionada com a matriz *Hamiltoniana*  $H \in \mathbb{R}^{2n \times 2n}$ ,

$$H := \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} > 0. \quad (3.6)$$

Maiores detalhes sobre a obtenção das propriedades desta matriz podem ser obtidos em (Yu 1983), (Lancaster e Rodman ,1995), (Athans e Falb ,1966) e (Petkov e Konstantinov ,1991). Em (Ferreira ,2003), a solução desta equação em regime permanente é analisada em termo da seleção das matrizes  $Q$  e  $R$  ótimas. Esta análise é direcionada para a escolha do método, que permite reduzir a quantidade de operações de valores dessas matrizes.

Os métodos de *Schur* e *Autovetor*, são classificados como abordagem de Auto-sistemas para a solução da equação de *Riccati*. Estes métodos são baseados na estrutura da matriz *Hamiltoniana*.

### 3.3.1 Método de *Newton*

O método de *Newton* quando aplicado na solução da *EAR*, produz uma seqüência de matrizes, que sob certas condições, aproxima-se da solução  $P$  simétrica e

definida positiva da Eq.(3.5) sendo o par  $(A, B)$ :  $B^T B = S$ ,  $\text{rank}(B) = \text{rank}(S)$  estabilizável e o par  $(A, C)$ :  $(C^T C = Q)$ ,  $\text{rank}(C) = \text{rank}(Q)$  detectável.

A solução da equação de *Riccati* é única e definida positiva para o caso contínuo. O método de *Newton* baseia-se no Teorema 3.3.1 e sua prova é dada em (Kleinman ,1968) e (Wonham ,1986).

**Teorema 3.3.1.** *Seja  $P_k$ ,  $k=0,1,\dots$ , uma única solução semidefinida positiva da equação  $A_k^T P_k + P_k A_k + Q - K_k^T R K_k = 0$ , sendo recursivamente  $K_k = R^{-1} B^T P_{k-1}$ ,  $k=1,2,\dots$ ,  $A_k = A - B K_k$ ,  $k=0,1,\dots$ . O índice  $k_0$  é escolhido de modo que a matriz  $A - B K_0$  possua autovalores com parte real negativa. Portanto:*

1.  $0 \leq P \leq P_{k+1} \leq P_k \dots < P_0, k = 0, 1, \dots$

2.  $\lim_{n \rightarrow \infty} P_k = P$

**Prova** Veja (Sandel ,1970).

A convergência da seqüência até a matriz  $P$  é quadrática, considerando as imposições do teorema, ela é válida para a desigualdade dada por (3.7) para alguma constante  $C_1$  e  $\forall k$ . Isto garante uma convergência rápida de  $P_k$  até  $P$  na vizinhança de  $P$ ,

$$\|P_{k+1} - P\| \leq C_1 \|P_k - P\|. \quad (3.7)$$

É possível determinar a matriz  $K_0$ , sendo a matriz  $(A - B K_0)$  estabilizável e o par de matrizes  $(A, B)$  estabilizável. Esta condição é essencial, porém se a matriz  $(A - B K_0)$  não é estável, então a iteração pode convergir para uma solução instável da Eq. (3.5) ou ela pode divergir. A determinação de  $K_0$  pode ser feita por algum algoritmo estabelecido. O algoritmo estabelecido serve para referenciar o método que determina a matriz de realimentação do sistema de malha fechada. (Petkov e Konstantinov ,1991) introduzem dois métodos que são: utilização do gramiano de controlabilidade e o segundo método de *Lyapunov*. Os algoritmos para os casos contínuos e discretos são introduzidos também por (Petkov e Konstantinov ,1991).

### 3.3.2 Método da Função Sinal Matricial

O algoritmo que implementa o método da função sinal matricial é um dos métodos simples para resolver a equação algébrica de *Riccati*. Este método apenas

requer a inversão de matrizes e a solução da equação para um sistema consistente e sobrestimado. Ao contrário do método de *Newton*, ele não requer uma aproximação inicial para a solução. A seguir é apresentado o método e suas principais características.

Considere a matriz  $A \in \mathbb{R}^{n \times n}$  expressa na forma canônica de *Jordan*

$$J = X^{-1}AX \equiv D + N \quad (3.8)$$

sendo  $X$  uma matriz constituída de autovetores e autovetores principais da matriz  $A$ . A matriz  $D = \text{diag}(d_1, d_2, \dots, d_n)$  é uma matriz diagonal cujos elementos são os autovalores da matriz  $A$  e  $N$  é uma matriz nilpotente.

**Definição 3.3.1.** A Eq. (3.9) define a função sinal matricial

$$\text{sign}(A) = X^{-1}YX \quad (3.9)$$

sendo  $Y = \text{diag}(y_1, y_2, \dots, y_n)$ ,  $y_i = +1$ , se  $\text{Re}(d_i) > 0$  e  $y_i = -1$ , se  $\text{Re}(d_i) < 0$ .

Se a matriz  $A$  possui autovalores no eixo imaginário, então o  $\text{sign}(A)$  não está definido.

Propriedades da Função Sinal:

1.  $[\text{Sign}(A)]^2 = I_n$ .
2.  $\text{Sign}(cA) = \text{Sign}(c)\text{Sign}(A)$ .
3.  $\text{Sign}(TAT^{-1}) = T\text{Sign}(A)T^{-1}$ .
4.  $A\text{Sign}(A) = \text{Sign}(A)A$ .
5. Os autovalores da  $\text{Sign}(A) = \pm 1$ .

Aplicando a função sinal matricial na solução da *EAR*, tem-se a Eq. (3.5). A matriz *Hamiltoniana* correspondente a equação de *Riccati*, possui a propriedade:

$$W^{-1}HW = -H^T \quad (3.10)$$

sendo



$$W := \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

e  $W^{-1} = W^T = -W$ .

A Eq. (3.10) mostra que as matrizes  $H$  e  $-H^T$  são similares, isto é, possuem o mesmo conjunto de autovalores. Uma vez que os autovalores das matrizes  $H$  e  $-H^T$  são os mesmos, conclui-se que se  $\lambda$  é um autovalor de  $H$ , então  $-\lambda$  também é um autovalor de  $H$  com a mesma multiplicidade. Além disso, é possível mostrar que os  $n$  autovalores da matriz  $H$  coincidem com a região polar do sistema ótimo de malha fechada.

De fato, considere a transformação matricial dada pela Eq. (3.11)

$$T := \begin{bmatrix} I_n & 0 \\ P^0 & I_n \end{bmatrix}. \quad (3.11)$$

Realizando a transformação não singular na matriz  $H$ , tem-se

$$T^{-1}HT = \begin{bmatrix} I_n & 0 \\ -P^0 & I_n \end{bmatrix} \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} I_n & 0 \\ P^0 & I_n \end{bmatrix}. \quad (3.12)$$

Desenvolvendo-se, tem-se:

$$T^{-1}HT = \begin{bmatrix} A - BK^0 & -BR^{-1}B^T \\ 0 & -(A - BK^0)^T \end{bmatrix}. \quad (3.13)$$

Os autovalores da matriz *Hamiltoniana*  $H$ , são os autovalores da matriz  $(A - BK^0)$  juntamente com os autovalores da matriz  $-(A - BK^0)$  devido a Eq. (3.13) matricial possuir a forma triangular em blocos.

Da Eq. (3.13), verifica-se que a matriz *Hamiltoniana* pode ser decomposta na forma matricial a seguir:

$$H = \begin{bmatrix} I_n & 0 \\ P^0 & I_n \end{bmatrix} \begin{bmatrix} A_1 & -BR^{-1}B^T \\ 0 & A_1^T \end{bmatrix} \begin{bmatrix} I_n & 0 \\ P^0 & I_n \end{bmatrix} \quad (3.14)$$

sendo  $A_1 = A - SP$ . Uma vez que os autovalores da matriz  $A_1$  possuem parte real negativa, a função sinal da matriz  $H$  é definida. Aplicando esta função na Eq. (3.14) e usando a terceira propriedade da função sinal matricial, obtém-se:

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \text{Sign}(H)$$

$$\text{Sign}(H) = \begin{bmatrix} I_n & 0 \\ P & I_n \end{bmatrix} \begin{bmatrix} -I_n & Z \\ 0 & I_n \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -P & I_n \end{bmatrix} \quad (3.15)$$

sendo  $w_{ij}$ ,  $i, j = 1, 2$  e  $Z \in \mathbb{R}^{n \times n}$ .

Explorando a quarta identidade da função sinal matricial, tem-se que a matriz  $Z$  é a solução da Equação de *Lyapunov* a seguir:

$$A_1 Z + Z A_1^T = 2S. \quad (3.16)$$

Da Eq. (3.15) obtém-se

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} I_n \\ P \end{bmatrix} = - \begin{bmatrix} I_n \\ P \end{bmatrix}, \quad (3.17)$$

isto é,

$$MP = N. \quad (3.18)$$

A abordagem de Auto-Sistema é formada por dois métodos que são utilizados na resolução da *EAR*: Método dos Autovetores e Método de *Schur*. Esses métodos exploram a estrutura da matriz *Hamiltoniana*.

### 3.3.3 Método de Autovetores

A principal vantagem desse método, é que ele permite calcular a solução  $P$  da *EAR* sem iterações na matriz  $P$ . Como um resultado da decomposição, pode-se

determinar autovalores e autovetores da matriz  $A_1$ . Descreve-se agora em linhas gerais o método.

Considere a Eq. (3.5) na forma contínua, sendo  $(A, B)$  estabilizável e  $(A, C)$  detectável. A matriz *Hamiltoniana* associada a Eq.(3.5), possui os autovalores

$$-\lambda_1, -\lambda_2, \dots, -\lambda_n, \lambda_1, \lambda_2, \dots, \lambda_n$$

sendo,  $\lambda_i, i = 1, \dots, n$  autovalores com parte real positiva. Considerando essas premissas, a matriz  $H$  não possui autovalores puramente imaginários. Se os divisores elementares da matriz  $H$  correspondente a  $\lambda_1, \lambda_2, \dots, \lambda_n$  são lineares, então a matriz  $H$  pode ser transformada em uma forma diagonal dada pela Eq. (3.19)

$$= V^{-1}HV = \begin{bmatrix} -\Lambda & 0 \\ 0 & -\Lambda \end{bmatrix} \quad (3.19)$$

sendo  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ .

Suponha que a matriz dos autovetores da matriz  $V \in \mathbb{R}^{n \times n}$  seja particionada em blocos. Então:

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \quad (3.20)$$

e  $V_1 = \begin{bmatrix} V_{11}^T & V_{21}^T \end{bmatrix}^T$  é uma matriz que consiste dos autovetores correspondentes aos autovalores com parte real negativa. Quando a matriz  $V_{11}$  é não-singular, a solução simétrica e definida positiva da Eq. (3.5) é dada por

$$P = V_{21}V_{11}^{-1}. \quad (3.21)$$

Levando em consideração os autovalores da matriz de malha fechada do sistema  $A_1$ , a região polar do sistema ótimo são os autovalores  $-\lambda_1, -\lambda_2, \dots, -\lambda_n$  e  $U_{11}$  é a matriz dos autovetores da matriz  $A_1$ . Note que o ordenamento atual dos autovetores da matriz  $V_1$  não é significativa.

Se algum dos autovetores da matriz  $H$  for complexo, então a relação dada pela Eq.( 3.21) é verdadeira. Para evitar a utilização de aritmética complexa, ao

invés de usar autovetores complexos conjugados  $v_k = y_k + iz_k$  e  $v_{k+1} = y_k - iz_k$  correspondentes aos autovalores complexos conjugados  $\lambda_k = \sigma_k + i\omega_k$  e  $\lambda_{k+1} = \sigma_k - i\omega_k$ , pode-se colocar consecutivamente na matriz  $V_1$  os vetores  $y_k$  e  $z_k$ . Então ordenando  $\lambda_{k+1}$  e  $\lambda_k$  na forma de bloco diagonal 2x2 da matriz  $H$ , tem-se a forma

$$\begin{bmatrix} \sigma_k & \omega_k \\ -\omega_k & \sigma_k \end{bmatrix}.$$

Mostramos agora como as transformações de *Householder* e *Givens* podem ser usada para computar várias transformações, começando com a fatorização  $QR$ . A fatorização  $QR$  de uma matriz  $A \in \mathbb{R}^{m \times n}$  é dada por  $A = QR$  sendo  $Q \in \mathbb{R}^{m \times m}$  ortogonal e  $R \in \mathbb{R}^{m \times n}$  é triangular superior. Assume-se que  $m \geq n$ . Se a matriz  $A$  possui posto coluna completo, então as primeiras  $n$  colunas da matriz  $Q$  forma uma base ortonormal para o posto da matriz  $A$ . Assim, a computação da fatorização  $QR$  é uma forma de computar uma base ortonormal para um conjunto de vetores. Esta computação pode ser disposta de diversas formas. Podemos citar os métodos baseados em *Householder*, blocos de *Householder*, *Givens* e transformações rápidas de *Givens*, (Golub first edition 1983, second edition 1989 and third edition 1996).

Portanto, a solução da Eq. (3.5) possui o seguinte algoritmo:

1.

$$\text{Faça } H := \begin{bmatrix} A & -S \\ -Q & -A^T \end{bmatrix}. \quad (3.22)$$

2. Calcule os autovalores da matriz  $H$  pelo método  $QR$ .

3. Selecione os autovetores correspondentes a partir dos autovalores correspondentes com parte real negativa e os coloque na forma matricial dada por

$$\begin{bmatrix} V_{11} \\ V_{21} \end{bmatrix}. \quad (3.23)$$

4. Resolva  $V_{11}^T P = V_{21}^T$  para a matriz  $P$ .

Esse método é numericamente instável quando a matriz *Hamiltoniana* é deficiente,  $H$  tem pouco autovalores múltiplos, uma vez que neste caso a matriz  $U_{11}$  é

mal condicionada. Deve ser enfatizado que o mal condicionamento da matriz  $U_{11}$ , provocado pela perda de alguns autovalores é independente do condicionamento da equação de *Riccati*. Pode-se salientar como uma desvantagem, o fato da matriz *Hamiltoniana* não possuir *rank* completo.

As dificuldades no caso dos divisores lineares não elementares da matriz  $H$ , podem ser evitadas se a matriz  $H$  for reduzida para a forma canônica de *Jordan*. A solução tem a forma dada pela Eq. (3.21), porém a matriz  $V$  pode ser constituída pelos autovalores e pelos vetores principais da matriz  $H$ . Entretanto, a redução da matriz  $H$  para a forma de *Jordan* requer  $n^4$  flops, tornando esse método inadequado. Além disso, em alguns casos, a forma de *Jordan* pode ser mal condicionada, o que resultará em um mal condicionamento da matriz  $V_{11}$ .

Considere as matrizes  $M = \begin{bmatrix} w_{12} \\ w_{22} + I_n \end{bmatrix}$  e  $N = \begin{bmatrix} w_{11} + I_n \\ w_{12} \end{bmatrix}$ . A Eq.(3.16) é um sistema consistente de  $2n^2$  equações com  $n^2$  elementos ignorados da matriz  $P$ . Da Eq. (3.15), tem-se que

$$\begin{bmatrix} w_{12} \\ w_{22} \end{bmatrix} = \begin{bmatrix} Z \\ PZ + I_n \end{bmatrix}, \quad (3.24)$$

$$M = \begin{bmatrix} Z \\ Z + 2I_n \end{bmatrix} \quad (3.25)$$

e

$$\begin{bmatrix} -P & I_n \end{bmatrix} M = \begin{bmatrix} -P & I_n \end{bmatrix} \begin{bmatrix} Z \\ PZ + 2I_n \end{bmatrix} = 2I_n \quad (3.26)$$

Da Eq. (3.26) temos que  $\text{rank}(M) = n$ . A Eq. (3.18) é resolvida por um modo confiável usando a decomposição  $QR$  da matriz  $M$ .

A função sinal matricial da equação *Hamiltoniana* pode ser calculada da seguinte maneira: a primeira identidade da propriedade da função sinal mostra que  $\text{Sign}(H)$  é a raiz quadrada da matriz identidade. Aplicando o método de *Newton* na raiz quadrada da matriz identidade, é obtida a seqüência dada pelas Eqs. (3.27-3.28):

$$W_0 = H \quad (3.27)$$

$$W_{k+1} = \frac{1}{2}(W_k + W_k^{-1}) \quad (3.28)$$

que convergem quadraticamente para  $Sign(H)$ . Se  $Sign(H)$  não é definido, então as iterações dadas pelas Eqs. (3.27-3.28) não convergem ou alguma das matrizes  $W_k$  são singulares. É possível mostrar que se a matriz  $H$  é *Hamiltoniana*, então a matriz  $H^{-1}$  também é *Hamiltoniana*, então desta maneira  $\frac{1}{2}(H + H^{-1})$  é uma matriz *Hamiltoniana*. Portanto, deste modo as propriedades da matriz *Hamiltoniana*  $H_0$  se propaga em  $W_{K+1} = \frac{1}{2}(H + H^{-1})$ .

Resultados mais precisos para o cálculo da  $Sign(H)$  podem ser obtidos usando a seqüência de Eqs. (3.29-3.30) a seguir

$$W_0 = H \quad (3.29)$$

$$W_{K+1} = W_k - \frac{1}{2}(W_k + W_k^{-1}). \quad (3.30)$$

Neste caso, o erro de arredondamento é reduzido e a correção é menor. As considerações são resumidas no algoritmo a seguir:

1.

$$\text{Faça } H := \begin{bmatrix} A & S \\ -Q & -A^T \end{bmatrix}. \quad (3.31)$$

2. Calcule  $W = Sign(H)$  usando as Eqs.(3.29-3.30).

3. Resolva a Eq.(3.18) para  $P$ , solução da *EAR*.

### 3.3.4 Abordagem do Método de *Schur*

Usaremos o algoritmo de *Schur* para resolver a Equação Algébrica de *Riccati*. O método de *Schur* é estudado como uma variante da abordagem clássica de autovetores e utiliza ao invés disso, um conjunto adequado de vetores de *Schur*. Desta maneira adquire uma vantagem numérica substancial. A abordagem dos vetores de *Schur* fornece uma técnica confiável, proveitosa e muito mais eficiente em relação ao método de *Newton* para resolver numericamente a equação algébrica de *Riccati* (Laub ,1979).

O método de *Schur* possui diversas vantagens em relação ao método dos autovetores. Inicialmente, a redução para a forma quasi-triangular é um passo intermediário no cálculo dos autovetores usando a decomposição *QR*, de tal forma que o método de *Schur* requer menos cálculo que o método dos autovetores. Independentemente da matriz *Hamiltoniana* ser defectiva, o método de *Schur* não é afetado pelas dificuldades inerentes dos autovetores da matriz *Hamiltoniana*.

Matrizes defectivas não possuem um conjunto completo de autovetores e, devido a isso, não são diagonalizáveis por uma transformação de similaridade. Esta deficiência é contornada por meio do Teorema de *Schur*, que é um resultado que se aplica a todas as matrizes simples, (matrizes com  $n$  autovetores linearmente independentes) ou defectivas, e isto é quase tão efetivo quanto o Teorema da Diagonalização para analisar a estrutura da matriz *Hamiltoniana*.

As matrizes simples são usadas em extensivas aplicações de Equações Diferenciais e Equações de Diferença. A ferramenta que torna isso possível é o Teorema da Diagonalização para matrizes simples, isto é, existe uma matriz não-singular  $S$  tal que  $S^{-1}AS = \Lambda$ . Não existe teorema que efetue tal diagonalização para matrizes cujos vetores colunas não formem uma base. No Teorema de *Schur*, a matriz  $\Lambda$  é substituída por uma matriz triangular superior  $T$ . Além de fornecer uma ferramenta para o estudo de matrizes defectivas, o teorema de *Schur* é útil na análise de uma variedade de importantes matrizes simples.

**Definição 3.3.2.** Diz-se que a matriz  $A \in \mathbb{R}^{2n \times 2n}$  é *Hamiltoniana* se  $J^{-1}A^T J = -A$  e que  $A \in \mathbb{R}^{2n \times 2n}$  é *Simplética* se  $J^{-1}A^T J = A^{-1}$ . Sendo a matriz  $A \in \mathbb{R}^{m \times n}$ , define-se o seu espectro (conjunto de autovalores e autovetores) por  $\sigma(A)$ .

**Teorema 3.3.2.** (1) Seja  $A \in \mathbb{R}^{2n \times 2n}$  uma matriz *Hamiltoniana*. Então  $\lambda \in \sigma(A) \Rightarrow -\lambda \in \sigma(A)$  com a mesma multiplicidade. (2) Seja  $A \in \mathbb{R}^{2n \times 2n}$  uma matriz

*Simplética. Então  $\lambda \in \sigma(A) \Rightarrow \frac{1}{\lambda} \in \sigma(A)$  com a mesma multiplicidade.*

**Prova** Veja (Laub ,1972).

**Teorema 3.3.3.** *Seja  $A \in \mathbb{R}^{2n \times 2n}$  uma matriz Simplética ou Hamiltoniana. Seja  $U \in \mathbb{R}^{2n \times 2n}$  uma matriz Simplética. Então  $U^{-1}AU$  é uma matriz Hamiltoniana ou Simplética.*

**Prova** Veja (Laub ,1972).

Apresentam-se dois teoremas clássicos da teoria de similaridade que formam a base teórica da álgebra linear numérica moderna.

**Teorema 3.3.4.** *(Forma Canônica de Schur): Suponha que  $\lambda_1, \dots, \lambda_n$  sejam autovalores da matriz  $A \in \mathbb{R}^{2n \times 2n}$ . Então existe uma transformação de similaridade unitária  $U$  tal que  $U^H AU$  é triangular superior com elementos diagonais  $\lambda_1, \dots, \lambda_n$  na ordem.*

**Prova** Veja (Stewart ,1979) e (Goldberg ,1991).

**Corolário 3.3.1.** *Se  $U^T AU = T$ , então os autovalores de  $A$  são as entradas diagonais de  $T$ .*

**Prova** Os autovalores de  $A$  e  $T$  são idênticos devido  $A$  e  $T$  serem similares. As entradas diagonais de  $T$  são os seus autovalores devido  $T$  ser uma matriz triangular superior.

Realmente é possível trabalhar somente sobre o conjunto dos números reais, reduzindo para a forma quasi-triangular com blocos 2x2 na diagonal correspondendo a autovalores complexos conjugados e blocos 1x1 correspondentes a autovalores reais. Refere-se a esta forma canônica como a forma real de *Schur* (*RSF*) ou a forma canônica *Murnaghan-Winter*.

**Teorema 3.3.5.** *(Forma Real de Schur): Seja  $A \in \mathbb{R}^{2n \times 2n}$ . Então existe uma transformação de similaridade ortogonal  $U$  tal que  $U^T AU$  é uma matriz quasi-triangular superior. Além disso, a matriz  $U$  pode ser escolhida de modo que os blocos diagonais  $2 \times 2$  e  $1 \times 1$  aparecem em qualquer ordem desejada.*

**Prova** Veja (Murnaghan ,1991).

As dificuldades numéricas relacionadas a utilização de autovetores da matriz *Hamiltoniana* quase defectiva podem ser contornadas aplicando o método de *Schur*. A equação contínua de *Riccati*, pode ser resolvida por meio da redução da matriz *Hamiltoniana*  $H$ , para a forma quasi-triangular, denominada Forma Real



de *Schur*. Para isso, fazemos uso das transformações ortogonais de similaridade. Para maiores detalhes, veja (Laub ,1979).

O método de *Schur* possui uma série de vantagens em relação ao método dos autovetores. Inicialmente, a redução para a forma quasi-triangular é um passo intermediário no cálculo dos autovetores pelo método *QR*, de tal maneira que o método de *Schur* requer menos cálculos que o método dos autovetores. É importante também enfatizar que o método de *Schur* não é afetado pelas dificuldades inerentes nos autovetores da matriz *Hamiltoniana* defectiva. Esse método tem um bom desempenho na presença de divisores lineares da matriz *H*.

O método de *Schur* é significativamente mais rápido que o método de *Newton* e também o método da *Função Sinal Matricial* com refinamento iterativo. Existe uma matriz ortogonal *U*, que transforma a matriz *Hamiltoniana H* na forma real de *Schur* dada por

$$T = U^T H U = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \quad (3.32)$$

sendo  $T_{11}$  e  $T_{22}$  matrizes quasi-triangulares superiores. Os blocos na diagonal das matrizes  $T_{11}$  e  $T_{22}$  são no máximo  $2 \times 2$ . A redução da matriz Eq. (3.5) não é única, e sempre é possível escolher a matriz *U* de tal modo que  $T_{11}$  tenha autovalores com parte real negativa, enquanto os autovalores da matriz  $T_{22}$  possui parte real positiva. Particionemos a matriz *U* da seguinte maneira:

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \quad (3.33)$$

sendo que cada bloco possua dimensão  $n \times n$ . Se a matriz  $U_{11}$  é não-singular, então a solução da equação algébrica de Riccati, *EAR*, é semidefinida positiva e dada pela expressão

$$P = U_{21} U_{11}^{-1}. \quad (3.34)$$

Além disso, os autovalores da matriz de bloco  $T_{11}$  formam o espectro da malha fechada da matriz  $(A - BR^{-1}B^T P)$ , isto é, a região polar do sistema de malha fechada ótimo é devida a relação  $(A - BR^{-1}B^T P) = U_{11} T_{11} U_{11}^{-1}$ .

A redução ortogonal da matriz *Hamiltoniana* na Forma Real de *Schur* é feita pelo método de decomposição *QR*. Se a matriz *H* não é simétrica, não é pos-

sível garantir qualquer ordenação especial dos autovalores na diagonal da forma quasi-triangular. É possível reordenar esta forma arbitrariamente pela mudança sistemática dos pares de autovalores adjacentes, implementando transformações ortogonais de similaridades.

A seguir, mostra-se um algoritmo para resolver o problema pelo método de *Schur* baseado na teoria introduzida nos parágrafos anteriores.

1.

$$\text{Faça } H := \begin{bmatrix} A & S \\ -Q & -A^T \end{bmatrix}. \quad (3.35)$$

2. Reduza a matriz  $H$  para a forma real de *Schur*  $T = U^T H U$  usando transformações ortogonais de similaridade.
3. Reordene a forma real de *Schur* usando transformação ortogonal, de tal forma que o bloco quasi triangular  $T_{11}$  no lado superior esquerdo de  $T$  tenha autovalores com parte real negativa e acumule as transformações em  $U$ .
4. Resolver  $U_1^T P = U_{21}^T$  para  $P$ , sendo  $[U_{11}^T U_{21}^T]^T$  os correspondentes vetores de *Schur* para  $T_{11}$ .

### 3.3.5 Método Simplético

A abordagem de Auto-Sistema possui a desvantagem de não explorar completamente a estrutura da matriz *Hamiltoniana*  $H$ . Nesse método o algoritmo da decomposição  $QR$  é aplicado na matriz  $H$  como se ela fosse uma matriz geral do tipo  $2n \times 2n$ , isto é, sem explorar a estrutura da matriz  $H$ , que é ineficiente no armazenamento do trabalho e armazenamento computacional. Os erros de arredondamento que surgem no algoritmo  $QR$  não consideram a estrutura da matriz *Hamiltoniana* do problema, de tal forma que os autovalores não aparecem exatamente mais em pares. Se na redução da matriz *Hamiltoniana* para a forma de *Schur* usa-se transformações de similaridade, que são unitárias ou simpléticas, as dificuldades do parágrafo anterior são contornadas e esses métodos são denominados de métodos simpléticos.

Para qualquer matriz  $A$ , as notações  $A \in \mathbb{R}^{m \times n}$  e  $A \in \mathbb{C}^{m \times n}$ , designam as matrizes com entradas reais e complexas,  $A^T$  e  $A^H$  representam as matrizes

transpostas e conjugadas, respectivamente, enquanto que  $A^{-T}$  denota  $(A^T)^{-1} = (A^{-1})^T$ .

**Definição 3.3.3.** Diz-se que a matriz  $A \in \mathbb{R}^{n \times n}$  é ortogonal se  $A^T = A^{-1}$ , enquanto que a matriz  $A \in \mathbb{R}^{n \times n}$  é unitária se  $A^H = A^{-1}$ .

Seja a matriz  $J =: \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$ ,  $J \in \mathbb{R}^{2n \times 2n}$  sendo  $I_n$  a matriz identidade de ordem  $n$  e que  $J^{-1} = H^H = J^T = -J$ .

**Definição 3.3.4.** Diz-se que a matriz  $H \in \mathbb{R}^{2n \times 2n}$  é *Hamiltoniana* se  $J^{-1} = J^H = J^T = -J$  e que a matriz  $Z \in \mathbb{R}^{2n \times 2n}$  é *Simplética* se  $J^{-1}Z^T J = Z^{-1}$ .

As matrizes *Hamiltonianas* e *Simpléticas* estão intrinsecamente relacionadas. Para uma maior discussão e revisão da Álgebra Simplética, veja (Laub, 1972).

Aplicando a transformação simplética na matriz *Hamiltoniana*, obtém-se:

$$\begin{aligned} J^{-1}(Z^{-1}HZ)J &= (J^{-1}ZJ)^{-1}(J^{-1}HJ)(J^{-1}ZJ) \\ &= -Z^H H^H Z^{-H} = -(Z^{-1}HZ)^H. \end{aligned} \quad (3.36)$$

Se a matriz  $U \in \mathbb{R}^{2n \times 2n}$  é unitária e simplética, então assume-se que  $J^{-1}UJ = U^{-H}$  e  $U$  tem a forma seguinte:

$$T = U^T H U = \begin{bmatrix} U_1 & U_2 \\ -U_2 & U_1 \end{bmatrix} \quad (3.37)$$

sendo  $U_1 \in \mathbb{R}^{n \times n}$  e  $U_2 \in \mathbb{R}^{n \times n}$ .

Partindo de  $U^H U = I_{2n}$ , tem-se:

$$\begin{aligned} U_1^H U_1 + U_2^H U_2 &= I_n \\ U_1^H U_2 + U_2^H U_1 &= 0. \end{aligned} \quad (3.38)$$

Note que usando a matriz  $U$ , é necessário apenas armazenar as matrizes de blocos  $U_1$  e  $U_2$ . As transformações simpléticas podem ser implementadas para formar elementos específicos de vetores ou matrizes. Para esse propósito é possível usar dois tipos de matrizes simpléticas unitárias, que são as matrizes de *Householder* e *Givens*.

Matrizes *simpléticas* de *Householder* possuem a seguinte forma:

$$U(k, v) = \begin{bmatrix} I_{k-1} & 0 & 0 & 0 \\ 0 & V & 0 & 0 \\ 0 & 0 & I_{k-1} & 0 \\ 0 & 0 & 0 & V \end{bmatrix} \quad 1 \leq k \leq n \quad (3.39)$$

sendo  $V = I_k - 2vv^H/(2vv^H)$  uma matriz de *Householder* de  $k$ -ésima ordem.

**Definição 3.3.5.** As matrizes simpléticas de *Givens* são definidas da seguinte maneira:

$$J(k, c, s) = \begin{bmatrix} C & S \\ -S & C \end{bmatrix} \quad (3.40)$$

sendo  $C = \text{diag}(1, 1, \dots, 1, c, 1, 1, \dots, 1)$ ,  $S = \text{diag}(0, 0, \dots, 0, s, 0, 0, \dots, 0)$  e  $|c|^2 + |s|^2 = 1$ .

As matrizes *simpléticas* de *Householder* podem ser usadas em vários componentes de um vetor enquanto as outras componentes do vetor permanecem inalteradas. Especificamente, é possível construir a matriz  $U(k, v)$  de forma tal que dado o vetor  $k$ , os vetores  $a \in \mathbb{R}^{n \times n}$  e  $b \in \mathbb{R}^{n \times n}$  são determinados da seguinte maneira:

$$U(k, v) \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.41)$$

sendo  $y_i = 0$  para  $i = k + 1, \dots, n$ .

As matrizes simpléticas podem ser exploradas para entradas nulas de um vetor ou uma matriz. Especificamente, para  $k$  e  $n$ , dados  $a$  e  $b$  é possível determinar uma matriz  $J(k, c, s)$  tal que

$$J(k, c, s) \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.42)$$

sendo  $y_k = 0$ .

Mostra-se que cada matriz simplética unitária pode ser representada como um produto de matrizes simpléticas de *Householder* e *Givens*. A seguir é introduzido um teorema que exhibe a estrutura canônica da matriz *Hamiltoniana* quando ela sofre ações sobre transformações simpléticas unitárias.

**Teorema 3.3.6.** *Teorema da Decomposição Hamiltoniano-Schur.* Seja  $H$  uma matriz *Hamiltoniana* cujos autovalores não possuem parte real nula. Então:

$$U^H H U = T = \begin{bmatrix} T_1 & T_2 \\ 0 & -T_1^H \end{bmatrix} \quad (3.43)$$

sendo  $T_1$  uma matriz triangular superior,  $T_2$  é *Hamiltoniana* e  $T$  é referenciada como uma matriz triangular superior em relação a  $H$ .

**Corolário 3.3.2.** Se  $Z$  é uma matriz simplética que não possui autovalores de magnitude 1, então existe uma matriz simplética unitária tal que

$$U^H Z U = R = \begin{bmatrix} R_1 & R_2 \\ 0 & -R_1^{-H} \end{bmatrix} \quad (3.44)$$

sendo  $R_1$  uma matriz triangular superior e  $R_2 R_1^H$  é *Hermitiana*. A matriz  $R$  é simplética triangular.

Este corolário pode ser demonstrado usando-se a seguinte transformação:

$$H = (Z + I_{2n})(Z - I_{2n})^{-1}. \quad (3.45)$$

A Decomposição (3.46) pode ser usada para resolver a equação de *Riccati*, Eq. (3.5).

Sabendo-se que o par de matrizes  $(A, B)$  é estabilizável e que o par  $(A, C)$  é detectável, os termos do Teorema da Decomposição *Hamiltoniana-Schur* são satisfeitos.

Se a matriz

$$U^H H U = T = \begin{bmatrix} U_1 & U_2 \\ -U_2 & U_1 \end{bmatrix} \quad (3.46)$$

é escolhida de tal forma que os autovalores da matriz  $T$  possuam parte real negativa, então a solução semidefinida positiva é determinada pela equação

$$P = -U_2 U_1^{-1}. \quad (3.47)$$

No caso geral, a solução da decomposição *Hamiltoniana-Schur* é um problema difícil. Para o caso de uma única entrada ( $\text{rank}(b) = 1$ ) existe uma variante do algoritmo *QR*, que nos permite determinar eficientemente a forma *Hamiltoniana* triangular da matriz  $H$  usando uma seqüência de transformações simpléticas unitárias. Como no algoritmo *QR* o volume de cálculos pode ser substancialmente reduzido, se a matriz *Hamiltoniana* original é preliminarmente reduzida para alguma forma condensada. No caso que ( $\text{rank}(B) = 1$ ), a matriz *Hamiltoniana* pode ser reduzida para a forma *Hamiltoniana-Hessenberg*. Então,

$$W^H H W = R = \begin{bmatrix} F & Y \\ 0 & -F^H \end{bmatrix} \quad (3.48)$$

sendo  $F$  uma matriz de *Hessenberg* triangular superior.

O algoritmo *Hamiltoniano QR*, consiste de diversas transformações simpléticas aplicadas iterativamente na forma *Hamiltoniana-Hessenberg*. Este algoritmo é numericamente estável. Entretanto, é verificado que em alguns casos, grandes perturbações nas matrizes individuais  $A$ ,  $Q$  e  $B R^{-1} B^T$  podem levar a instabilidade numérica do método como um todo, similar ao que ocorre na abordagem do auto-sistema.

### 3.4 Conclusão

Neste capítulo foi apresentada a formulação e solução do controlador *LQR* clássico, tomando como referência (Athans e Falb, 1966). Sua solução está relacionada com a determinação de uma matriz simétrica e definida positiva da equação algébrica de *Riccati*. Para isso, diversas abordagens foram enfatizadas, dando destaque ao método de *Schur* para solução da EAR, que explora as propriedades da matriz *Hamiltoniana* e os métodos heurísticos, e em particular o método de *Bryson* para realizar a escolha das matrizes de ponderação. Muitas das tendências

---

atuais na computação das redes neurais estão enraizadas na teoria de otimização, como a desenvolvida na década de 1970. O campo de controle ótimo, é uma parte importante desta teoria que explora diversas capacidades de aproximações não-linear de métodos de computação neural (White ,1992).

# METODOLOGIA NEURO - GENÉTICA PARA SOLUÇÃO DO LQR

---

A metodologia neuro-genético representa uma fusão de um algoritmo genético e uma rede neural recorrente para realizar a seleção das matrizes de ponderação e a solução da equação algébrica de *Riccati*, respectivamente. O objetivo principal desta metodologia é estabelecer uma busca automática das matrizes de ponderação  $QR$  e em seguida a resolução da *EAR* via RNA em tempo mínimo de modo que atenda a ação de controle do projeto *LQR*. Especificamente, o controle *LQR* almeja impor uma alocação de autoestrutura específica em sistemas MIMO.

A metodologia de fusão coordena a computação da solução Neural-EAR, que tem que estar sincronizada com os valores das matrizes  $Q$  e  $R$  fornecidos pelo AG segundo mostra a Fig. 4.1. Este fluxograma está associado com uma primeira abstração da metodologia apresentada na estrutura (4.1):



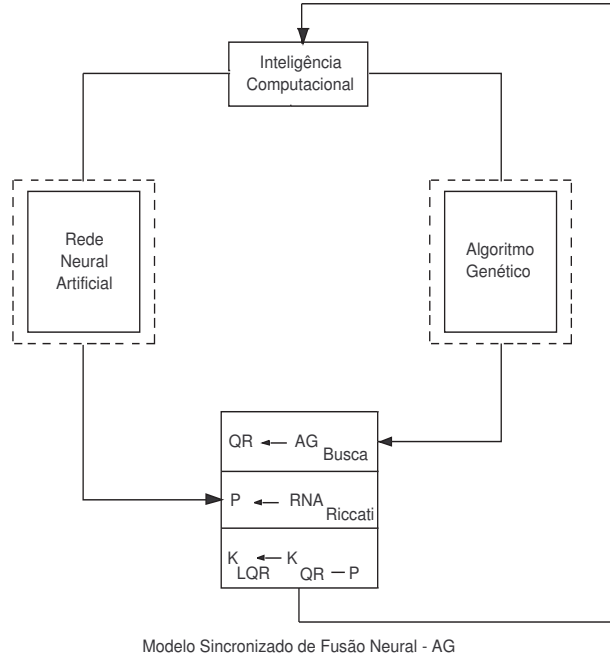


Figura 4.1: Modelo de fusão neuro-genético.

$$\begin{aligned}
 QR &\leftarrow AG_{Busca} \\
 P &\leftarrow RNA_{Riccati} \\
 K_{LQR} &\leftarrow K_{QR-P}
 \end{aligned} \tag{4.1}$$

sendo  $QR$  as matrizes selecionadas pelo AG,  $P$  é a solução da equação algébrica de *Riccati* que é obtida com a rede neural artificial e  $K_{LQR}$  é o vetor de ganhos do controlador que é calculado com as matrizes  $R$  e  $P$ .

A metodologia de fusão neuro-genético é designado para coordenar as iterações entre o  $AG_{Busca}$  que representa a metodologia do AG para a seleção das matrizes de ponderação, a  $RNA_{Riccati}$  que representa a solução da equação algébrica de *Riccati* e  $K_{QR-P}$ , a computação dos ganhos ótimos.

Pelo ponto de vista da metodologia de fusão, as vantagens em usar o AG, consiste em obter melhores soluções viáveis em tempo menor que as heurísticas de tentativas e erro. A vantagem de usar uma rede neural artificial, consiste

em evitar operações com inversões de matrizes na solução da equação de *Riccati*. Além do paralelismo natural, a rede neural artificial possui a vantagem de poder modelar sistemas do mundo real.

O restante deste capítulo está organizado em Seções. A Seção 4.1 é baseada nos paradigmas de IC, que apresenta uma abordagem proposta pelo projeto *LQR* que está fundamentada na fusão do AG e da RNA. A Seção 4.2 define um AG canônico realçando sua esquematização, operadores genéticos e estratégia elitista. A Seção 4.3 destaca o modelo de busca das matrizes de ponderação que é utilizado para sintonizar o ganho do controlador e a solução do LQR, e também os modelos que compõem o AG para a realizar a busca das matrizes de ponderação.

A Seção 4.4 apresenta alguns conceitos básicos e definições de termos da genética artificial dentro do contexto da modelagem genética das matrizes de ponderação. A Seção 4.5 discute os modelos que compõem o algoritmo genético para a busca das matrizes de ponderação. A Seção 4.6 apresenta uma estrutura algorítmica comum modificada para resolução da EAR, sendo que: seus parâmetros matriciais iniciais são matrizes identidades, a taxa de convergência (tamanho do passo) está associada aos parâmetros de sintonia que são responsáveis pela velocidade de convergência da RNR e a direção de busca é fornecida pela dinâmica das camadas de saída e oculta.

A Seção 4.7 introduz uma base informativa a respeito da EAR no contexto de síntese dos sistemas do projeto *LQR*. Já a Seção 4.8 apresenta a formulação do problema para realizar a solução da equação de *Riccati*, enquanto que a Seção 4.9 descreve a arquitetura da RNR por meio de um sistema de equações diferenciais não lineares.

A Seção 4.10 fornece comentários e observações à respeito do treinamento da rede neural recorrente. Teoremas sobre a estabilidade da rede e condição necessária e suficiente de convergência para uma solução simétrica e definida positiva da EAR em quase todos os estados iniciais são apresentados na Seção 4.11. Este capítulo é complementado com a Seção 4.13 que diz respeito a conclusões e observações finais.

## 4.1 Metodologia Neuro-Genética

Os controladores ótimos *LQR* são baseados em paradigmas de inteligência computacional. A Fig. 4.2 mostra a abordagem proposta de projeto inteligente LQR, que tem por base a concatenação do AG e da RNR. O algoritmo genético é orientado à busca das matrizes de ponderação e detalhes da avaliação do desempenho de sua convergência, podem ser encontrados em (Abreu ,2008b). Uma avaliação do ajuste dos parâmetros está relatado em (Silva Junho, 2008), sendo que a EAR é avaliada em termos de estabilidade e solvabilidade para diversos sistemas dinâmicos. A matriz de ganho de realimentação ótima é dada por  $K = R^{-1}B^T P$ , sendo  $K \in \mathbb{R}^{m \times n}$ .

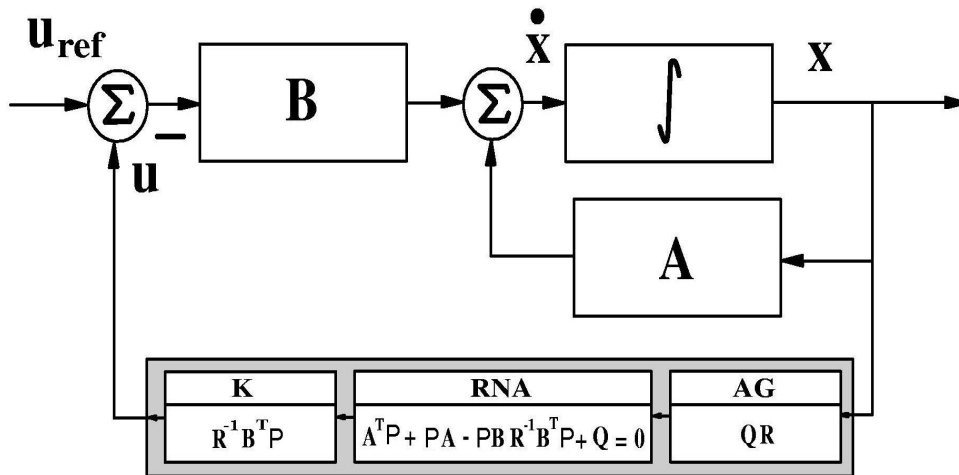


Figura 4.2: Modelo do controlador ótimo neuro-genético.

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 a lei de controle dada por  $u_{LQR} = -K_{LQR}x$ .

Nesta abordagem propõe-se uma rede neural recorrente com múltiplas camadas para resolver a equação algébrica matricial de *Riccati* contínua no tempo. A rede neural recorrente aqui proposta, apresenta quatro camadas conectadas bidirecionalmente, sendo que cada camada é formada de arranjos de neurônios, segundo a Fig. 4.3.

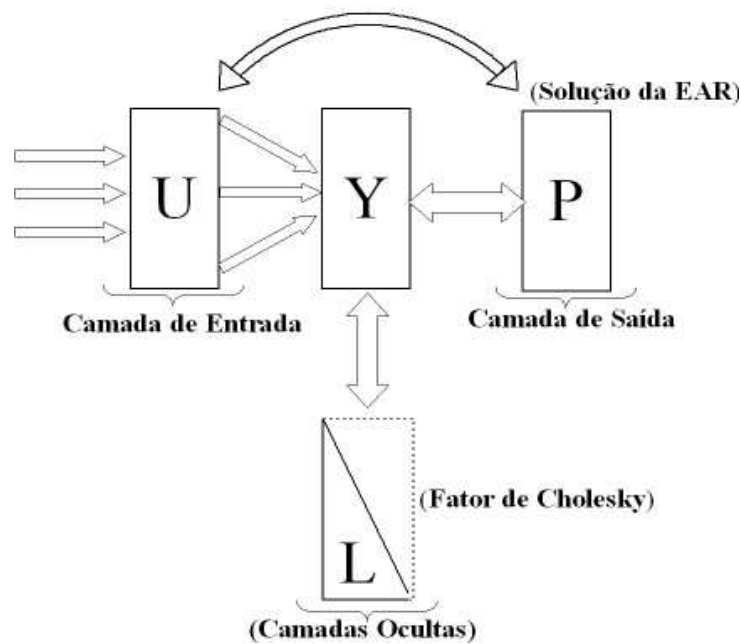


Figura 4.3: Estrutura algébrica de arquitetura da RNR

Mostra-se que a rede neural recorrente é capaz de resolver a *EAR*. Discutem-se resultados analíticos a respeito de estabilidade da rede neural recorrente *RNR* e solvabilidade da *EAR* pelo uso da *RNR*. As características de operação da rede neural são demonstradas pelos experimentos computacionais.

As redes neurais são formadas de neurônios simples e conectados maciçamente. Tendo as estruturas semelhantes a sua contraparte biológica, as redes neurais artificiais são modelos representacionais e computacionais de processo de informações de modo distribuído e paralelo. Demonstra-se que as redes neurais treinadas com um número limitado de amostras de treinamento possuem uma boa capacidade de generalização. As redes neurais possuem a vantagem sobre as abordagens tradicionais devido o processamento de informação ser inerentemente concorrente.

Na literatura de controle, têm-se dado muita atenção para os problemas de projeto de sistemas de controle *LQR*, como formulação do problema, solução do acompanhamento do sinal de referência e propriedades de robustez com relação a variações razoavelmente amplas de parâmetros do sistema. O problema de

projetar um sistema de controle de realimentação linear, minimizando um índice de desempenho quadrático, pode ser reduzido a se obter uma solução simétrica e definida positiva para a resolução da equação de *Riccati* (Gardiner e Laub ,1991a), (Laub ,1979), (Bunse-Gerstner A. ,1997) e outros.

Devido a computação neural de natureza distribuída paralela, as redes neurais podem ser um modelo computacional viável para sintetizar sistemas de controle lineares em tempo real. Atualmente, as redes neurais recorrentes têm sido desenvolvidas para resolver uma ampla variedade de problemas de álgebra matricial, (Wang e Jerry ,1992) e outros. Em particular, as redes neurais têm sido desenvolvidas por decomposição *LU* e fatoração de *Cholesky*, (Wang e Guang. ,1993). Os resultados dessas investigações têm estabelecido a base para solucionar a equação algébrica de *Riccati* via rede neural em tempo real.

Esta pesquisa apresenta uma abordagem sobre RNR para resolver *EAR* contínua no tempo baseada em uma rede neural recorrente com quatro camadas. Resolvendo-se duas equações matriciais utilizando redes neurais recorrentes, a abordagem proposta sobre redes neurais, é capaz de obter uma solução simétrica e definida positiva para a *EAR* conforme vê-se em experimentos computacionais.

## 4.2 Metodologia Genética para Busca das Matrizes QR

A dinâmica complexa dos sistemas e o aumento da demanda de requisitos de um sistema de malha fechada, exige o uso de controladores mais sofisticados e complexos. Os sistemas não lineares exigem o uso de controladores mais sofisticados que os sistemas lineares de baixa ordem quando devem ser satisfeitos objetivos que vão além da estabilidade. Entretanto, somente durante as últimas décadas tem ocorrido um desenvolvimento significativo nas áreas de computação, habilitando novas abordagens para projeto de controladores de sistemas dinâmicos complexos. Por exemplo, abordagens que tem por base a simulação de processos biológicos (Zicouchian ,2005).

Segundo (Passino ,2004), o controle inteligente é baseado em simulação de processos biológicos. As áreas de biologia, neurociência cognitiva, psicologia 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, possuem conhecimento significativo em como resolver um problema de automação e este conhecimento pode ser automatizado em algoritmos computacionais devido a tarefa ser extremamente árdua.

O controle inteligente é um procedimento e uma abordagem da teoria computacional eficiente para atingir metas de um sistema dinâmico complexo de controle, baseado em certos princípios de inteligência computacional (Meystel, 1985). As metas podem ser estendidas, por exemplo, como requisitos desejados de desempenho e estabilidade.

Uma abordagem recente no desenvolvimento dos algoritmos e dos processos de modelagem dos sistemas, estão relacionados com a observação dos processos biológicos, evolutivos, aprendizagem do ser humano e evolução das espécies. A modelagem do conhecimento, da inteligência e discussões sobre as variáveis envolvidas nesse processo, tem sido largamente estudada nas últimas décadas. A inteligência computacional IC, é a área que engloba as mais diversas técnicas e métodos para esses estudos. Técnicas de IC usadas em controle, são frequentemente denominadas de controle inteligente. Lógica *fuzzy*, algoritmos genéticos, redes neurais e sistemas especialistas são as principais técnicas de IC aplicadas em controle.

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 foi construído por (Neto, 2000), para resolver o problema de alocação de autoestruturas em sistemas dinâmicos multivariáveis. O método proposto é uma alternativa para alocar a autoestrutura por meio do  $LQR$ , superando as buscas por tentativa e erro do projetista para determinar as matrizes de ponderação. Este Capítulo está fundamentado nos algoritmos dos modelos propostos.

Esta seção tem por objetivo apresentar os modelos que compõem o algoritmo genético para a busca das matrizes de ponderação  $Q$  do estado e  $R$  do projeto de controle  $LQR$ , de maneira a determinar um controlador que satisfaça as restri-

ções de alocação de autoestrutura, conduzindo a um sistema dinâmico com bom desempenho e robustez de estabilidade. Esse modelo está alicerçado em uma estrutura de otimização para busca dessas matrizes. Estes modelos abordam desde as representações genético artificial das matrizes, passando pelas operações cromossômicas até chegar nas avaliações de *fitness*. O modelo representa a estrutura de otimização combinatória para busca das matrizes de ponderações, em termos de algoritmos e estruturas de dados de computação evolutiva. Apresenta-se alguns conceitos básicos e definições relacionados com a genética artificial dentro do contexto do modelo genético direcionado para a busca das matrizes de ponderação  $Q$  e  $R$  tais como: cromossomos, genótipos, fenótipos, genes e outros termos que foram criados para sintetizar as idéias que estão envolvidas no contexto do algoritmo proposto.

#### 4.2.1 Definição Geral de um AG Canônico

Podemos definir formalmente um algoritmo genético canônico como um  $n$ -upla do tipo

$$AG = (\Sigma, \Omega, m, P, f, X_S, X, \Psi, \tau), \quad (4.2)$$

sendo  $\Sigma$  o espaço de busca (fenótipos);  $\Omega$  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,

$$f : b_i \rightarrow \mathfrak{R}^+, \quad i = 1, \dots, m, \quad (4.3)$$

o operador de seleção de  $r$  pais para o cruzamento é

$$X_s : P \rightarrow \{p_1, p_2, \dots, p_r\}, \quad (4.4)$$

$X$  é o conjunto de operadores  $X_C$  (*crossover*) e  $X_M$  (mutação) que produzem  $s$  filhos à partir de  $r$  pais,

$$X = \{X_C, X_M\} : \{p_1, p_2, \dots, p_r\} \rightarrow \{f_1, f_2, \dots, f_s\}, \quad (4.5)$$

$\Psi$  o operador de substituição dos  $s$  indivíduos selecionados da geração  $P_t$  pelos  $s$  indivíduos obtidos da aplicação dos operadores  $X_C$  e  $X_M$  na nova geração  $P_{t+1}$ ,

$$P_{t+1} = P_t - \Psi(P_t) + \{f_1, f_2, \dots, f_s\} \quad (4.6)$$

e  $\tau$  o critério de parada (número predefinido de etapas), (Davis ,1971).

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 *Mutação*  $X_M$ ;

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

Determinar a nova geração  $P_t$  por meio de  $\Psi$ ;

**ATÉ** que se aplique algum critério de parada.

### 4.3 Método de Busca da Estrutura de Otimização Genética LQR<sub>Q,R</sub>

Por várias décadas, as matrizes de ponderação foram selecionadas por métodos de tentativa e erro. Um levantamento dos métodos de tentativa e erro para determinação de tais matrizes foi apresentado em (Johnson e Grimble ,1987), porém não de forma automatizada. Uma contribuição com a teoria *LQR*, foi dada por (Neto ,2000), que desenvolveu um método para alocação de autoestrutura em sistemas dinâmicos multivariáveis.

A escolha das matrizes de ponderação para o projeto *LQR* são as variáveis que são usadas para sintonizar os ganhos ótimo do controlador. Alguns aspectos na determinação e importância no contexto do desenvolvimento do projeto LQR é introduzido em (Ferreira ,2003)



O foco principal desta seção, são os métodos de busca das matrizes de ponderação, que são as componentes do índice de desempenho quadrático  $J$ . Esta formulação é adequada para ser resolvida por técnicas de combinação evolutiva, para se determinar o ganho do controlador  $K = R^{-1}B^T P_{LQR}(Q, R)$ . A estrutura de otimização combinatória para busca das matrizes de ponderação é dada por

$$\min_{Q,R} \sum_{i=1}^n p_i s_i(Q, R) \quad (4.7)$$

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 e Patton ,1998) e (Bottura e Fonseca Neto ,1999). Cada par  $(\lambda_i, v_i)$  de autovalores e autovetores, representa um objetivo para o sistema com  $n$ -estados. O problema LQR já foi formulado como uma estrutura de otimização cujo objetivo é determinar a lei de controle  $u(t) = -K_{LQR}x(t)$ , (Neto e Rêgo ,2008) e (Rêgo ,2007).

## 4.4 Conceitos Básicos e Definições

A similaridade entre algoritmo genético e o sistema natural é apresentada para explicar os termos básicos da computação evolutiva usados nesta tese. Essa analogia é feita por meio das matrizes de ponderação  $Q$  de estado e  $R$  de controle.

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  $Q \in \mathbb{R}^{n \times n}$  e  $R \in \mathbb{R}^{m \times m}$ . 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 associado ao problema 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 4.4 ilustra um caso particular das matrizes  $Q$  e  $R$  sendo transformadas em *string*, com  $n = m = 2$ .

$$Q \cup R = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \cup \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \xrightarrow{\text{Transformação}} \begin{array}{|c|c|c|c|c|c|} \hline & \text{cromossomo-}QR & & & & \\ \hline q_{11} & q_{12} & q_{22} & r_{11} & r_{12} & r_{22} \\ \hline \end{array}$$

Figura 4.4: Método 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- $QR$ , de modo que cada indivíduo consiste 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 seqüenciais 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 4.5 ilustra o caminho do ciclo de busca genética. O ciclo de busca genética é inicializado com as matrizes iniciais de ponderação, sendo que após serem efetuadas as operações cromossômicas e aplicada a função de fitness para avaliar o desempenho de cada cromossomo, o AG fornece as matrizes  $Q$  e  $R$  que serão utilizadas para exibir a solução da *EAR*, a matriz de ganho. Em seguida, verifica-se a especificações de projeto, ou seja, se a autoestrutura encontrada está dentro da faixa estabelecida pelo projetista. Uma vez que as especificações de projeto estejam satisfeitas, pontua-se cada autovalor e autovetor, caso contrário, o AG reinicializa todo o processo.

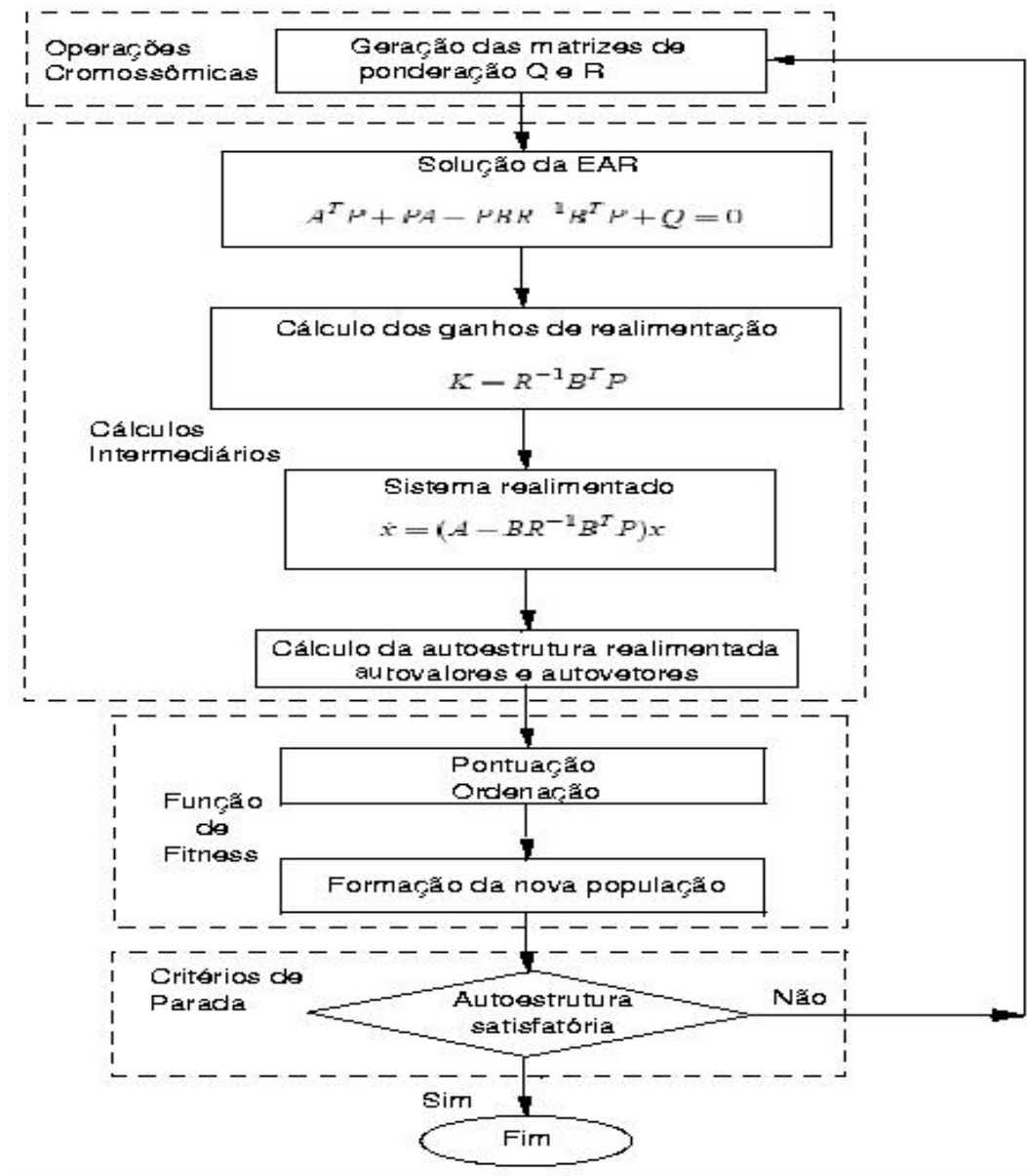


Figura 4.5: Ciclo de Busca Genética.

## 4.5 Modelos Genéticos Artificiais

Os modelos genéticos geram as estratégias que produzem os elementos funcionais que realizam a busca das matrizes de ponderação. Estes elementos estão agrupados de acordo com suas funcionalidades: representações genéticas artificiais das operações cromossômicas, avaliação da função de *fitness* dirigida por mode-

lagem da estrutura de otimização (4.7). Estes elementos estão ordenados para realizar a busca das matrizes  $Q$  e  $R$  usando o AG. Este algoritmo é dado por:

**Algoritmo Genético- $QR$**   $f(s_i^s, \lambda_i^s, Q, R)$

1. AG inicialização;
2. Geração da população inicial;
3. while  $n_{ger} \neq n_{max}$ ;
4.  $AG_{reprod}$ ;
5.  $AG_{x-over}$ ;
6.  $AG_{mut}$ ;
7.  $AG_{cal}$ ;
8.  $AG_{desemp}$ ;
9. Fim do algoritmo.

As entradas do algoritmo genético  $QR$  são especificadas como as sensibilidades  $s_i^s$  e os autovalores  $\lambda_i^s$ . As matrizes  $Q$  e  $R$  são as saídas do procedimento realizado pelo AG.

A inicialização do AG, linha 1, implementa o modelo das matrizes  $Q$  e  $R$  que estabelece a estrutura dos cromossomos do algoritmo genético e o modelo da população  $QR$  que estabelece o tamanho da população. A linha 2, implementa o modelo da população inicial que representa uma estratégia para gerar as matrizes de ponderação simétricas, semi-definida positiva e definida positiva, respectivamente. A linha 3, implementa o critério de parada que é baseado no número de iterações de busca.

As próximas instruções realizam a busca das matrizes até que o critério de parada alcance o número máximo de iterações. A linha 4, implementa a estratégia de seleção de forma aleatória dos modelos de operação genética, que representam a escolha dos indivíduos para participar do cruzamento. As linhas 5 e 6 implementam as operações de crossover e mutação, respectivamente, definidas pelos modelos de operações genéticas, sendo que o *crossover* realiza a combinação de

dois cromossomos individuais e a mutação realiza uma modificação aleatória nos genes dos indivíduos.

A estrutura da função de *fitness* do modelo de avaliação do cromossomo é implementada na linha 7, que representa a solução  $QR$ , a autoestrutura de realimentação do estado e a pontuação individual. A linha 8 implementa um critério de sobrevivência, que consiste na comparação da função de *fitness* entre os descendentes do crossover e os indivíduos cromossômicos sob a operação de mutação. Os melhores indivíduos se juntam na população da geração seguinte. A linha 9 finaliza o algoritmo.

#### 4.5.1 Modelo das matrizes $Q$ e $R$

Inicialmente, considere as matrizes  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$  e  $C \in \mathbb{R}^{p \times n}$  do sistema dinâmico. As dimensões das matrizes do modelo e as especificações do projeto  $LQR$  nos levam a agrupar as matrizes de ponderação  $Q \in \mathbb{R}^{n \times n}$  e  $R \in \mathbb{R}^{m \times m}$ . Estas matrizes devem obedecer as restrições do problema de otimização do  $LQR$ : as matrizes  $Q$  e  $R$  devem ser simétricas, semidefinida positiva e definida positiva, respectivamente.

A característica relativa a simetria dessas matrizes, nos conduzem a um modelo com matrizes triangulares superiores ou inferiores. Os indivíduos  $QR_z$  são modelados em termos de representação dos genes  $Q$  e  $R$ , que possui a seguinte forma:

$$\begin{aligned} QR_z &= \bigcup_{j,i=1}^n q_{ij} \wedge \bigcup_{j,i=1}^m r_{ij}, \quad i \leq j \\ z &= 1, \dots, n_{indiv} \end{aligned} \quad (4.8)$$

sendo  $n$  a dimensão da matriz  $A$ , isto é, a ordem do modelo do sistema dinâmico, e  $m$  (número de entradas) o número de colunas da matriz  $B$ . Os elementos  $q_{ij}$  e  $r_{ij}$  representam os genes da matriz  $QR_z$ ; e  $n_{indiv}$  é o número de cromossomos individuais de uma certa população.

### 4.5.2 Modelo da População $QR$

Considerando que o cromossomo  $QR_z$ , Eq. (4.8), é um elemento composto por  $g$  genes, o qual representa  $Q \in \mathbb{R}^{n \times n}$  e  $R \in \mathbb{R}^{m \times m}$ , as matrizes de estado e de controle, respectivamente, 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

$$QR_{n_{indiv} \times g} = \left[ QR_1; QR_2; QR_3; \dots; QR_{n_{indiv}} \right]. \quad (4.9)$$

A notação  $n_{indiv} \times g$  indica a quantidade de indivíduos do cromossômicos de uma população e a quantidade ( $g$ ) 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,

$$qr_z^w = \begin{cases} q_{i,j} & , w < n_Q + 1, \quad i, j = 1, \dots, n_Q \\ r_{i,j} & , w > n_Q, \quad i, j = 1, \dots, n_R \end{cases} \quad z = 1, \dots, n_{indiv} \quad (4.10)$$

sendo  $n_Q$  e  $n_R$  as quantidades dos elementos das matrizes simétricas  $Q$  e  $R$ , respectivamente, o parâmetro  $w$  representa a posição do alelo no cromossomo  $QR$  e  $n_{indiv}$  é o número de indivíduos.

### 4.5.3 Modelo da População Inicial

O modelo de geração aleatório das matrizes iniciais  $Q$  e  $R$  é dado por

$$q_{i,j} = \begin{cases} p_{Q\alpha} + p_{Q\beta} \kappa_{Qii} & i = j \\ p_{Q\gamma} \kappa_{Qij} & i \neq j \end{cases} \quad i, j = 1, \dots, n \quad (4.11)$$

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  $\kappa_{Qii}$  é uma variável pseudo 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 diagonal e o parâmetro  $\kappa_{Qij}$  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,

$$r_{i,j} = \begin{cases} p_{R\alpha} + p_{R\beta}\kappa_{Rii} & i = j \\ p_{R\gamma}\kappa_{Rij} & i \neq j \end{cases} \quad i, j = 1, \dots, n \quad (4.12)$$

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  $R$  que são:  $Q \geq 0$  e  $R > 0 \forall q_{ij}$  e  $r_{ij}$ .

#### 4.5.4 Modelo de Avaliação dos Cromossomos

A avaliação da função de *fitness* é realizada pontuando cada indivíduo  $QR_z$  da população  $QR_{n_{indiv}}$  em relação a alocação de autoestrutura e sensibilidades dos autovalores, de acordo com as métricas da função de *fitness* das Eqs. (4.7-4.8). A avaliação cromossômica é um elemento funcional do AG que é dado por

$$K_z = LQR_z(A, B, Q_z, R_z) \quad (4.13)$$

$$A_z = (A - BK_z) \quad (4.14)$$

$$W_z \lambda_z V_z = A_z \quad (4.15)$$

$$S_z = \frac{\|V_z\|^2 \|W_z\|^2}{\langle V_z, W_z \rangle} \quad (4.16)$$

$$F_{S_z} = \sum S_z \quad (4.17)$$

$$R_{S_z} = rank(S_z, F_{S_z}) \quad (4.18)$$

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

sendo  $K_z$  o ganho atribuído por um indivíduo  $QR_z$ .  $A_z$  é a matriz de malha fechada para o ganho  $K_z$ .  $\lambda_z$ ,  $V_z$  e  $W_z$  são os autovalores e autovetores à direita e esquerda, respectivamente.  $S_z$  são as sensibilidades associadas que são atribuídas pela autoestrutura,  $\langle . \rangle$  e  $\|.\|^2$  são o produto escalar e a norma 2 dos autovetores  $V_z$  e  $W_z$ , respectivamente.  $F_{S_z}$  é o índice de desempenho designado aos indivíduos e finalmente  $R_{S_z}$  é o índice de pontuação atribuído ao indivíduo  $QR_z$ .

Os passos a seguir, explicam as ações do modelo da função de *fitness*:



1. O ganho ótimo é dado pela Eq. (4.13);
2. A autoestrutura de realimentação de estado é fornecida pelas Eqs. (4.14-4.15);
3. As sensibilidades e o índice de desempenho são dados pelas Eqs. (4.16-4.17);
4. O *rank* do indivíduo  $QR_z$  é dado pela Eq. (4.16)

### 4.5.5 Modelos de Operações Genéticas

As operações genéticas são representadas por três modelos. O primeiro modelo está direcionado para a seleção de indivíduos, e os outros modelos lidam com operadores genéticos de cruzamento (*crossover*) e mutação.

#### Modelo de Seleção

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

$$\kappa_z = Fit_{pop}\kappa_{rand} \quad (4.19)$$

$$QR_{select} = QR_j, \quad \max_j \kappa_z < \sum_{j=1}^{n_{indiv}} Fit_j, \quad (4.20)$$

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

sendo  $\kappa_z$  escolhido aleatoriamente para o  $z$ -ésimo indivíduo que é baseado no número pseudo-aleatório  $\kappa_{rand}$ ,  $Fit_{pop}$  é o desempenho populacional total e  $QR_{select}$  é o  $j$ -ésimo indivíduo selecionado.

A seleção dos indivíduos é feita comparando o valor da função de *fitness* da roleta  $k_z$  que é associado com a função de *fitness* da população e é escolhido aleatoriamente pela Eq. (4.19). A expressão  $Fit_{par} = \sum_{j=1}^{n_{indiv}} Fit_j$  representa a função de *fitness* parcial da Eq. (4.20) que é realizada até a condição de seleção

do indivíduo  $Fit_{par}(j) \geq k_z$  tornar-se verdadeira. Conseqüentemente, o indivíduo  $QR_j$  da Eq. (4.20) é selecionado para o cruzamento, de acordo com o valor aleatório da Eq. (4.19).

### Modelo de Seleção Elitista

O modelo de seleção elitista normalmente é acoplado a outros métodos de seleção, na tentativa de 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:

$$QR_{G+1,super} = \begin{cases} QR_{G,melhor} \Leftrightarrow (\sum s_i(QR_{G,melhor}) < \sum s_i(QR_{G,super}) \text{ e} \\ QR_{G,melhor} \in \Omega_f) \text{ ou } (QR_{G,melhor} \in \Omega_f \\ \text{e } QR_{G,super} \notin \Omega_f \\ QR_{G,super} \Leftrightarrow (\sum s_i(QR_{G,melhor}) \geq \sum s_i(QR_{G,super}) \text{ e} \\ QR_{G,super} \in \Omega_f) \text{ ou } (QR_{G,melhor} \notin \Omega_f), \end{cases} \quad (4.21)$$

sendo  $\Omega_f = \{QR_z; \lambda_{ei} \leq \lambda_{ic}(QR_z) \leq \lambda_{di} \text{ e } s_i(QR_z) \leq 1\}$  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 responsável pela combinação de dois indivíduos no AG. Neste estágio, os indivíduos  $QR_{l_1}$  e  $QR_{l_2}$  da população  $QR_{crom \times g}$  trocam informações genética. O modelo para esta operação considera os  $l_1$  e  $l_2$  ésimos indivíduos de uma dada população  $G$ ,  $l_1 \neq l_2$ ,

$$QR_{G+1,l_1} \leftarrow \alpha(QR_{G,l_1}) + (1 - \alpha)(QR_{G,l_2}) \quad (4.22)$$

$$QR_{G+1,l_2} \leftarrow \alpha(QR_{G,l_2}) + (1 - \alpha)(QR_{G,l_1}) \quad (4.23)$$

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  $\alpha$  é um número aleatório uniformemente distribuído entre 0 e 1 sob o ponto de vista de combinação convexa.

### Modelo de Mutação

A mutação é um elemento essencial neste método de busca, de modo que, 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,

$$q_{ij}^{novo} = q_{ij}^l b^{x_{local}} \quad (4.24)$$

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

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

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, Eq.(4.24).

## 4.6 Metodologia Neuronal para Solução EAR-LQR

Apresentamos as condições de otimalidade no Apêndice B com as quais podemos identificar um ponto ótimo aplicando as condições de otimalidade de primeira

ordem (Princípio Mínimo de *Pontryagin*) e de segunda ordem. Iremos agora estudar o algoritmo que, partindo de um ponto inicial  $(P, L)^0$ , nos leve a um ponto  $(P, L)^*$  de mínimo local da função em estudo, no caso, a função energia que é o erro médio quadrático.

A programação não-linear (PNL) caracteriza-se por não possuir um método geral de resolução dos seus problemas. São diversos algoritmos e quase sempre voltados para problemas específicos, focalizados em as características diversas tais como continuidade, unimodalidade, diferenciabilidade de primeira e / ou segunda ordem, etc.

Uma outra característica é a inexistência de critérios absolutos para comparação entre os vários algoritmos existentes. O que se faz é comparar algumas características não definidas precisamente, tais como: simplicidade computacional, tempo de máquina necessário para atingir um ponto de mínimo a partir de um ponto inicial qualquer, memória necessária, rapidez de convergência, sensibilidade a erros computacionais, etc. A rapidez de convergência está ligada a fatores tais como o número de iterações necessárias para atingir um ponto desejável. Existem algoritmos que garantem um número finito de iterações, enquanto em outros a terminação finita nem sempre pode ser obtida. Pode-se associar a cada algoritmo uma razão de convergência, um parâmetro característico do algoritmo no limite de convergência. Maiores detalhes podem ser obtido em (Luenberger, 1984). Neste capítulo nos propomos à resolução de Problema Irrestrito (PI), porém, ressaltamos desde já que muitos dos algoritmos aplicados poderão ser utilizados na resolução de Problemas Restritos (PR), após transformá-los em PI.

Desenvolveremos o algoritmo para o PI em minimização

$$\begin{aligned} \min E(P, L) & \qquad (4.25) \\ (P, L) & \in \mathbb{R}^{n \times n} \end{aligned}$$

porém, todos são aplicáveis à casos de maximização. Os métodos, para resolução deste problema, denominados algoritmos de minimização irrestrita, possuem uma estrutura comum, descrita a seguir: Inicialização, processo iterativo, escolha da direção e executar a busca. O ponto inicial  $(P, L)^0$  deve ser convenientemente escolhido, pois influência no número de iterações e na convergência do algoritmo,

que no nosso caso são matrizes identidades. Muitas vezes é bom usar conhecimento a priori sobre o comportamento da função. A função energia é quadrática e convexa.

A condição de parada do processo iterativo consiste em verificar se  $(P, L)^k$  é um ponto desejável, ou ainda se  $(P, L)^k = (P, L)^*$ , sendo  $k$  um contador de iterações. Se  $(P, L)^*$  é um ponto de mínimo local, então basta verificar se  $\nabla E[(P, L)^*] = 0$ . Muitas vezes não estamos interessados em um mínimo local exatamente, sendo desejável apenas um ponto em que a função energia "decreceu" bastante. Outras vezes, por razões de erros computacionais, não podemos atingir o zero absoluto que é o desejável.

A escolha de direções é o que realmente diferencia os algoritmos de minimização irrestrita: a definição de uma regra fixa para determinar uma direção de busca. A direção é fornecida pelas taxas de variações associadas às camadas  $P$  e  $L$ , sendo  $P$  um parâmetro matricial cujas entradas são reais, simétrica e definida positiva por imposição do projeto  $LQR$ , que representa a saída da RNR enquanto que  $L$  representa o Fator de *Cholesky*.

As equações dinâmicas de todos os algoritmos executam um processo de minimização na direção determinada. Este processo de busca de um ponto desejável naquela direção denomina-se Busca Unidirecional. O ponto  $(P, L)^{k+1}$  deve ser tal que  $E(P, L)^{k+1} < E(P, L)^k$ , uma vez que estamos minimizando o erro médio quadrático.

Após uma descrição desta estrutura algorítmica, podemos verificar que o problema de minimizar uma função de  $n$  variáveis se reduz a minimização de uma seqüência de problemas a uma variável. No processo convencional de busca, a cada iteração, determina-se um passo  $\lambda$  (taxa de aprendizagem); na abordagem proposta, esses parâmetros de aprendizagens  $\eta_v$  e  $\eta_z$  são fixos de modo que  $\eta_v < \eta_z$  uma vez que  $(P, L)^{k+1} = (P, L)^k + (\eta_v, \eta_z)(dP, dL)$  seja um ponto desejável na direção  $(dP, dL)$  e  $E(P, L)^{k+1} < E(P, L)^k$ . Esta seqüência deverá convergir para o ponto  $(P, L)^*$  após a escolha correta da direção.

Existem diversos métodos utilizados na resolução de PNL (minimização) irrestrito (4.25). Dentre eles podemos destacar os métodos do gradiente, *Newton*, direções conjugadas e *Quase-Newton*. Para estes métodos devemos ter que a função objetivo  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  seja continuamente diferenciável  $f \in C^1$ , utilizando

derivadas de primeira ordem. Alguns desses métodos usam derivadas parciais de segunda ordem, exigindo também que  $f \in C^2$ . Todos caracterizam-se por usarem derivadas, ou aproximações, para determinar direções de pesquisa.

Dentro desta estrutura geral, todos os métodos foram particularizados. Ressalta-se que o método, de um modo geral, foi desenvolvido para resolver um problema de programação não linear (PPNL) cuja função objetivo é quadrática e que a convergência do método de um modo geral, está intrinsicamente relacionada ao fato que a função objetivo (função energia) é convexa. Porém supõe-se que as seqüências de pontos gerados (P e L), solução da equação algébrica de *Riccati* e o Fator de *Cholesky*, respectivamente são convergentes ou possuem subsequências convergentes para um ponto estacionário. Porém a extensão para os casos não-quadráticos, que é imediata, é apresentado em (Bazaraa *et al.*, 1993).

O método do Gradiente é também conhecido por método de *Cauchy* ou de Máximo Declive (*Steepest Descent*). É um dos métodos mais antigos e conhecidos para minimização de uma função de  $n$  variáveis. Devido a sua simplicidade ainda é bastante aplicado. Pode convergir muito lentamente. Ainda assim é importante, devido à sua base teórica e à sua facilidade de análise. Vale ainda ressaltar que muitos dos métodos com melhores condições de convergência são resultantes de modificações no método de gradiente.

O método utiliza somente derivadas parciais de primeira ordem de  $E(P, L)$ . Como o gradiente aponta na direção na direção de maior crescimento da função no ponto, o método procura em cada ponto caminhar na direção oposta. Portanto, a direção de busca é a direção oposta ao gradiente,  $-\nabla E(P, L)$ .

A partir de  $(P, L)^0$  procuramos um ponto estacionário  $(P, L)^k$  tal que

$$\nabla E[(P, L)^k] = 0.$$

Esta é uma condição necessária para identificação de um ponto de mínimo local já estabelecido. Porém, sabe-se que em um ponto de sela esta condição é atendida. Portanto, a importância do conhecimento do comportamento da função objetivo. Faz-se necessário verificar na condição de parada em que ponto estamos. A função energia é quadrática e convexa e o critério de parada foi baseado no número de iterações.

O reconhecimento do ponto de mínimo pode ser feito por meio da matriz hessiana, se disponível. Em se tratando de um ponto de sela é necessária a utilização de

um método não gradiente para se mover deste ponto e prosseguir a minimização. O objetivo principal em apresentar no início do capítulo uma breve introdução a respeito da estrutura do algoritmo e métodos de otimização (minimização) irrestrito, em particular o de Máximo Declive (Steepest Descent) é estabelecer uma analogia do método com o treinamento da rede neural recorrente.

Os problemas de otimização surgem em uma larga variedade de aplicações científicas e de engenharia, inclusive em processamento de sinais, identificação de sistemas, projeto de filtro, aproximação de função, análise de regressão e assim por diante. Em diversos problemas práticos de otimização tais como o planejamento de sistemas de potência e sistemas de telecomunicações, o número de variáveis de decisão e restrições são geralmente muito grande.

É desafiador quando um procedimento de otimização de larga escala tem que ser realizado em tempo real para otimizar o desempenho do sistema dinâmico. Para tais aplicações, as técnicas de otimização clássicas não tem competência devido a dimensionalidade do problema e requisitos estritos em tempo computacional. Uma abordagem possível e desafiadora para otimização em tempo real é a aplicação de redes neurais artificiais.

Uma das equações matriciais não lineares estudadas com maior intensidade por matemáticos e engenheiros, é a Equação de *Riccati*. O termo genérico "Equação de *Riccati*" pode significar qualquer classe de matrizes. Por exemplo, quadráticas, algébrica, diferencial, diferenças finitas do tipo simétrica ou não simétrica, surgidas no estudo de sistemas dinâmicos contínuos ou discretos no tempo, (Arnold e Laub ,1984). Apresenta-se neste Capítulo, uma solução neuronal para a Equação Algébrica de *Riccati* (*EAR*). Sua resolução é de valiosa importância uma vez que seu cálculo surge geralmente na análise e síntese de sistemas de controle ótimo linear quadrático (*LQR*), (Wuang e Wu ,1998), sendo utilizada pelos mais diversos setores da indústria, ciência e tecnologia. As referências seguintes mostram a importância industrial do *LQR* (Lee ,1996) e a aplicação em sistemas embarcados para implementação deste controlador (Paschall ,1994).

O problema de projetar um sistema de controle de realimentação linear que minimiza um índice de desempenho quadrático, pode ser reduzido ao problema de obter uma solução simétrica e definida positiva da equação matricial de *Riccati*. Uma variedade de algoritmos numéricos têm sido desenvolvidos para solucionar a

*EAR* (Gardiner e Laub ,1991a). Existem poucos relatórios sobre soluções neurais para equações de *Riccati*.

## 4.7 Caracterização da Equação Algébrica de Riccati

A necessidade para resolver a equação algébrica de *Riccati* geralmente surge na análise e síntese em sistemas de controle ótimo linear quadrático. Exibe-se o modelo da *RNR*, as abordagens para converter uma estrutura de otimização restrita em um problema de otimização irrestrita. Faz-se necessário salientar a importância da garantia da solução da *EAR* ser simétrica e definida positiva, e para isso, é necessário usar a fatoração de *Cholesky*, visto que a *EAR* possui soluções múltiplas, e então uma restrição adicional é incluída para se obter esta solução simétrica e definida positiva para equação algébrica de *Riccati*.

Apresenta-se um sistema dinâmico no espaço de estado, para o problema do regulador do valor assumido na saída do sistema de modo que o sinal de erro gerado pela diferença entre o valor de saída e o desejado seja mínimo.

Considere um sistema de equações lineares de estado, completamente observável e sem ruído, dado pelo Problema de Valor Inicial (PVI):

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad (4.26)$$

sendo  $x(t) \in \mathbb{R}^n$  o vetor de estado,  $u(t) \in \mathbb{R}^m$  é o vetor de controle,  $A \in \mathbb{R}^{n \times n}$  e  $B \in \mathbb{R}^{n \times m}$  são as matrizes associadas aos vetores de estado  $x(t)$  e ao vetor de entrada respectivamente,  $x_0$  representa o vetor de estado inicial dado de modo que  $m \leq n$ . Se todas as variáveis de estado são mensuráveis, pode-se então determinar uma lei de controle de realimentação de estado linear descrita pela relação  $u(t) = -Kx(t)$  que é aplicada ao sistema dado pela Eq. (4.26). Dessa maneira, o sistema toma a seguinte forma:

$$\dot{x}(t) = Ax(t) + Bu(t) = Ax(t) - BKx(t) = (A - BK)x(t), \quad x(0) = x_0 \quad (4.27)$$

sendo  $K \in \mathbb{R}^{m \times n}$  a matriz de ganho de realimentação de estado.



A fim de minimizar-se os sinais de controle e estado do sistema de controle de realimentação, de modo geral, minimiza-se um índice de desempenho quadrático, Eq. (B.7).

Quando as matrizes  $A$  e  $B$  são constantes, a matriz de ganho ótimo do sistema realimentado pode ser deduzida a partir da expressão  $K^* = R^{-1}B^T P \in \mathbb{R}^{m \times n}$ , sendo  $P \in \mathbb{R}^{n \times n}$  a solução simétrica e definida positiva da equação algébrica de *Riccati* contínua no tempo, Eq. (3.5) de maneira que as matrizes  $A$  e  $B$  são as matrizes do sistema dinâmico (Wuang e Wu ,1998).

## 4.8 Formulação da Solução da Equação de *Riccati*

A seguir apresenta-se a formulação do problema a ser resolvido para a solução da equação de *Riccati*, Eq. (3.5). Pelo fato desta equação possuir múltiplas soluções, uma restrição adicional será incluída para se obter uma única solução  $P$  simétrica e definida positiva. Uma vez que a solução da *EAR* é real, simétrica e definida positiva, pode-se escrevê-la na forma fatorada de *Cholesky*, ou seja,  $LL^T = P$  sendo  $L \in \mathbb{R}^{n \times n}$  uma matriz triangular inferior com elementos diagonais positivos (Fator de *Cholesky*).

Definindo-se  $\tilde{P}$  e  $\Lambda$  como blocos de matrizes, reescreve-se a *EAR* na seguinte forma:

$$\tilde{P}\Lambda\tilde{P}^T = Q + I, \quad (4.28)$$

sendo  $I$  a matriz identidade,  $\tilde{P} \triangleq (P, I)$  e

$$\Lambda \triangleq \begin{bmatrix} S & -A \\ -A^T & I \end{bmatrix}.$$

Uma vez que  $Q$  é simétrica e definida positiva (ou semidefinida positiva),  $Q + I$  é sempre simétrica e definida positiva, e também a matriz  $\Lambda$  é simétrica. Desta forma, o problema de síntese do *LQR* é reformulado por meio da solução das Eqs. matriciais (4.29-4.30)

$$\tilde{P}\Lambda\tilde{P}^T = Q + I \quad (4.29)$$

e

$$LL^T = P. \quad (4.30)$$

Estas equações possuem a mesma dimensão.

## 4.9 Arquitetura da Rede Neuronal Recorrente

A resolução da equação algébrica de *Riccati* do controlador LQR pode ser considerada como um problema de otimização irrestrito. Dadas as matrizes  $A$ ,  $B$ ,  $Q$  e  $R$ , deseja-se obter as matrizes  $P$  e  $L$  associado ao problema de programação não-linear, em que almeja-se minimizar a função energia dada a seguir:

$$\min_{P,L} \xi(PSP - A^T P - PA - Q + LL^T - P). \quad (4.31)$$

Observe que  $S = BR^{-1}B^T$ ,  $e_{ij} : \mathbb{R} \rightarrow \mathbb{R}$  é uma função energia, convexa e limitada inferiormente que verifica a soma algébrica da *EAR* com o Fator de *Cholesky* sendo expressa na forma escalar por

$$\left[ \sum_{k=1}^n \sum_{l=1}^n p_{ik} s_{kl} p_{lj} - \sum_{k=1}^n (a_{ki} p_{kj} + p_{ik} a_{kj}) - q_{ij} \right] + \left[ \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk} - p_{ij} \right]. \quad (4.32)$$

O sistema dinâmico de equações não-lineares da rede neural recorrente para resolver a equação de *Riccati* é descrito a seguir:

Para  $i, j = 1, \dots, n$ ;

$$\begin{aligned} \frac{dp_{ij}(t)}{dt} = & -\eta_v \left\{ \sum_{k=1}^n \sum_{l=1}^n [p_{ik}(t) s_{kl} u_{lj}(t) + u_{ik}(t) s_{kl} p_{lj}(t)] \right. \\ & \left. - \sum_{k=1}^n [a_{ik} u_{kj}(t) + u_{ik}(t) a_{jk}] - y_{ij}(t) \right\} \end{aligned} \quad (4.33)$$

$$\frac{dl_{ij}(t)}{dt} = -\eta_z \sum_{k=j}^n y_{ik}(t) l_{kj}(t), \quad i \geq j \quad (4.34)$$

$$u_{ij}(t) = f_{ij} \left[ \sum_{k=1}^n \sum_{l=1}^n p_{ik}(t) s_{kl} p_{lj}(t) - \sum_{k=1}^n [a_{ki} p_{kj}(t) + p_{ik}(t) a_{kj}] + q_{ij} \right] \quad (4.35)$$

$$y_{ij}(t) = f_{ij} \left[ \sum_{k=1}^{\min\{i,j\}} l_{ik}(t) l_{jk}(t) - p_{ij}(t) \right]. \quad (4.36)$$

Escreve-se ainda as Eqs. (4.33-4.36) na forma matricial segundo as Eqs. (4.37-4.40),

$$\frac{dP(t)}{dt} = -\eta_v [P(t)SU(t) + U(t)SP(t) - AU(t) - U(t)A^T - Y(t)] \quad (4.37)$$

$$\frac{dL(t)}{dt} = -\eta_z Y(t)L(t) \quad (4.38)$$

$$U(t) = F[P(t)SP(t) - A^T P(t) - P(t)A + Q] \quad (4.39)$$

$$Y(t) = F[L(t)L(t)^T - P(t)], \quad (4.40)$$

sendo  $U(t) = [u_{ij}(t)] \in \mathbb{R}^{n \times n}$ ,  $P(t) = [p_{ij}(t)] \in \mathbb{R}^{n \times n}$  e  $Y(t) = [y_{ij}(t)] \in \mathbb{R}^{n \times n}$  matrizes dos estados de ativação,  $L(t) = [l_{ij}(t)] \in \mathbb{R}^{n \times n}$  é uma matriz triangular inferior dos estados de ativação,  $P(0) = P(0)^T$ ,  $P(0) \neq 0$ ,  $\eta_v$  e  $\eta_z$  são parâmetros positivos do projeto. Uma vez que  $F = f_{ij}$  é uma matriz simétrica dos estados de ativação, as matrizes  $P(t)$ ,  $U(t)$  e  $Y(t)$  também serão simétricas. A matriz  $S$  apresentada na Eq. (4.31) representa a influência da matriz de ponderação  $R$  na forma  $S = BR^{-1}B^T$ , considerando que os ganhos do controlador ótimo são ajustados com base nas variações dos parâmetros matriciais  $Q$  e  $R$ . O método de

*Runge-Kutta* é usado para computar os valores das camadas de saída  $P$  (solução da EAR) e oculta  $L$  (Fator de *Cholesky*), uma vez que este método é um dos mais populares e também precisos para obter soluções aproximadas de problemas de valor inicial. Essas condições de simetria são utilizadas para a redução do número de neurônios usados para a construção da rede neural artificial.

A função de saturação usada pelos elementos de processamento, neurônios, é a função de saturação linear, Fig. 4.6, que é dada por

$$f(x) = \begin{cases} -1, & \text{se } x \leq -1 \\ x, & \text{se } -1 < x < 1 \\ 1, & \text{se } x \geq 1 \end{cases}$$

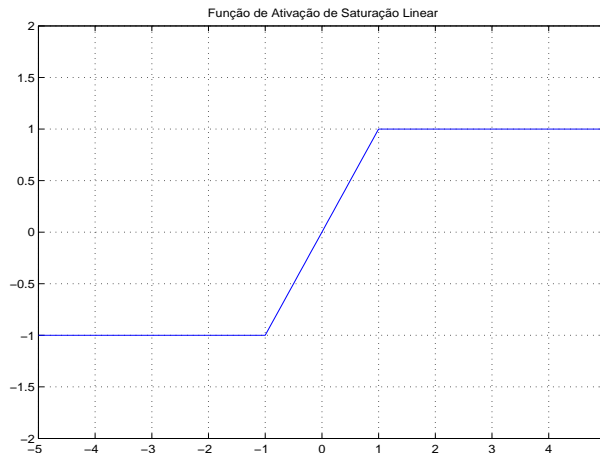


Figura 4.6: Função de ativação linear simétrica

A função de desempenho para mensurar o processo de treinamento da RNR é o EMQ (Erro Médio Quadrático), e esta função representa a função energia. O erro médio quadrático  $EMQ_{RNR}$  de aprendizagem da RNR é dado por

$$EMQ_{RNR} = \frac{1}{2} \|e_{EAR} + e_{FC}\|^2 \tag{4.41}$$

sendo  $e_{EAR}$  o erro associado a equação algébrica de *Riccati*  $A^T P + PA - PSP + Q$  e  $e_{FC}$  é o erro associado ao Fator de *Cholesky* que é dado por  $LL^T - P$ .

Conforme mostra a Fig. 4.3, a arquitetura da rede neuronal artificial recorrente para resolução da equação algébrica de *Riccati*, é formada por quatro camadas conectadas bidirecionalmente de modo que a camada de saída é representada por  $P(t)$ , uma camada de entrada  $U(t)$ , e duas ocultas,  $Y(t)$  e  $L(t)$  que realizam a interconexão entre as camadas de saída  $P(t)$  e de entrada  $U(t)$ . Segundo comentamos, o treinamento da rede neuronal pode ser feito por meio de um critério de parada do algoritmo ou até mesmo por estipular um número de iterações. Para esta arquitetura, utiliza-se o treinamento não supervisionado, ou seja, estabelece-se um número de iterações e à medida que o treinamento ocorre, são apresentadas os valores da energia, a norma do infinito da solução da EAR e o Fator de *Cholesky*, pois desta maneira pode-se verificar os pontos usados para a construção dos gráficos, e então verificar a convergência das três seqüências obtidas.

Uma vez que a solução é uma matriz, especifica-se uma métrica, pois desta forma podemos comparar as diversas soluções que o algoritmo retorna a cada iteração. Dentre essas métricas, usamos a norma do máximo e a do infinito para verificar o comportamento do problema específico.

As camadas  $P(t) \in \mathbb{R}^{n \times n}$ ,  $U(t) \in \mathbb{R}^{n \times n}$  e  $Y(t) \in \mathbb{R}^{n \times n}$  são matrizes quadradas de neurônios e  $L(t)$  é uma matriz triangular. A matriz  $P$  de estado de ativação representa a solução da equação de *Riccati*, Eq. (3.5), enquanto que a matriz  $L$  designa o fator de *Cholesky*.

Resultados anteriores citados em artigos científicos de (Wang e Guang. ,1993) sugerem que quaisquer funções de ativação  $F$  não-decrescentes podem ser usadas. Como exemplo temos: função degrau, linear, quadrática e trigonométricas. A escolha de uma função de ativação contínua, diferenciável por partes e monótona não-decrescente possui uma relação direta com a função energia convexa, explícita pela relação  $\frac{de_{ij}(\xi)}{d\xi} = f_{ij}(\xi)$ . Observe que a função  $e_{ij}$  é convexa e limitada inferiormente, porém não necessariamente alcança o zero.

## 4.10 Treinamento da Rede Neural Recorrente

O processo de treinamento da rede neural direcionado para o sistema dinâmico não-linear tem por objetivo, determinar a solução da EAR ( $P$ ) e do fator de *Cholesky* ( $L$ ) do problema de otimização irrestrito. Inicializa-se o processo com

os parâmetros matriciais  $A$ ,  $B$ ,  $Q$  e  $R$  que são fixos e os demais parâmetros, são variáveis, ou seja, são fornecidas pelas camadas  $P$  e  $L$ . O treinamento tem por fim atualizar as camadas  $P$  e  $L$  a partir de uma condição inicial  $P^0$  e  $L^0$  em algum ponto da superfície da função energia (quadrática e convexa). Para definir a direção de treinamento da rede neural recorrente usa-se, o Gradiente Descendente, que é fornecido pelas Eqs. (4.37-4.38). As taxas de aprendizagens são fornecidas pelos parâmetros  $(\eta_z, \eta_v)$ , que são responsáveis pela velocidade de convergência e também em refinamento em torno da solução, logo que a diferença entre duas interações subsequentes seja pequena.

A Fig. 4.7 mostra a influência de cada camada no treinamento da RNR. Exige-se que a camada oculta  $L$ , Fator de *Cholesky*, tem que convergir mais rápido que a camada de saída  $P$ , solução da EAR, ou seja, a camada  $L$  tem que chegar primeiro em  $Y$  do que  $P$ , pois senão a camada  $L$  não será atualizada e a camada  $Y$ , responsável por uma ponderação na camada de saída  $P$  poderá convergir para uma solução instável devido a Eq. (4.40). Por isso, justifica-se o fato que a camada  $L$  ter que convergir mais rápido que a camada  $P$  tomando  $\eta_z \geq \eta_v$ , a fim de garantir que no estado estacionário, a matriz  $P$  seja definida positiva.

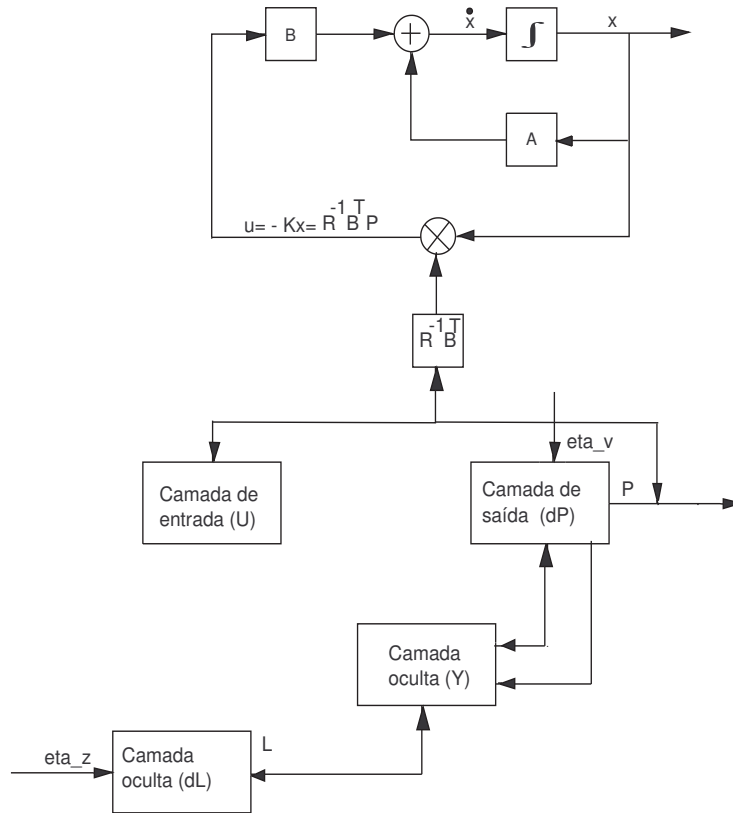


Figura 4.7: Diagrama de blocos do processo dinâmico do projeto LQR

A RNR possui integrais para calcular os valores dessas camadas e, para estimá-las utiliza-se o método de *Runge-Kutta*. Em seguida, ocorre o processo de atualização das camadas de entrada  $U$  e oculta  $Y$  respectivamente, por meio da função de ativação linear  $f_{ij}$ .

Uma vez que o método de *Runge-Kutta* retorna as camadas  $P$  e  $L$ , a cada iteração gera-se uma nova solução  $(P, L)$  e obtém-se uma nova energia fornecida pela Eq. (4.31). No processo de treinamento a função energia gera seqüências de matrizes a cada iteração, de modo que elas decrescem a cada iteração conforme observa-se nas figuras 5.14, 5.15 e 5.16 referentes a função energia, norma do infinito associada a EAR e norma do infinito associada ao fator de *Cholesky*, respectivamente.

A aplicação do algoritmo é feita de modo a minimizar a função energia a um

valor pré-determinado ou estipular um número fixo para o laço de atualizações das camadas de modo a obter um resultado satisfatório. Com já dissemos, o treinamento da RNR é não supervisionado, ou seja, é baseado em um número fixo de iterações.

Esses parâmetros de projeto devem ser especificados, e existe uma séria problemática entre velocidade de convergência e exatidão na solução. Existem algumas heurísticas sobre quais os valores iniciais para a escolha desses parâmetros, isto é, se inicialmente se escolheria um passo fixo ou um passo variável, com o intuito de melhorar o desempenho. Independente da heurística utilizada é necessário que os valores dos parâmetros  $(\eta_z, \eta_v)$  sejam suficientemente grandes para garantir a convergência da rede em regime permanente. Além do mais, devemos garantir a positividade da solução em regime permanente, deste modo a camada oculta  $L$  tem que convergir mais rápido que a camada  $P$ , Fig. 4.7, pois a mesma fornece informação usada no cálculo da camada de saída  $P$ , a fim de garantir a restrição acima citada. Portanto, devemos escolher os valores de  $(\eta_z, \eta_v)$  de modo que  $\eta_z \geq \eta_v$ .

Visto que agora o problema se resume a uma estrutura de otimização irrestrita, a rede neuronal é utilizada para realizar a busca de uma trajetória que convirja para um estado estável de energia mínima baseado no sentido inverso do gradiente, conforme suas equações dinâmicas.

A matriz de estado gerada pela camada  $P$  representa o resultado computacional e a matriz de estado gerada pela camada oculta  $L$  representa a solução do Fator de *Cholesky*. Uma abordagem sobre a construção de redes neurais para problemas de otimização é encontrada em (Xia e Wuang, 2001).

Especificamente, quando os parâmetros  $(\eta_v, \eta_z)$  assumem valores grandes, mais rápida é a convergência. Portanto, a convergência do processo de computação neural está relacionada à seleção de um par de parâmetros com valores suficientemente grandes.

## 4.11 Estabilidade e Solvabilidade

Nesta seção discute-se os resultados referentes à análise de estabilidade da RNR utilizando o método direto de *Lyapunov* e uma condição necessária e suficiente



para convergência global da *EAR*. A rede neural recorrente é um sistema dinâmico não linear.

**Teorema 4.11.1.** *Se toda função de ativação é contínua, pelo menos diferenciável por partes e monótona não-decrescente, isto é,  $\frac{df_{ij}(\xi)}{d\xi} \geq 0$ , para  $-\infty < \xi < +\infty$ , e  $f_{ij}(\xi) = 0$  se  $\xi = 0$ , ( $i, j = 1, \dots, n$ ), então a rede neural é assintoticamente estável ou seja,  $\forall P(0), L(0), \exists \bar{L}, \bar{P}$  tal que  $\lim_{t \rightarrow \infty} L(t) = \bar{L}$  e  $\lim_{t \rightarrow \infty} P(t) = \bar{P}$ , (Wuang e Wu, 1998).*

*Prova:* Seja  $f_{ij}(\xi) \triangleq \frac{de_{ij}(\xi)}{d\xi}$ . Desde que  $f_{ij}$  seja contínua, pelo menos diferenciável e  $\frac{df_{ij}(\xi)}{d\xi} \geq 0$ ,  $e_{ij}(\xi) > -\infty$ , ou seja,  $e_{ij}$  é limitada inferiormente e  $\frac{d^2e_{ij}(\xi)}{d\xi^2} \geq 0$ , (conseqüentemente  $e_{ij}$  é convexa) para  $-\infty < \xi < +\infty$  e  $i, j = 1, \dots, n$ . Uma vez que  $\xi$  é definida como a soma de  $e_{ij}$  convexa e limitada inferiormente,  $\xi$  também é convexa e limitada inferiormente. Além disso,  $\xi$  é convexa e limitada inferiormente,  $\xi \rightarrow \infty$  quando  $\|P\| \rightarrow \infty$  ou  $\|L\| \rightarrow \infty$ , ou seja,  $\xi$  é ilimitada radialmente. De acordo com os resultados: Eq. (4.32),  $s_{ij} = s_{ji}$ ,  $y_{ij}(t) = y_{ji}$ , simetria das matrizes  $R$  e  $F$ , para  $i, j = 1, \dots, n$  e supondo-se que

$$g_{ij}(P) = \sum_{k=1}^n \sum_{l=1}^n p_{ik} s_{kl} p_{lj} - \sum_{k=1}^n (a_{ki} p_{kj} + p_{ik} a_{kj}) - q_{ij}$$

pode-se escrever,

$$\begin{aligned} \sum_{k=1}^n \sum_{l=1}^n u_{kl}(t) \frac{\partial g_{kl}}{\partial p_{ij}} &= \sum_{k=1}^n \left\{ \sum_{l=1}^n [p_{ij}(t) s_{kl} u_{lj}(t) + \right. \\ &\quad \left. + u_{ik}(t) s_{kl} p_{lj}] - a_{ik} u_{kj} - u_{ik} a_{jk} \right\} \end{aligned} \quad (4.42)$$

$$\sum_{k=1}^n \sum_{l=1}^n y_{kl}(t) \frac{\partial h_{kl}}{\partial p_{ij}} = -y_{ij} \quad (4.43)$$

$$\sum_{k=1}^n \sum_{l=1}^n y_{kl}(t) \frac{\partial h_{kl}}{\partial l_{ij}} = 2 \sum_{p=j}^n y_{ip}(t) l_{pj}(t). \quad (4.44)$$

Assume-se que  $\eta_v > 0$ ,  $\eta_z > 0$  e que  $h_{ij}(P, L) = \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk} - p_{ij}$ ,  $\frac{de_{ij}(h_{ij})}{dh_{ij}} \triangleq f_{ij}(h_{ij})$ .

As Eqs. (4.31-4.36) e Eqs. (4.43)-(4.44) implicam que

$$\begin{aligned} \frac{d\xi(P, L)}{dt} &= \sum_{k=1}^n \sum_{l=1}^n \left[ \frac{de_{kl}(g_{kl})}{dt} + \frac{de_{kl}(h_{kl})}{dt} \right] \\ &= \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \left\{ \sum_{j=1}^n \left[ \frac{\partial e_{kl}(g_{kl})}{\partial p_{ij}} \frac{dp_{ij}(t)}{dt} + \right. \right. \\ &\quad \left. \left. + \frac{\partial e_{kl}(h_{kl})}{\partial p_{ij}} \frac{dp_{ij}(t)}{dt} + \sum_{j=1}^i \frac{\partial e_{kl}(h_{kl})}{\partial l_{ij}} \frac{dl_{ij}(t)}{dt} \right] \right\} \end{aligned}$$

$$\begin{aligned} \frac{d\xi(P, L)}{dt} &= \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \left\{ \sum_{j=1}^n \left[ \frac{de_{kl}(g_{kl})}{dg_{kl}} \frac{\partial g_{kl}}{\partial p_{ij}} + \right. \right. \\ &\quad \left. \left. + \frac{de_{kl}(h_{kl})}{dh_{kl}} \frac{\partial h_{kl}}{\partial p_{ij}} \right] \frac{dp_{ij}(t)}{dt} + \right. \\ &\quad \left. + \sum_{j=1}^i \frac{de_{kl}(h_{kl})}{dh_{kl}} \frac{\partial h_{kl}}{\partial l_{ij}} \frac{dl_{ij}(t)}{dt} \right\} \end{aligned} \quad (4.45)$$

$$\begin{aligned} \frac{d\xi(P, L)}{dt} &= \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^n \left\{ \sum_{j=1}^n \left[ f_{kl}(g_{kl}) \frac{\partial g_{kl}}{\partial p_{ij}} + \right. \right. \\ &\quad \left. \left. + f_{kl}(h_{kl}) \frac{\partial h_{kl}}{\partial p_{ij}} \right] \frac{dp_{ij}(t)}{dt} + \right. \\ &\quad \left. + \sum_{j=1}^i f_{kl}(h_{kl}) \frac{\partial h_{kl}}{\partial l_{ij}} \frac{dz_{ij}(t)}{dt} \right\} \end{aligned} \quad (4.46)$$

$$\begin{aligned} \frac{d\xi(P, L)}{dt} &= \sum_{l=1}^n \left\{ \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n \left[ u_{kl}(t) \frac{\partial g_{kl}}{\partial p_{ij}} + \right. \right. \\ &\quad \left. \left. + y_{kl}(t) \frac{\partial h_{kl}}{\partial p_{ij}} \right] \right\} \end{aligned} \quad (4.47)$$

$$\begin{aligned}
 \frac{d\xi(P, L)}{dt} &= \sum_{i=1}^n \sum_{j=1}^n \left\{ \sum_{k=1}^n \sum_{l=1}^n [p_{ik}(t) s_{kl} p_{lj}(t) + \right. \\
 &+ u_{il}(t) s_{kl} p_{lj}(t)] - \sum_{k=1}^n [a_{ik} u_{kj}(t) \\
 &+ u_{ik}(t) a_{jk}(t) - y_{ij}(t)] \left. \right\} \frac{dp_{ij}(t)}{dt} + \\
 &+ 2 \sum_{i=1}^n \sum_{j=1}^n \left[ \sum_{k=j}^n y_{ik}(t) l_{kj}(t) \right] \frac{dz_{ij}(t)}{dt}
 \end{aligned} \tag{4.48}$$

$$\begin{aligned}
 \frac{d\xi(P, L)}{dt} &= - \sum_{i=1}^n \left\{ \frac{1}{\eta_v} \sum_{j=1}^n \left[ \frac{dp_{ij}(t)}{dt} + \right]^2 \right. \\
 &+ \left. \frac{2}{\eta_z} \sum_{j=1}^n \left[ \frac{dl_{ij}(t)}{dt} \right]^2 \right\}
 \end{aligned}$$

$$\begin{cases} < 0, & \text{se } \exists i, j, dp_{ij}(t)/dt \neq 0, \text{ ou } dl_{ij}/dt \neq 0; \\ = 0, & \text{se } \forall i, j, dp_{ij}(t)/dt = 0 \text{ e } dl_{ij}/dt = 0. \end{cases}$$

Portanto,  $\xi$  é uma função de *Lyapunov* e a rede neural recorrente é assintoticamente estável no sentido amplo.

#### 4.11.1 Análise de Solvabilidade

A análise de estabilidade indica que a transição do estado de ativação forma um fluxo de gradiente que minimiza a função energia  $\xi$ . O teorema a seguir fornece uma condição necessária e suficiente para que o fluxo de gradiente convirja para a solução simétrica e definida positiva da equação algébrica de *Riccati*,  $P$ , a partir de quase qualquer estado inicial.

**Teorema 4.11.2.** *Suponha que toda função de ativação seja contínua, pelo menos diferenciável por partes, monótona não decrescente, isto é,  $\frac{df_{ij}(\xi)}{d\xi} \geq 0$ , para  $-\infty < \xi < +\infty$ , e  $f_{ij}(\xi) = 0$ , se  $\xi = 0$ . A matriz de estado estacionário da rede neural recorrente é sempre a solução simétrica e definida positiva para a equação*

de *Riccati* para qualquer estado inicial simétrico  $P(0)$  e quaisquer estados iniciais não-nulos  $L(0)$ , ( ou seja,  $\forall P(0) = P(0)^T, \forall L(0) \neq 0, \bar{P} = P$ ) se e somente se

$$\text{rank}(\bar{P}BR^{-1}B^T - A) = n \quad (4.49)$$

e todos os elementos diagonais em  $\bar{L}$  são não nulos, isto é,  $\forall i, \bar{l}_{ii} \neq 0$  para  $i = 1, 2, \dots, n$ , (Wuang e Wu, 1998).

**Prova** Os requisitos relativos a simetricidade da matriz de estado inicial  $P(0)$  e matriz de estado inicial não-nula  $L(0)$ , são devidos a assimetria da matriz de estado inicial  $P(0)$ , a qual resulta em uma assimetria de  $P(t)$ , e uma matriz de estado inicial nula não permite qualquer estado de transição. Visto que  $f_{ij}$  é contínua,  $f_{ij}(\xi) = 0$  se  $\xi = 0$ ,

$$\begin{aligned} U(t) &\triangleq F[P(t)SP(t) - A^T P(t)P(t)A - Q] \\ Y(t) &\triangleq F[L(t)L(t)T - P(t)] \\ \bar{U} &= F(\bar{P}S\bar{P} - A^T\bar{P} - \bar{P}A - Q) = 0, \end{aligned}$$

se e somente se  $\bar{P}S\bar{P} - A^T\bar{P} - \bar{P}A - Q = 0$  e

$$\bar{Y} = F(\bar{L}\bar{L}^T - \bar{P}) = 0, \quad \text{se e somente se } \bar{L}\bar{L}^T = \bar{P}.$$

A Eq. (4.37) implica que  $\bar{P}S\bar{U} + \bar{U}S\bar{P} - A\bar{U} - \bar{U}A^T - \bar{Y} = 0$  quando  $P(t)$  converge, isto é,  $\frac{dP(t)}{dt} = 0$ . Similarmente, a Eq. (4.38) implica que  $\bar{Y}\bar{L} = 0$  quando  $L(t)$  converge, ou seja,  $\frac{dL(t)}{dt} = 0$ . Uma vez que

$$\begin{aligned} &\bar{P}S\bar{U} + \bar{U}S\bar{P} - A\bar{U} - \bar{U}A^T - \bar{Y} = \\ &= (\bar{P}S - A)\bar{U} + [(\bar{P}S - A)\bar{U}]^T - \bar{Y} \\ &= (\bar{P}S - A)\bar{U} + [(\bar{P}S - A)\bar{U}]^T - \bar{Y}, \end{aligned}$$

a condição necessária e suficiente para  $\bar{P}S\bar{U} + \bar{U}S\bar{P} - A\bar{U} - \bar{U}A^T - \bar{Y} = 0$  é se e somente se  $\bar{U} = 0 \wedge \bar{Y} = 0$  é que  $(\bar{P}BR^{-1}B^T - A)$  possua *rank* completo. Similarmente, a condição necessária e suficiente para  $\bar{Y}\bar{L} = 0$  é se e somente se  $\bar{Y} = 0$  isto é,  $\bar{L}$  tenha *rank* cheio. Desde que  $L(t)$  seja uma matriz triangular,  $\text{rank}(\bar{L}) = n$  se e somente se  $\forall i, \bar{l}_{ii} \neq 0$ . Portanto, a matriz de estado da rede

neural recorrente converge para uma solução  $P$  simétrica e definida positiva da equação algébrica de *Riccati* se e somente se  $\text{rank}(\bar{P}BR^{-1}B^T - A) = n, \forall i, \bar{l}_{ii} \neq 0$ .

**Observação** O Teorema 4.11.2 estabelece uma condição necessária e suficiente para convergência global da solução definida positiva da equação algébrica de *Riccati* com quase todos os estados iniciais arbitrários, ou seja, a matriz de estado  $P(t)$  pode ainda alcançar a solução desejada dos dados estados iniciais sem satisfazer a condição. Um método de projeto é fixar  $P(0)$  e  $L(0)$  com valores relativamente grandes de modo que  $P(t)$  e  $L(t)$  sejam inicialmente definidas positivas.

## 4.12 Limitações de Vantagens da Metodologia Neuro-Genética

A metodologia da RNA tem como potencial usar a informação prévia para computar a solução da EAR, de modo que para uma condição operacional estável, o número de iterações necessárias para computar a nova solução da EAR será menor que as condições iniciais.

A convergência da RNA será mais rápida se comparada ao método de *Schur*, que necessita computar a inversão da matriz, ordenação e realizar transformações de similaridade.

As limitações da RNA estão nas condições iniciais, dependência das matrizes de ponderação  $Q$  e  $R$ , e o valor inicial da solução da EAR no processo de treinamento da rede neural recorrente. Para superar o último problema, podemos usar a estratégia da homotopia para sistemas muito ruidosos ou altamente não lineares. Na pesquisa decorrente, computou-se a primeira condição inicial usando o método de *Schur*.

O AG tem como desvantagem baixa velocidade, que está associada ao número de parâmetros e também às condições iniciais. Outras limitações que necessitam ser discutidas, são as definições dos operadores genéticos, afim de manter a diversidade da população em todo o processo de evolução para as gerações.

O AG pode ser usado como esquema escalonado, tal que possamos computar as matrizes  $Q$  e  $R$  *off-line* para os modos de operações mais freqüentes e então trocar as matrizes entre esses modos operacionais para um processo *on-line*.

## 4.13 Conclusão

Uma fusão do AG e da RNR para resolver o problema de autoestrutura é uma boa alternativa para realizar a seleção das matrizes de ponderação  $QR$  e computar a solução da EAR. Uma avaliação desacoplada dos paradigmas de inteligência computacional, apresentou uma boa eficiência para realizar a alocação de autovalores e autovetores. Como uma contribuição para a teoria  $LQR$ , o método proposto promove melhorias no problema de controle de autoestrutura pelo algoritmo genético e na solução da equação algébrica de Riccati pela rede neural artificial recorrente.

O desempenho da RNR foi avaliado em termos das variações dos parâmetros e ilustrações das superfícies das normas do erro e função energia. Estas avaliações mostram a importância dos ajustes desses parâmetros para melhorar a convergência da rede neural recorrente. A análise da convergência está centrada em dois tipos de resultados. O primeiro resultado vem do melhor ajuste dos parâmetros e o segundo lida com a melhor escolha das condições iniciais. Outro ponto que pode ser considerado, é o número de iterações usadas para computar a solução. Como o número de iterações é crescente, uma melhor solução é alcançada com grande esforço computacional. De modo geral, a fusão proposta mostrou que é possível uma solução viável, para realização em tempo real das metodologias de controle inteligente baseada na teoria do projeto  $LQR$  e paradigmas de inteligência computacional.

# PROJETO LQR NEURO - GENÉTICO: EXPERIMENTOS COMPUTACIONAIS E VALIDAÇÕES

---

Apresentam-se nesta tese modelos de sistemas multivariáveis representando as dinâmicas de um circuito elétrico de 4ª ordem, um gerador de indução duplamente alimentado (*DFIG*) de 6ª e de uma aeronave de 6ª ordem para avaliar o desempenho dos paradigmas de inteligência computacional e a metodologia do projeto de controle. A avaliação do desempenho da busca genética é realizada em termos de estatísticas da função de *fitness* enquanto que a convergência da rede neural recorrente é avaliada por variações que apresentam os resultados dos experimentos computacionais para o circuito de quarta ordem da norma do infinito relacionada a solução da equação algébrica de *Riccati* e da função energia. A solução do problema de controle é avaliada no domínio do tempo e frequência pela resposta ao impulso, análise modal e valores singulares.

A Seção 5.1, apresenta o desenvolvimento do procedimento dos projetos que caracterizam os controladores *LQR* neuro-genético. As questões relevantes desta seção são discussões e análise sobre o comportamento do modelo do sistema dinâmico, isto é, análise no domínio do tempo e da frequência, e computação dos ganhos do controlador. A Seção 5.2 apresenta os sistemas dinâmicos de sexta ordem cujo resultados e análises encontram-se nos Apêndices E e F. Na Seção 5.3 encontram-se discussões sobre o modelo de fusão neuro-genético. A Seção 5.4

apresenta algumas conclusões e observações sobre o compromisso da metodologia.

## 5.1 Projeto *LQR* Neuro-Genético: Circuito Elétrico

O modelo de quarta ordem referente a um circuito elétrico, é usado para avaliar o desempenho da metodologia proposta. O circuito elétrico é apresentado na Fig. 5.1, (Abreu ,2008a). Optou-se por este sistema devido à facilidade de mostrar os efeitos de integração do modelo de fusão (AG e RNA). Os elementos básicos são capacitores, indutores, resistores e fontes, sendo que os resistores são responsáveis pelo desempenho do amortecimento e os outros elementos pelas oscilações. As leis da tensão e corrente de *Kirchhoff* são aplicadas para obter as EDOs. O comportamento da tensão e corrente deste circuito são dadas por

$$C \frac{dx_1}{dt} = -\frac{x_1}{R_1} + x_3 \quad (5.1)$$

$$C \frac{dx_2}{dt} = -\frac{x_2}{R_1} - x_4 \quad (5.2)$$

$$L \frac{dx_3}{dt} = -u_1 - x_1 - R_2(x_3 + x_4) \quad (5.3)$$

$$L \frac{dx_4}{dt} = -u_2 - x_2 - R_2(x_3 + x_4) \quad (5.4)$$

sendo  $x_1$  e  $x_2$  as variáveis de estado que representam as tensões nos capacitores, enquanto que as variáveis de estado  $x_3$  and  $x_4$  representam as correntes nos indutores.



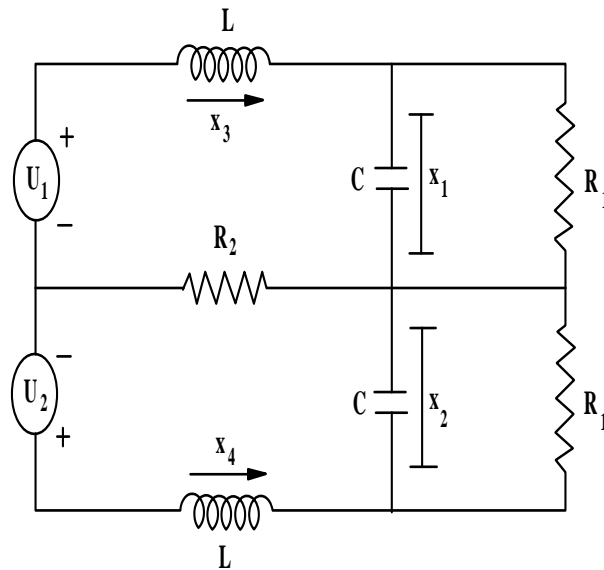


Figura 5.1: Diagrama de um circuito elétrico de 4ª ordem.

Seguindo a metodologia clássica de controle, o projeto  $LQR$  consiste na determinação de uma lei de controle ótimo que passa pelas seguintes etapas:

- Representação do sistema dinâmico por um conjunto de equações diferenciais lineares e ordinárias;
- Descrições em função de transferência e no espaço de estado;
- Análise de controle;
- Sintonia do ganho do projeto  $LQR$ : (tentativa e erro - método de *Bryson*) e AG;
- RNA.

### 5.1.1 Descrição no Espaço de estados

O modelo das EDOs do circuito elétrico Eqs. (5.1) é escrito na forma matricial ou no espaço de estados, que é dada por

$$\dot{x} = Ax + Bu, \quad t \geq t_o. \quad (5.5)$$

$$y = Cx \quad (5.6)$$

sendo

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T, u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T, y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$$

o vetor de estado, o vetor de estado e o vetor de entrada, respectivamente. Considerando o conjunto dos parâmetros do circuito elétrico  $\Omega_{RLC} = \{L, C, R\}$  e supondo que seus elementos assumem valores unitários, as matrizes do sistema dinâmico de representação no espaço de estado são:

$$A = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 \\ -1 & 0 & -1 & -1 \\ 0 & 1 & -1 & -1 \end{bmatrix}, \quad (5.7)$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (5.8)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.9)$$

e

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (5.10)$$

### 5.1.2 Análise Modal

A seguir apresenta-se as restrições do projeto *LQR* mostradas na Tabela 5.1. Elas constituem os limites dos autovalores e suas sensibilidades associadas. Estas

restrições foram montadas a partir da autoestrutura para verificar a metodologia proposta, (Davis e Clarke ,1995). A Tabela 5.1 exibe as restrições de projeto para os autovalores e as sensibilidades associada a cada autovalor.

Tabela 5.1: Restrições de projeto para o  $LQR$  e sensibilidades

Nº.	Autovalor	Sensibilidade
1	$-3.00 \leq Re \leq -1.50$	1.6
	$-1.0 \leq Imag \leq 1.0$	
2	$-3.00 \leq Re \leq -1.5.00$	1.5
	$-1.5 \leq Imag \leq 1.00$	
4	$-2.00 \leq Re \leq -0.6.00$	1.5
	$0.8 \leq Imag \leq 1.00$	
5	$-2.5.00 \leq Re \leq -0.6.00$	1.8
	$-1.00 \leq Imag \leq 1.00$	

A análise modal caracteriza-se por verificar os efeitos provocados pelas interações entre a autoestrutura, condições iniciais e as matrizes de entrada do sistema. Os autovalores do circuito elétrico são apresentados na Tabela 5.2. Como pode ser observado, o sistema é estável e as sensibilidades dos autovalores estão na linha de fronteira (eixo  $j\omega$ ) da autoestrutura especificada.

Tabela 5.2: Circuito Elétrico: Autovalores e sensibilidades.

Nº	Autovalores	Sensibilidades
1	$-1.500 + j 0.866$	1.000
2	$-1.500 - j 0.866$	1.000
3	$-0.500 + j 0.866$	1.000
4	$-0.500 - j 0.866$	1.000

### 5.1.3 Análise no Domínio do Tempo

As saídas definidas do circuito elétrico são os estados  $x_1$  e  $x_2$ . O comportamento dinâmico dos estados pode ser avaliado pelas respostas ao sinal impulso. A

Fig. 5.2 mostra as respostas ao impulso para o sistema teste. As legendas  $y_1 - ch1$  e  $y_1 - ch2$  da referida figura representam os canais associados às saídas devido as contribuições das entradas  $u_1$  e  $u_2$ . A soma das contribuições de saída é a  $y_1$  que corresponde a saída total. A mesma interpretação é dada para a saída  $y_2$  com relação aos canais  $y_2 - ch1$  e  $y_2 - ch2$ .

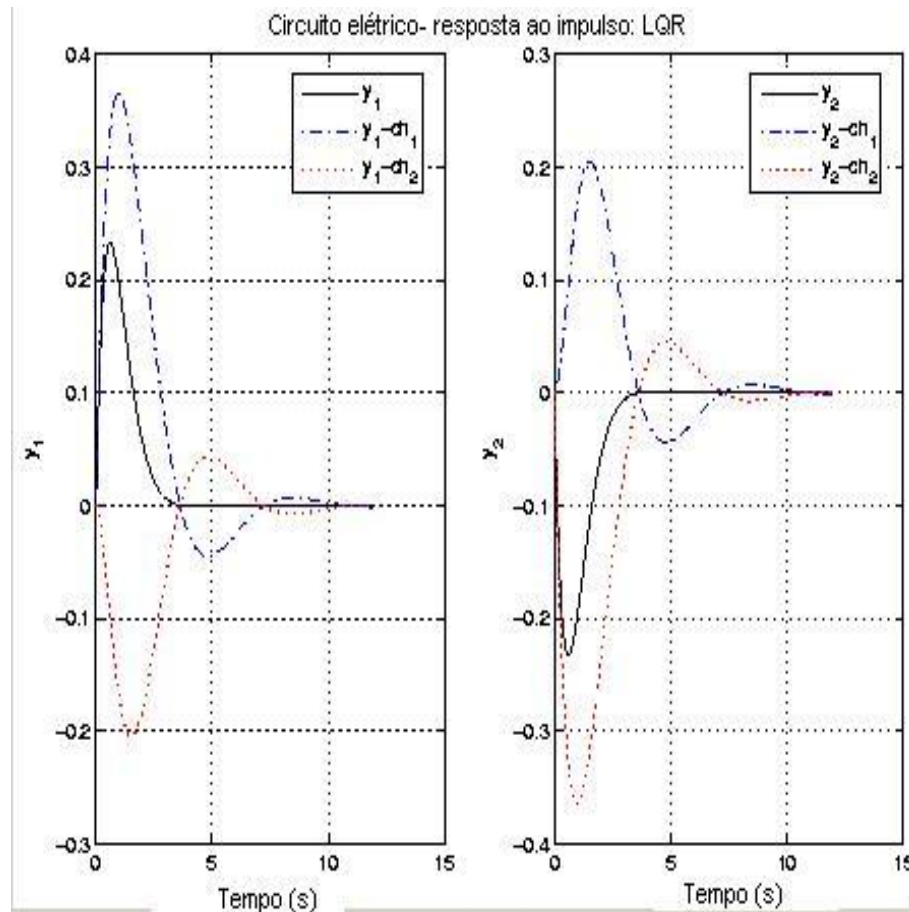


Figura 5.2: Modelo do controlador ótimo neuro-genético.

#### 5.1.4 Análise de Valores Singulares

A análise no domínio da frequência é realizada em termos da decomposição dos valores singulares, SVD, da matriz  $A$  e em termos da variação de frequência na função de transferência. A decomposição em valores singulares da matriz do sistema é dada por

$$A = U_{planta} \sigma_{planta} V_{planta}^H \quad (5.11)$$

sendo  $\sigma_{planta} \in \mathbb{F}^{n \times n}$  uma matriz diagonal com todos os seus elementos diagonais não-negativos dispostos em ordem decrescente, enquanto que as matrizes  $U_{planta} \in \mathbb{F}^{m \times n}$  e  $V_{planta}^H \in \mathbb{F}^{n \times n}$  são unitárias. Os valores singulares da matriz  $A$  são dados por

$$\sigma_{planta} = \begin{bmatrix} 2.303 & & & \\ & 1.618 & & \\ & & 1.303 & \\ & & & 0.618 \end{bmatrix}. \quad (5.12)$$

Sob o ponto de vista de grande e pequenos ganhos, pode-se observar que a função de transferência está amplificando nos três primeiros valores singulares e atenuando no último valor singular.

### 5.1.5 Sintonia de Ganho do Projeto *LQR*

Os parâmetros do AG são os parâmetros do sistema dinâmico, especificações de controle, dimensão do cromossomo individual e o número de indivíduos da população. O AG é codificado numa base decimal devido a natureza do problema, que é formado pelos elementos decimais das matrizes de ponderação.

A fim de resolver o índice de desempenho, necessita-se inicialmente resolver a EAR, e esta equação é dependente das matrizes  $Q$  e  $R$ . Estas matrizes devem ser semidefinida e definida positiva, respectivamente. Os métodos por tentativa estabelecem que essas matrizes não são suficientes.

A alocação de autoestrutura é imposta pelos controladores *LQR*, porém ajustando os controladores que são desenvolvidos por AG. Na essência, as ações de ajuste são realizadas pelas matrizes de ponderação, que são escolhidas pelo AG. Conseqüentemente, as matrizes  $Q$  e  $R$ , são usadas para ajustar os ganhos  $K_{LQR}$  ótimos do controlador.

O propósito principal desta seção é assinalar as características de desempenho de um AG direcionado a busca das matrizes de ponderação. A busca das matrizes

de ponderação de estado e controle, são sustentadas por um AG baseado na estrutura de otimização.

### Inicialização do Algoritmo Genético

As condições iniciais da população do AG estão associadas na pesquisa com a caracterização das espécies. Considerando as hipóteses, os elementos básicos que constroem a espécie proposta é um gerador pseudo-aleatório com as seguintes características: seqüência pseudo-aleatória com distribuição uniforme no intervalo  $[2^{-53}, 1 - 2^{-53}]$ , sendo  $2^{1492}$  o máximo de números aleatórios.

### Parâmetros da População

Os parâmetros da população inicial são: o tamanho da população, as dimensões das matrizes  $Q$  e  $R$  que são dadas por  $Q_{ordem} = 4$ ,  $R_{ordem} = 2$ , e uma população constante de 50 indivíduos.

### Parâmetros da População Inicial

Os parâmetros fixos e variáveis que são usados para gerar a população inicial, de acordo com as regras das Eqs. (4.11) e (4.12), são mostrados nas Tabelas 5.3 e 5.4.

Tabela 5.3: Parâmetros da Matriz  $Q$  - População Inicial.

	Fixo	Variável		Fixo	Variável
	$p_{Q\alpha}$	$p_{Q\beta}$		$p_{Q\alpha}$	$p_{Q\beta}$
$q_{11}$	10.500	40.000	$q_{22}$	30.500	30.000
$q_{33}$	10.000	50.000	$q_{44}$	20.000	10.000
	$p_{Q\gamma}$				
$q_{ij}$	10.270				

AS regras de polarização para os parâmetros da população inicial são dadas por:  $Q_{ii} = Q_{iia} + Q_{jjb}\kappa_{rand}$  para os elementos diagonais e  $R_{ij} = R_{ijb}\kappa_{rand}$  para os elementos não diagonais. A mesma regra é aplicada para construir os cromossomos  $R$  dos indivíduos  $QR$ .

Tabela 5.4: Parâmetros da Matriz  $R$  - População Inicial.

	Fixo	Variável		Fixo	Variável
	$p_{R\alpha}$	$p_{R\beta}$		$p_{R\alpha}$	$p_{R\beta}$
$r_{11}$	10.00	15.000	$r_{22}$	8.00	6.00
	$p_{R\gamma}$				
$r_{12}$	5.00				

### Avaliação da População Inicial

O perfil da população inicial é definido pelos valores da função objetivo apresentado na Fig. 5.3

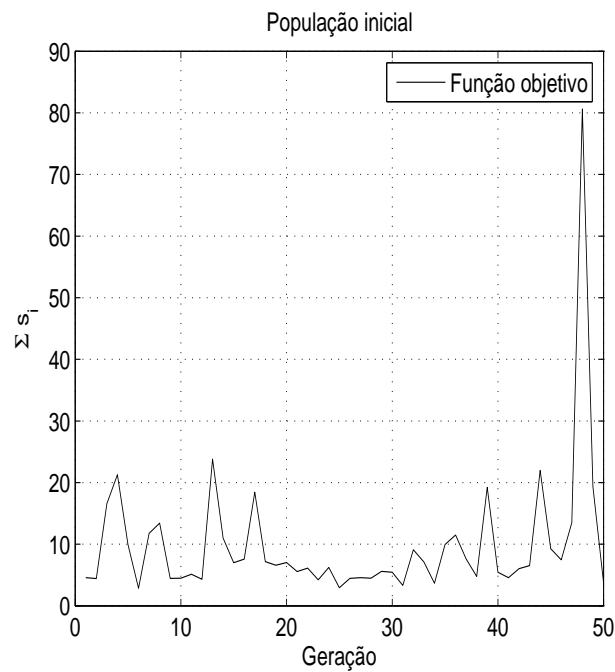


Figura 5.3: Sensibilidade total de cada indivíduo da população inicial

Detalhes de busca da população inicial, podem ser visto no processo de inicialização do referido AG. A diversidade genética pode ser observada na Fig. 5.3;

do indivíduo 20 até aproximadamente ao indivíduo 30 é observado um vale cujo significado, implica em uma baixa diversidade genética, que por possuir genes similares não consegue evoluir, ou seja, quanto maior for a perda de diversidade, mais rápida será a convergência genética do AG. Estas observações implicam em uma população satisfatória para iniciar o processo de busca das matrizes  $QR$ . Neste processo de inicialização do AG, diversos indivíduos possuem sensibilidade que satisfaz a condição associada à estrutura de otimização combinatória para realizar a busca das matrizes.

As estatísticas de sensibilidades associadas aos autovalores são dadas na Tabela 5.5. Percebe-se claramente uma variabilidade dos dados em menor e maior grau, sendo que a variabilidade dos dados sendo maior, sua média terá uma confiabilidade pequena que será inadequada calculá-la. Por outro lado, um alto grau de uniformidade ou pequena dispersão, em geral, costuma ser considerado como algo de qualidade desejável no AG. É inconveniente qualquer produção que apresente muita variabilidade no processo obtido. A média de valores das sensibilidades normalizadas, mostram que a maioria dos indivíduos não alcançaram a sensibilidade especificada, ou seja,  $s_i(Q, R) \leq 1$ ; os valores mínimos de todas as quatro sensibilidades alcançaram as especificações de projeto para indivíduos diferentes;

Tabela 5.5: População inicial-LQR - estatísticas da sensibilidade normalizada.

$s_i$	Máx.	Média	Mín.	Desv. P.
1	14.320	2.060	0.729	2.051
2	13.550	2.228	0.752	2.804
3	30.185	2.759	0.721	4.354
4	46.931	2.875	0.589	6.677

### 5.1.6 População Final - AG

As estatísticas da população final estão apresentadas na Tabela 5.6. Ela exhibe os indivíduos viáveis das gerações foram obtidos com a evolução do AG; percebe-se claramente que as restrições de sensibilidades foram satisfeitas, ou seja,  $s_i(Q, R) \leq 4.04$  e que as matrizes  $Q$  e  $R$  ainda continuam com as suas características de



projeto positiva semidefinida e positiva e definida por meio de seus respectivos traços.

Tabela 5.6: População final - função objetivo e traços de matrizes.

Ger.	Indiv	$\Sigma s_i$	$Q_{trac.}$	$R_{trac.}$
1	20	3.073	137.289	26.793
3	13	2.997	122.136	26.443
5	17	3.082	128.276	27.218
7	14	3.151	135.854	26.933
9	48	3.110	132.304	26.773

### Avaliação da População Final

Na Fig. 5.4 é apresentado o perfil da função objetivo (soma das sensibilidades) versus a média da função de *fitness* de cada indivíduo da população final durante o processo de busca. A média da função objetivo da população, apresentou uma boa melhoria até a trigésima geração; a partir daí, a média da função objetivo apresentou melhorias até o trigésimo indivíduo. Pode-se perceber claramente na figura que a partir desta geração, o processo não apresentou melhorias e que a soma das sensibilidades dos indivíduos factíveis em cada geração é igual a 12.54 e a soma das médias da função de *fitness* equivale a 3.905. Isto mostra que o algoritmo apresenta resultados satisfatórios.

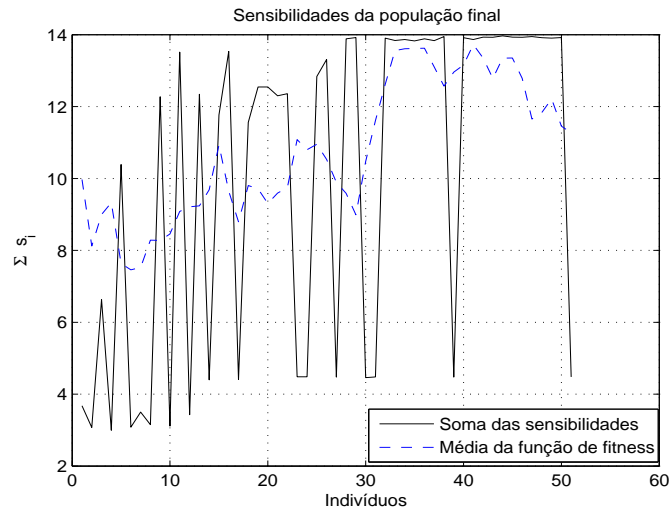


Figura 5.4: Processo de evolução-média da função objetivo e melhor função objetivo de gerações

O AG de busca, encontrou cinco (05) indivíduos que são apresentados na Tabela 5.7. Os autovalores associados estão na Tabelas 5.8 e 5.9. As partes real têm uma diferença de  $10^{-2}$  e a parte imaginária, uma diferença de  $10^{-1}$ . Conseqüentemente, como esperado, os resultados computacionais para resposta ao impulso, apresentaram uma ligeira diferença entre todas as soluções viáveis que são apresentadas na Tabela 5.7.

Tabela 5.7: População final AG - sensibilidades e função objetivo dos indivíduos viáveis

Ger.	Indiv	$s_1$	$s_2$	$s_3$	$s_4$	$\Sigma s_i$
1	20	0.975	0.815	0.694	0.589	3.073
3	13	0.919	0.831	0.677	0.570	2.997
5	17	0.967	0.845	0.684	0.586	3.082
7	14	0.996	0.883	0.686	0.586	3.151
9	48	0.986	0.855	0.683	0.586	3.110

Tabela 5.8: População Final - Indivíduos Viáveis - Parte Real dos Autovalores.

Ger.	Indiv	$\lambda_{1r}$	$\lambda_{2r}$	$\lambda_{3r}$	$\lambda_{4r}$
1	20	-1.762	-1.762	-1.249	-1.249
3	13	-1.765	-1.765	-1.183	-1.183
5	17	-1.743	-1.743	-1.301	-1.301
7	14	-1.762	-1.762	-1.322	-1.322
9	48	-1.758	-1.758	-1.308	-1.308

Tabela 5.9: População final - indivíduos viáveis - parte imaginária dos autovalores.

Ger.	Indiv	$\lambda_{1i}$	$\lambda_{2i}$	$\lambda_{3i}$	$\lambda_{4i}$
1	20	0.800	-0.800	0.911	-0.911
3	13	0.790	-0.790	1.069	-1.069
5	17	0.900	-0.900	0.848	-0.848
7	14	0.737	-0.737	1.024	-1.024
9	48	0.725	-0.725	0.954	-0.954

As estatísticas de sensibilidade normalizada da população final são apresentadas na Tabela 5.10. Observando esta tabela e as estatísticas da população inicial da Tabela 5.5, nota-se uma melhoria promovida pelo AG, uma vez que os valores mínimos dos cinco indivíduos associados a sensibilidades apresentam valores inferiores a um (1) e o desvio padrão também apresentou melhoria quando comparado com o da população inicial apresentado na Tabela 5.5.

Tabela 5.10: População final  $LQR$  - estatística da sensibilidade normalizada.

$s_i$	Máx.	Média	Mín.	Desv. P.
1	4.247	2.991	0.919	1.220
2	2.040	1.167	0.815	0.186
3	5.773	3.848	0.677	2.228
4	3.055	2.217	0.570	1.088

As sensibilidades  $s_i$  associada aos quatro autovalores e a média da função de *fitness* de cada geração da população final, são apresentadas na Fig. 5.5. Nesta figura têm-se as sensibilidades dos quatro autovalores e as médias das sensibilidades do indivíduo 20 que foi considerado o melhor. Esses valores correspondem a:  $s_1 = 3.206$  e média  $s_{20}^1 = 2.594$ ,  $s_2 = 1.162$ , média  $s_{20}^2 = 1.205$ ,  $s_3 = 5.290$ , média  $s_{20}^3 = 3.487$  e  $s_4 = 2.887$  média  $s_{20}^4 = 2.020$ .

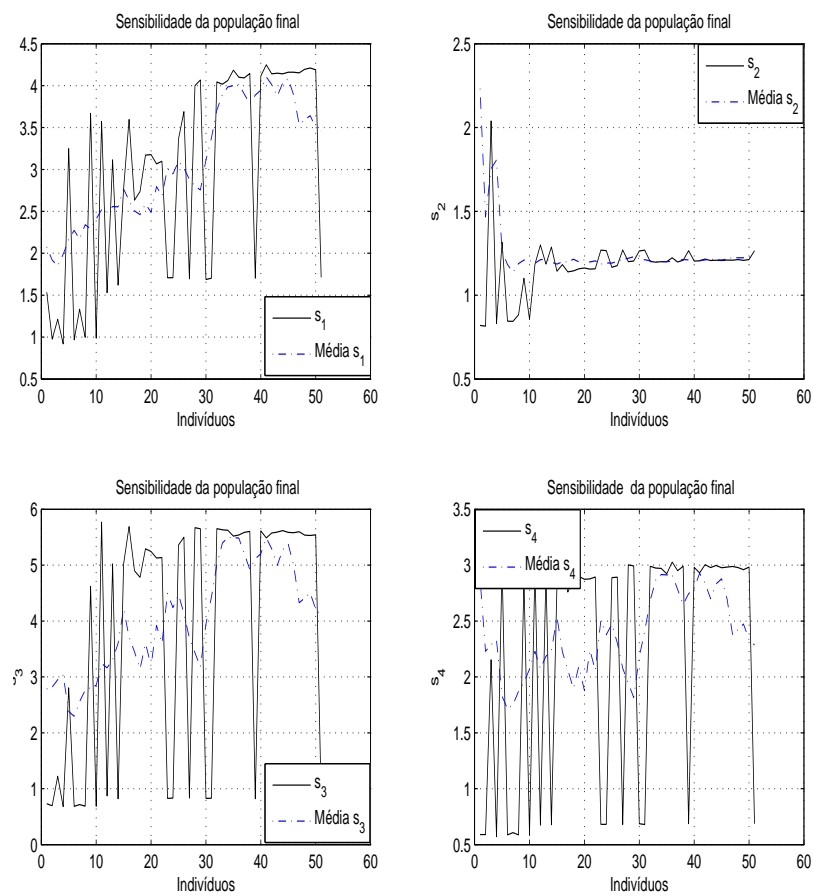


Figura 5.5: Comparação da média das sensibilidades com a sensibilidade de cada autovalor.

### 5.1.7 Avaliação do Controlador $LQR$

Um dos melhores indivíduos é escolhido para avaliar o desempenho dos controladores  $LQR$  para designar a autoestrutura. Este indivíduo é denominado de melhor indivíduo. O indivíduo número 20 da primeira geração da população final, Tabela 5.7, é escolhido para realizar os testes de avaliação. Como pode ser visto nesta tabela, todos os indivíduos viáveis estão próximos às restrições e função objetivo. Além disso, a positividade das matrizes são mostradas em termos de seus traços e autovalores.

As matrizes de ponderação  $Q$  e  $R$  do melhor individuo viável são dadas por

$$Q_{LQR_{AG}} = \begin{bmatrix} 41.436 & 5.494 & 4.069 & 5.123 \\ 5.494 & 44.103 & 9.148 & 3.102 \\ 4.069 & 9.148 & 29.682 & 6.951 \\ 5.123 & 3.102 & 6.951 & 22.067 \end{bmatrix} \quad (5.13)$$

e

$$R_{LQR_{AG}} = \begin{bmatrix} 15.225 & 2.517 \\ 2.517 & 11.568 \end{bmatrix} \quad (5.14)$$

Na Tabela 5.11, os autovalores e traços confirmam a positividade das matrizes de ponderação. A positividade pode ser usada para guiar a busca do AG, evitando a indesejada formação de nichos.

Tabela 5.11: População final do melhor indivíduo - autovalores e traços das matrizes de ponderação.

Nº	$Q - Matriz$		$R - Matriz$	
	$\lambda_Q$	$Trao_Q$	$\lambda_R$	$Trao_R$
1	17.466	137.288	10.285	26.793
2	27.654		16.508	
3	37.660			
4	54.508			

O AG de busca com seus parâmetros de inicialização produziu soluções muito próximas uma da outra, ou seja, as matrizes solução apresentadas pelo AG pos-

suem ligeiras discrepâncias. isto significa que qualquer outro indivíduo da Tabela 5.9 poderia ser escolhido.

### A Solução da EAR-*Schur*

O método de *Schur* pode ser resumido pela Eq. (3.21), que fornece a solução da EAR. Substituindo as matrizes (5.17) e (5.18) na EAR, Eq. (3.5). O método de *Schur* é aplicado de acordo com o algoritmo desenvolvido por (Laub ,1979). Após todas as computações impostas pela forma real de *Schur*, a solução da EAR é dada por

$$P_{schur} = \begin{bmatrix} 17.884 & 2.228 & 2.533 & 1.838 \\ 2.228 & 19.229 & 2.760 & -2.245 \\ 2.533 & 2.760 & 13.596 & -4.100 \\ 1.838 & -2.245 & -4.100 & 10.859 \end{bmatrix}. \quad (5.15)$$

Na hierarquia do projeto *LQR* da Fig. 4.2, os ganhos do controlador, de acordo com a Eq. (3.5), são dados por

$$K_{schur} = \begin{bmatrix} 0.145 & 0.221 & 0.987 & -0.440 \\ 0.127 & -0.242 & -0.569 & 1.035 \end{bmatrix}. \quad (5.16)$$

As matrizes de ponderação implementadas pelo método de *Bryson*, são dadas por

$$Q_{LQR_{Bryson}} = \begin{bmatrix} 1.3346 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 18.5231 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.5436 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.2118 \end{bmatrix} \quad (5.17)$$

e

$$R_{LQR_{Bryson}} = \begin{bmatrix} 18.5960 & 0.0000 \\ 0.0000 & 11.568 \end{bmatrix}. \quad (5.18)$$

Na Tabela 5.12 apresenta-se os autovalores e traços das matrizes de ponderação usando o método de *Bryson*. A seguir, tem-se a solução de *Riccati* e a matriz de ganho respectivamente.

Tabela 5.12: Método de *Bryson*: Autovalores e traços das matrizes de Ponderação.

Nº	<i>Q – Matriz</i>		<i>R – Matriz</i>	
	$\lambda_Q$	$Traco_Q$	$\lambda_R$	$Traco_R$
1	1.2118	22.6131	17.4610	36.0570
2	1.3346		18.5960	
3	1.5436			
4	18.5231			

As Eqs. (5.19) e (5.20) nos fornecem a solução da EAR e o ganho do projeto *LQR* usando as matrizes de ponderação implementadas pelo método de *Bryson*, respectivamente.

$$P_{Bryson} = \begin{bmatrix} 1.3133 & 0.5308 & -0.6847 & 0.9553 \\ 0.5308 & 7.5108 & -0.0156 & -1.6708 \\ -0.6847 & -0.0156 & 1.9093 & -2.0392 \\ 0.9553 & -1.6708 & -2.0392 & 3.7923 \end{bmatrix}. \quad (5.19)$$

$$K_{Bryson} = \begin{bmatrix} -0.0368 & -0.0008 & 0.1027 & -0.1097 \\ 0.0547 & -0.0957 & -0.1168 & 0.2172 \end{bmatrix} \quad (5.20)$$

### Resposta ao Impulso

As respostas ao impulso são aplicadas para verificar o desempenho do projeto *LQR*. Conseqüentemente os ganhos  $K_{Bryson}$  do controlador da Eq. (5.20) é o produto final do projeto *LQR*. Na Fig. 5.6 são apresentadas as respostas ao impulso para os canais 1 e 2 para o controlador *LQR*. Como pode ser observado, o projeto *LQR* utilizando a metodologia neuro-genética promoveu um aumento no fator de amortecimento devido a ambas entradas, ou seja, o projeto apresentou menor *overshoot* e entrou em regime permanente primeiro que a planta sem o controlador.

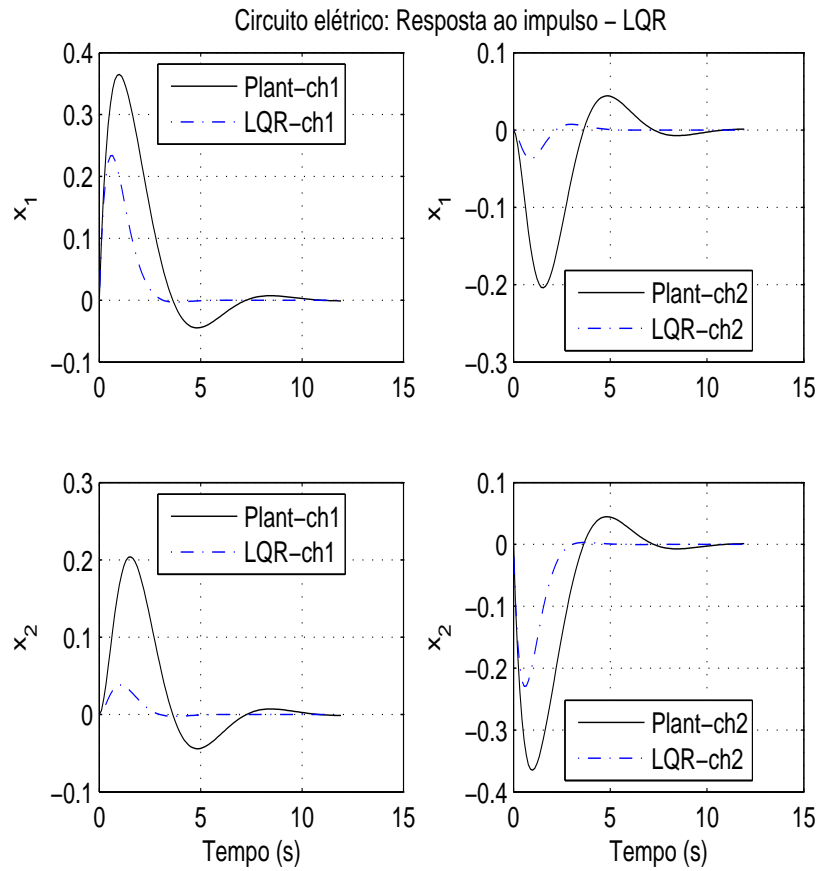


Figura 5.6: Resposta ao impulso de  $x_1$  e  $x_2$  - circuito elétrico de 4<sup>a</sup> ordem.

Na Fig. 5.7 são apresentadas as respostas ao impulso para os canais 3 e 4. Similarmente, o projeto de controle  $LQR$  também promoveu um aumento no amortecimento de ambas saídas.



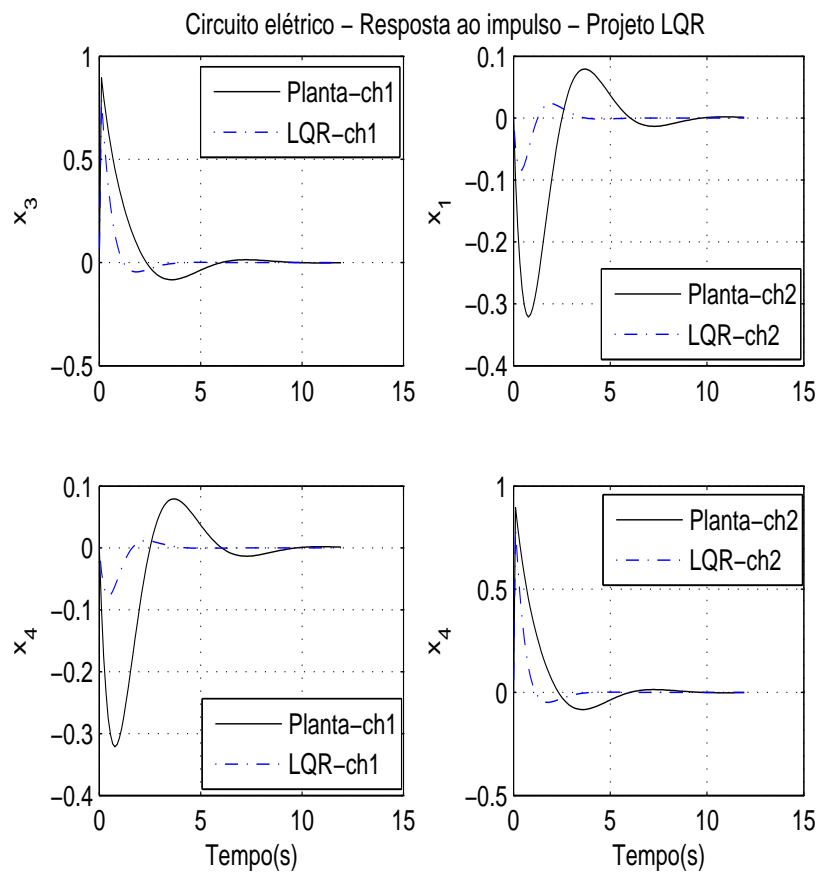


Figura 5.7: Resposta ao impulso de  $x_3$  e  $x_4$  - circuito elétrico de 4<sup>a</sup> ordem.

A Fig. 5.8 exibe a resposta ao impulso com relação ao circuito elétrico, utilizando as matrizes  $QR$  dos cinco melhores indivíduos implementadas pelo AG do projeto LQR e também as matrizes obtidas pelo método de *Bryson*. Nota-se que a ação de controle realizada pelo AG, também promoveu um aumento no amortecimento comparado ao método de *Bryson*.

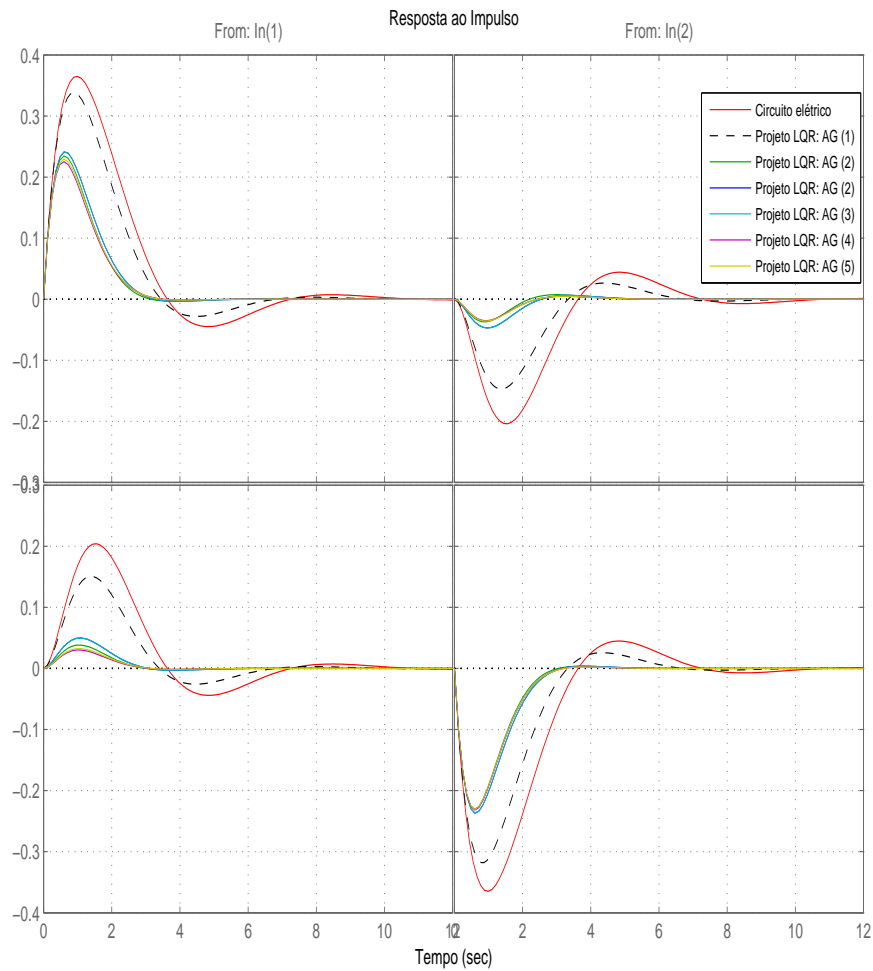


Figura 5.8: Resposta ao impulso - Circuito Elétrico de 4ª Ordem: Bryson e Controladores.

### Análise Modal

Os autovalores e sensibilidades do melhor indivíduo viável, apresentados na Tabela 5.14, mostra sua influência no aumento do fator de amortecimento das variáveis de estado. Como pode ser observado, todos os autovalores estão mais a esquerda no semi-plano complexo que os autovalores iniciais, dado na Tabela 5.2, bem como suas associadas sensibilidades dos autovalores.

Tabela 5.13: Autovalores do circuito elétrico usando  $Q_{AG}$  e  $R_{AG}$

Nº	Autovalores	Sensibilidades
1	-1.762 +j 0.790	0.975
2	-1.762 -j 0.790	0.815
3	-1.249 +j 1.069	0.694
4	-1.249 -j 1.069	0.589

Tabela 5.14: Autovalores do circuito elétrico usando  $Q_{Bryson}$  e  $R_{Bryson}$

Nº	Autovalores	Sensibilidades
1	-1.5233 + j0.8864	0.6649
2	-1.5233 - j0.8864	0.7372
3	-0.6366 + j0.9294	0.7407
4	-0.6366 - j0.9294	0.6062

### Análise de Valor Singular

As matrizes dos valores singulares  $\sigma_{AG}$  e  $\sigma_{Bryson}$  são dadas por

$$\sigma_{AG} = \begin{bmatrix} 2.818 & & & \\ & 1.990 & & \\ & & 1.398 & \\ & & & 1.285 \end{bmatrix} \quad (5.21)$$

e

$$\sigma_{Bryson} = \begin{bmatrix} 2.3661 & & & \\ & 1.5505 & & \\ & & 1.3163 & \\ & & & 0.8164 \end{bmatrix} \quad (5.22)$$

Segundo comentamos anteriormente, observa-se que a função de transferência está amplificando em todos os valores singulares, matriz  $\sigma_{LQR}$ , Eq. (5.21), do projeto  $LQR$  em comparação com os valores singulares das matrizes  $\sigma_{planta}$  (Eq. 5.12) e  $\sigma_{Bryson}$  (Eq. 5.22).

As respostas em frequências ( $rad/sec$ ) em termos do maior e menor valor singular em  $dB$  da Fig. 5.9, mostram que os controladores gerados por todos os cinco indivíduos viáveis da população final, Tabela 5.7, promovem uma resposta em SVD muito próxima. Um outro fator que pode-se destacar nesta figura é que a distância em módulo entre o maior e menor autovalor da planta e do controlador usando as matrizes de ponderação do método de *Bryson* são maiores que a distância dos controladores usando a metodologia neuro genética. Isso em termos computacionais pode levar o sistema a instabilidade.

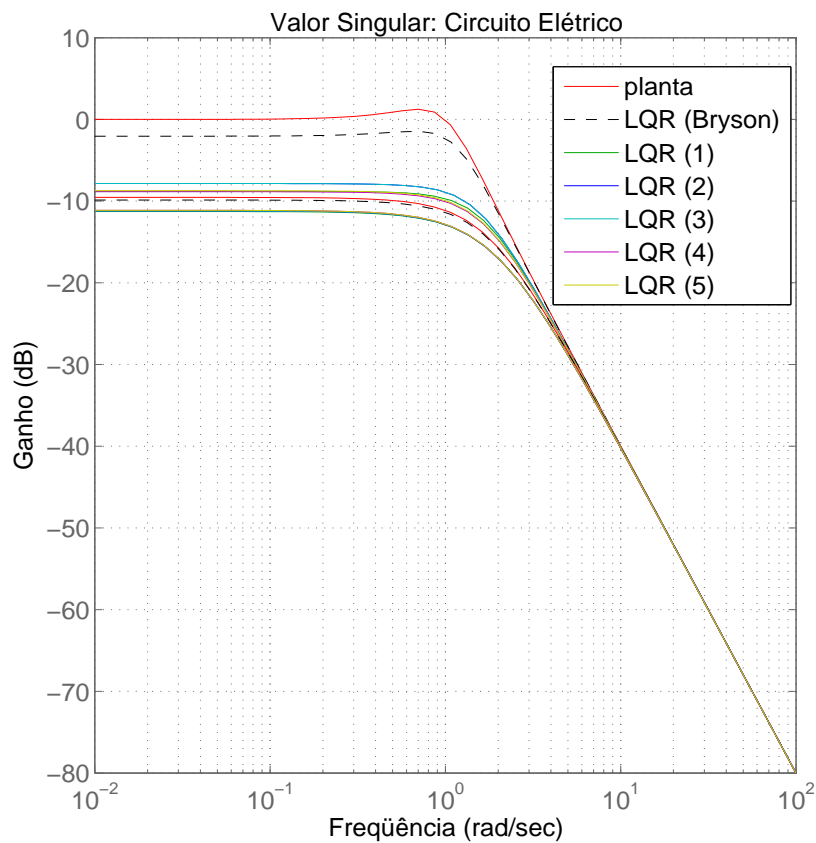


Figura 5.9: Valores Singulares: Circuito elétrico de 4ª Ordem.

### 5.1.8 Rede Neural Recorrente para Solução de *Riccati*

Esta seção enfoca os métodos de sintonia para convergência e análise de uma rede neural recorrente para resolver a EAR. A análise da RNR para resolver a EAR é orientada para mostrar as propriedades de convergência da rede. As propriedades de convergência são classificadas em termos da velocidade medida pelas quantidades de iterações e em termos da convergência de uma solução única e estável. Os parâmetros de ajustes  $\eta_v$  e  $\eta_z$ , Eqs (4.37-4.38), são sustentados pela norma do infinito e superfícies da energia da matriz mismatch da solução da EAR. Para um par de parâmetros,  $\eta_v$  e  $\eta_z$ , a análise de convergência é realizada em termos de velocidade e unicidade da solução da equação algébrica de *Riccati*.

#### Sintonia dos Parâmetros $\eta_v$ e $\eta_z$

As superfícies da norma do infinito e da energia, como função dos parâmetros de ajustes  $\eta_v$  e  $\eta_z$ , são avaliadas para definir uma inicialização para melhorar as características de estabilidade de convergência e solvabilidade da rede neural recorrente.

O impacto de variações dos parâmetros de sintonia  $\eta_v$  e  $\eta_z$ , são avaliados em primeiro momento, pelas superfícies da função energia e norma do infinito. Conseqüentemente, as superfícies possuem significado qualitativo no ajuste da RNR pelos parâmetros de sintonia. Os alvos seguintes, podem ser requeridos para o desempenho da RNR: velocidade de convergência e a solvabilidade da RNR. Na Fig. 5.10 é apresentada a superfície da norma do infinito da matriz de entrada  $U$ , sendo [1000 10000] e [1 10] os intervalos de variação dos parâmetros  $\eta_v$ , e  $\eta_z$ , respectivamente. Os passos de  $\eta_v$  e  $\eta_z$  são 1000 e 1, respectivamente. A mesma faixa de variação dos parâmetros  $\eta_v$ ,  $\eta_z$  e passos, são estabelecidos para a superfície da função energia apresentada na Fig. 5.11.

Numa inspeção visual da superfície da função energia dada na Fig.5.11, é observado que a região definida pelo escalonamento dos parâmetros  $\eta_v=[5000\ 8000]$  e  $\eta_z=[5\ 8]$ , apresenta uma mínima perda de energia. Enquanto isso, a superfície da norma do infinito, Fig. 5.10, mostra um ponto de vista diferente, levando em conta apenas a função energia, Fig. 5.11, como uma métrica para definir os parâmetros  $\eta_v$  e  $\eta_z$ .

Devido ao seu comportamento interno variável, representado pela superfície da

norma do infinito, é observado que o *mismatch* da solução da EAR nesta região é muito pobre, porém uma redução drástica do parâmetro  $\eta_v$ , conduz a otimização neural numa região de pequeno *mismatch* da norma do infinito da EAR.

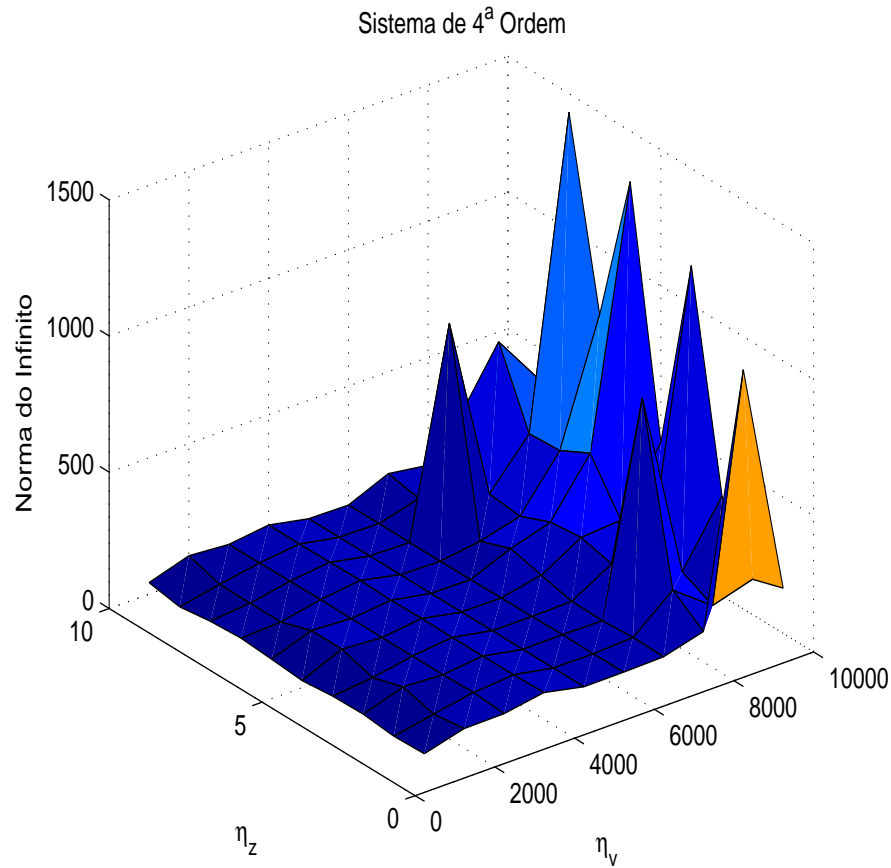


Figura 5.10: Superfície da norma do infinito para variação dos parâmetros  $\eta_v = 1000 - 10000$  e  $\eta_z = 1 - 10$ .

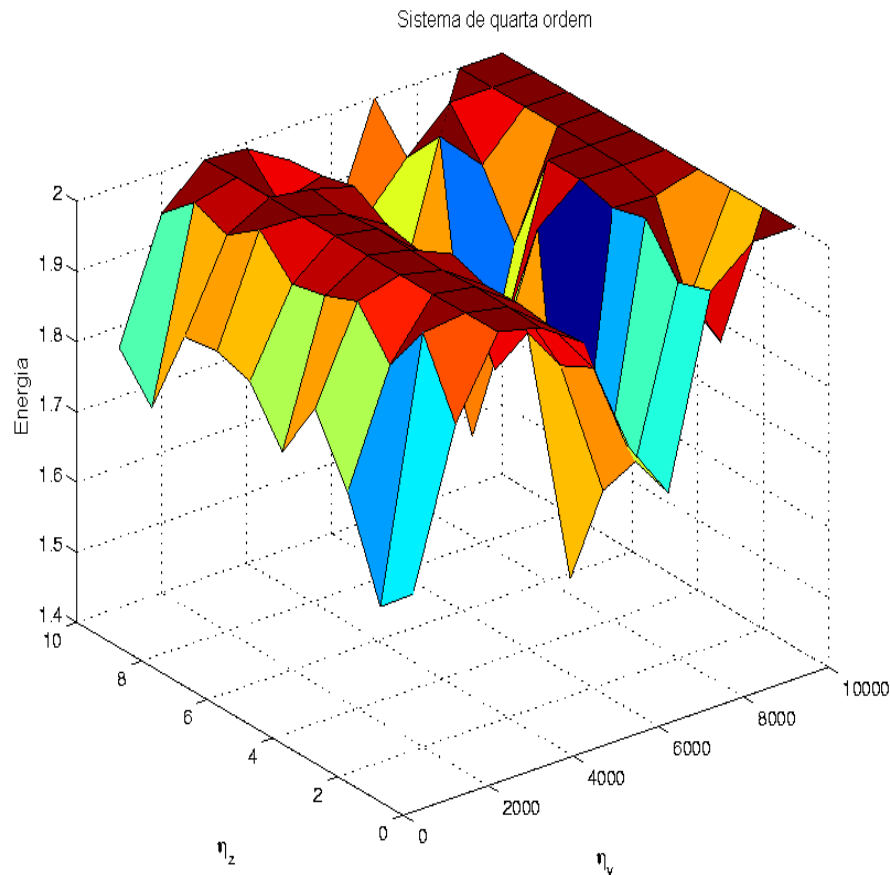


Figura 5.11: Superfície da energia para variação dos parâmetros  $\eta_v = 1000 - 10000$  e  $\eta_z = 1 - 10$ .

As superfícies da função energia, Fig. 5.11 e a norma do infinito Fig. 5.10, são insatisfatórias por estabelecer como primeiras condições de inicialização de sintonia dos parâmetros  $\eta_v$  e  $\eta_z$  para os valores descritos anteriormente.

Pode-se dizer que com a energia consumida, o erro mostra-se satisfatório sob o ponto de vista da solução da equação de *Riccati* com esta margem de erro e prevendo o ponto de operação do sistema.

Conseqüentemente, uma nova superfície é plotada para os parâmetros  $\eta_v$  e  $\eta_z$  nos intervalos  $[2000 \ 10000]$  e  $[1 \ 10^4]$ , respectivamente. Nas Figs. 5.12 e 5.13 são mostradas as superfícies da norma do infinito e energia, respectivamente. Estas

novas superfícies mostram uma concordância entre a norma do infinito e a função energia que permite estabelecer os parâmetros  $\eta_v$  e  $\eta_z$  que fornecem a solução única da EAR.

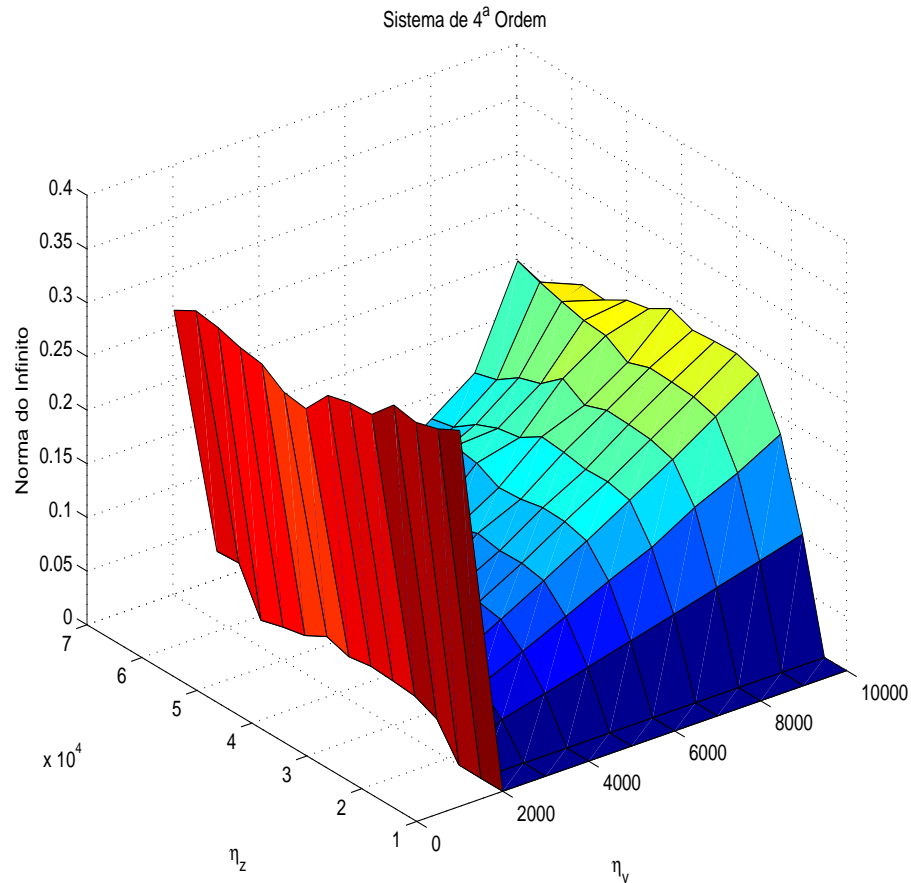


Figura 5.12: Superfície da norma do infinito para variação dos parâmetros  $\eta_v$  e  $\eta_z$ .

O primeiro ajuste para os parâmetros  $\eta_v$  e  $\eta_z$ , são escolhidos com base nessas superfícies. Os valores  $\eta_v = 100$  e  $\eta_z = 8$  são uma boa escolha para minimizar a função energia, bem como, o *mismatch* da norma do infinito da solução da equação de *Riccati*.



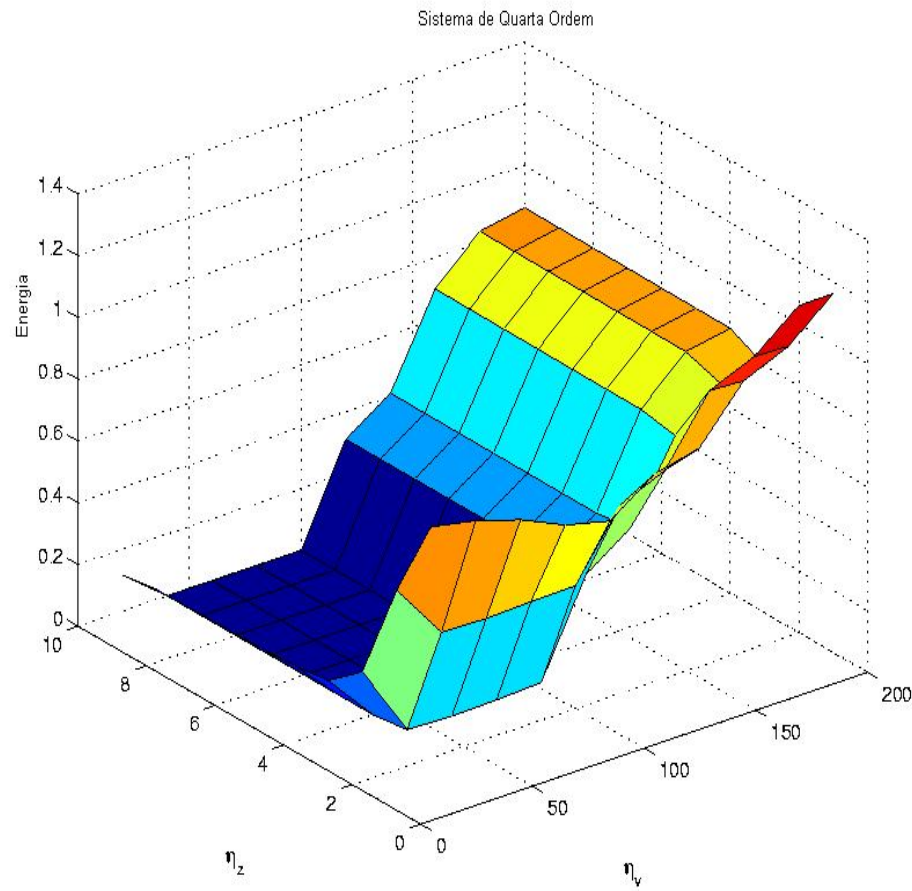


Figura 5.13: Superfície da energia para os parâmetros  $\eta_v = 20 - 200$  e  $\eta_z = 3 - 10$ .

Para as Figs. 5.14, 5.15 e 5.16, foram escolhidos um intervalo conveniente que resultou em uma boa solução:  $\eta_v = 100$ ,  $\eta_z = 8$ ,  $N_{amostra} = 800$  e  $\Delta T = 5 \times 10^{-5}$ . Ambas figuras representam o impacto do treinamento no comportamento da solução da EAR, ou seja, com o treinamento da rede neural, a função energia, a norma do infinito associada a EAR e a norma do infinito do fator de *Cholesky* diminuem.

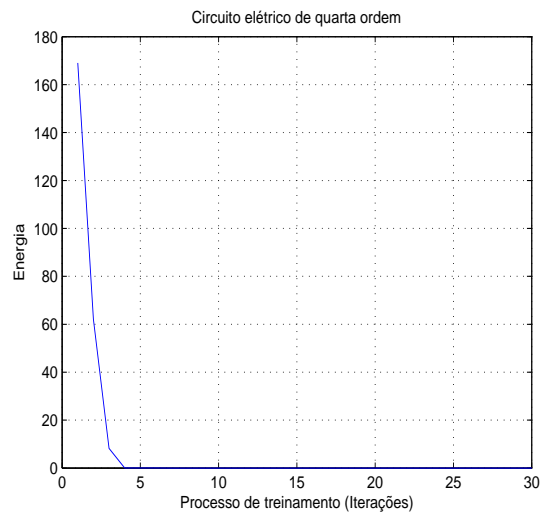


Figura 5.14: Energia em função do número de iterações.

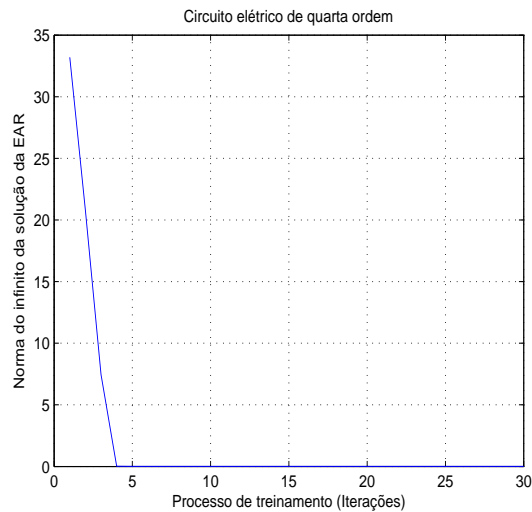


Figura 5.15: Norma do infinito da solução da EAR.

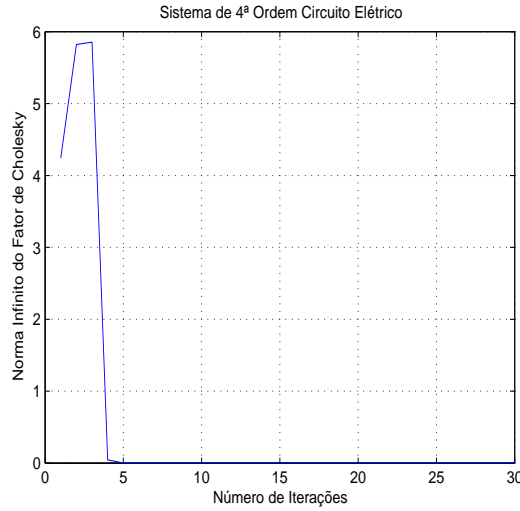


Figura 5.16: Norma do infinito do Fator de Cholesky

### Solução Neural da Equação de Riccati

A solução neural da equação algébrica de Riccati é baseada nas Eqs. (4.37-4.40) para resolver a estrutura de otimização que minimiza a função energia dada pela Eq. (4.31). A solução da EAR é a camada de saída da RNR dada por

$$P_{RNA} = \begin{bmatrix} 17.884 & 2.228 & 2.533 & 1.838 \\ 2.228 & 19.229 & 2.760 & -2.245 \\ 2.533 & 2.760 & 13.596 & -4.100 \\ 1.838 & -2.245 & -4.100 & 10.859 \end{bmatrix}. \quad (5.23)$$

Comparando a solução da EAR via RNA, Eq. (5.23), e a solução da EAR via método de *Schur*, Eq. (5.15), observa-se que todos os elementos são equivalentes pelo menos até na terceira casa. A taxa de convergência da solução  $P_{RNA}$  da Eq. (5.23) é estabelecida em termos da minimização função energia e norma do infinito da EAR. A avaliação da convergência é baseada na norma do infinito que sustenta o processo de otimização e expressa a dinâmica do processo de treinamento da rede neural.

A matriz de ganho usando a solução da equação algébrica de Riccati fornecida pela RNR, é dada por

$$K_{RNA} = \begin{bmatrix} 0.14531 & 0.22132 & 0.98711 & -0.44029 \\ 0.12725 & -0.24224 & -0.56916 & 1.0345 \end{bmatrix} \quad (5.24)$$

e que os ganhos  $K_{Schur}$ ,  $K_{RNA}$  e as soluções da EAR dadas por  $P_{Schur}$  e  $P_{RNA}$  verificam a Eq.  $A^T P + PA - PBR^{-1}B^T P = 0$ , como mostram as expressões a seguir:

$$Mismatch_{RicSchur} = \begin{bmatrix} -6.608e-013 & 3.1086e-013 & -4.2988e-013 & 4.9649e-013 \\ -1.4744e-013 & -2.8422e-013 & -2.9132e-013 & 2.82e-013 \\ -1.4477e-013 & 3.9435e-013 & -2.0961e-013 & 6.1284e-014 \\ 3.1974e-014 & 5.0182e-013 & 3.1086e-014 & 1.1369e-013 \end{bmatrix} \quad (5.25)$$

$$Mismatch_{RicRNA} = \begin{bmatrix} 1.9185e-013 & -2.4869e-014 & -4.885e-014 & -8.8818e-015 \\ -2.3981e-014 & 1.9895e-013 & -1.5987e-014 & 5.1958e-014 \\ -4.885e-014 & -1.5987e-014 & 8.5265e-014 & 2.6645e-015 \\ -7.9936e-015 & 5.5511e-014 & 2.6645e-015 & 9.2371e-014 \end{bmatrix} \quad (5.26)$$

O *Mismatch* associado à EAR é definido como os valores matriciais que a satisfazem. Para efeito de comparação, tomou-se os valores matriciais obtidos pelo método de *Schur* e da RNA, segundo as Eqs. (5.25-5.26). Pode-se dizer que ambas soluções da EAR,  $P_{Schur}$  e  $P_{RNA}$  são satisfatórias segundo os resultados exibidos pelos seus *mismatch*. Usamos a norma do infinito para as Eqs. (5.25)-(5.26) para mensurar o quão uma solução é melhor que a outra e obtemos os seguintes resultados:  $norm_{inf}(RNA) = 2.944e-013$  e  $norm_{inf}(Schur) = 1.898e-012$ . Percebe-se que a solução da EAR obtida pela RNR apresenta menor erro.

### Análise de Convergência

A análise de convergência conduz a dois aspectos na solução da EAR. O primeiro diz respeito a unicidade da solução, simetria e definida positiva e o segundo em relação a taxa para atingir um valor estável. A análise de desempenho, focalizada na velocidade da rede neural, tem como preocupação principal, atingir habilidades para avaliação do valor da solução em regime permanente, ou seja, a estabilização da norma do infinito das quatro camadas da RNR durante o processo iterativo.

Na Fig. 5.17 são apresentadas as normas do infinito das quatro camadas da RNR. Os valores da norma do infinito tendem a uma solução estável. Uma

observação importante está relacionada com as camadas  $U$  e  $Y$ . Estes dois gráficos mostram que todos os elementos das camadas  $U$  e  $Y$  são nulos quando  $t \rightarrow \infty$ . Conseqüentemente, uma solução estável da EAR atingiu a convergência, sendo esta alcançada em torno de 200 iterações.

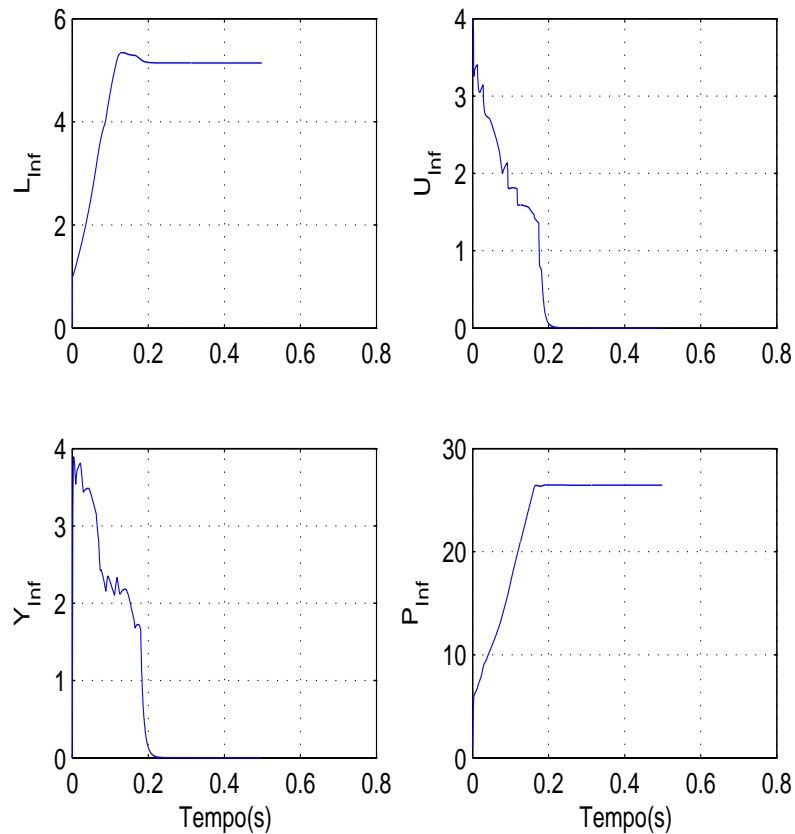


Figura 5.17: Norma do infinito das camadas de entrada, ocultas e saída para  $\eta_v = 100$  e  $\eta_z = 8$ .

Como um exemplo de uma solução estável, porém que não apresente comportamento da solução da EAR, é mostrada na Fig. 5.18, para os parâmetros  $\eta_v = 1000$  e  $\eta_z = 8$ . Como pode ser visto, as normas do infinito das camadas  $U$  e  $Y$  são acompanhadas por valores que possuem uma norma do infinito estável.

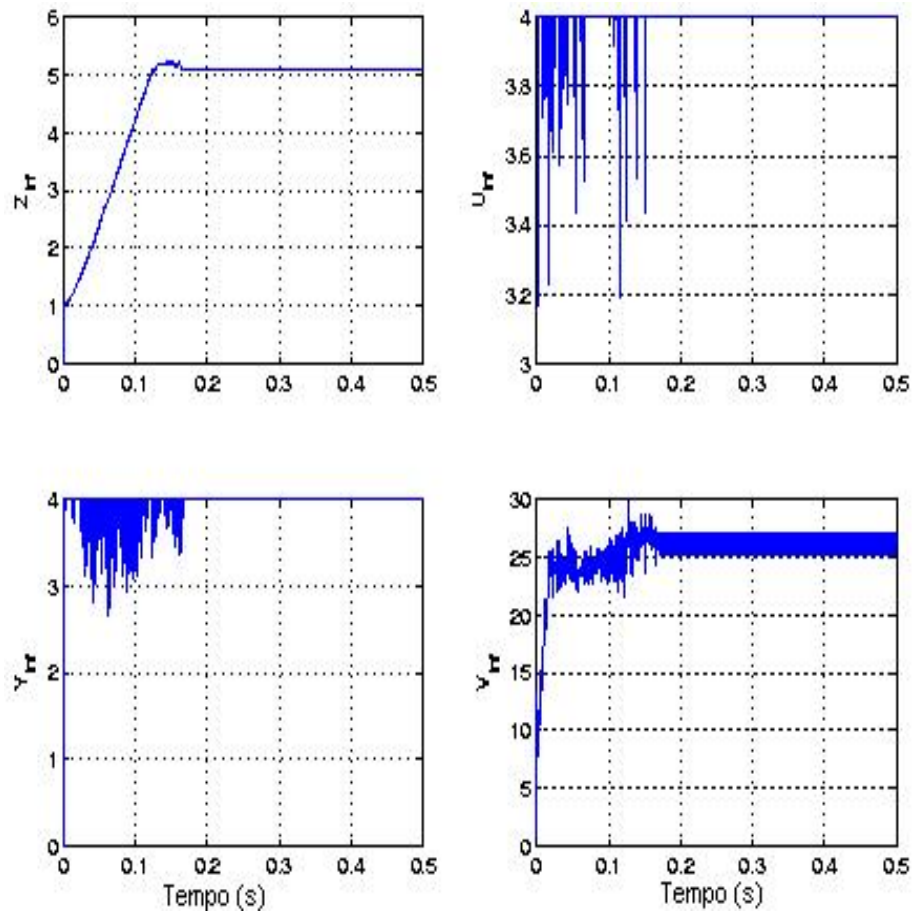


Figura 5.18: Solução associada à norma do infinito das camadas de entrada, oculta e saída para  $\eta_v = 1000$  e  $\eta_z = 8$ .

## 5.2 Projetos LQR Neuro-Genético: *DFIG* e Aeronave

Um modelo de sexta ordem representado por um gerador de indução duplamente alimentado para planta eólica, *DFIG*, Apêndice E e um modelo de sexta ordem representado por uma planta de uma aeronave, Apêndice F foram utilizados na aplicação do projeto LQR neuro-genético, com o objetivo de validarmos a metodologia por meio das abordagens de inteligência computacional, alocação de autoestrutura e resolução da equação de *Riccati*.

### 5.3 Discussão do Modelo de Fusão

Em uma primeira instância, a convergência do algoritmo genético e da rede neural recorrente foi avaliada de maneira desacoplada, como pode ser notado nas últimas subseções. A convergência do AG apresentou um bom desempenho para impor a especificada autoestrutura, porém a velocidade de convergência pode ser melhorada pelo seu paralelismo intrínseco. (Neto ,2000) e (M. Alba ,2002), realizam a seleção das matrizes de ponderação em tempo real. Do mesmo modo, a RNR pode ser melhorada pelo paralelismo natural e a implementação em tempo real foi desenvolvida por (Rocco Furferi e Daou ,2007) e outros.

O desempenho desacoplado do modelo de fusão proposto, conduz a uma discussão sobre algumas melhorias para acelerar a convergência em atender aos requisitos em tempo real, em que o impulsor da estratégia de controle deve executar ações e processar informações em um intervalo de tempo menor que as reações da planta controlada, (Mehdi Amirijoo e Son ,2008) e (Mehdi Amirijoo e Gunnarsson ,2007), ou seja, mudanças operacionais ou perturbações. Os desenvolvimentos adicionais da metodologia vão em direção à busca das matrizes  $QR$  e a solução da EAR via RNR, em atender os prazos limites dos sistemas dinâmicos.

### 5.4 Conclusão

Uma fusão de um AG e uma RNR para resolver o problema de autoestrutura, mostrou-se ser uma boa alternativa para realizar a seleção das matrizes de ponderação  $QR$  e computar a solução da EAR para sistemas dinâmicos de quarta ordem e de sexta ordem. Uma avaliação desacoplada dos paradigmas de inteligência computacional, apresentaram uma boa eficiência para realizar a alocação de autovalores e autovetores. Como uma contribuição para a teoria  $LQR$ , o método proposto promove melhorias no problema de controle de autoestrutura pelo algoritmo genético e na solução da equação algébrica de *Riccati*, pela rede neural artificial recorrente.

## CAPÍTULO 6

# CONCLUSÃO

---

Apresentou-se a estrutura montada e definida de pesquisa que está fundamentada nos pilares dos capítulos desta tese. Nesta abordagem apresentou-se uma metodologia para sintonia e síntese de sistemas de controle ótimo, que tem por base paradigmas de inteligência computacional, algoritmos genéticos e redes neurais artificiais. As matrizes de ponderação de estado e controle selecionadas pelo  $AG$  são usadas como parâmetros da equação algébrica de *Riccati* que é resolvida pela rede neural recorrente.

Realizou-se a avaliação da fusão de um modelo neuronal e um algoritmo genético que estão integrados na parametrização parcial da equação de *Riccati* e da sua solução. Esta avaliação é realizada por estatísticas de medida de tendência central e de dispersão, respectivamente na função de *fitness* associada ao modelo de busca das matrizes de ponderação do projeto  $LQR$  e também por superfícies da norma do infinito da EAR e função energia tendo como base as variações paramétricas nos pesos de aprendizagem da rede neural.

Para análise de desempenho em termos de solução do problema de controle, foram realizadas análises de estabilidade. Mostrou-se os modelos e um método, para análise de desempenho de um  $AG$  destinado à seleção das matrizes de ponderação para alocação de autoestrutura em sistemas MIMO. Percebe-se que a análise da autoestrutura é importante, pois mostra claramente a influência das relações entre os autovalores, autovetores, condições iniciais e entrada na composição da resposta temporal do sistema.

A análise de convergência evidenciou que o mapeamento da função *fitness* é



uma alternativa satisfatória para verificar o desempenho do *AG* em relação a formação de nichos, desta forma, busca-se contornar o problema relacionado com convergência prematura não factível. A análise estatística realizada por momentos de 1ª e 2ª ordem, pode incrementar melhorias no desenvolvimento do algoritmo, no sentido de reduzir o tempo de convergência e aumentar a quantidade de soluções viáveis. Portanto, o modelo de fusão mostra o comprometimento da metodologia utilizada no controle inteligente.

A comparação entre as metodologias realizadas pelos métodos de *Schur* e da RNR para resolver a equação algébrica de *Riccati* e entre as metodologias realizadas pelos métodos de *Bryson* e Algoritmos Genéticos para realizar a seleção das matrizes de ponderação, mostrou que a fusão é uma alternativa promissora para a sintonia de controladores robustos do tipo *LQR*. A abordagem proposta evidenciou, que as soluções computacionais inteligentes podem ser uma alternativa eficaz para o projeto e controle de sistemas em tempo real.

Apresentou-se uma abordagem de controle inteligente orientada para alocação de autoestrutura em sistemas dinâmicos multivariáveis. O método está apoiado em inteligência computacional, especificamente, redes neurais artificiais e algoritmos genéticos. Com a abordagem proposta da fusão, conclui-se que a mesma apresentou desempenho aceitável, uma vez que esta fusão está fundamentada em paradigmas das abordagens de computação evolutiva, *soft computing* e inteligência computacional.

## 6.1 Propostas Futuras

Como perspectiva futura, propõe-se continuar a pesquisa abordando restrições temporais que visam aplicações em tempo real: a convergência dos Algoritmos Genéticos e das Redes Neurais Artificiais, sincronização entre tempo de busca e solução da Equação Algébrica de *Riccati*. Especificamente, a pesquisa tende para os seguintes desenvolvimentos e investigações:

- Enfocar sob o ponto de vista de *AG*, sua convergência com pequenas populações, desenvolver estratégias de busca, que tem por base normas das matrizes de ponderação, utilizando pareto, articulação de preferências e lógica *fuzzy*;

- 
- Sob o ponto de vista das RNA, ajuste dos parâmetros  $\eta_v$  e  $\eta_z$  usando AG e lógica fuzzy;
  - Modelos de coordenação das atividades de busca das matrizes  $Q$ ,  $R$  e o ajuste dos parâmetros da RNA para fins de projeto e implementação.

# PROJETO DE SISTEMA LINEAR COM CRITÉRIO QUADRÁTICO

---

O estudo de projeto de sistemas ótimo no tempo, indica que existem dificuldades consideráveis na determinação da lei de controle ótimo, e dificuldade posterior na implementação do sistema de realimentação ótimo. Até mesmo se o sistema controlado é linear, é ainda quase impossível obter um resultado analítico geral para o caso ótimo no tempo. Pode-se perguntar se existe algum sistema e critério para um resultado geral que esteja disponível. Este Apêndice está fundamentado nos livros do (Athans e Falb ,2007) e (Lewis e Syrmos ,1995).

A seguir, obtém-se um resultado analítico para uma classe particular de sistemas e critérios de desempenho. De fato, examina-se sistemas lineares variantes no tempo cujo vetor de estado  $x(t)$ , vetor de controle  $u(t)$  e saída  $y(t)$  estão relacionados pelo sistema de equações

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t).\end{aligned}$$

A forma mais geral da função custo funcional que deve-se considerar é:

$$\begin{aligned}J(u) &= \frac{1}{2} \langle [z(T) - y(T)], F[z(T) - y(T)] \rangle + \\ &\frac{1}{2} \int_{t_0}^T \{ \langle [z(t) - y(t)], Q(t)[z(t) - y(t)] \rangle + \langle u(t), R(t)u(t) \rangle \} dt\end{aligned} \quad (\text{A.1})$$

sendo que  $z(t)$  representa a saída desejada.

Assume-se que não existem restrições nas magnitudes das componentes do vetor de controle  $u(t)$ , e deve-se assumir certas hipóteses a respeito das matrizes  $F$ ,  $Q(t)$  e  $R(t)$ . Sob essas hipóteses, deve-se estar apto a obter uma expressão analítica para o controle ótimo, e portanto, garantir que o sistema de realimentação ótimo é linear.

## A.1 Formulação do Problema

A seguir, discute-se a formulação do problema ótimo. Para a escolha da função custo funcional  $J(u)$  enfatiza-se a importância do significado físico, sob algumas hipóteses matemáticas das matrizes já citadas.

Suponha que seja dado um sistema dinâmico linear variante no tempo Eq. (A.1). Assume-se que  $x_1(t), x_2(t), \dots, x_n(t)$  são as componentes do vetor de estado  $x(t)$ ,  $u_1(t), u_2(t), \dots, u_r(t)$  são as componentes do vetor  $u(t)$  e que  $y_1(t), y_2(t), \dots, y_m(t)$  são as componentes do vetor de saída  $y(t)$ . Portanto,  $A(t) \in R^{n \times n}$ ,  $B(t) \in R^{n \times r}$  e  $C(t) \in R^{m \times n}$ . Além disso, assume-se que

$$0 < m \leq r \leq n \quad (\text{A.2})$$

e que  $u(t)$  é irrestrito.

Define-se o objetivo do sistema a partir de um ponto de vista físico. Seja  $z(t)$  um vetor com  $m$  componentes. O objetivo é controlar o sistema Eq.(A.1) de modo que o vetor de saída  $y(t)$  esteja próximo ao vetor  $z(t)$ . Convenciona-se o vetor  $z(t)$  como a saída desejada. Pode-se então definir um *vetor erro* fazendo

$$e(t) = z(t) - y(t). \quad (\text{A.3})$$

Deste modo, o objetivo do controle é: Obter uma lei de controle  $u(t)$  de maneira que o erro  $e(t)$  seja pequeno.

Conforme suposto anteriormente, o vetor de controle  $u(t)$  é irrestrito em magnitude; portanto, existem casos em que o controle  $u(t)$  é extremamente grande. A fim de evitar tais situações não realísticas (que requer ganhos extremamente grandes na malha de controle), pode-se desejar incluir em nosso objetivo de controle

uma caracterização a respeito do fato que sinais de controle alto tem custo elevado. Em outras palavras, gostaríamos, por um lado, de manter o erro  $e(t)$  pequeno, porém, por outro lado, não devemos usar sinais de controle desnecessariamente grandes.

A translação dessas especificações físicas e exigências sobre um particular funcional de desempenho matemático é, de fato, até engenhoso. Escolhe-se uma classe particular de custos funcionais, quadrático por natureza, que realmente corresponda as exigências físicas. É certamente verdadeiro que esta escolha dos custos funcionais pode não ser sensata em uma dada situação; em tal caso, o engenheiro deve desenvolver seu próprio critério de desempenho e ele mesmo descobrir o problema. Para auxiliar o engenheiro a compreender o significado dos custos funcionais deve-se considerar, a discussão a seguir do significado físico sobre os termos matemáticos.

Para ser preciso, considere o custo funcional

$$J(u) = \frac{1}{2} \langle e(T), Fe(T) \rangle + \frac{1}{2} \int_{t_0}^T \{ \langle e(t), Q(t)e(t) \rangle + \langle u(t), R(t)u(t) \rangle \} dt \quad (\text{A.4})$$

sendo  $t_0$  o tempo inicial,  $T$  o tempo final especificado,  $F \in \mathbb{R}^{m \times m}$  uma matriz constante,  $Q(t) \in \mathbb{R}^{n \times n}$ ,  $Q(t) \geq 0$  e  $R(t) \in \mathbb{R}^{r \times r}$ ,  $R(t) > 0$ .

Considere agora cada termo na função custo funcional  $J(u)$ , Eq.(A.4), e veja de que modo ela representa uma translação matemática razoável das especificações e exigências físicas. Primeiro, considere o termo  $L_e = \frac{1}{2} \langle e(t), Q(t)e(t) \rangle$ . Claramente  $L_e \geq 0 \forall e(t)$  e, em particular,  $L_e = 0$  quando  $e(t) = 0$ . Se o erro  $e(t)$  é pequeno  $\forall t \in [t_0, T]$ , então a integral de  $L_e$  será pequena.

Desde que  $L_e = \frac{1}{2} \langle e(t), Q(t)e(t) \rangle = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m q_{ij}(t) e_i(t) e_j(t)$  sendo  $q_{ij}(t)$  as entradas da matriz  $Q(t)$ ,  $e_i(t)$  e  $e_j(t)$  as componentes do vetor  $e(t)$ , é claro que o custo  $L_e$  pondera erros maiores muito mais fortemente que erros menores, e assim o sistema é penalizado muito mais severamente para erros maiores que para erros menores.

Em seguida, considere o termo  $L_u = \frac{1}{2} \langle u(t), R(t)u(t) \rangle$ . Este termo pondera o custo do controle e penaliza o sistema muito mais severamente para sinais de controle grandes que para sinais de controle menores. Como  $R(t) > 0$ , o custo

funcional  $L_u > 0, \forall u(t) \neq 0$ . Este termo geralmente é denominado de potência de controle, e  $\int_{t_0}^T L u dt$  é chamado de energia de controle. A razão para essa nomenclatura é a seguinte: Suponha que  $u(t)$  seja um escalar proporcional a voltagem ou corrente; então  $u^2(t)$  é proporcional à potência e  $\int_{t_0}^T dt$  é proporcional à energia despendida no interval  $[t_0, T]$ . A exigência que a matriz  $R(t)$  seja definida positiva em vez de semidefinida positiva é, como deve-se ver, uma condição para a existência de uma lei de controle finita.

Finalmente, dirige-se a atenção para o termo  $\frac{1}{2}\langle e(T), Fe(T) \rangle$ . Este termo geralmente é denominado de custo final, e seu propósito é garantir que o erro  $e(T)$  no instante final  $T$  seja pequeno. Em outras palavras, este termo deve ser incluído se o valor de  $e(t)$  no tempo final é particularmente importante; se isto não é o caso, então pode se estabelecer  $F = 0$  e confiar no resto do custo funcional para garantir que o erro final não seja excessivamente grande.

Uma questão que pode surgir é a seguinte: por que razão as matrizes  $Q(t)$  e  $R(t)$  são variantes no tempo em vez de serem constantes? Se desenvolvermos uma teoria para matrizes  $Q(t)$  e  $R(t)$  variantes no tempo em vez de matrizes constantes  $Q$  e  $R$ , então pode-se usar o custo funcional  $J(u)$  para problemas mais realísticos. Por exemplo, suponha que em  $t = t_0$ , o estado  $x(t_0)$  é dado e conseqüentemente a saída  $y(t_0) = x(t_0)$  seja conhecida; então pode acontecer que a saída desejada  $z(t_0)$  seja tal que o erro inicial  $e(t_0)$  seja muito grande. O fato que  $e(t_0)$  seja grande, certamente não é por imperfeição do sistema controle. Por essa razão, pode-se querer escolher  $Q(t)$  de tal maneira que erros iniciais grandes sejam penalizados menos severamente que erros grandes subseqüentes. Para fazer isso, pode-se escolher  $Q(t)$  de tal modo que, dado os tempos

$$t_0 < t_1 \ll t_2 < T \quad (\text{A.5})$$

e qualquer vetor constante  $b$ , os produtos escalares  $\langle b, Q(t_1)b \rangle$  e  $\langle b, Q(t_2)b \rangle$  satisfaz a relação

$$\langle b, Q(t_1)b \rangle \leq \langle b, Q(t_2)b \rangle. \quad (\text{A.6})$$

É nossa esperança que a discussão precedente das propriedades físicas do custo funcional  $J(u)$  ajudará a compreender suas propriedades. Deseja-se enfatizar se

isto deve ser usado ou não para um dado problema até o projetista. Todavia, o custo funcional possui duas propriedades muito desejáveis: Primeira, é matematicamente tratável, e segundo, conduz a sistemas de realimentação ótimo que são lineares. Com a motivação física, iremos estabelecer a caracterização formal do problema.

O problema que deve-se considerar é o seguinte: Dado o sistema linear Eq. (A.1) e o custo funcional Eq. (A.4) que satisfaçam as hipóteses relativas às Eqs. (A.2 - A.3) e matrizes  $F$ ,  $Q(t)$  e  $R(t)$  respectivamente. Obter o controle ótimo, isto é, a lei de controle que governará o sistema Eq. (A.1) de modo a minimizar o custo funcional, Eq. (A.4).

Antes de tentar a solução deste problema, vamos obter alguns resultados úteis, porém evidentes. Primeiro de todos, se  $Q(t) = F = 0$ , então o controle ótimo  $u(t) = 0$ ; neste caso, o custo é  $J(u) = \frac{1}{2} \int_{t_0}^T \langle u(t), R(t)u(t) \rangle dt$ , e uma vez que não especificamos o vetor de estado final  $x(T)$ , o custo é minimizado se e somente se  $u(t) = 0$ . Para excluir este caso trivial, assume-se que  $Q(t)$  e  $F$  são matrizes não-nulas, embora possamos permitir que a matriz  $Q(t)$  ou  $F$  possa ser a matriz nula individualmente.

Agora suponha que  $\Phi(t; t_0)$  seja a matriz fundamental do sistema linear Eq.(A.1) e seja  $x(t_0)$  o estado inicial. Então o estado  $x(t)$  para  $t \in [t_0, T]$ , é dado pela Eq.(A.7)

$$x(t) = \Phi(t; t_0)[x(t_0) + \int_{t_0}^t \Phi^{-1}(\tau; t_0)B(\tau)u(\tau)d\tau] \quad (\text{A.7})$$

e conseqüentemente, o erro  $e(t)$  é dado pela equação

$$\begin{aligned} e(t) &= z(t) - y(t) = z(t) - C(t)x(t) \\ &= z(t) - C(t)\Phi(t; t_0)[x(t_0) + \int_{t_0}^t (\tau; t_0)B(\tau)u(\tau)d\tau] \end{aligned} \quad (\text{A.8})$$

Se o estado inicial  $x(t_0)$  e a saída desejada  $z(t)$  estão relacionadas por  $z(t) = C(t)\Phi(t; t_0)x(t_0)$ ,  $\forall t \in [t_0, T]$ , então o controle ótimo é  $u(t) = 0$  e o custo mínimo  $J^* = 0$ .

**Prova** Substitua  $u(\tau) = 0$  na Eq. (A.8) para obter  $e(t) = 0$ .

**Lema A.1.1.** *Se  $u(t) \neq 0$ , então o custo funcional  $J > 0$ .*

**Prova** Segue da hipótese que as matrizes  $F$  e  $Q(t)$  são semidefinidas positiva e que  $R(t) > 0$ .

**Lema A.1.2.** *Se para todo  $u(t)$ , o custo funcional  $J(u)$  não está definido, isto é, se  $J(u) = \infty, \forall u(t)$ , então não existe uma solução ótima.*

**Prova** Suponha que dois controles geram um custo infinito, então não podemos distinguir quais dos dois controles é o melhor.

## A.2 O Problema do Regulador de Estado

Inicia-se esta Seção com o desenvolvimento matemático para o problema do regulador de estado (isto é, o problema de manter o estado próximo a zero). Usa-se o princípio mínimo para desenvolver as condições necessárias sobre o controle ótimo. Mostra-se que as equações canônicas relacionadas aos vetores de estado  $x(t)$  e co-estado  $p(t)$  são lineares. A linearidade das equações canônicas nos permite concluir que esses vetores estão relacionados pela equação da forma  $p(t) = K(t)x(t)$ , sendo  $K(t)$  uma matriz real simétrica. Considerações relativas a computação da matriz  $K(t)$  nos levam a lei de controle Eq. (A.1). Estabelece-se que as hipóteses nos permitem concluir que as condições necessárias são suficientes globalmente. Portanto, estamos aptos a provar que o controle ótimo existe, e é além disso, único.

Considere o problema do regulador de estado. Basicamente, a solução do problema do regulador de estado leva a um sistema de realimentação ótimo com a propriedade que as componentes do vetor de estado  $x(t)$  são mantidas próximas a zero sem consumo excessivo de controle de energia. Agora resume-se a estrutura desta seção.

Inicia-se aplicando-se as condições necessárias fornecidas pelo princípio mínimo para obter os controles extremos. Mostra-se que:

1. O controle mínimo  $H$  resulta em uma combinação linear do vetor de co-estado.
2. As equações canônicas se reduzem a um sistema de equações diferenciais lineares homogêneo do tipo  $2n$ .



3. A condição de transversalidade sobre o vetor de co-estado  $p(t)$  no tempo final  $T$  especifica que o co-estado  $p(T)$  é uma função linear do não especificado estado final  $x(T)$ .

Motivado por considerações computacionais, deve-se então mostrar que o controle extremo é uma função linear dos estados. Deve-se descobrir que a matriz variante no tempo relacionada ao controle extremo ao vetor de estado, pode ser encontrada resolvendo uma equação diferencial matricial (a equação matricial de *Riccati*). Prova-se que o controle extremo é pelo menos local ótimo e único. A existência do controle ótimo será deduzida a partir do fato que a solução da equação de *Hamilton-Jacobi* é definida em toda parte.

Considere o sistema linear variante no tempo

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (\text{A.9})$$

e o custo funcional  $J_1$  dado por

$$J_1 = \frac{1}{2} \langle x(T), Fx(T) \rangle + \frac{1}{2} \int_{t_0}^T [\langle x(t), Q(t)x(t) \rangle + \langle u(t), R(t)u(t) \rangle] dt \quad (\text{A.10})$$

Note que se fixarmos  $C(t) = I$  e  $z(t) = 0$ , então  $y(t) = x(t) = -e(t)$ . De fato:  $e(t) = z(t) - y(t)$ . Mas  $y(t) = C(t)x(t) = I.x(t) = x(t)$ . Como  $z(t) = 0$ ,  $e(t) = 0 - y(t) = -y(t)$ . Daí,  $y(t) = -e(t)$ . Portanto  $y(t) = x(t) = -e(t)$  e o custo funcional  $J(u)$  da Eq. (A.4) reduz-se ao funcional  $J_1$  da Eq. (A.10). A interpretação física do funcional  $J_1$  é que: deseja-se manter o vetor de estado próximo de zero sem consumo excessivo do controle de energia. Ainda assume-se que as hipóteses a respeito das matrizes  $F$ ,  $Q(t)$  e  $R(t)$  são satisfeitas.

Vamos mostrar que a lei de controle ótimo é uma função linear dos estados, ou seja, é da forma

$$u(t) = G(t)x(t), \quad \forall t \in [t_0, T] \quad (\text{A.11})$$

sendo  $G(t) \in R^{r \times n}$ .

Assume-se que existe um controle ótimo para qualquer estado inicial. Pode-se usar o princípio mínimo para obter as condições necessárias para o controle

ótimo e assim derivar os controles extremos. O *Hamiltoniano*  $H$  para o sistema da Eq.(A.9) e o custo funcional  $J_1$  da Eq. (A.10) é

$$H(x, u, p) = \frac{1}{2}\langle x(t), Q(t)x(t) \rangle + \frac{1}{2}\langle u(t), R(t)u(t) \rangle + \langle A(t)x(t), p(t) \rangle + \langle B(t)u(t), p(t) \rangle. \quad (\text{A.12})$$

O vetor de co-estado  $p(t)$  é a solução da equação diferencial vetorial

$$\dot{p}(t) = -\frac{\partial H}{\partial x(t)} \quad (\text{A.13})$$

que leva a

$$\dot{p}(t) = -Q(t)x(t) - A^T(t)p(t). \quad (\text{A.14})$$

Ao longo da trajetória ótima, deve-se ter  $\frac{\partial H}{\partial u(t)} = 0$  o que implica em

$$\frac{\partial H}{\partial u(t)} = R(t)u(t) + B^T(t)p(t) = 0. \quad (\text{A.15})$$

Da Eq. (A.15), deduz-se que

$$u(t) = -R^{-1}(t)B^T(t)p(t). \quad (\text{A.16})$$

A hipótese que  $R(t) > 0 \forall t \in [t_0, T]$  garante a existência da  $R^{-1}(t) \forall t \in [t_0, T]$ . Sabe-se que o controle ótimo deve minimizar o *Hamiltoniano*  $H$ . A condição necessária  $\frac{\partial H}{\partial u(t)} = 0$  gera somente um ponto extremo de  $H$  com relação a  $u(t)$ . A fim de que o extremo de  $H$  seja um ponto de mínimo com relação a  $u(t)$ , a matriz  $\frac{\partial^2 H}{\partial u^2(t)} \in R^{r \times r}$  deve ser definida positiva. De fato: Da Eq. (A.15),

$$\frac{\partial^2 H}{\partial u^2(t)} = R(t) > 0, \quad (\text{A.17})$$

e conseqüentemente, uma vez que  $R(t) > 0$ , segue que a lei de controle  $u(t)$  é dada pela Eq.(A.16) que realmente minimiza o *Hamiltoniano*  $H$  e, conseqüentemente é  $H$ -minimal.

O próximo passo é obter as equações canônicas reduzidas; para isso, substitui-se a Eq. (A.16) na Eq. (A.9) para obter a Eq. (A.18)

$$\dot{x}(t) = A(t)x(t) - B(t)R^{-1}(t)B^T(t)p(t). \quad (\text{A.18})$$

As Eqs. (A.18-A.14) são as canônicas reduzidas. Define-se a matriz

$$S(t) \triangleq B(t)R^{-1}(t)B^T(t). \quad (\text{A.19})$$

Usando a matriz  $S(t)$ , pode-se combinar as Eqs. canônicas (A.18-A.14) na forma matricial

$$\begin{bmatrix} \dot{x}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} A(t) & -S(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{bmatrix} x(t) \\ p(t) \end{bmatrix} \quad (\text{A.20})$$

A Eq. (A.20) é um sistema de equações de  $2n$  diferenciais lineares homogêneas variando no tempo. Sabe-se que podemos obter uma solução única deste sistema de equações diferenciais, estabelecendo que se conhece um total de  $2n$  condições de contorno. Um total de  $n$  condições de contorno é fornecido pelo estado inicial em  $t_0$ ,  $x(t_0)$ . O restante das  $n$  condições de contorno são fornecidas pelas condições de transversabilidade, que exige que (desde que  $x(T)$  não seja especificado), no tempo final  $T$ , o vetor de co-estado  $p(T)$  deve satisfazer a Eq.(A.21)

$$p(T) = \frac{\partial}{\partial x(T)} \left[ \frac{1}{2} \langle x(T), Fx(T) \rangle \right] \quad (\text{A.21})$$

Dessa forma, deduz-se que

$$p(T) = Fx(T) \quad (\text{A.22})$$

Seja  $\Omega(t; t_0)$  a matriz fundamental do tipo  $2n \times 2n$  para o sistema da Eq. (A.20). Seja  $p(t_0)$  o vetor de co-estado inicial (desconhecido), então a solução da Eq. (A.20) é da forma

$$\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = \Omega(t; t_0) \begin{bmatrix} x(t_0) \\ p(t_0) \end{bmatrix}. \quad (\text{A.23})$$

Portanto, em  $t = T$ , tem-se a relação

$$\begin{bmatrix} x(T) \\ p(T) \end{bmatrix} = \Omega(T; t) \begin{bmatrix} x(t) \\ p(t) \end{bmatrix}. \quad (\text{A.24})$$

Em seguida, particiona-se a matriz  $\Omega(T; t) \in \mathbb{R}^{2n \times 2n}$  em quatro sub-matrizes do tipo  $n \times n$  como segue:

$$\Omega(T; t) = \begin{bmatrix} \Omega_{11}(T; t) & \Omega_{12}(T; t) \\ \Omega_{21}(T; t) & \Omega_{22}(T; t) \end{bmatrix}. \quad (\text{A.25})$$

Então a Eq. (A.24) pode ser escrita na forma (usando  $p(T) = Fx(T)$ )

$$\begin{aligned} x(T) &= \Omega_{11}(T; t)x(t) + \Omega_{12}(T; t)p(t) \\ p(T) &= \Omega_{21}(T; t)x(t) + \Omega_{22}(T; t)p(t) = Fx(T) \end{aligned} \quad (\text{A.26})$$

Das Eqs. (A.26), encontra-se após algumas manipulações algébricas, que

$$p(t) = [\Omega_{22}(T; t) - F\Omega_{12}(T; t)]^{-1}[F\Omega_{11}(T; t) - \Omega_{21}(T; t)]x(t) \quad (\text{A.27})$$

desde que a inversa exista. A Eq. (A.27) sugere que os vetores de estado e co-estado estejam relacionados pela Eq. (A.28)

$$p(t) = K(t)x(t), \quad \forall t \in [t_0, T]. \quad (\text{A.28})$$

A matriz  $K(t) \in \mathbb{R}^{n \times n}$  é uma matriz variante no tempo, que depende do tempo final  $T$  e da matriz  $F$ , porém não depende do estado inicial. De fato,

$$K(t) = [\Omega_{22}(T; t) - F\Omega_{12}(T; t)]^{-1}[F\Omega_{11}(T; t) - \Omega_{21}(T; t)]. \quad (\text{A.29})$$

Neste ponto, pequenos comentários podem ser úteis. Não mostramos que a matriz  $[\Omega_{22}(T; t) - F\Omega_{12}(T; t)]$  admite inversa. Examinemos a situação em  $t = T$ .

$$\Omega(T : T) = I \quad (\text{A.30})$$

de modo que as relações

$$\Omega_{11}(T; T) = \Omega_{22}(T; T) = I \quad (\text{A.31})$$

$$\Omega_{12}(T; T) = \Omega_{21}(T; T) = 0 \quad (\text{A.32})$$

se mantém. Portanto, a matriz  $[\Omega_{22}(T; t) - F\Omega_{12}(T; t)]$  é não-singular. Além disso,

$$K(T) = [\Omega_{22}(T; T) - F\Omega_{12}(T; T)]^{-1}[F\Omega_{11}(T; T) - \Omega_{21}(T; T)]. \quad (\text{A.33})$$

de modo que, devido as Eqs. (A.30-A.31), tem-se

$$K(T) = F. \quad (\text{A.34})$$

Portanto, a Eq. (A.28) mantém-se em  $t = T$ . Pode ser mostrado que a requerida inversa existe  $\forall t, t_0 \leq t \leq T$ , de modo que a relação fornecida pela Eq. (A.28) é válida.

Agora, faz-se alguns comentários a respeito da avaliação da matriz  $K(t)$ . Se as matrizes  $A(t)$ ,  $S(t)$  e  $Q(t)$  são variantes no tempo, então é impossível, em geral, obter uma expressão analítica para a matriz fundamental  $\Omega(T; t) \in \mathbb{R}^{2n \times 2n}$ . Neste caso, deve-se avaliar a matriz  $K(t)$  usando um computador digital. Se, entretanto, as matrizes  $A(t)$ ,  $S(t)$  e  $Q(t)$  são invariantes no tempo, então a matriz  $\Omega(T; t)$  pode ser avaliada analiticamente usando, por exemplo, Transformadas de *Laplace*. Todavia, mesmo nesse caso, a avaliação da matriz inversa na Eq. (A.27) é uma tarefa extremamente laboriosa, especialmente se a ordem do sistema é alta, isto é, se  $n$  é um número grande.

Parece até agora, que as dificuldades envolvidas a partir do ponto de vista computacional são quase insuperáveis. A questão natural que surge é a seguinte: justificada a relação  $p(t) = K(t)x(t)$ , pode-se obter propriedades adicionais sobre

a matriz  $K(t)$  que não envolva computação da inversa de uma matriz do tipo  $n \times n$ ? A resposta para essa questão é sim.

Considere a relação  $p(t) = K(t)x(t)$ ,  $t \in [t_0, T]$ , e diferencia-se com relação ao tempo; deve-se então substituir as equações derivadas nas equações canônicas, e deve-se chegar a conclusão que a matriz  $K(t)$  deva satisfazer uma certa equação diferencial matricial. Suponha que  $x(t)$  e  $p(t)$ , as soluções das Equações canônicas estejam relacionadas pela Eq. (A.35)

$$p(t) = K(t)x(t), \quad t \in [t_0, T]. \quad (\text{A.35})$$

Diferenciando em relação ao tempo, obtém-se

$$\dot{p}(t) = \dot{K}(t)x(t) + K(t)\dot{x}(t). \quad (\text{A.36})$$

Das Equações canônicas, conhece-se as Eqs. (A.37-A.38)

$$\dot{x}(t) = A(t)x(t) - S(t)p(t) \quad (\text{A.37})$$

$$\dot{p}(t) = -Q(t)x(t) - A^T(t)p(t). \quad (\text{A.38})$$

Substituindo-se a Eq. (A.35) na Eq. (A.37), obtém-se a Eq. (A.39)

$$\dot{x}(t) = [A(t) - S(t)K(t)]x(t). \quad (\text{A.39})$$

Substituindo-se a Eq. (A.35) na Eq. (A.37), chega-se a Eq. (A.40)

$$\dot{p}(t) = [-Q(t) - A^T(t)K(t)]x(t). \quad (\text{A.40})$$

Das Eqs. (A.39-A.40), chega-se a Eq. (A.41)

$$[\dot{K}(t) + K(t)A(t) - K(t)S(t)K(t) + A^T(t)K(t) + Q(t)]x(t) = 0, \quad \forall t \in [t_0, T] \quad (\text{A.41})$$

Uma vez que: a Eq. (A.41) mantém-se para qualquer escolha do estado inicial, a matriz  $K(t)$  não depende do estado inicial, e que  $x(t)$  é a solução da Eq. (A.39)

homogênea, segue que a Eq. (A.41) deve não se manter não importando qual seja o valor de  $x$ . Isto significa que a matriz  $K(t)$  deve satisfazer a Eq. (A.42) diferencial matricial

$$\dot{K}(t) + K(t)A(t) + A^T(t)K(t) - K(t)S(t)K(t) + Q(t) = 0 \quad (\text{A.42})$$

Mas,  $S(t) = B(t)R^{-1}(t)B^T(t)$ , e então a Eq. (A.42) pode ser escrita como Eq. (A.43)

$$\dot{K}(t) = -K(t)A(t) - A^T(t)K(t) + K(t)B(t)R^{-1}(t)B^T(t)K(t) - Q(t). \quad (\text{A.43})$$

Desta maneira, estabelece-se o seguinte lema.

**Lema A.2.1.** *Se os vetores de estado e co-estado  $x(t)$ ,  $p(t)$  são soluções das Eqs. canônicas (A.20) e se  $p(t) = K(t)x(t)$ ,  $\forall t \in [t_0, T]$  e  $\forall x(t)$ , então  $K(t)$  satisfaz a Eq. (A.43)*

Voltamos agora a nossa atenção para a condição de contorno em  $t = T$ . As condições de transversabilidade requerem que

$$p(T) = Fx(T). \quad (\text{A.44})$$

Mas em  $t = T$ , temos também a relação

$$p(T) = K(T)x(T) \quad (\text{A.45})$$

e deste modo conclui-se que

$$[K(T) - F]x(T) = 0, \quad \forall x(T) \quad (\text{A.46})$$

e, portanto,

$$K(T) = F \quad (\text{A.47})$$

o que é equivalente ao encontrado na Eq. (A.34)

A Eq. (A.43) é denominada equação diferencial matricial de *Riccati*, ou simplesmente de equação de *Riccati*. A Eq. (A.47) nos fornece as condições de contorno para a equação de *Riccati*, e então pelo teorema da existência e unicidade, a solução  $K(t)$  da equação de *Riccati* existe e é única.

A primeira vista, pode parecer que a matriz  $K(t) \in R^{n \times n}$ , Eq. (A.43), represente um sistema de  $n^2$  equações diferenciais de primeira ordem. Mostraremos a seguir que a matriz  $K(t)$  é simétrica.

**Lema A.2.2.** *Se a matriz  $K(t)$  é a solução da equação de Riccati, Eq. (A.43) e se  $K(T) = F$ , então  $K(t)$  é simétrica  $\forall t \in [t_0, T]$ , isto é,*

$$K(t) = K^T(t). \quad (\text{A.48})$$

**Prova** Tomando a transposta em ambos os lados da Eq. (A.43), temos que:

$$\left[\frac{d}{dt}K(t)\right]^T = -K^T(t)A(t) - A^T(t)K^T(t) + K^T(t)B(t)R^{-1}(t)B^T(t)K^T(t) - Q(t). \quad (\text{A.49})$$

desde que  $Q(t)$  e  $B(t)R^{-1}(t)B^T(t)$  sejam simétricas. Mas, para qualquer matriz  $K(t)$ , a equação

$$\left[\frac{d}{dt}K(t)\right]^T = \frac{d}{dt}[K^T(t)] \quad (\text{A.50})$$

é verdadeira. Usando a Eq. (A.50), compara-se as Eqs. (A.49-A.43), e observa-se que ambas matrizes  $K(t)$  e  $K^T$  são soluções da mesma equação diferencial. Em  $t = T$ , tem-se a condição de contorno  $K(T) = K$ . Desde que  $F$  seja uma matriz simétrica,  $F = F^T$ ; porém,  $K^T(T) = F^T = F$ , assim conclui-se que

$$K^T(T) = F^T = F. \quad (\text{A.51})$$

Portanto,  $K(t)$  e  $K^T(t)$  são soluções da mesma equação diferencial com as mesmas condições de contorno. A partir da unicidade de soluções de equações diferenciais, conclui-se que  $K(t) = K^T(t)$ .



### Lei de Controle: Solução para o Problema do Regulador de Estado

Dado o sistema linear

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (\text{A.52})$$

e o custo funcional dado por

$$J_1 = \frac{1}{2} \langle x(T), Fx(T) \rangle + \frac{1}{2} \int_{t_0}^T [\langle x(t), Q(t)x(t) \rangle + \langle u(t), R(t)u(t) \rangle] dt \quad (\text{A.53})$$

sendo  $u(t)$  irrestrito,  $T$  é especificado,  $F$ ,  $Q(t)$  são matrizes semi-definidas positivas e  $R(t)$  definida positiva. Então existe uma lei de controle ótimo, única, e é dada pela Eq. (A.54)

$$u(t) = -R^{-1}(t)B^T(t)K(t)x(t) \quad (\text{A.54})$$

sendo  $K(t) \in R^{n \times m}$  a única solução da equação de *Riccati*, Eq. (A.43), satisfazendo a condição de contorno

$$K(T) = F. \quad (\text{A.55})$$

O estado do sistema ótimo é então a solução da equação diferencial linear

$$\dot{x}(t) = [A(t) - B(t)R^{-1}(t)B^T(t)K(t)]x(t), \quad x(t_0) = \xi. \quad (\text{A.56})$$

Estabelece-se inicialmente que o controle extremo  $u(t)$  dado pela Eq. (A.54) represente pelo menos um mínimo local do custo funcional  $J_1$ . Suponha que a matriz (A.57)

$$\begin{bmatrix} \frac{\partial^2 H}{\partial x^2(t)} & \frac{\partial^2 H}{\partial x(t)\partial u(t)} \\ \frac{\partial^2 H}{\partial u(t)\partial x(t)} & \frac{\partial^2 H}{\partial u^2(t)} \end{bmatrix} \quad (\text{A.57})$$

seja definida positiva, então o controle que torna  $\frac{\partial H}{\partial u(t)} = 0$ , será pelo menos local ótimo. Da Eq. (A.12), obtém-se que

$$\begin{aligned} \frac{\partial H}{\partial x(t)} &= Q(t)x(t) + A^T(t)p(t) & (A.58) \\ \frac{\partial^2 H}{\partial x^2(t)} &= Q(t) \\ \frac{\partial H}{\partial u(t)} &= R(t)u(t) + B^T(t)p(t) \\ \frac{\partial^2 H}{\partial u^2(t)} &= R(t) \\ \frac{\partial^2 H}{\partial x(t)\partial u(t)} &= \frac{\partial^2 H}{\partial u(t)\partial x(t)} = 0 \end{aligned}$$

Se substituirmos a Eq. (A.58) na Eq. (A.57), gera-se a matriz

$$\begin{bmatrix} Q(t) & 0 \\ 0 & R(t) \end{bmatrix} \quad (A.59)$$

Uma vez que a matriz  $R(t)$  é definida positiva,  $Q(t)$  também é definida positiva, então a matriz Eq. (A.59) é definida positiva. Se, entretanto,  $Q(t)$  é somente semidefinida positiva, então a matriz Eq. (A.59) é somente semidefinida positiva. Uma vez que as derivadas de ordem superior do *Hamiltoniano*  $H$  são nulas, a hipótese para este problema, que  $R(t)$  é uma matriz definida positiva é bastante forte para garantir que o controle  $u(t) = -R^{-1}(t)B^T(t)K(t)x(t)$  minimiza, pelo menos localmente, o custo funcional.

**Lema A.2.3.** *O controle  $u(t) = -R^{-1}(t)B^T(t)K(t)x(t)$  gera pelo menos um mínimo local para o custo funcional  $J_1$ .*

O próximo passo é provar que o controle extremo  $u(t) = -R^{-1}(t)B^T(t)K(t)x(t)$  é único. Esta demonstração depende do fato que a solução  $K(t)$  da equação de *Riccati*, sujeito a condição de contorno  $K(T) = K$ , é única.

**Lema A.2.4.** *Se um controle ótimo existe, então ele é único e é dado pela Eq. (A.54).*

**Prova** Nesta dedução mostraremos a unicidade. Sabe-se que o controle extremo  $u(t) = -R^{-1}(t)B^T(t)K(t)x(t)$  resultará pelo menos em um mínimo local

de  $J_1$ . Uma vez que  $K(t)$  é única, o controle ótimo (local) é uma função única do estado. Suponha que existam dois controles ótimos  $u_1(t)$  e  $u_2(t)$ , duas trajetórias ótimas  $x_1(t)$  e  $x_2(t)$  tais que  $x_1(t_0) = x_2(t_0)$ . Então  $x_1(t)$  e  $x_2(t)$  devem ser duas soluções distintas da equação diferencial

$$\dot{x}(t) = [A(t) - B(t)R^{-1}(t)B^T(t)K(t)]x(t), \quad x(t_0) = \xi \quad (\text{A.60})$$

originando no mesmo estado inicial  $x_1(t_0) = x_2(t_0)$ . Mas, desde que  $K(t)$  é uma solução única, a matriz  $A(t) - B(t)R^{-1}(t)B^T(t)K(t)$  é também única. Uma vez que a Eq. (A.60) é um sistema linear e homogêneo, sua solução iniciando de qualquer estado inicial é única; conseqüentemente,  $x_1(t) = x_2(t)$  e  $u_1(t) = u_2(t)$ ,  $\forall t \in [t_0, T]$ . Portanto o lema está demonstrado.

**Teorema A.2.1.** *Dado o sistema linear Eq. (A.52) e o custo funcional  $J_1$  de Eq. (A.53). Seja  $J_1^*$  o valor mínimo de  $J_1$ ; então  $J_1^*$  é dado por*

$$J_1^*[x(t), t] = \frac{1}{2} \langle x(t), K(t)x(t) \rangle \quad (\text{A.61})$$

sendo  $K(t) \in R^{n \times n}$  que é a solução da equação de *Riccati*, Eq. (A.43), satisfazendo a condição de contorno  $K(T) = F$ . Além disso, se o controle ótimo  $u(t) \neq 0$  para todos os estados, então a matriz  $K(t) > 0 \forall t, t_0 \leq t < T$ , e  $K(T) = F \geq 0$ . Uma vez que o custo mínimo  $J_1^* = \frac{1}{2} \langle x(t), K(t)x(t) \rangle$  é definido para todo  $x(t)$  e  $t$ , existe um controle ótimo e o custo mínimo é de fato  $J_1^*$ .

**Prova** Consulte (Athans e Falb, 1966).

Podemos realmente dizer que verificamos completamente a Lei de Controle. Mostramos que existe um ótimo e que o controle extremo único Eq. (A.54), corresponde ao mínimo do custo funcional  $J_1(u)$ . Portanto, o controle ótimo deve ser dado pela Eq. (A.54).

## FORMULAÇÃO CLÁSSICA DO *LQR*

---

Esta seção apresenta os fundamentos utilizados na formulação do problema *LQR* como um problema de otimização multiobjetivo para alocação de autoestruturas. A metodologia de controle clássica direcionada para o *LQR*, pode ser visualizada como uma aplicação particular de um problema de otimização restrito.

No cálculo existem métodos para determinar o valor mínimo de uma função em um espaço vetorial de dimensão finita. No controle ótimo, entretanto, as variáveis independentes são os vetores de estado e o vetor de entrada (controle), que são funções do tempo. Para encontrar a trajetória de controle que minimiza a medida de desempenho, exige-se obter um valor mínimo de um funcional, isto é, uma função sobre um espaço funcional. A otimização no espaço funcional é um problema de cálculo variacional.

A medida do desempenho na teoria de controle geralmente adota a forma da integral no tempo de uma medida de 'custo' instantâneo:

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt \quad (\text{B.1})$$

sendo  $x(t)$  o vetor de estado,  $u(t)$  o vetor de controle,  $t_0$  e  $t_f$  são o tempo inicial e final respectivamente. A função  $h$  é usada em problemas de horizonte finito para introduzir uma ponderação no estado final. A função  $g$  quantifica o custo instantâneo de um valor particular do sinal de controle e um valor particular do estado. Esta função assume várias formas em diferentes problemas; geralmente problemas envolvendo energia, deslocamento ou tempo.

Os problemas de controle ótimo inserem complexidades adicionais em problemas de otimização devido a relação entre o estado e o controle. Essas quantidades são relacionadas pelo sistema de equações dinâmicas:

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

o qual implica que os problemas são necessariamente de otimização restritos. A condição de restrição entre o estado e o controle é realizada na teoria de controle introduzindo o vetor de co-estado  $\lambda(t)$ . O vetor co-estado é variante no tempo e análogo aos multiplicadores de *Lagrange* do cálculo tradicional. Similar ao cálculo, eles têm como propósito transformar um problema de otimização restrito em problema de otimização irrestrito. Eles aparecem como multiplicadores de apoio indeterminados no custo funcional aumentado  $J_0$ :

$$J_0 = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) + \lambda(t)^T [f(x(t), u(t), t) - \dot{x}(t)] dt. \quad (\text{B.2})$$

Mostra-se que extrematizar o custo funcional aumentado com relação as variáveis independentes  $x$ ,  $\lambda$  e  $u$ , é equivalente ao cálculo do valor extremo do custo funcional original  $J$  com relação ao vetor  $u$ , (Kirk ,1970). A realização desta extremização é uma aplicação do Teorema Fundamental do Cálculo de Variações e gera as seguintes condições necessárias de otimalidade, que na realidade constituem o princípio mínimo de *Pontryagin*, (Kirk ,1970), (Athans e Falb ,1966):

$$\dot{x}^*(t) = f(x^*(t), u^*(t), t) \quad (\text{B.3})$$

$$\dot{p}(t) = -\left[\frac{\partial f}{\partial x}(x^*(t), u^*(t), t)\right]^T p(t) - \frac{\partial g}{\partial x}(x^*(t), u^*(t), t) \quad (\text{B.4})$$

e

$$0 = -\frac{\partial g}{\partial u}(x^*(t), u^*(t), t) + \left[\frac{\partial f}{\partial x}(x^*(t), u^*(t), t)\right]^T \dot{\lambda}(t) \quad (\text{B.5})$$

sendo que o asterisco refere-se aos valores ótimos. Uma equação adicional, conforme ((Kirk ,1970)) fornece valores de contorno para as equações diferenciais.

Em geral, a equação de valor de contorno fornece condições iniciais para a equação de estado (B.3) e condições finais para o co-estado de Eq.(B.4). Portanto, a resolução das equações de controle ótimo está associada a solução de um problema de valor de contorno em fronteira com dois pontos.

Esta formulação é caracterizada pela busca do vetor de controle  $u$  que minimiza um funcional  $J(\cdot)$ , denominado índice de desempenho quadrático cuja restrição é o sistema dinâmico representado por  $\dot{x}$ . As justificativas em focar o desenvolvimento da pesquisa voltada para implementações de sistemas de controle do tipo *LQR* podem ser constatadas pelas referências bibliográficas que estão alicerçadas em aplicações da metodologia *LQG/LQR*. No setor industrial, (Weerasooriya ,1995), utilidade pública-geração de energia (Lee ,1996), (Rab-Alibeik ,2003), atenuação de ruídos auditivos devido às perturbações externas dos automóveis e outros, (Garcia ,2004), controle de máquinas de corrente contínua, controle de processos, (Gasem ,1995) e pode-se destacar a aplicação de sistemas embarcados para implementação deste controlador, (Paschall ,1994).

Considere o problema de otimização restrito a seguir:

$$\begin{aligned} & \min_u J(x, Q, u, R) \\ & \text{sujeito a} \\ & \dot{x} = Ax + Bu. \end{aligned} \tag{B.6}$$

A estrutura de otimização (B.6), é denominada Problema de Controle Ótimo. No caso do Regulador Linear Quadrático, o índice de desempenho  $J$  é um mapeamento dos vetores de estado  $x$  e de entrada  $u$ , respectivamente de modo que este funcional é ponderado pelas matrizes  $Q$  e  $R$ .

A metodologia de projeto LQR tem como vantagem margens de estabilidade garantida, isto é margem de ganho infinita e de  $60^\circ$  de margem de fase, possuindo portanto, a qualidade de estabilidade robusta. Entretanto, é indicado como desvantagem da metodologia LQR, a necessidade de avaliar alguns dos estados para medição do sinal de controle de realimentação, de tal maneira que não exista ruído e nem perturbação no sistema, o que em geral não é possível, (M.Maciejowski ,1989). A metodologia de projeto de controle LQG/LTR tem superado esta deficiência na formulação do projeto LQR, em termos de implementações práticas,

(K. Zhou e Glover ,1996).

O problema do Regulador Linear Quadrático é apresentado sob o ponto de vista da natureza das matrizes de ponderação de estado  $Q$  e de controle  $R$ . O procedimento de solução da equação algébrica matricial de *Riccati EAR* ser estável, é devido as matrizes de ponderação e outros termos obedecerem rigorosamente as restrições que garantem a existência de uma solução ótima.

O problema *LQR* é formulado por meio de um índice de desempenho quadrático e tem como restrição a equação de estado linear invariante no tempo dada por,

$$J(t_o) = \frac{1}{2}x^T(T)S(T)x(T) + \frac{1}{2} \int_{t_o}^T [x^T Q(t)x + u^T R(t)u]dt \quad (\text{B.7})$$

s.a

$$\dot{x} = Ax + Bu \quad (\text{B.8})$$

sendo  $x \in \mathbb{R}^n$  o vetor de estado e  $u \in \mathbb{R}^m$  o vetor de controle, o par  $(A, B)$  é controlável,  $S(T) \geq 0, Q(T) \geq 0$  e  $R > 0$  todas são simétricas. As matrizes de ponderação  $Q \in \mathbb{R}^{n \times n}$  e  $R \in \mathbb{R}^{m \times m}$  são positiva semidefinida e positiva definida, respectivamente.  $[t_o, T]$  é o intervalo de tempo de interesse do problema.

## B.1 A Solução Variacional do *LQR*

Para resolver o Problema de Otimização Restrito dado por (B.6) , deve-se inicialmente convertê-lo em um Problema de Otimização Irrestrito e utilizar os métodos de Cálculo Variacional, (Athans e Falb ,1966). A variável  $t$  que indica o tempo, será suprimida para simplificação do entendimento das equações. Inicialmente, a estrutura de otimização é usada para construir a Equação Energia *Hamiltoniano*, que é fornecido pela seguinte expressão:

$$H(x, u, \lambda) = \frac{1}{2}(x^T Qx + u^T Ru) + \lambda^T (Ax + Bu). \quad (\text{B.9})$$

Aplicando as condições de otimalidade de primeira ordem na Eq. ( B.9), (Bryson e Jr ,1975), obtém-se os vetores de estado e co-estado na forma matricial:

$$\dot{x} = \frac{\partial H(x, u, \lambda)}{\partial \lambda} = Ax + Bu, \quad (\text{B.10})$$

$$-\dot{\lambda} = \frac{\partial H(x, u, \lambda)}{\partial x} = Qx + A^T \lambda, \quad (\text{B.11})$$

$$0 = \frac{\partial H(x, u, \lambda)}{\partial u} = Ru + B^T \lambda. \quad (\text{B.12})$$

Da Eq. (B.12) pode-se obter

$$u(t) = -R^{-1}B^T \lambda. \quad (\text{B.13})$$

A hipótese que a matriz  $R$  seja definida positiva, isto é,  $R > 0 \forall t \in [t_o, T]$ , garante a existência da matriz inversa  $R^{-1} \forall t$ , ou seja, não singular  $\in [t_o, T]$ . Sabe-se que o controle ótimo minimiza o *Hamiltoniano*.

A condição necessária  $\frac{\partial H}{\partial u}$ , produz apenas um valor extremo de  $H$  com relação ao vetor de controle  $u(t)$ . Afim de que o valor extremo de  $H$  seja um valor mínimo com relação ao vetor  $u(t)$ , a matriz  $\frac{\partial^2 H}{\partial u^2(t)} \in \mathbb{R}^{n \times n}$  deve ser definida positiva. Da Eq.(B.12), obtém-se que

$$\frac{\partial^2 H}{\partial u^2(t)} = R(t) > 0, \quad (\text{B.14})$$

(Condição Suficiente), e conseqüentemente segue que a lei de controle  $u(t)$  dada pela Eq. (B.13) de fato minimiza o *Hamiltoniano*  $H$ .

A seguir, o próximo passo é obter as Equações Canônicas reduzidas. Substituindo-se a Eq. (B.13) na Eq. (B.10) obtém-se a relação

$$\dot{x} = Ax + B(-R^{-1}B^T \lambda) = Ax - BR^{-1}B^T \lambda. \quad (\text{B.15})$$

As Eqs. (B.15) e (B.11) são as Equações Canônicas Reduzidas. Combinando essas equações na forma matricial, obtém-se um Sistema  $2n$  Equações Diferenciais Lineares Homogêneo variante com o tempo, que na realidade é um Problema de valor de Contorno Linear em Fronteira com Dois-Pontos



$$\begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} = \begin{bmatrix} A(t) & -S(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix}, \quad (\text{B.16})$$

sendo  $S(t) \triangleq B(t)R^{-1}(t)B^T(t)$  e

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}, \quad (\text{B.17})$$

a matriz *Hamiltoniana*,  $x(t_0)$  é dado e  $\lambda(T) = S(T)x(T)$ . Desde que  $\text{rank}(H) = n$ , então  $H$  possui  $n$  vetores colunas e linearmente independentes de modo que um conjunto  $S$  formado por esses vetores-colunas forma uma base do  $\mathbb{R}^n$ . Diz-se então que a matriz  $H$  é diagonalizável pois existe uma matriz inversível  $S$  tal que  $S^{-1}HS = \Lambda$  é uma Matriz Diagonal. As entradas diagonais desta matriz são seus autovalores. Então os vetores colunas da matriz  $S$  são os autovetores  $v_i$  à direita que estão associados aos autovalores  $\lambda_i$  que satisfazem a relação  $(\lambda I - H)v_i = 0$ .

Já os autovetores recíprocos ou à esquerda  $w_i$ , são obtidos das colunas da matriz  $S^{-1}$  e satisfazem a equação  $w_i^T(\lambda I - H) = 0$ .

Uma vez que o problema é linear, as equações diferenciais e condições de contorno no instante final  $T$  são homogêneas, então  $x(t)$  e  $\lambda(t)$  são proporcionais a  $x(t_0)$ . Resolve-se o problema de valor de contorno pela Superposição Linear (Matriz de Transição), Método da varredura (*Sweep Method*) e ainda pelo Método das Equações Diferenciais Parciais de *Hamilton-Jacobi-Bellman*, (Bryson e Jr ,1975).

Pode-se obter uma única solução do sistema Eq. (B.16) estabelecendo que conhece-se um total de  $2n$  condições de contorno, (Athans e Falb ,1966). Um total de  $n$  condições de contorno é fornecida pelo estado inicial  $x(t_0)$  em  $t_0$ . As  $n$  condições de contorno restantes são fornecidas pela condição de transversalidade, que exige no tempo final  $T$ , que o vetor de co-estado  $\lambda(T)$  satisfaça a relação  $\lambda(T) = P(T)x(T)$ , sendo  $P \in \mathbb{R}^{n \times n}$  uma matriz semidefinida positiva.

Suponha-se que os vetores  $x(t)$  e  $\lambda(t)$ , as soluções do sistema de Eqs. (B.16), estejam realmente relacionados pela equação

$$\lambda(t) = K(t)x(t) \quad t \in [t_0, T] \quad (\text{B.18})$$

Diferenciando a Eq.(B.18) com relação ao tempo, obtém-se

$$\dot{\lambda}(t) = \dot{K}(t)x(t) + K(t)\dot{x}(t) \quad (\text{B.19})$$

Da Eq. (B.16), sabe-se que

$$\dot{x}(t) = A(t)x(t) - S(t)\lambda(t) \quad (\text{B.20})$$

$$\dot{\lambda}(t) = -Q(t)x(t) - A^T\lambda(t). \quad (\text{B.21})$$

Substituindo-se a Eq.(B.18) em  $\dot{x}(t) = A(t)x(t) - S(t)\lambda(t)$ , obtém-se

$$\dot{x}(t) = [A(t) - S(t)K(t)]x(t) \quad (\text{B.22})$$

e novamente substituindo-se a Eq.(B.22) na Eq.(B.19), encontra-se que

$$\dot{\lambda}(t) = [\dot{K}(t) + K(t)A(t) - K(t)S(t)K(t)]x(t). \quad (\text{B.23})$$

Levando a Eq.(B.18) na Eq.(B.21), tem-se:

$$\dot{\lambda}(t) = [-Q(t) - A^T(t)K(t)]x(t). \quad (\text{B.24})$$

Das Eqs.(B.23-B.24), conclui-se que

$$[\dot{K}(t) + K(t)A(t) - K(t)S(t)K(t) + A^TK(t) + Q(t)]x(t) = 0, \quad (\text{B.25})$$

$\forall t \in [t_0, T]$ . A Eq. (B.25) mantém-se para qualquer escolha do estado inicial, desde que a matriz  $K(t)$  não dependa do estado inicial, e uma vez que  $x(t)$  seja uma solução da Eq. (B.22) homogênea, segue que a Eq. (B.25) deva se manter não importando o valor de  $x(t)$ . Portanto, a matriz  $K(t)$  satisfaz a equação diferencial matricial

$$\dot{K}(t) = -K(t)A(t) - A^TK(t) + K(t)B(t)R^{-1}(t)B^T(t)K(t) - Q(t) \quad (\text{B.26})$$

**Lema B.1.1.** *Se  $x(t)$  e  $\lambda(t)$  são as soluções das Eqs. (B.16) Canônicas, e se  $\lambda(t) = K(t)x(t)$ ,  $\forall t \in [t_0, T]$  e  $\forall x(t)$ , então  $K(t)$  satisfaz a Eq. (B.26)*

**Prova** Maiores detalhes, consulte (Athans e Falb ,1966) Capítulo 9, página 761.

**Lema B.1.2.** *Se  $K(t)$  é a solução da equação de Riccati (B.26) e se  $K(t) = F$ , então  $K(t)$  é simétrica  $\forall t \in [t_0, T]$ , isto é  $K(t) = K^T(t)$*

**Prova** Consulte (Athans e Falb ,1966) página 762.

Dado o sistema linear  $\dot{x} = A(t)x(t) + B(t)u(t)$  e o índice de desempenho quadrático  $J_1$ , sendo o vetor de entrada  $u(t)$  irrestrito,  $T$  é especificado,  $F$  e  $Q(t)$  são matrizes semidefinidas positivas e  $R(t)$  definida positiva, respectivamente. Então o controle ótimo existe, é único e dado por  $u(t) = -R^{-1}(t)B^T(t)K(t)x(t)$  de modo que a matriz  $K(t) \in \mathbb{R}^{n \times n}$  é a solução da equação de *Riccati* satisfazendo a condição de contorno  $K(T) = F$ , (Athans e Falb ,1966), (Bryson e Jr ,1975). O estado do sistema ótimo é a solução da equação diferencial linear

$$\dot{x}(t) = [A(t) - B(t)R^{-1}B^T(t)K(t)]x(t), \quad x(t_0) = \xi \quad (\text{B.27})$$

**Lema B.1.3.** *O controle  $u(t) = -R^{-1}(t)B^T K(t)x(t)$  gera pelo menos um mínimo local para  $J_1$ .*

**Prova** Consulte (Athans e Falb ,1966), Capítulo 9 página 764.

**Lema B.1.4.** *Se o controle ótimo existe, então ele é único e dado por  $u(t) = -R^{-1}(t)B^T(t)K(t)x(t)$ .*

**Prova** Veja (Athans e Falb ,1966), página 764.

**Teorema B.1.1.** *Dado o sistema linear  $\dot{x} = A(t)x(t) + B(t)u(t)$  e o custo funcional  $J_1$ . Seja  $J_1^*$  o valor mínimo de  $J_1$ ; então  $J_1^*$  é dado por*

$$J_1^*[x(t), t] = \frac{1}{2} \langle x(t), K(t)x(t) \rangle . \quad (\text{B.28})$$

**Prova** Veja (Athans e Falb ,1966), página 765.

A realimentação variante no tempo, nem sempre permite uma implementação conveniente na prática, devido à variação do ganho do controlador com o tempo. Entretanto, podemos utilizar ao invés disso, um ganho de realimentação subótimo

(ganho constante) que minimize o índice de desempenho, Eq. (B.7). Considerando o desenvolvimento da solução apresentada em (Athans e Falb ,1966), a lei de controle de realimentação de estado linear que minimiza o índice de desempenho  $J$  e satisfaz as restrições  $\dot{x}$ , na estrutura (B.6) é dada por:

$$u(t) = -Kx(t), \quad (\text{B.29})$$

desde que todas as componentes do vetor de entrada sejam medidas, sendo  $K \in \mathbb{R}^{m \times n}$  a matriz de ganho de realimentação no sistema de controle dada por

$$K(t) = -R^{-1}B^T P(t). \quad (\text{B.30})$$

Note que o ganho é função da solução da equação algébrica de Riccati  $P \in \mathbb{R}^{n \times n}$  simétrica e definida positiva,

$$A^T P + PA - PBR^{-1}B^T P + Q = 0, \quad t \leq T. \quad (\text{B.31})$$

Uma desvantagem da abordagem de *Auto-Sistema*, é que este método não explora totalmente a estrutura da matriz *Hamiltoniana*. Neste método, aplica-se o algoritmo *QR* à matriz *Hamiltoniana*  $H \in \mathbb{R}^{2n \times 2n}$  como se ela fosse uma matriz geral, isto é, sem explorar a estrutura da matriz *Hamiltoniana*  $H$ , que é ineficiente para o trabalho de armazenamento computacional. Os erros de arredondamento surgem no curso da aplicação do algoritmo *QR*, de tal forma que, os autovalores não aparecem exatamente mais em pares.

## B.2 Abordagem Clássica para EAR

Esta Seção introduz de forma sucinta uma pesquisa nos métodos numéricos usados na solução da Equação Matricial Algébrica de *Riccati*. A descrição desses métodos são baseados em, (Petkov e Konstantinov ,1991), (Gardiner e Laub ,1991b) e (Kleinman ,1968). Os métodos numéricos tomados para solução da *EAR*, são:

- Método de *Newton*.
- Método da Função Sinal.

- Abordagem de Auto-Sistemas que é formada por dois métodos para resolução da *EAR*: Métodos do Autovetor e *Schur* (baseados na matriz *Hamiltoniana*).
- Método Simplético

A *EAR* é analisada em termo da seleção das melhores matrizes  $Q$  e  $R$ . Esta análise é tratada mais adiante por métodos que permitem reduzir as operações para essas matrizes.

A equação algébrica de Riccati, *EAR*, é amplamente utilizada na síntese de sistemas de controle, enquanto que as equações de *Lyapunov*, são usadas na análise de estabilidade e existência de soluções de controle para esses sistemas, (Doyle e Stein ,1979). Apresenta-se a Equação Matricial Algébrica de *Riccati*, Eq. (B.32)

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (\text{B.32})$$

cuja solução está intrinsicamente relacionada com a matriz *Hamiltoniana*  $H \in \mathbb{R}^{2n \times 2n}$ ,

$$H := \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} > 0 \quad (\text{B.33})$$

Maiores detalhes sobre a obtenção das propriedades desta matriz podem ser obtidos em (Yu 1983), (Lancaster e Rodman ,1995), (Athans e Falb ,1966) e (Petkov e Konstantinov ,1991). Em (Ferreira ,2003) a solução desta equação em regime permanente é analisada em termo da seleção das matrizes  $Q$  e  $R$  ótimas. Esta análise é direcionada para a escolha do método, que permite reduzir a quantidade de operações de valores dessas matrizes.

Os métodos de *Schur* e *Autovetor*, são classificados como abordagem de Auto-sistemas para a solução da equação de *Riccati*. Estes métodos são baseados na estrutura da matriz *Hamiltoniana*.

# COMPUTAÇÃO EVOLUTIVA

---

O termo computação evolutiva, criado em 1991, representa o esforço para unir diferentes abordagens desenvolvidas para simular vários aspectos do processo de seleção natural, reprodução, variável aleatória, competição e seleção dos indivíduos em uma população que constituem a essência da evolução e são os pontos comuns das técnicas desenvolvidas tais como: algoritmos genéticos, programação evolutiva e estratégias evolutivas. A criação destas técnicas foi estimulada pela necessidade de solucionar problemas com relação à otimização, inteligência de máquina, adaptação robusta, (Bäck e Michalewicz ,1997).

Os autores (Chipperfield e Peter ,1996) optam pelo termo algoritmos evolucionistas para referenciar as abordagens que simulam o processo de evolução natural e afirmam que esses métodos têm por base modelos computacionais do processo fundamental de evolução, de modo que classificam os algoritmos evolucionistas como métodos de busca estocástica e otimização.

A metodologia canônica de computação evolutiva é formada de duas etapas:

- A primeira etapa consiste em gerar uma população inicial (solução) e realizar sua avaliação.
- A segunda etapa consiste em evoluir a população gerada na primeira fase por meio da seleção de indivíduos, criação de novos indivíduos e da avaliação desses novos indivíduos. O processo de evolução continua até certos indivíduos atingirem um nível de adequabilidade aceitável ou a população mostrar-se inapta em adaptar-se ao ambiente proposto.

## C.1 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).

São insensíveis à forma da função objetivo, sendo igualmente eficientes em funções descontínuas, ruidosas, multidimensionais e multimodais, e são capazes de explorar várias soluções ao mesmo tempo (paralelismo).

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.

## C.2 Conceitos Básicos

Os Algoritmos Genéticos (AGs) constituem uma técnica de busca e otimização, altamente paralela, inspirada no princípio *Darwiniano* de seleção natural e reprodução genética, (Goldberg, 1985).

Os princípios da natureza nos quais os AGs se inspiram são relativamente simples. De acordo com a teoria de *Charles Darwin*, o princípio de seleção privilegia os indivíduos mais aptos com maior longevidade e, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações. Tais códigos genéticos constituem a identidade de cada indivíduo e estão representados nos cromossomos.

Estes princípios são imitados na construção de algoritmos computacionais que buscam uma melhor solução para um determinado problema, por meio da evolução de populações de soluções codificadas através de cromossomos artificiais. Nos AGs, um cromossomo é uma estrutura de dados que representa uma das possíveis soluções do espaço de busca do problema. Portanto, os cromossomos são então submetidos a um processo evolucionário que envolve: avaliação (função de *fitness*), seleção (regra da roleta), e os operadores genéticos *crossover* e mutação. Após vários ciclos de evolução a população deverá conter indivíduos mais aptos.

Os AGs são particularmente aplicados em problemas complexos de otimização: problemas com diversos parâmetros ou características que precisam ser combinadas em busca da melhor solução, problemas com diversas restrições que não podem ser representadas matematicamente e problemas com espaços de busca grandes.

Os AGs ocupam lugar de destaque entre os paradigmas de Computação Evolutiva, devido a uma série de razões dentre as quais:

1. Apresentam-se como o paradigma mais completo da Computação Evolutiva visto que englobam de forma simples e natural todos os conceitos nela contidos;
2. Apresentam resultados bastante aceitáveis, com relação a precisão e recursos empregados (fáceis de implantar em computadores de porte médio), para uma ampla gama de problemas de difícil resolução por outros métodos;
3. São muito flexíveis, aceitam sem grandes dificuldades uma infinidade de alterações na sua implementação e permitem fácil hibridização (vantagem



importante no caso de aprendizagem) inclusive com técnicas não relacionadas à Computação Evolutiva;

4. Em relação aos outros paradigmas da Computação Evolutiva, são os que exigem menor conhecimento específico do problema em questão para o seu funcionamento, o que os torna altamente versáteis e, além disso, agregam conhecimento específico com pouco esforço;
5. São o paradigma mais usado dentro da Computação Evolutiva e, junto com as Redes Neurais, os mais usados de toda a Computação Natural.

As principais vantagens dos AGs podem ser resumidas no que segue:

1. São procedimentos de busca paralela que podem ser implementados em máquinas de processamento paralelo, acelerando em muito o processo;
2. São aplicáveis a problemas de otimização contínuos e discretos;
3. São estocásticos e, portanto, menos sujeitos a ficarem presos em mínimos locais, freqüentes em diversos problemas práticos de otimização;
4. Sua flexibilidade facilita a estruturação e identificação de parâmetros em modelos complexos tais como redes neurais e sistemas de inferência nebulosos.

## APÊNDICE D

# ALGORITMOS AG-LQR<sub>Q,R</sub>

---

Apresentam-se os algoritmos relativos aos modelos do AG que são apresentados na Seção 4.3.

## D.1 Programa Principal do AG para o LQR

### D.1.1 Inicializações

#### Sistema Dinâmico

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

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

- $A$  ← matriz de estado
- $B$  ← matriz de controle
- $C$  ← matriz de saída

#### Especificações de Projeto

- Sens ← Parâmetros de sensibilidade
- Ydir ← Limite direito dos autovalores em malha fechada
- Yesq ← Limite esquerdo dos autovalores em malha fechada

## Parâmetros

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

## Indicação do Número de Indivíduos Classificados

cont	←	0
contger	←	0

### D.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

### D.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

```

## D.2 Algoritmo da População Inicial

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

### D.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)
```

### D.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

```

### D.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

```

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

```

## D.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 espe-

cificações 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 D.3.1 e D.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 D.3.3.

### D.3.1 Recuperação da Matriz Simétrica $Q$

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

```

For z ← 1 : nindiv
  If w < nQ + 1
    For i ← n
      c ← i
      For j ← c : n
        Q(i, j) ← QR(z, w)
        w ← w + 1
      End
    End
  End
  For i ← 1 : n
    For j ← 1 : n
      If i < j
        Q(j, i) ← Q(i, j)
      End
    End
  End
End

```

### D.3.2 Recuperação da Matriz Simétrica $R$

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

```

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

```

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

```

## D.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 D.4.1, e



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

### D.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 - Fitp
    End
  End
  For i ← 1 : nQ + nR
    QR1(z, i) ← QR(elem, i)
  End
End

```

### D.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

```

### D.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

```

## D.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 : nidiv
  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_chav ← 2
    Else
      si_xover_B ← [ si_xover_B; z si_xover(z,1)
      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  $\text{imag}(Y_{esq}(k)) \sim 0$ 
     $lab\_si\_I_{esq}(k) \leftarrow \text{abs}(\text{imag}(Y_{i\_xover}(z,k))/\text{imag}(Y_{esq}(k)))$ 
    If  $lab\_si\_I_{esk}(k) < 1$ 
       $lab\_si\_ie(k) \leftarrow 1$ 
    Else
       $lab\_si\_ie(k) \leftarrow 0$ 
    End
  Else
    If  $\text{imag}(Y_{i\_xover}(z,k)) \sim 0$ 
       $lab\_si\_I_{esk}(k) \leftarrow 1.5$ 
       $lab\_si\_ie(k) \leftarrow 0$ 
    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  $si\_min\_melhor \sim 20000$ 
   $melhor\_indiv\_si\_A1 \leftarrow [si\_xover\_B(zz, 1) \quad si\_xover\_B(zz, 2) \quad si\_xover\_B(zz, 3)$ 
     $si\_xover\_B(zz, 5) \quad si\_xover\_B(zz, 6) \quad si\_xover\_B(zz, 7) \quad \text{sum}(si\_xover\_B(zz, 2$ 
  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
```

# PROJETO LQR NEURO-GENÉTICO: *DFIG*

---

As atividades de pesquisa e desenvolvimento voltadas para geração de energia tem importância significativa para minimização dos danos ambientais e promoção do bem-estar social. As fontes convencionais de geração de energia elétrica são as usinas hidrelétricas, termoeletricas e nucleares. Outras fontes de energia, como a energia solar, eólica, biomassa e energia das marés, são classificadas como fontes alternativas de energia. As fontes alternativas de energia mostram seu potencial de aplicação em face de duas grandes questões: a questão da eletrificação de áreas isoladas e a questão da esgotabilidade das fontes convencionais, (Abreu ,2007) e (Pinto Março, 2007).

Iremos aplicar a metodologia do projeto *LQR* Inteligente a um gerador de indução duplamente alimentado de uma planta eólica. Este projeto consiste em mostrar a viabilidade de integrar as abordagens de inteligência computacional, tais como: algoritmos genéticos e redes neurais, para síntese de sistemas de controle ótimo do tipo *LQR*. A seleção das matrizes de ponderação é realizada por algoritmo genético e solução da EAR é dada por uma rede neuronal recorrente. Desta forma, o AG contribui para impor uma lei de controle que aloca a autoestrutura e rede neuronal que contribui para a computação dos ganhos, evitando problemas com inversão de matrizes.

A importância do desenvolvimento de pesquisa voltada para síntese de sistemas de controle que estão fundamentadas no projeto *LQR*, envolve os mais variados

segmentos das atividades tais como setor industrial, utilidade pública, geração de energia, (Lee ,1996), atenuação de ruídos auditivos, devido a perturbações externas dos automóveis e outros (Garcia ,2004), controle de máquinas de corrente contínua e controle de processos, (Gasem ,1995). A utilização do projeto *LQR* tem-se mostrado uma alternativa em aplicações a sistemas eólicos, enfatizando as tecnologias de geração eólica que adotam os geradores de indução duplamente alimentados (*DFIG*), (Pinto Março, 2007).

## E.1 Geradores Eólicos *DFIG*

A energia eólica pode ser convertida em energia elétrica através de aerogeradores, que são sistemas que captam a energia do vento através das hélices, transformando-a em energia mecânica que então é convertida em energia elétrica através de um gerador.

Os geradores de indução com rotor bobinado para aplicações em sistemas eólicos podem ser encontrados em dois tipos básicos de configuração. Uma em que o escorregamento ou torque é controlado através da eletrônica de potência no circuito do rotor, caracterizado pela variação da resistência rotórica. A outra configuração consiste na conexão de um circuito de extração de potência pelo rotor, conhecido como *Double Fed Induction Generator (DFIG)*. Para maiores detalhes, veja as referências, (Nunes ,2003), (Marques ,2003) e (Hansen ,2001).

A Fig. E.1 ilustra um diagrama geral das conexões da turbina com gerador que adota o gerador com rotor bobinado, sendo que um dos conversores é conectado à rede e o outro é conectado aos enrolamentos do rotor, de modo que os dois conversores são interligados através de um circuito capacitivo e com controle de modulação por largura de pulso *PWM*. Para o conjunto gerador e conversores de potência existem várias concepções para interligação dos geradores de indução duplamente alimentados às turbinas eólicas.

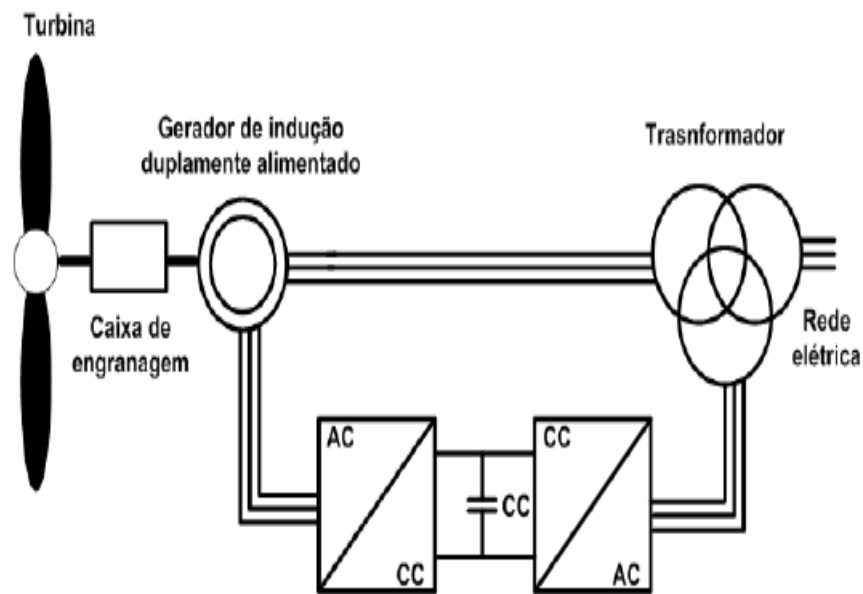


Figura E.1: Gerador de Indução duplamente alimentado.

O conversor interligado ao gerador controla a velocidade do rotor e a potência reativa injetada ou consumida pelo gerador através do estator. Já o conversor interligado à rede controla a tensão no barramento CC e a potência ativa que o rotor troca com a rede. A configuração que utiliza o *DFIG* possui como uma das principais vantagens sobre as demais tecnologias, a utilização de conversores de frequência da ordem de trinta por cento (30%) da potência nominal da turbina, reduzindo substancialmente o seu custo de instalação (Pinheiro ,2004).

Segundo (Marques ,2003), as topologias mais utilizadas para geração eólica com velocidade variável são as que utilizam o *DFIG* e gerador síncrono, mas dentre os fatores mais importantes a serem considerados são o custo e as propriedades dinâmicas de cada esquema (Nunes ,2003).

O princípio de funcionamento do *DFIG* é ser dado em maiores detalhes em (Hansen ,2003). Pode ser explicado por meio do escorregamento  $s = \frac{\omega_s - \omega_m}{\omega_s}$  e do fluxo de potência. Quando o *DFIG* encontra-se na velocidade super-síncrona ( $s < 0$ ) a potência será fornecida à rede simultaneamente pelo rotor por meio



dos conversores e pelo estator. Deste modo ( $P_{rotor} > 0$ ) indica o fornecimento de potência. Já quando o *DFIG* encontra-se na velocidade sub-síncrona ( $s > 0$ ) o fluxo da potência será da rede para o rotor. Desta forma, ( $P_{rotor} < 0$ ) indica que o rotor está absorvendo potência da rede. Em ambos os casos (velocidade super-síncrona e sub-síncronas) o estator fornece potência à rede.

## E.2 Modelo Gerador-Turbina

Após a apresentação do motor na Subseção E.1, mostra-se aqui o modelo gerador-turbina, mais precisamente o modelo linearizado, visto que, deste modo podemos utilizar a teoria de sistemas lineares para realizar a análise da dinâmica, estabilidade, controlabilidade e observabilidade, resposta em frequência e robustez do sistema não linear. O modelo linearizado assume a seguinte forma, (Pinto Março, 2007),

$$\Delta \dot{x} = A\Delta x + B\Delta u \quad (\text{E.1})$$

$$\Delta y = C\Delta x \quad (\text{E.2})$$

sendo  $\Delta x$  e  $\Delta u$  vetores de estado e entrada, respectivamente, que são  $\Delta x = [\Delta i_{dr} \Delta i_{qr} \Delta \omega_r \Delta i_d \Delta i_q \Delta V_{dc}]^T$  e  $\Delta u = [\Delta v_{dr} \Delta v_{qr} \Delta T_e \Delta v_d \Delta v_q]^T$ . A matriz  $A$  de estados é constituída por matrizes de bloco  $A_r^{3 \times 3}$  e  $A_s^{3 \times 3}$  que representam o lado do rotor e o lado do estator, respectivamente. A matriz de controle  $B$  é constituída pelas matrizes de bloco  $B_r^{3 \times 2}$  e  $B_s^{3 \times 2}$ . As matrizes da Eq. (E.1) de estado,

$$A_r = \begin{bmatrix} -\frac{R_r}{L_{rr}\sigma} & s_0\omega_0 & 0 \\ -s_0\omega_0 & -\frac{R_r}{L_{rr}\sigma} & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

$$B_r = \begin{bmatrix} -\frac{1}{L_{rr}\sigma} & 0 & 0 \\ 0 & -\frac{1}{L_{rr}\sigma} & 0 \\ 0 & 0 & -\frac{1}{2H} \end{bmatrix}$$

$$A_s = \begin{bmatrix} -\frac{R}{L} & \omega_0 & 0 \\ -\omega_0 & -\frac{R}{L} & 0 \\ -\frac{\nu_{d0}}{CV_{cc0}} & -\frac{\nu_{q0}}{CV_{cc0}} & \frac{\nu_{d0}i_{d0} + \nu_{q0}i_{q0}}{CV_{cc0}^2} \end{bmatrix}$$

e

$$B_s = \begin{bmatrix} \frac{1}{L} & \omega_0 \\ 0 & \frac{1}{L} \\ -\frac{i_{d0}}{CV_{cc0}} & -\frac{i_{q0}}{CV_{cc0}} \end{bmatrix},$$

ou na forma compactada, têm-se as matrizes

$$A = \begin{bmatrix} -\frac{R_r}{L_{rr}\sigma} & s_0\omega_0 & 0 & 0 & 0 & 0 \\ -s_0\omega_0 & -\frac{R_r}{L_{rr}\sigma} & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{R}{L} & \omega_0 & 0 \\ 0 & 0 & 0 & -\omega_0 & -\frac{R}{L} & 0 \\ 0 & 0 & 0 & -\frac{\nu_{d0}}{CV_{cc0}} & -\frac{\nu_{q0}}{CV_{cc0}} & \frac{\nu_{d0}i_{d0} + \nu_{q0}i_{q0}}{CV_{cc0}^2} \end{bmatrix},$$

e

$$B = \begin{bmatrix} -\frac{1}{L_{rr}\sigma} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{L_{rr}\sigma} & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2H} & 0 & 0 \\ 0 & 0 & 0 & c1L & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{L} \\ 0 & 0 & 0 & -\frac{i_{d0}}{CV_{cc0}} & -\frac{i_{q0}}{CV_{cc0}} \end{bmatrix}$$

sendo o coeficiente  $\mu$  definido abaixo por

$$\mu = -\frac{\frac{1}{2}\rho A\nu^3 C_p(\lambda, \beta)}{2H\omega_r^2} + \frac{\frac{1}{2}\rho A\nu^3}{2H\omega_r} \cdot \frac{dC_p(\lambda, \beta)}{d\lambda} \cdot \frac{R_p}{\nu} \quad (\text{E.3})$$

Aplica-se o método *LQR* ao modelo de um gerador *DFIG*, Fig. E.2. Este método é constituído pelo estágio de busca das matrizes de ponderação via AG e pela determinação do ganho ótimo do controlador utilizando uma RNA Recorrente.

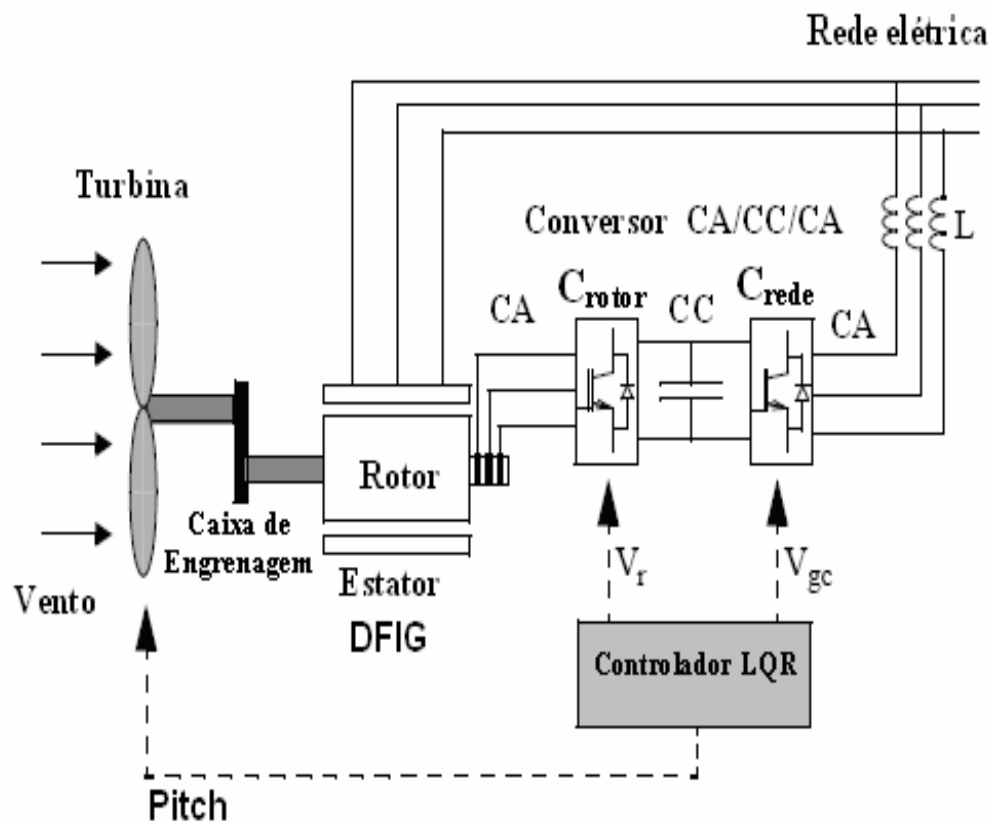


Figura E.2: Esquema do gerador de indução com o controlador  $LQR$ .

A metodologia está organizada em uma seqüência de métodos para a realização do projeto  $LQR$  inteligente. O levantamento do modelo linearizado gerador-turbina é o primeiro passo para o projeto. O segundo passo é seleção das matrizes de ponderação por um algoritmo genético que garante a alocação da autoestrutura. O terceiro passo é solução da equação de *Riccati* que garante a robustez do controlador.

### E.3 Modelo Completo

O modelo completo e os parâmetros da turbina eólica do *DFIG* e do inversor que são substituídos nas matrizes, Eqs. (E.1) e (E.2), encontram-se em (Pinto Março, 2007). Desta forma, as matrizes da equação de estado para o sistema linearizado em um ponto de operação são:

$$A = \begin{bmatrix} -39.4228 & 0.3905 & 0 & 0 & 0 & 0 \\ -0.2761 & -55.7513 & 0 & 0 & 0 & 0 \\ 0 & 0 & -21.9508 & 0 & 0 & 0 \\ 0 & 0 & 0 & -24.5578 & 0.6380 & 19.8695 \\ 0 & 0 & 0 & -1.2701 & -18.0866 & 0.7299 \\ 0 & 0 & 0 & -6.9175 & -0.0803 & -1.7817 \end{bmatrix}$$

$$B = \begin{bmatrix} -12.46 & 0 & 0 & 0 & 0 \\ 0 & -12.46 & 0 & 0 & 0 \\ 0 & 0 & -2.19 & 0 & 0 \\ 0 & 0 & 0 & 5.71 & 0 \\ 0 & 0 & 0 & 0 & 5.71 \\ 0 & 0 & 0 & -0.71 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}.$$

Para o caso contínuo, os autovalores da planta modelo *DFIG* são apresentados na Tabela E.1. Nota-se que o segundo autovalor apresenta uma dinâmica mais rápida em relação aos demais.

Tabela E.1: Autovalores do sistema dinâmico sem controlador *DFIG*

Nº	Autovalores
1	-39.4294
2	-55.7447
3	$-13.3703 + j2.6033$
4	$-13.3703 - j2.6033$
5	-17.6856
6	-21.9508

Na Tabela E.2 observa-se os autovalores do sistema de malha fechada. Nela, os autovalores apresentam uma dinâmica mais rápida em relação ao sistema não compensado.

Tabela E.2: Autovalores do Sistema Dinâmico Compensado *DFIG*

Nº	Autovalores
1	-56.6226
2	-40.2041
3	-22.0073
4	-18.6784
5	$-13.9865 + j1.8650$
6	$-13.9865 + j1.8650$

Utilizou-se o sistema dinâmico *DFIG* para obtenção das matrizes de ponderação. Para o algoritmo genético, iremos usar as restrições impostas pelo projetista que são as faixas de autovalores, suas respectivas sensibilidades associadas e a matriz de ganho do controlador *LQR*, segundo as Tabelas E.3 e E.4 com o intuito de verificar a abordagem.

Tabela E.3: Restrições do Projeto *LQR* para *DFIG*

Nº.	Autovalor	Sensibilidade
1	$-60.00 \leq Re \leq -25.00$	3.0
2	$-45.00 \leq Re \leq -25.00$	2.5
3	$-15.00 \leq Re \leq -10.00$	2.0
4	$-15.00 \leq Re \leq -10.00$ $-2.0 \leq Imag \leq 2.00$	1.5
5	$-25.00 \leq Re \leq -10.00$ $-2.6033 \leq Imag \leq 2.6033$	3.0
6	$-25.00 \leq Re \leq -10.00$ $-2.6033 \leq Imag \leq 2.6033$	2.0

Tabela E.4: Ganhos do Controlador *LQR*.

<i>K</i>	Ganho do Controlador					
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$K_1$	0.000	0.000	-0.117	-3.299	-0.458	2.827
$K_2$	0.000	0.000	-0.294	-0.964	-0.682	3.899
$K_3$	0.1300	1.1100	-0.235	-0.793	-0.328	1.435
$K_4$	0.5950	0.2734	-0.294	-0.9103	-0.6791	4.7281
$K_5$	0.8110	0.1620	-0.6297	-0.652	-0.3722	4.1186

As matrizes de ponderação  $Q$  do estado e  $R$  do controle foram aqui geradas por um algoritmo genético, para vinte (20) gerações, baseada na estrutura formulada na Seção 4.3. São elas:

$$Q_{LQR} = \begin{bmatrix} 12.9929 & 1.7324 & 5.6853 & 2.7946 & 1.7676 & 9.1792 \\ 1.7324 & 21.9909 & 7.1443 & 1.4841 & 7.9915 & 22.8901 \\ 5.6853 & 7.1443 & 17.4770 & 25.0252 & 1.9513 & 1.8292 \\ 2.7946 & 1.4841 & 25.0252 & 55.7782 & 1.6904 & 1.8271 \\ 1.7676 & 7.9915 & 1.9513 & 1.6904 & 61.7301 & 1.4894 \\ 9.1792 & 22.8901 & 1.8292 & 1.8271 & 1.4894 & 74.6812 \end{bmatrix} \quad (E.4)$$

e

$$R_{LQR} = \begin{bmatrix} 32.6687 & 0.6732 & 0.5810 & 0.8181 & 0.6199 \\ 0.6732 & 34.6592 & 0.7615 & 0.4161 & 0.6527 \\ 0.5810 & 0.7615 & 37.2962 & 0.6119 & 0.7926 \\ 0.8181 & 0.4161 & 0.6119 & 48.4470 & 0.6957 \\ 0.6199 & 0.6527 & 0.7926 & 0.6957 & 52.8291 \end{bmatrix} \quad (E.5)$$

A Tabela E.5, exibe-se os autovalores de malha fechada, as sensibilidades dos autovalores e os ganhos usando o controlador *LQR*, obtidos usando-se as matrizes de ponderação (E.4) e (E.5), geradas pelo *AG*. O controlador ótimo é avaliado sob o ponto de vista de controle. Inicialmente, verifica-se que os autovalores de malha fechada e as sensibilidades obtidas pelo controlador gerado pelo *AG* satisfazem, as restrições de projeto conforme a tabela referida, isto é, os autovalores estão na faixa especificada pelo projetista e satisfazem as sensibilidades normalizadas.

Tabela E.5: Autovalores, Sensibilidades e Ganhos do Controlador-*DFIG*.

No	Autovalores	$s_i$	Ganhos
1	-56.6226	0.0010	-0.0621 -0.0054 -0.0349 -0.0111 -0.0135 -0.0767
2	-40.2041	0.1380	-0.0052 -0.0703 -0.0324 0.0043 -0.0407 -0.1348
3	-13.9865 + 1.1743j	0.5244	-0.0044 -0.0040 -0.0229 -0.0297 -0.0058 -0.0202
4	-13.9865 - 1.1743j	0.6109	-0.0000 -0.0070 0.0479 0.1800 -0.0023 -0.2442
5	-22.0073	0.8599	0.0040 0.0124 0.0062 -0.0014 0.1801 0.0122
6	-18.6784	0.9132	

No algoritmo genético algumas estratégias são usadas para realizar a seleção dos indivíduos no processo evolutivo. Essas estratégias são expressas pelas funções de *fitness*, *crossover* e mutação. Essas funções podem ser reformuladas dependendo do problema e das especificações do projeto. Essas funções influenciam na dinâmica da determinação das matrizes de ponderação *Q* e *R*. Nestes experimentos foram usados a função de *fitness* que pontua cada indivíduo a partir da satisfação das especificações. Estudos sobre a utilização da função de *fitness* usando o método elitista, que mantém os melhores indivíduos no decorrer das gerações, pode ser uma alternativa interessante para implementação.

## E.4 Análise de Convergência-*DFIG*

O desempenho da população inicial relatado pela função objetivo, representado pelas sensibilidades de cada indivíduo, é mostrado na Fig. E.3. A população inicial é gerada com 50 indivíduos cromossômicos, sendo que 12%, aproximadamente seis (06), dos indivíduos da população possuem valores alto da sensibilidades e a maioria dos indivíduos, possuem sensibilidade menor que dois (02). Pode-se dizer que, para a abordagem proposta, bons indivíduos devem ter  $s_i \leq 0.6$ .

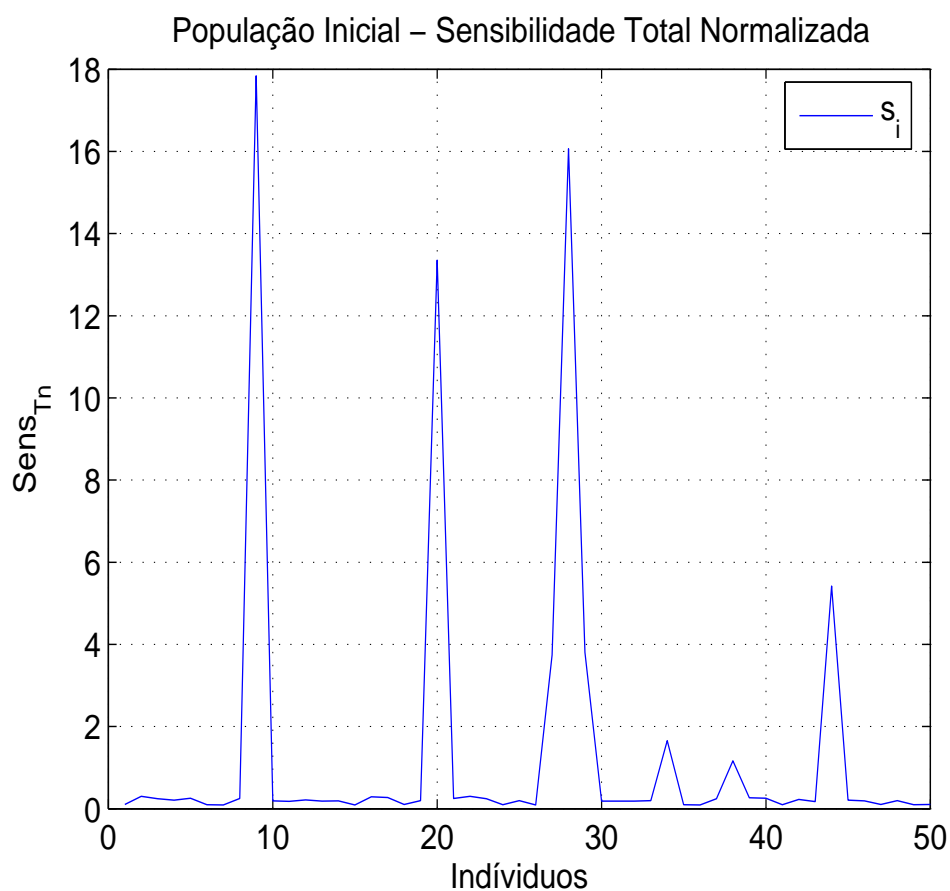


Figura E.3: Sensibilidade total normalizada da população inicial.

Os melhores indivíduos apresentam maior *fitness* total, o equivalente a *fitness* dos autovalores e das sensibilidades mapeadas para cada indivíduo conforme a Fig.



E.4. Iremos agora analisar o comportamento da população inicial com relação à avaliação de desempenho sob o ponto de vista de concentração de indivíduos. Esta procedimento leva em consideração os efeitos da concentração global dos objetivos dos autovalores e sensibilidades.

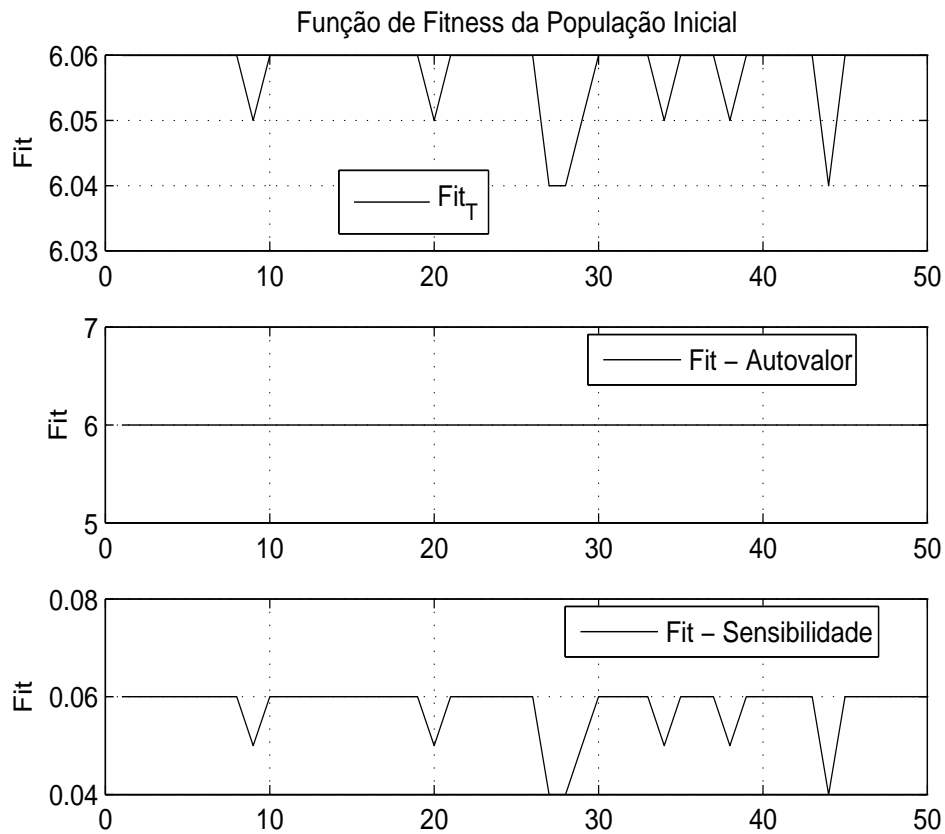


Figura E.4: Função de *fitness* total população inicial.

Na tabela E.6 exibe-se os indivíduos que não alcançaram as restrições de sensibilidade. Neste caso, oito (08) indivíduos deixaram de satisfazer as restrições de sensibilidades.

Tabela E.6: *Fitness* da População Inicial

Indivíduos	$Fitness_A$	$Fitness_S$	$Fitness_T$
9	6.000	0.050	6.060
20	6.000	0.050	6.060
27	6.000	0.040	6.060
28	6.000	0.040	6.060
29	6.000	0.050	6.060
34	6.000	0.050	6.060
38	6.000	0.050	6.060
44	6.000	0.040	6.060

Apresentam-se os histogramas da Fig.E.5 que fornecem um indicativo de agrupamento das *fitness* de autovalores, sensibilidades e *fitness* total respectivamente, de modo que este agrupamento é construído em torno de um valor central. Observa-se que aproximadamente 84% dos indivíduos desta população possuem valor de *fitness* em torno de 6.059, isto é, 42 indivíduos satisfazem a relação  $6.058 \leq Fit_{total} \leq 6.06$ . Nota-se uma concentração de nichos, ou seja, pouca variedade do material genético, o que poderá levar o algoritmo à convergência prematura. Ainda na faixa de  $6.047 \leq Fit_{total} \leq 6.05$ , 6 indivíduos possuem valor de *fitness* em torno de 6.048, que corresponde a um percentual de 12%.

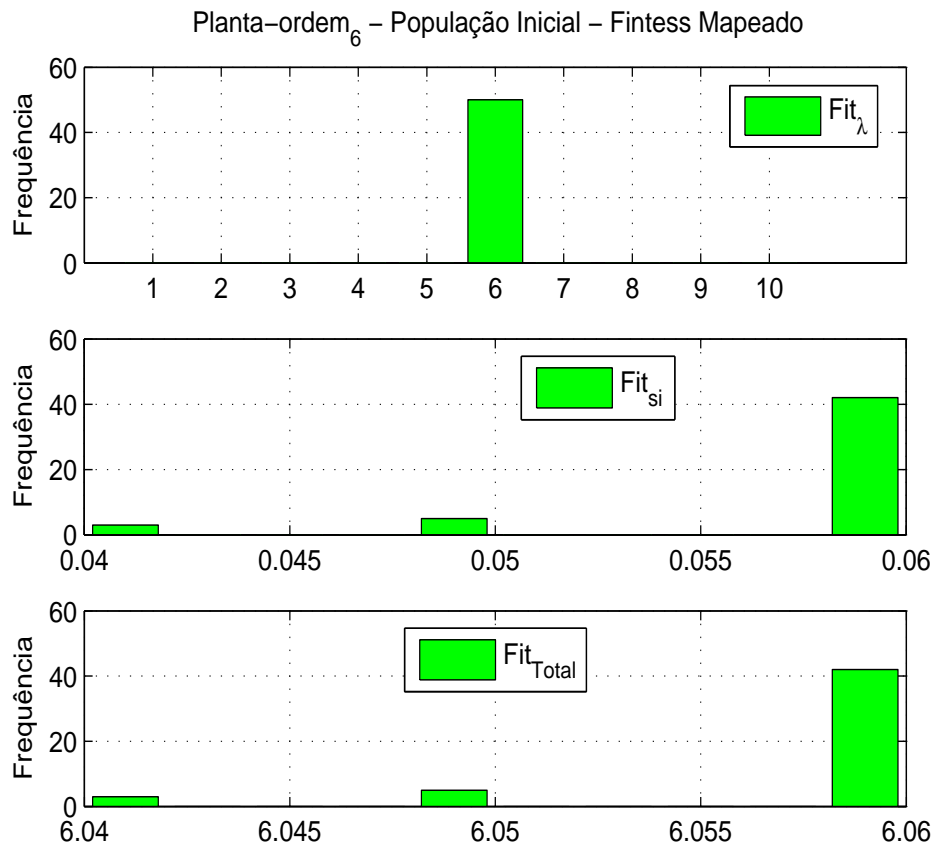


Figura E.5: Função de *fitness* população Inicial-*DFIG*.

## E.5 Processo de Busca Geração 10-*DFIG*

Avalia-se agora o desempenho do modelo sob o ponto de vista da construção da população do processo de busca, geração 10 por meio da soma das sensibilidades para cada indivíduo, sendo que cada indivíduo da população do processo de busca é representado pela estrutura de otimização, que são obtidos de acordo com as equações do modelo de *fitness*. Como já vimos, a segunda parte para análise do comportamento da população geração 10, é formulada pela avaliação do seu desempenho sob o ponto de vista da concentração de indivíduos. Na Fig. E.6, a maioria dos indivíduos possuem  $s_i \leq 0.6$ .

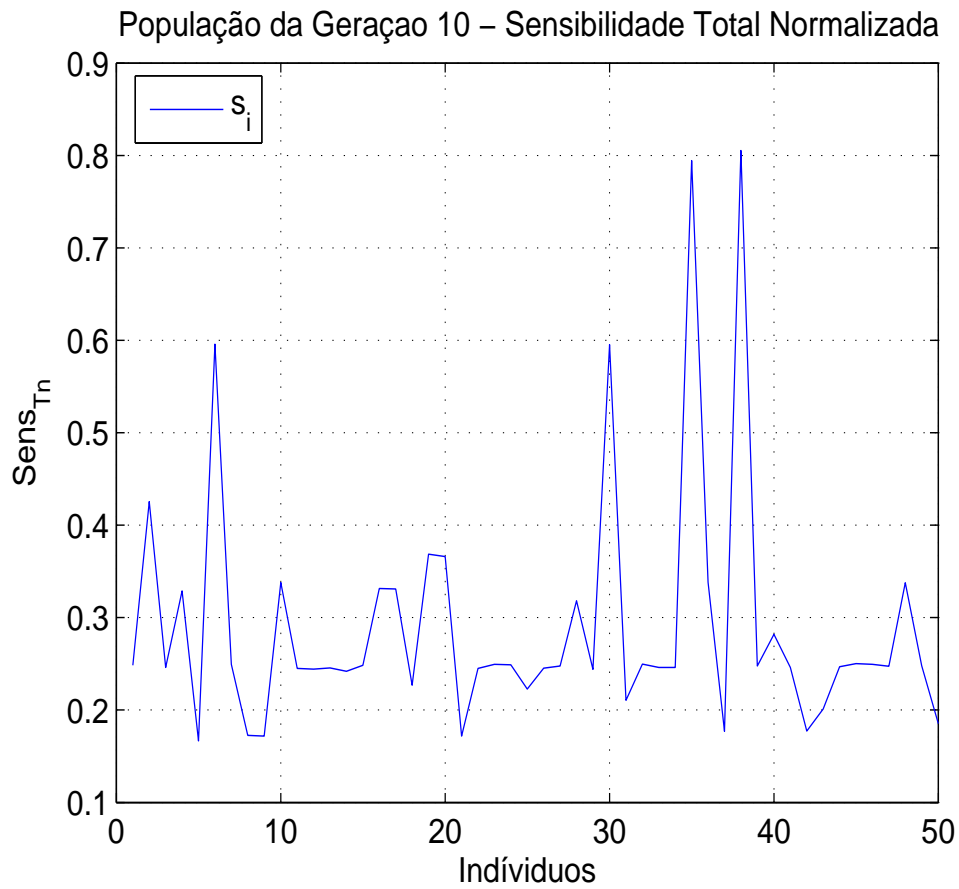


Figura E.6: Sensibilidade total normalizada população geração 10-*DFIG*.

A Fig. E.7 ilustra a função de *fitness* da população geração 10, sendo tomadas a *fitness* total, as *fitness* de autovalores e sensibilidades mapeadas. Diversos indivíduos satisfizeram as restrições de autovalores e sensibilidades.

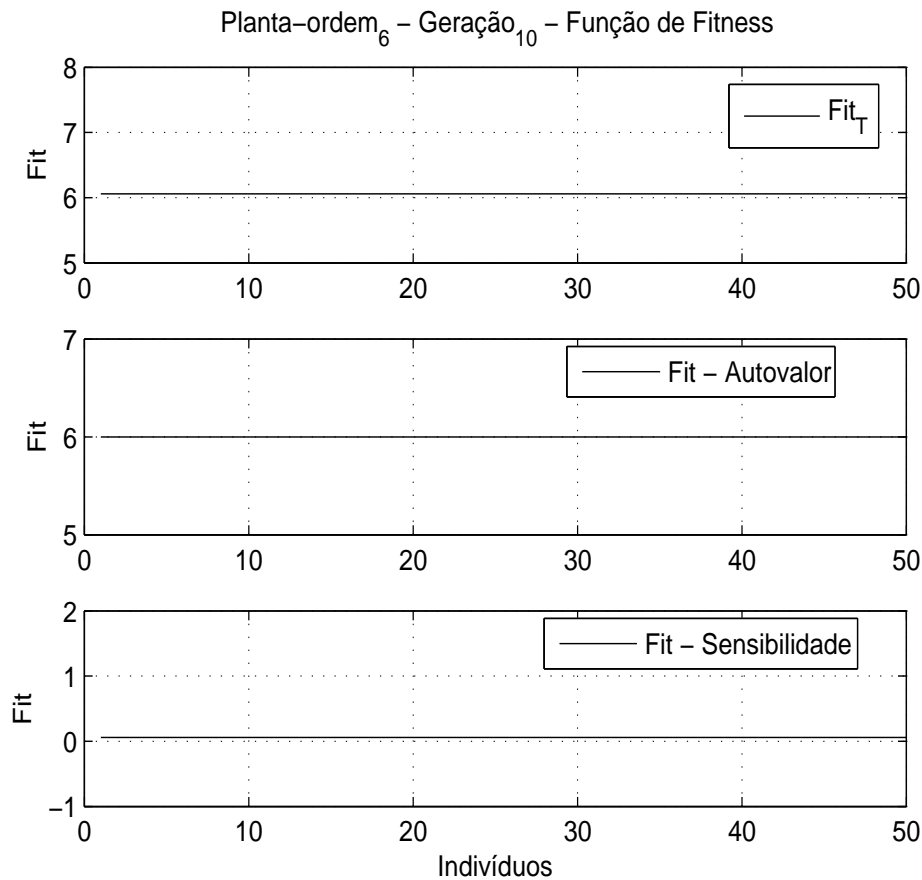


Figura E.7: Função de *fitness* população geração 10-*DFIG*.

Os histogramas da Fig. E.8 representam o agrupamento das *fitness* de autovalores, sensibilidades e *fitness* total respectivamente. Nota-se que este agrupamento é construído em torno de um valor central. Uma vez que o número de indivíduos está relacionado à frequência, observa-se que 100% dos indivíduos desta população possuem valor de fitness em torno de 6.06, isto é, os 50 indivíduos satisfazem à relação  $6.00 \leq Fit_{total} \leq 7.00$ . Percebe-se que existe uma concentração de nichos, isto é, pouca variedade do material genético. O algoritmo poderá convergir prematuramente.

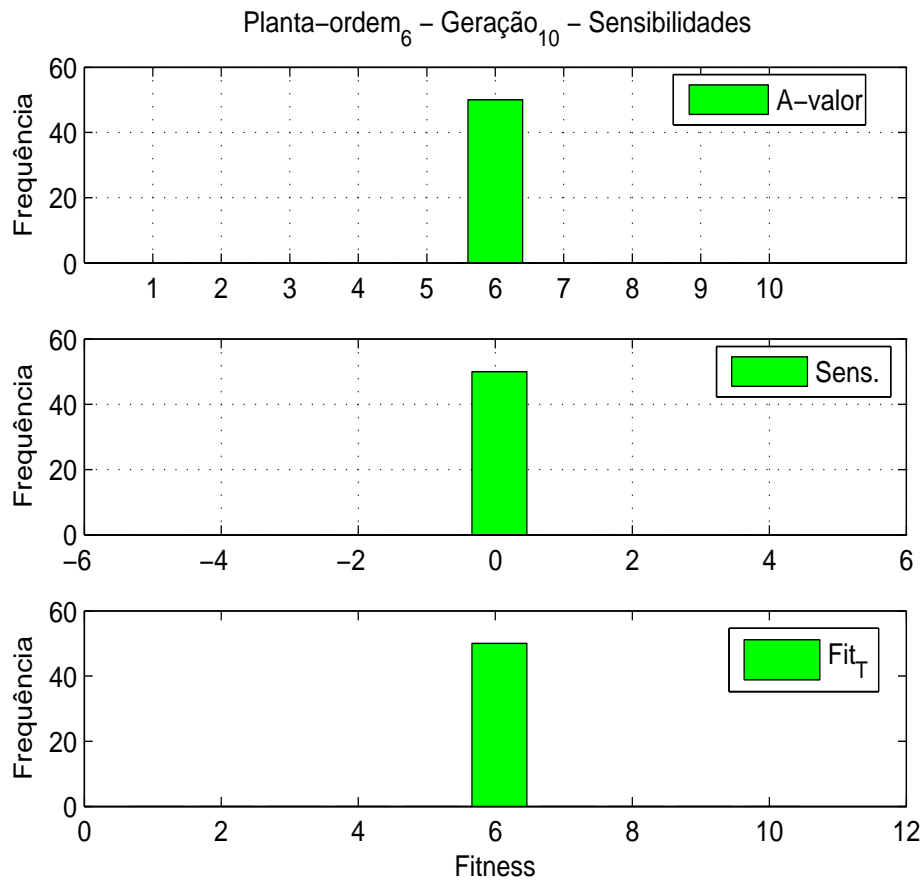


Figura E.8: Função de *Fitness* Mapeada População Geração 10-*DFIG*.

## E.6 População Final-*DFIG*

A Fig. E.9 ilustra a geração final do processo de evolução, relacionado a função objetivo que é representada pelas sensibilidades de cada indivíduo.

A maior parte do individual satisfaz os requisitos para as sensibilidades, mas apenas alguns deles são indivíduos viáveis, isto aconteceu devido estes indivíduos terem sensibilidades menor que 0,6. Alguns não podiam satisfazer as restrições de positividade, retornando as matrizes  $Q$  ou  $R$  que não são semidefinidas positivas ou definidas positivas, respectivamente. Apenas nove (09) indivíduos satisfizeram todas as restrições de positividade, sensibilidade e alocação de autoestrutura.

Tem-se a Fig. E.9 que ilustra a performance da população final por meio da soma das sensibilidades  $\sum p_i s_1(Q, R)$  para cada indivíduo da população final representado pela estrutura de otimização que são obtidas de acordo com as equações do modelo de *fitness*.

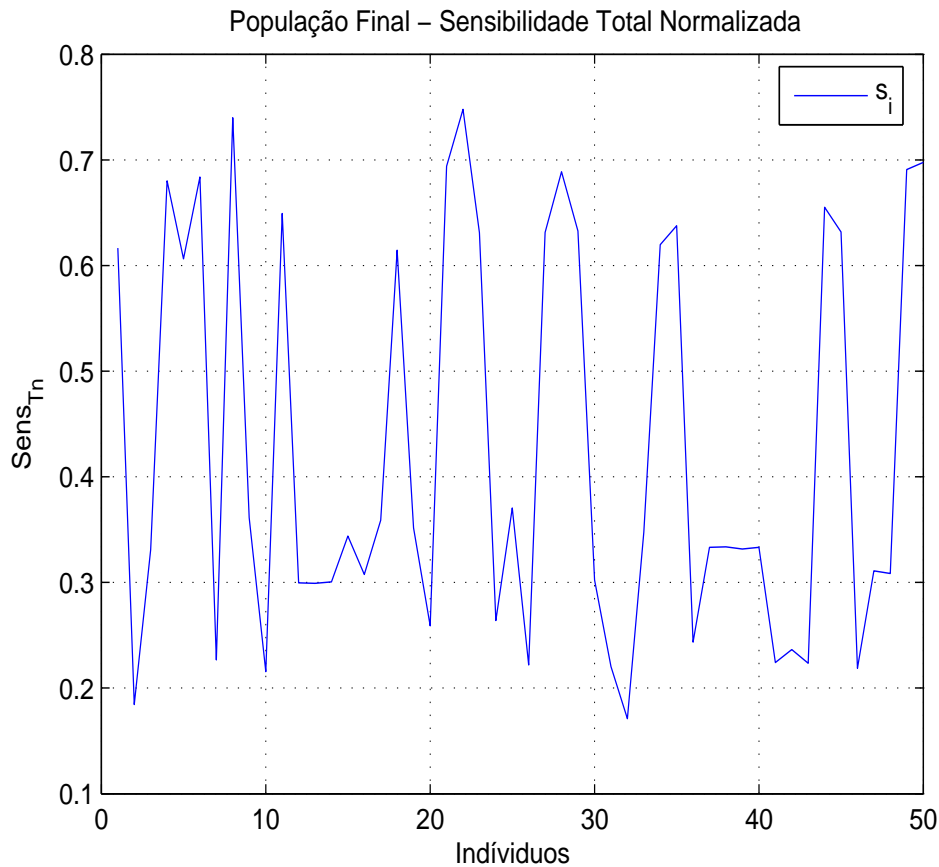
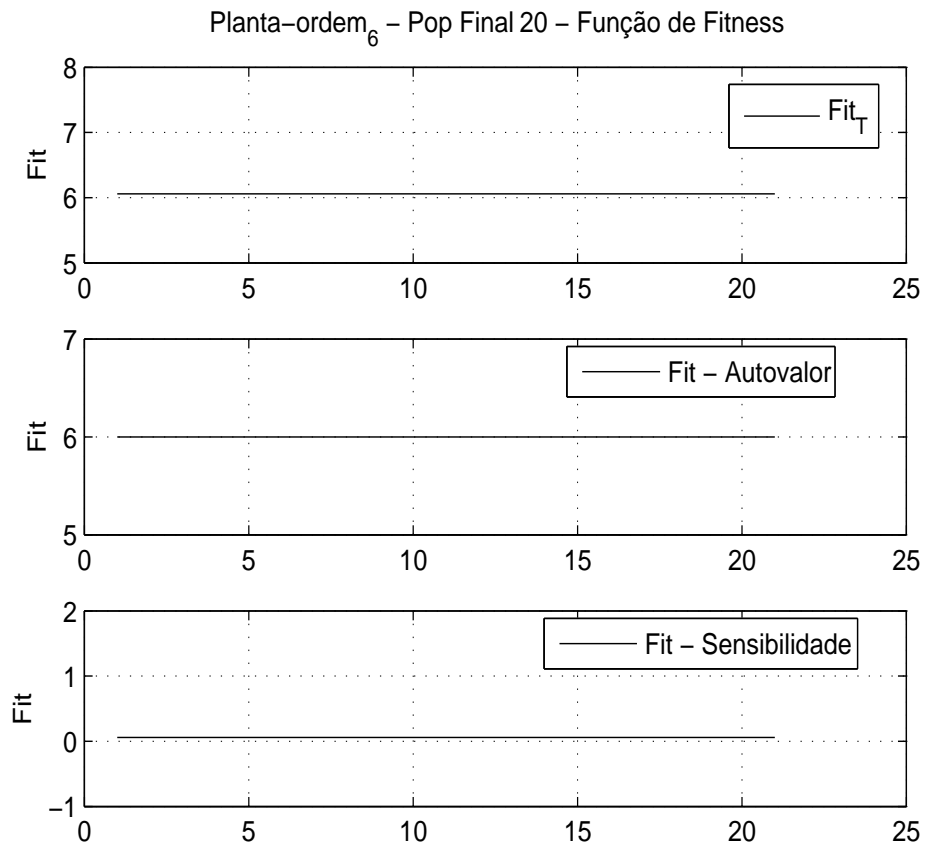


Figura E.9: Sensibilidade Total Normalizada da População Final-*DFIG*.

A Fig. E.10 mostra a função de *fitness* da população final, sendo considerados a *fitness* total, as *fitness* dos autovalores e sensibilidades mapeadas.

Figura E.10: Função de *Fitness* da População Final-*DFIG*.



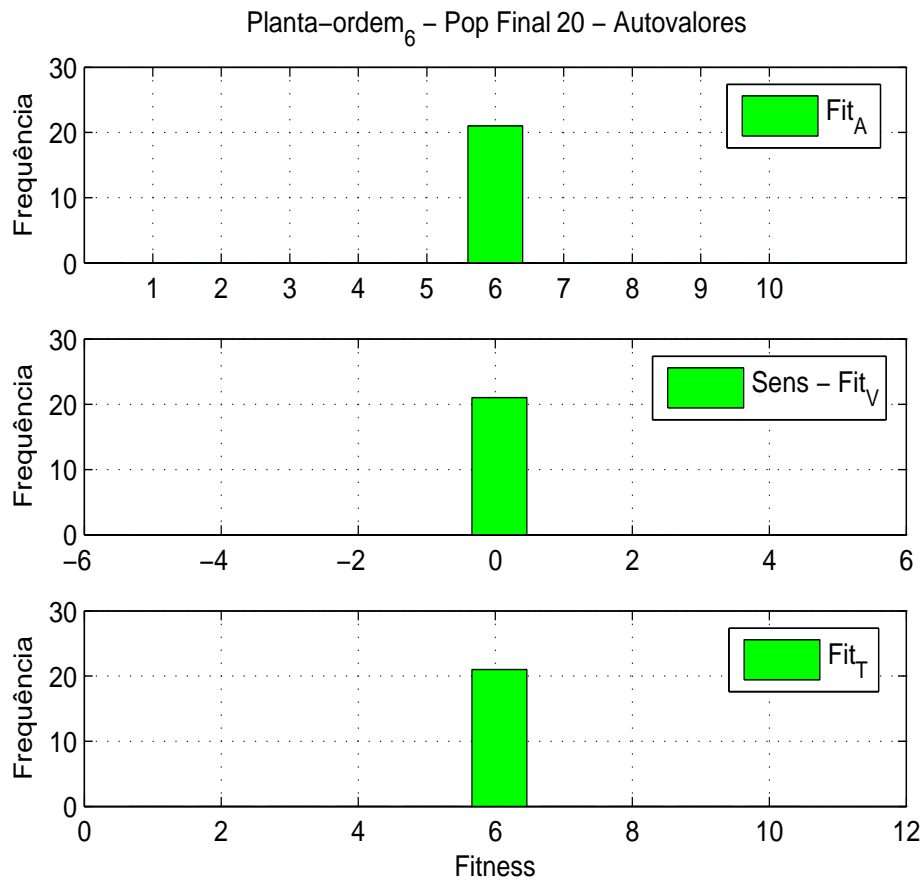


Figura E.11: Função de Fitness Mapeada da População Final-*DFIG*.

As respostas em frequências em termos dos maior e menor valor singular em *dB* apresentados na Fig. E.12, mostra que o controlador gerado por um indivíduo viável, da população final promove respostas em SVD bastante próximas.

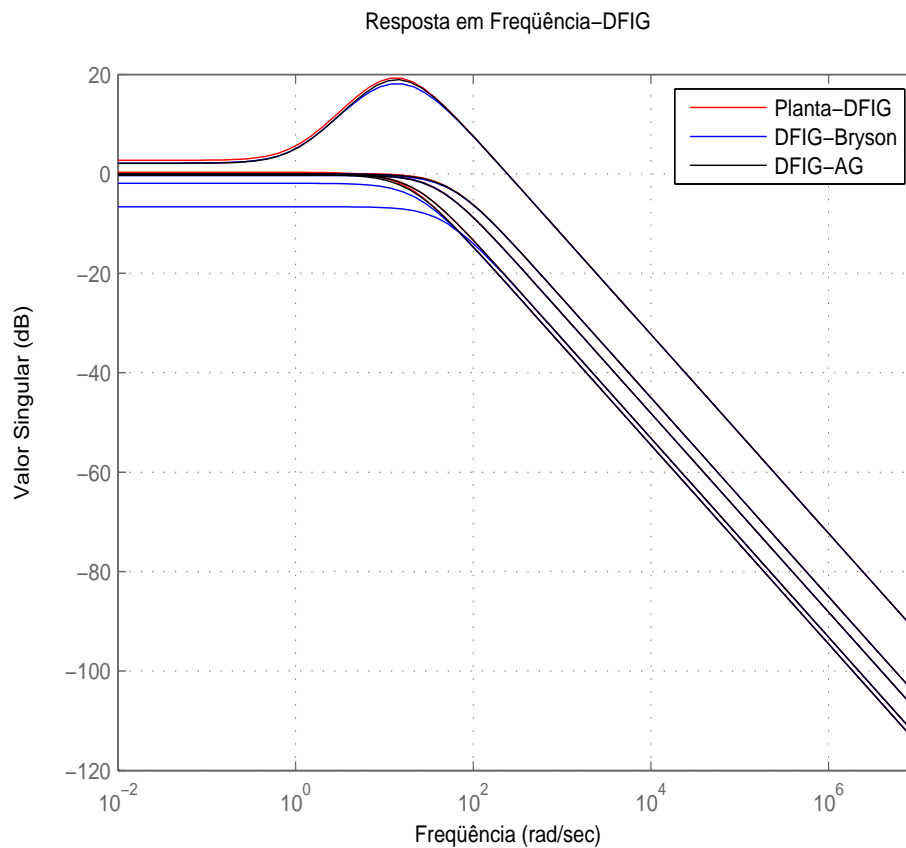


Figura E.12: Decomposição em Valores Singulares-*DFIG*.

Iremos agora apresentar a avaliação da rede neural recorrente. Neste item, é apresentada a superfície usada para inicializar a rede neural recorrente para o sistema *DFIG*. A Fig. E.13 mostra a energia total para uma faixa larga de valores dos parâmetros  $\eta_v$  e  $\eta_z$ . Assim podemos ter uma melhor noção da influência desses parâmetros, e deste modo, escolher a região e selecionar o par de valores  $\eta_v$  e  $\eta_z$  em que a energia é mínima.

A energia é mais dependente do parâmetro  $\eta_v$  que  $\eta_z$ . Para valores de  $\eta_v$  maior que 1550, a energia começa a aumentar e portanto, deve ser evitada a escolha de valores maiores que este.

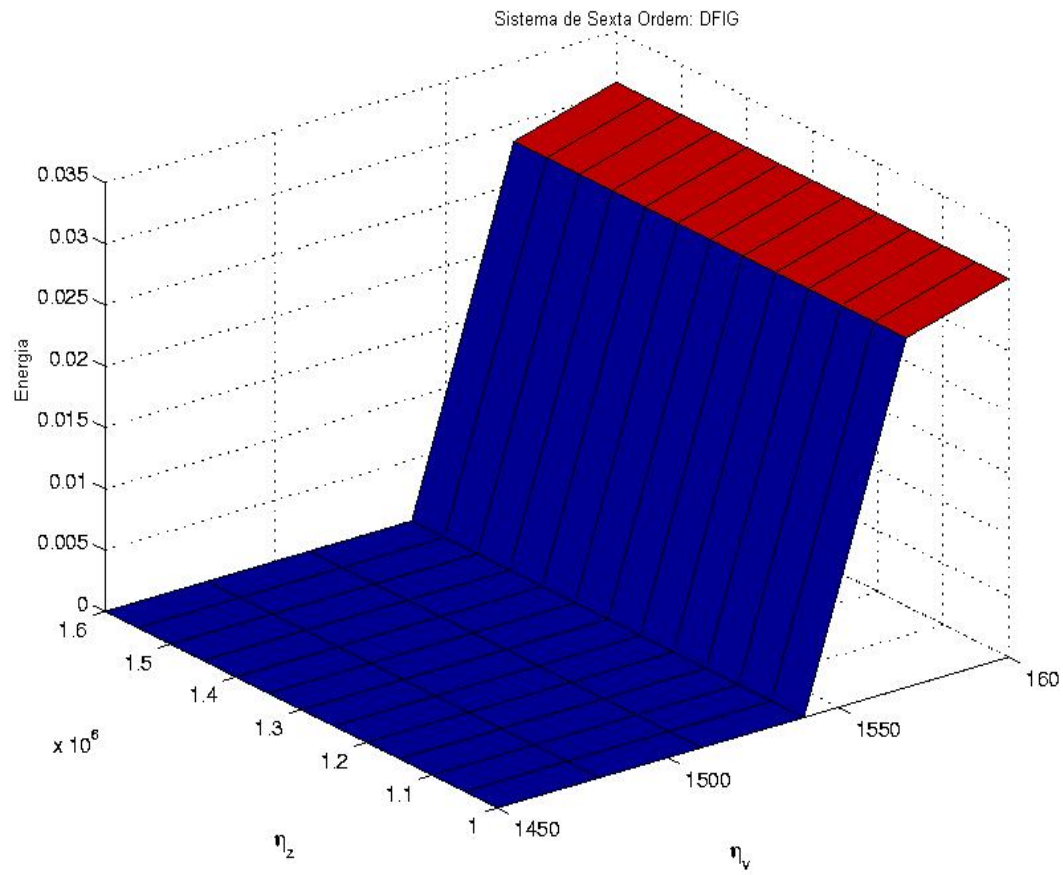


Figura E.13: Superfície da energia para variação dos parâmetros- $\eta_v=1450-1600$  e  $\eta_z=100000-1600000$ .

A Fig. E.14 exibe a norma do infinito da solução da equação de *Riccati* para diversos valores de  $\eta_v$  e  $\eta_z$ , a fim de observar o comportamento da RNR para valores diferentes de convergência e parâmetros de aprendizagem  $\eta_v$  e  $\eta_z$ . Os valores mínimos da solução da equação de *Riccati*, estão compreendidos na região onde  $1000 \ll \eta_v < 1500$ . Desta região, refina-se a superfície e escolhe-se os valores que melhor resolve a EAR e promova a solução mínima.

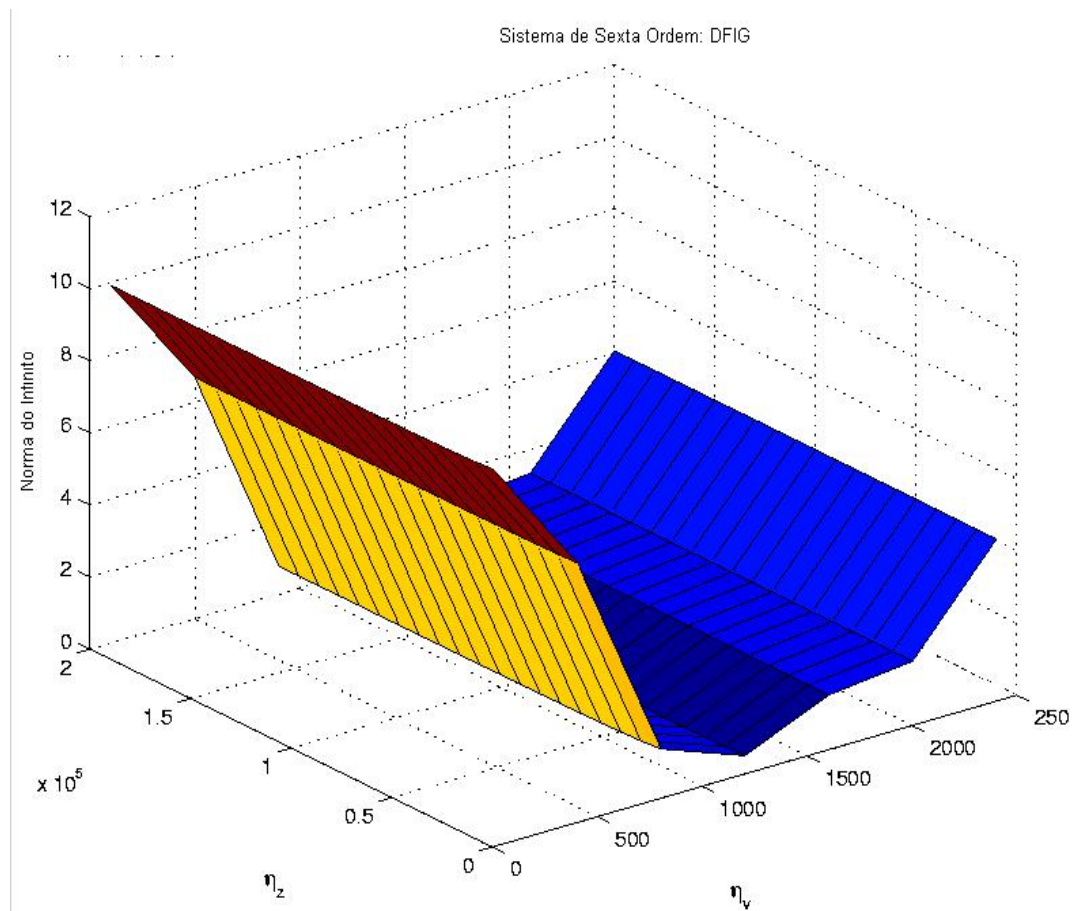


Figura E.14: Norma do infinito da solução da EAR para  $\eta_v=1000-2500$  e  $\eta_z=10000-2000000$ .

As Fig. E.15, E.16 e E.17 mostram de fato, a proposta da RNR: minimizar a função energia, obter a solução da EAR (P) e o Fator de *Cholesky* (L). À medida que o treinamento da RNR ocorre, gera-se uma seqüência de pontos da função energia  $Energia(\xi) = MSE\{g(P) + H(P, H)\}$ , de tal maneira que a energia decresça a cada iteração, uma vez que a direção de busca do treinamento é baseado no método Máximo Declive (*Steepest Descent*) e o tamanho do passo é fornecido pelos parâmetros ( $\eta_v$ ,  $\eta_z$ ) que são fixos no processo de busca. Nestas figuras, a cada iteração geram se as seqüências associadas a P e L, sendo que as mesmas deverão convergir para o ponto ótimo  $P^*$  e  $L^*$ .

Da fundamentação teórica deste método, sabe-se que o mesmo utiliza apenas derivadas de primeira ordem da função energia. Como o gradiente aponta na direção de maior crescimento da energia no ponto (P,L), o método procura em

cada ponto caminhar na direção oposta  $-(dP, dL)$ . A partir de um certo número de iterações, o decremento não apresenta maior significância.

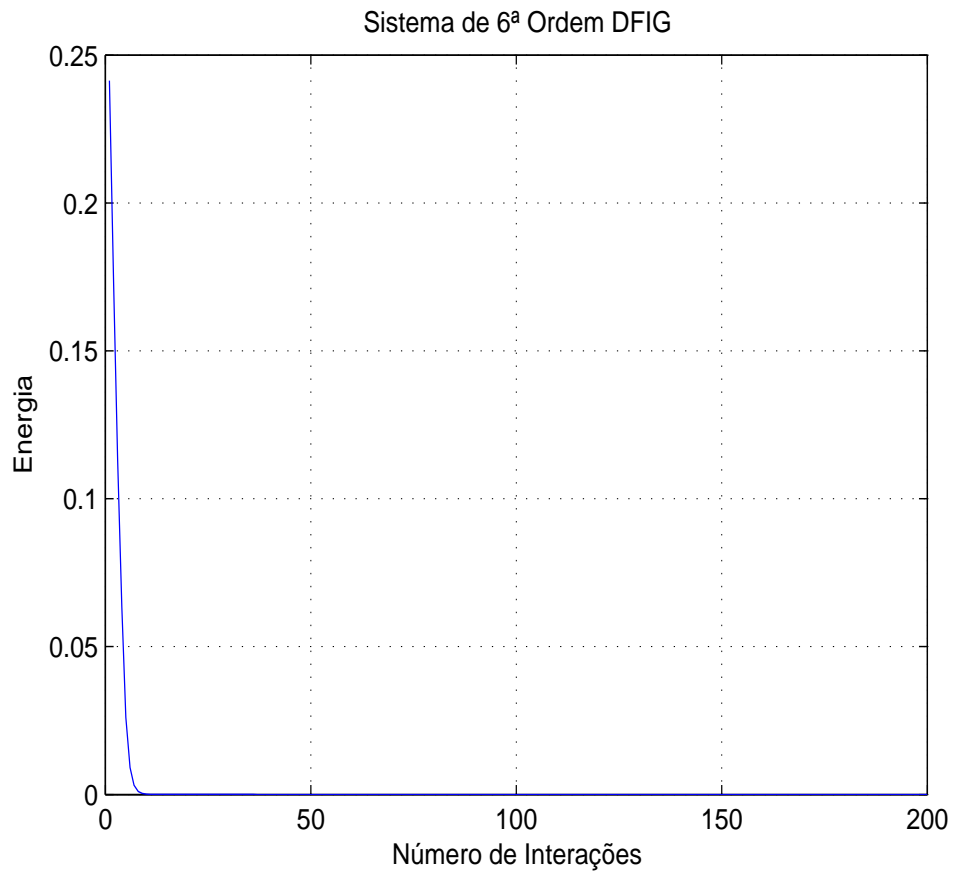


Figura E.15: Energia-*DFIG* em função do número de iterações.

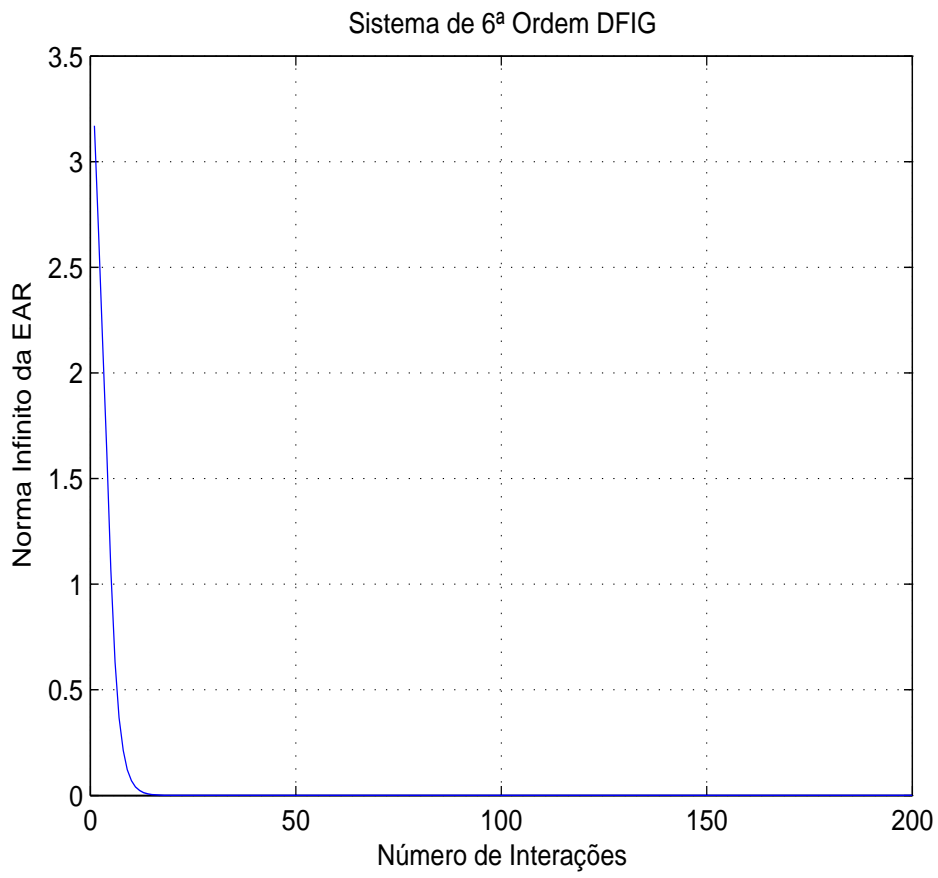


Figura E.16: Norma do infinito da solução da *EAR-DFIG* versus número de iterações.

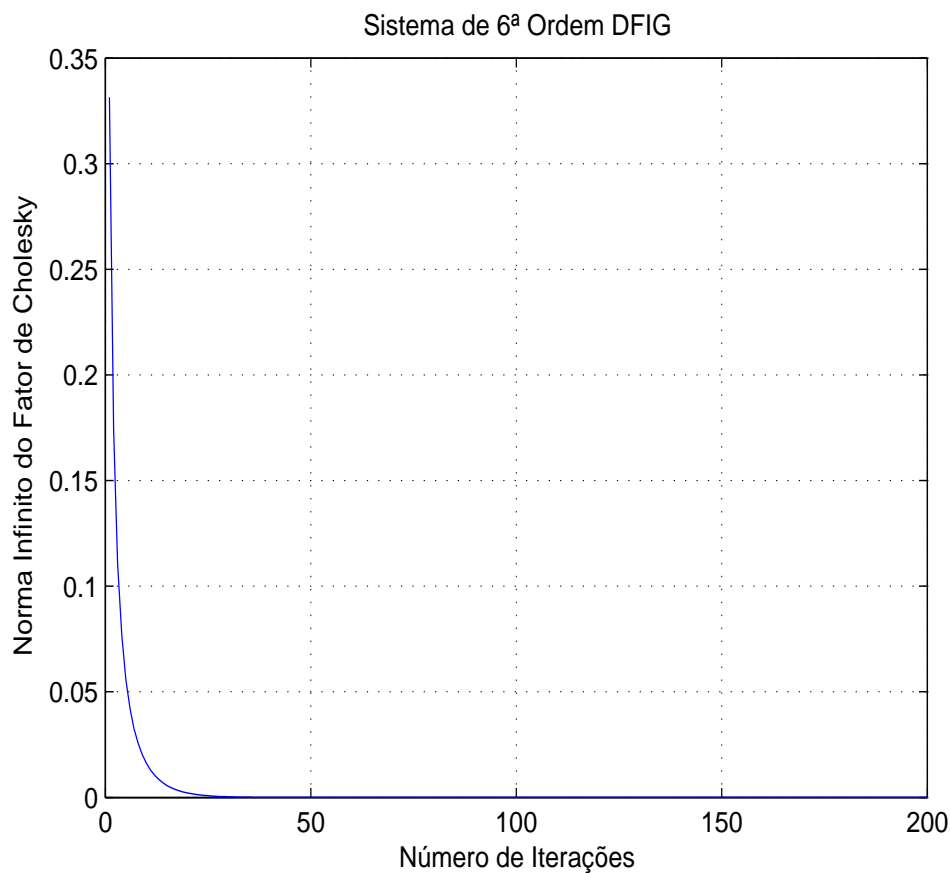


Figura E.17: Norma do infinito da solução do Fator de Cholesky - *DFIG*.

As respostas ao impulso são aplicadas para verificar o desempenho do projeto *LQR*. A Fig. E.18 compara as resposta ao impulso do sistema *DFIG* em relação ao método de seleção das matrizes  $Q$  e  $R$ . Comparamos os métodos de *Bryson* e *AG*, nota-se que ambas metodologias regulam o sistema. Como o sistema *DFIG* é um sistema multivariável, tem-se cinco entradas e seis saídas, porém, escolheu-se a quarta e quinta entradas para expressar o comportamento da resposta ao impulso. Uma melhor atenuação e resposta de velocidade, pode ser vista a partir do projeto *LQR* que utilizou o *AG* para selecionar as matrizes de ponderação  $Q$  e  $R$ .

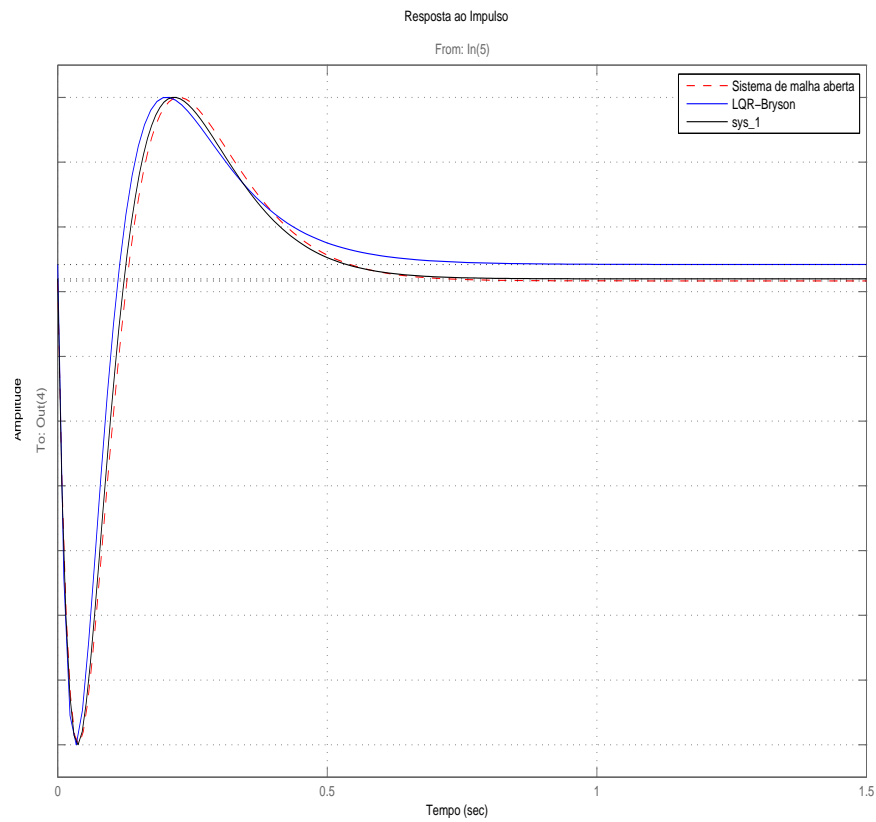


Figura E.18: Resposta ao impulso-planta eólica, método Bryson e LQR-AG .

## E.7 Solução Schur-EAR

As matrizes de ponderação  $Q$  e  $R$  selecionadas pelo AG são constantes contidas na estrutura da equação de *Riccati*, que é resolvida pelo método de *Schur*. A determinação dos ganhos do controle ótimo envolve uma série de operações complexas, tais como a solução da equação algébrica de *Riccati*, Eq. (3.34),



$$K_{Schur} = \begin{bmatrix} -0.5204 & 0.0230 & -0.0023 & -0.1788 & -0.0650 & 0.0937 \\ 0.0007 & -1.0507 & -0.0267 & -0.1150 & -0.0595 & 0.0544 \\ 0.0241 & 0.0105 & -1.0679 & -0.1567 & -0.1279 & 0.1058 \\ 0.0018 & 0.0256 & -0.0099 & 3.0516 & 0.0289 & -1.9801 \\ 0.0161 & 0.0342 & 0.0265 & -0.1032 & 1.8151 & 0.0494 \end{bmatrix}$$

$$P_{LQR} = \begin{bmatrix} 0.0968 & 0.0042 & 0.0184 & -0.0161 & -0.0020 & 0.0149 \\ 0.0042 & 0.1877 & 0.0101 & -0.0178 & -0.0098 & 0.0156 \\ 0.0184 & 0.0101 & 1.2800 & -0.0346 & -0.0327 & 0.0220 \\ -0.0161 & -0.0178 & -0.0346 & 1.0030 & 0.0732 & -0.5491 \\ -0.0020 & -0.0098 & -0.0327 & 0.0732 & 0.7966 & -0.0561 \\ 0.0149 & 0.0156 & 0.0220 & -0.5491 & -0.0561 & 1.1815 \end{bmatrix}$$

Os autovalores do *DFIG* com controlador *LQR* são apresentados na Tabela E.7. Os cinco primeiros autovalores apresentam uma dinâmica mais rápida que o sexto autovalor. O último autovalor é real e provoca um amortecimento lento diante aos demais autovalores.

Tabela E.7: Autovalores do Sistema de Malha Fechada *DFIG*

Nº	Autovalores
1	-13.2352
2	-10.0848
3	-9.2088 + j4.2459
4	-9.2088 + j4.2459
5	-6.5748
6	-2.5335

## E.8 Solução Neuronal-EAR

O algoritmo que implementa a solução neuronal da equação de *Riccati*, consiste na implementação das estrutura de otimização, Eq. (4.31), que minimiza a função

energia erro médio quadrático (EMQ). Os ganhos do *LQR* utilizando a RNA são obtidos a partir de  $K_{RNA} = R^{-1}B^T P_{rna}$  e  $K_{SCHUR} = R^{-1}B^T P_{schur}$ , sendo dados por:

$$K_{Rna} = \begin{bmatrix} -0.52037 & 0.022959 & -0.0023414 & -0.17877 & -0.065043 & 0.093719 \\ 0.00070022 & -1.0507 & -0.02668 & -0.11498 & -0.059479 & 0.054399 \\ 0.024136 & 0.01047 & -1.0679 & -0.15668 & -0.12792 & 0.1058 \\ 0.0017812 & 0.025559 & -0.009857 & 3.0516 & 0.028864 & -1.9801 \\ 0.016113 & 0.034199 & 0.026501 & -0.10323 & 1.8151 & 0.049405 \end{bmatrix} \quad (\text{E.6})$$

$$K_{Schur} = \begin{bmatrix} -0.52037 & 0.022959 & -0.0023414 & -0.17877 & -0.065043 & 0.093719 \\ 0.00070022 & -1.0507 & -0.02668 & -0.11498 & -0.059479 & 0.054399 \\ 0.024136 & 0.01047 & -1.0679 & -0.15668 & -0.12792 & 0.1058 \\ 0.0017812 & 0.025559 & -0.009857 & 3.0516 & 0.028864 & -1.9801 \\ 0.016113 & 0.034199 & 0.026501 & -0.10323 & 1.8151 & 0.049405 \end{bmatrix} \quad (\text{E.7})$$

a solução da equação algébrica de *Riccati*

$$P_{Schur} = \begin{bmatrix} 0.096762 & 0.0042152 & 0.0184 & -0.016147 & -0.0020058 & 0.014916 \\ 0.0042152 & 0.18765 & 0.010108 & -0.017782 & -0.0097501 & 0.015551 \\ 0.0184 & 0.010108 & 1.28 & -0.034605 & -0.03274 & 0.022049 \\ -0.016147 & -0.017782 & -0.034605 & 1.003 & 0.073223 & -0.54906 \\ -0.0020058 & -0.0097501 & -0.03274 & 0.073223 & 0.79662 & -0.056079 \\ 0.014916 & 0.015551 & 0.022049 & -0.54906 & -0.056079 & 1.1815 \end{bmatrix} \quad (\text{E.8})$$

$$P_{Rna} = \begin{bmatrix} 0.096762 & 0.0042152 & 0.0184 & -0.016147 & -0.0020058 & 0.014916 \\ 0.0042152 & 0.18765 & 0.010108 & -0.017782 & -0.0097501 & 0.015551 \\ 0.0184 & 0.010108 & 1.28 & -0.034605 & -0.03274 & 0.022049 \\ -0.016147 & -0.017782 & -0.034605 & 1.003 & 0.073223 & -0.54906 \\ -0.0020058 & -0.0097501 & -0.03274 & 0.073223 & 0.79662 & -0.056079 \\ 0.014916 & 0.015551 & 0.022049 & -0.54906 & -0.056079 & 1.1815 \end{bmatrix} \quad (\text{E.9})$$

Os ganhos  $K_{Rna}$ ,  $K_{Schur}$  e as soluções da equação algébrica de *Riccati*  $P_{rna}$ ,  $P_{schur}$  fornecidos pela rede neural artificial, de fato, satisfazem a Eq. (3.5) produzindo os erros abaixo:

$$E(Ric_{Rna}) = A^T P + PA - PBR^{-1}B^T P + Q$$

$$E(Ric_{Schur}) = A^T P + PA - PBR^{-1}B^T P + Q.$$

$$E(Ric_{Rna}) = \begin{bmatrix} -1.88e-015 & -3.46e-017 & 1.45e-016 & 2.42e-016 & 1.2e-016 & -5.96e-016 \\ -4.16e-017 & -8.88e-016 & 4.16e-017 & 8.32e-017 & 1.24e-016 & -2.77e-016 \\ 1.17e-016 & 1.38e-017 & -5.51e-014 & 4.16e-016 & 6.24e-016 & -2.47e-015 \\ 2.15e-016 & 6.93e-017 & 4.44e-016 & -2.66e-015 & -3.46e-016 & -2.34e-014 \\ 1.24e-016 & 1.66e-016 & 5.68e-016 & -5.68e-016 & -5.32e-015 & 2.83e-015 \\ -5.41e-016 & -2.35e-016 & -2.66e-015 & -2.52e-014 & 3.05e-015 & -2.50e-013 \end{bmatrix} \quad (E.10)$$

$$E(Ric_{Schur}) = \begin{bmatrix} -5.55e-016 & 4.70e-015 & -1.24e-016 & 4.80e-015 & -2.95e-015 & -9.21e-015 \\ -7.63e-016 & -3.99e-015 & 2.95e-015 & 2.33e-015 & -8.32e-017 & -4.69e-015 \\ -1.38e-016 & -1.22e-015 & -1.11e-014 & -2.52e-015 & 1.16e-015 & 7.07e-016 \\ -6.93e-018 & 2.05e-015 & -2.21e-014 & -8.88e-016 & -1.43e-014 & -1.28e-014 \\ 2.40e-015 & 5.5e-015 & -1.25e-014 & -1.24e-014 & -1.24e-014 & 1.83e-015 \\ 1.40e-015 & -3.33e-016 & 1.46e-014 & 1.55e-014 & 2.05e-015 & -3.55e-015 \end{bmatrix} \quad (E.11)$$

Os ganhos calculados tanto pelo método de *Schur* e pela RNA, foram próximos, sendo os da RNA mais estáveis, e não é percebido visto que a diferença das soluções está na precisão decimal a partir de casas decimais.

Observou-se que de acordo com os autovalores do sistema e a diagonal das matrizes  $Q$  e  $R$  influenciam diretamente no desempenho e na convergência da RNA, deste modo, os valores das matrizes  $Q$  e  $R$  podem ser reduzidos ou aumentados uniformemente, melhorando assim a solução da equação de *Riccati*.

Os ajustes das constantes  $\eta_z$  e  $\eta_v$  representam com que rapidez o algoritmo convergirá ou divergirá e o quão boa uma solução pode ser. A análise de resultados da solução *Neuronal-ARE*, são baseadas no comportamento da camada de saída considerando a entrada e as camadas ocultas.

# PROJETO *LQR* NEURO-GENÉTICO: AERONAVE

---

Os testes de avaliação dos modelos de busca genética e da eficiência da lei de controle são realizados em um modelo de uma aeronave *Lockheed*, tipo *L1011 Tristar*. Este modelo foi utilizado por (Davis e Clarke ,1995) para realizar uma alocação de autoestrutura, ou seja, alocar autovalores em uma determinada faixa e autovetores para o desacoplamento dos modos. Utiliza-se o projeto do regulador linear quadrático para determinar os ganhos do controlador por realimentação de estados e um algoritmo genético paralelo cujo objetivo é realizar a busca das matrizes de ponderação que devem satisfazer as especificações de projeto formuladas pelo projetista.

O objetivo de controle é a alocação de autoestruturas em sistemas dinâmicos multivariáveis que é imposta pela lei de Controle Ótimo e o objetivo do procedimento da análise é promover a aceleração da convergência por meio de métricas que estão fundamentadas em momentos estatísticos de primeira e segunda ordem. O desempenho do modelo é avaliado sob o ponto de vista da construção da população inicial, populações do processo de busca e população final.

A análise de convergência conduz ao desenvolvimento de regras baseadas em métricas da função de *fitness*. Os testes de avaliação dos modelos de busca genética e da eficiência da lei de controle são realizados em um modelo dinâmico de 6ª ordem. Na subseção F.1 apresenta-se o modelo da aeronave, autovalores do sistema, restrições de autoestrutura e os ganhos do controladores base e do *LQR*.

Na Seção F.2 discute-se a análise de convergência por meio de *fitness* da função objetivo sensibilidade normalizada e *fitness* mapeada com relação a população inicial. Na subseção F.3 apresenta-se o processo de busca (população geração 10) que é formulado por modelos genéticos que formam o seu núcleo, sendo que as operações cromossômicas realizam as operações genéticas. Na subseção F.4 apresenta-se a estatística da *fitness* da população final como um mapeamento dos autovalores e suas respectivas sensibilidades. A subseção F.5 mostra os resultados de desempenho do método de sintonia de ganhos de controladores *LQR* para AAE sob o ponto de vista de controle. Os testes de desempenho da resposta ao impulso são ações tomadas para avaliar o desempenho do método de busca. A subseção F.6 mostra os resultados em valores singulares do sistema dinâmico (aeronave) e o controlador. Na subseção F.7 discute-se à análise de convergência e estabilidade da RNA para solução da EAR. A variação paramétrica é apresentada como uma das responsáveis pelo desempenho e exatidão do funcionamento da RNA. A subseção F.8 estabelece a análise de estabilidade usando o método de *Lyapunov*. A subseção F.9 destaca o treinamento da rede e os parâmetros de inicialização. Na subseção F.10, destaca-se a influência dos parâmetros de sintonia da rede neural, mostrando que uma boa escolha realizada pela RNR conduz a solução do problema de otimização. Discute-se o modelo de fusão na subseção F.11 e finalmente conclui-se este capítulo com observações na Seção F.12.

## F.1 Modelo da Aeronave

O modelo da aeronave é linearizado em uma condição de cruzeiro (Sobel e Shapiro ,1985). As matrizes  $A$  de Estado,  $B$  de Entrada,  $C$  de Saída e  $D$  de Transmissão Direta do sistema dinâmico teste são dadas por:

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

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

$$C = \begin{bmatrix} 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

Os autovalores do modelo dinâmico aeronave são exibidos na Tabela F.1. Observa-se que os dois primeiros autovalores são reais e encontram-se próximos da origem, o que provoca um amortecimento lento. O terceiro e quarto autovalores são complexos com amortecimento lento e pequenas oscilações. O quinto e sexto autovalores apresentam uma dinâmica mais rápida.

Tabela F.1: Autovalores do Sistema Dinâmico *Aeronave*

Nº	Autovalores
1	-0.2276
2	-0.8955
3	-0.7670 - j0.9251
4	-0.7670 + j0.9251
5	-20.000
6	-25.000

O sistema dinâmico teste é utilizado para geração das matrizes de ponderação. Para o Algoritmo Genético proposto, utiliza-se as restrições estabelecidas pelo projetista que são os limites dos autovalores e suas respectivas sensibilidades, (Neto ,2000). Tais restrições foram formuladas a partir da autoestrutura fornecida pela implementação de um Controlador Base, (Davis e Clarke ,1995). Esses dados são mostrados na Tabelas F.2 e F.3

Tabela F.2: Restrições do Projeto *LQR*

Nº.	Autovalor	Sensibilidade
1	$-23.00 \leq Re \leq -13.00$	7.28
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	$-3.00 \leq Re \leq -2.00$	9.92
6	$-3.00 \leq Re \leq -0.20$	2.18

Tabela F.3: Ganhos do Controlador *LQR*.

<i>K</i>	Ganho do Controlador					
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$K_1$	0.185	0.006	-0.032	-1.179	-0.010	1.598
$K_2$	0.019	0.194	-0.422	-0.785	-1.205	2.624

Mostra-se os resultados do Algoritmo Genético desenvolvido para geração das matrizes de ponderação  $Q$  do estado e  $R$  do controle, utilizando o sistema teste. As matrizes de ponderação produzidas pelo *AG* para vinte (20) gerações são

$$Q_1 = \begin{bmatrix} 10.1 & 1.9 & 2.0 & 1.7 & 0.4 & 1.2 \\ & 8.2 & 2.6 & 0.7 & 0.6 & 2.5 \\ & & 7.3 & 1.7 & 2.2 & 1.6 \\ & & & 38.1 & 2.0 & 1.2 \\ & & & & 78.0 & 0.3 \\ & & & & & 29.2 \end{bmatrix} \quad (F.1)$$

$$R_1 = \begin{bmatrix} 31.9 & 4.6 \\ & 25.9 \end{bmatrix} \quad (F.2)$$

A Tabela F.4, mostra os autovalores de malha fechada, as sensibilidades dos autovalores e os ganhos do controlador, obtidos utilizando-se as matrizes de ponderação  $Q_1$  e  $R_1$  representadas por (F.1-F.2), geradas pelo *AG*. O controlador ótimo é avaliado sob o ponto de vista de controle. Inicialmente, verifica-se que os

autovalores de malha fechada e as sensibilidades obtidas pelo controlador gerado pelo *AG* satisfazem, as restrições de projeto conforme a Tabela F.4, isto é, os autovalores estão na faixa especificada pelo projetista e satisfazem as sensibilidades normalizadas.

Tabela F.4: Autovalores, Sensibilidades e Ganhos do Controlador-Aeronave.

No	Autovalores	$s_i$	Ganhos
1	-22.9010	0.2434	0.1851 0.0061 -0.0325 -1.1791 -0.0106 1.5988
2	-28.6366	0.1380	0.0198 0.1943 -0.4226 -0.7849 -1.2051 2.6245
3	-1.1774 + 1.1743i	0.5244	
4	-1.1774 - 1.1743i	0.6109	
5	-2.0351	0.8599	
6	-0.2895	0.9132	

## F.2 Análise de Convergência

Os modelos do *AG*, Capítulo 4, representado pelas relações de recorrência Eqs. (4.8-4.24) são avaliados para a busca das matrizes de ponderação para o projeto de controladores *LQR* em sistemas dinâmicos multivariáveis. A abordagem de análise de convergência está direcionada para promover melhorias, no sentido de realizar uma busca confiável e rápida, (Wolff ,2006) e (Abreu ,2008b).

Após realizado o cálculo da autoestrutura de malha fechada, pontua-se cada indivíduo da seguinte forma: para cada autovalor localizado dentro da margem de autovalores especificada pelo projetista, atribui-se 1 ponto ao indivíduo  $QR_z$ , da mesma forma que 0.01 para cada autovetor associado que possua sensibilidade normalizada  $s_z < 1$ . O indivíduo de maior pontuação é considerado o melhor, e para regra de desempate o melhor é aquele que possuir a menor soma das sensibilidades normalizadas.

### F.2.1 População Inicial

A população inicial é gerada aleatoriamente com cinquenta (50) indivíduos cromossômicos,  $QR_z$ , Eqs. (4.11-4.12). Na geração aleatória das matrizes  $Q$  e  $R$ , a



população inicial é gerada por meio de uma sequência pseudo-aleatória distribuída uniformemente. A transformação das matrizes  $Q$  e  $R$  em matrizes simétricas, é um procedimento projetado para garantir as especificações de projeto *LQR* e a construção das matrizes  $QR$ , é um procedimento que agrupa o modelo genético das soluções ou indivíduos  $QR_z$ , (Abreu ,2007).

### F.2.2 *Fitness* da População Inicial

A Fig. F.1 ilustra o comportamento da população inicial por meio da soma das sensibilidades  $\sum p_i s_i(Q, R)$  para cada indivíduo. Cada indivíduo da população inicial é representado pelo argumento da estrutura de otimização Eq. (4.7), que são obtidos de acordo com as Eqs. (4.13-4.16) do modelo de *fitness*. Observa-se na Fig. F.1, que vários indivíduos obtiveram sensibilidade total normalizada menor ou igual a um. O melhor indivíduo, é aquele que apresenta menor sensibilidade total normalizada. O melhor indivíduo foi o 17.

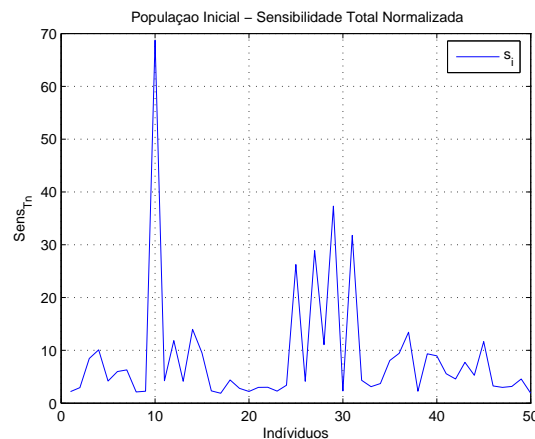


Figura F.1: Sensibilidade Total Normalizada.

A segunda etapa para análise do comportamento da população inicial consiste na avaliação do seu desempenho sob o ponto de vista da concentração de indivíduos. O método desenvolvido considera os efeitos da concentração global dos objetivos das sensibilidades e autovalores. O objetivo é atingir a melhor pontuação independente de qual  $s_i$  contribui para convergência do algoritmo.

Utiliza-se os momentos estatísticos no sentido de avaliar a formação de nichos que são aplicados na métrica *fitness* da população inicial, considerando um mapeamento da implementação dos autovalores  $\lambda_i$  e suas sensibilidades  $s_i$  do modelo de *fitness*, Eqs. (4.13-4.16). Após exaustivas análises para várias populações iniciais, (Wolff ,2006), tem-se informação para determinar as estatísticas da *fitness* da população inicial.

Na tabela F.5 são apresentados os indivíduos que alcançaram as restrições de sensibilidade. Neste caso, onze indivíduos satisfizeram as restrições de sensibilidades.

Tabela F.5: *Fitness* da População Inicial

Indivíduos	$Fitness_A$	$Fitness_S$	$Fitness_T$
3	5.000	0.060	5.060
7	4.000	0.060	4.060
8	4.000	0.060	4.060
9	4.000	0.060	4.060
14	5.000	0.060	5.060
15	5.000	0.060	5.060
17	5.000	0.060	5.060
25	5.000	0.060	5.060
30	4.000	0.060	4.060
36	5.000	0.060	5.060
42	4.000	0.060	4.060

A Fig. F.2 ilustra a função de *fitness* da população inicial, sendo considerados a *fitness* total, as *fitness* de autovalores e sensibilidades mapeadas. Considera-se a seguir alguns resultados. Observa-se que o *posto* do melhor indivíduo é 5.06, isto é, apenas um autovalor não alcançou o objetivo de modo que todas as sensibilidades foram satisfeitas. Como também, considerando uma população de cinquenta (50) indivíduos, percebeu-se que dez deles satisfizeram as restrições de sensibilidades. Desta amostra podemos destacar o indivíduo 17 como sendo o melhor por apresentar uma *fitness* de autovalores igual a 5.00 e além disso, apresenta

sensibilidade total normalizada de 1.880, sendo este o menor valor atingido por todos os indivíduos.

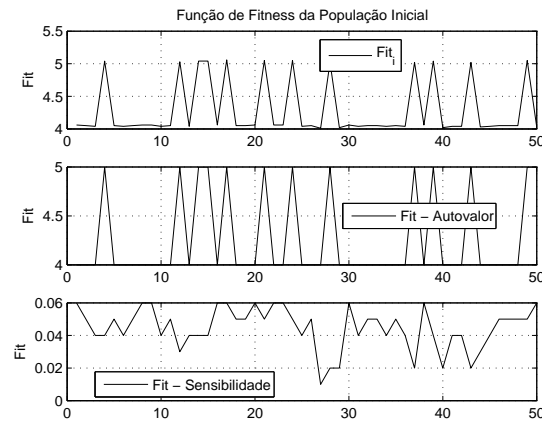


Figura F.2: Função de *Fitness* Mapeada - População Inicial - Sistema de Ordem 6.

Os histogramas da Fig. F.3 representam o agrupamento das *fitness* de autovalores, sensibilidades e *fitness* total respectivamente. Este agrupamento é construído em torno de um valor central. Observou-se que aproximadamente 72% dos indivíduos desta população possuem valor de *fitness* em torno de 4.05, ou seja, 36 indivíduos satisfazem a relação  $4 \leq Fit_{total} \leq 4.1$ . Existe uma concentração de nichos, ou seja, há pouca variedade do material genético, o que poderá levar o algoritmo à convergência prematura. Na faixa  $4.95 \leq Fit_{total} \leq 5.05$ , uma quantidade de 11 indivíduos possuem valor de *fitness* em torno de 5.02, que corresponde a um percentual de 22%.

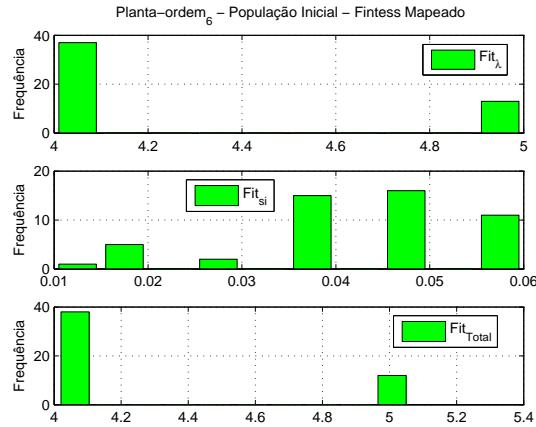


Figura F.3: Frequência da Função de *Fitness* - População Inicial - Sistema de ordem 6.

As frequências de ocorrência das sensibilidades e autovalores que atingiram as especificações do projeto podem ser obtidas por meio de observações do comportamento de *fitness* da população inicial. A sensibilidade  $s_5$  atingiu com maior frequência o objetivo, enquanto que a  $s_6$  com uma menor. O autovalor  $\lambda_1$  foi atingido com uma menor frequência. Pode-se obter mais informação a respeito da população inicial, verificando o conjunto de sensibilidades ( $s_1, s_2, s_3, s_4, s_5, s_6$ ) associadas com o conjunto (36, 35, 37, 41, 47, 28) de frequências que atingiram as especificações do projeto para um universo de cinquenta (50) indivíduos. Para os autovalores, as ocorrências ( $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ ) estão associados com o conjunto (6, 50, 50, 50, 7, 50).

### F.3 Processo de Busca

O algoritmo do processo de busca é constituído por três modelos que constituem o seu núcleo. Estes modelos são representados pelas Eqs. (4.9, 4.16, 4.19, 4.22 e 4.24) que estão associados com a estrutura genético artificial da população  $QR$ , avaliação dos cromossomos e as operações genéticas, respectivamente. As operações cromossômicas realizam as operações de seleção, cruzamento e mutação

que são representadas pelas Eqs. (4.19-4.24).

Na análise de convergência apresenta-se o comportamento do processo de busca em termos do *fitness* de uma população e pelos seus melhores indivíduos, para tal escolheu-se a população da geração 10. A estatística da *fitness* da população da geração 10 considera um mapeamento dos autovalores e suas sensibilidades nas variáveis escalares  $s_i$  e  $\lambda_i$ .

Durante o processo de busca, cada indivíduo recebe uma pontuação com valores escalares determinados na fase de pré-*fitness* ou de cálculos intermediários. As etapas do pré-*fitness* correspondem a determinação dos ganhos do controlador por meio de solução da equação algébrica de *Riccati*, montagem do sistema de malha fechada e computação dos autovalores e autovetores deste sistema. A função de *fitness* verifica as restrições, em seguida ordena os indivíduos de acordo com o grau de satisfabilidade das autoestruturas computadas em relação à autoestrutura especificada. Isto implica que esta ordenação estipula o grau de adequabilidade, de modo que este grau diz respeito à chance de sobrevivência de cada indivíduo no meio ambiente, sendo que a quantidade de indivíduos que são escolhidos para participar de uma população é limitada, visto que os indivíduos competem entre si durante as iterações do ciclo de busca.

Apresenta-se a estatística de *fitness* da população geração 10, que considera um mapeamento dos autovalores e suas sensibilidades nas variáveis escalares  $\lambda_i$  e  $s_i$ . Desenvolveu-se um algoritmo que nos forneceu a estatística da *fitness* da População da geração dez (10), sendo que foram realçados alguns resultados. Observou-se que o posto do melhor indivíduo é 6.04, isto é, duas sensibilidades não alcançaram o objetivo. Da população de cinquenta indivíduos, percebeu-se que dezesseis deles satisfizeram as restrições de sensibilidades. A tabela F.6 apresenta os indivíduos que alcançaram as restrições de sensibilidade. Neste caso, quatorze (14) indivíduos satisfizeram as restrições de sensibilidades.

Tabela F.6: *Fitness* da População Geração 10

<i>Individuos</i>	<i>Fitness<sub>S</sub></i>	<i>Fitness<sub>A</sub></i>	<i>Fitness<sub>T</sub></i>
5	0.060	5.000	5.060
6	0.060	5.000	5.060
8	0.060	5.000	5.060
18	0.060	5.000	5.060
19	0.060	3.000	3.060
26	0.060	5.000	5.060
27	0.060	5.000	5.060
28	0.060	5.000	5.060
35	0.060	5.000	5.060
38	0.060	5.000	5.060
40	0.060	5.000	5.060
42	0.060	5.000	5.060
47	0.060	5.000	5.060
50	0.060	5.000	5.060

A *fitness* total, as *fitness* de autovalor e sensibilidade são ilustradas na Fig. F.4 para cada indivíduo da população da geração dez (10). Podemos eleger o indivíduo trinta e cinco (35) como sendo o melhor, pois apresenta um maior desempenho, isto é, menor sensibilidade total normalizada e maior valor da função de *fitness* mapeada.

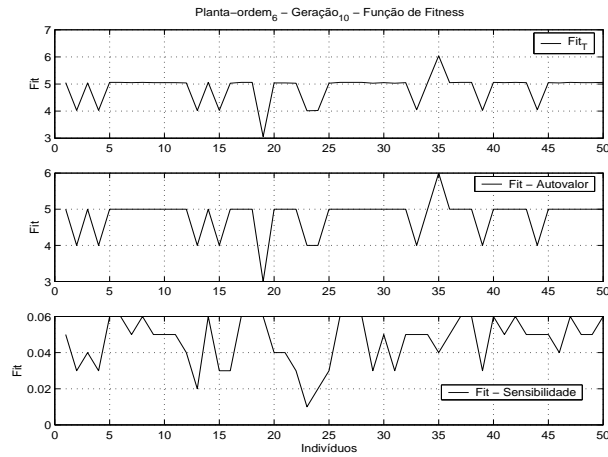


Figura F.4: Função de *Fitness* Mapeada - População Geração 10-Sistema de Ordem 6.

Os histogramas da Fig. F.5 representam o agrupamento das *fitness* de autovalores, sensibilidades e *fitness* total respectivamente, sendo que este agrupamento é construído em torno de um valor central. Observou-se que aproximadamente trinta e oito (38) indivíduos desta população possuem valor de *fitness* em torno de 5.02. Comparando este resultado com o da população inicial, existe uma realocação de concentração de nichos.

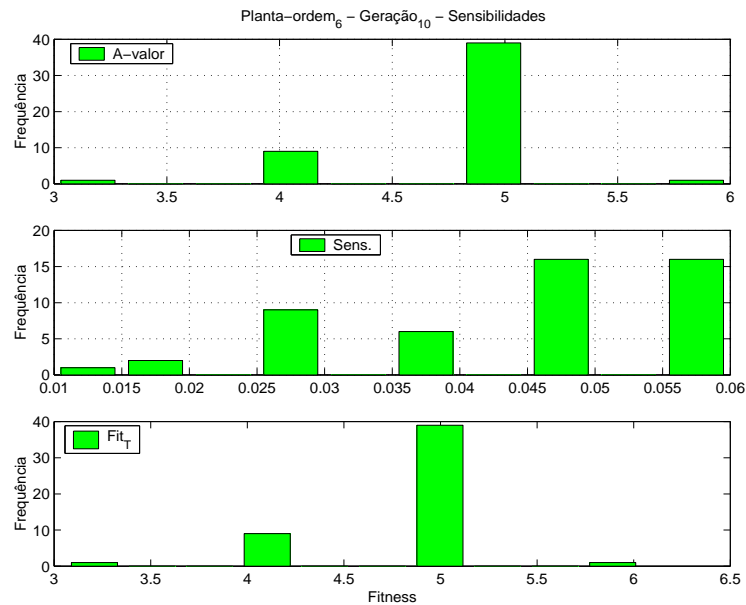


Figura F.5: Frequência da Função de *Fitness* - População Geração 10-Sistema de Ordem 6.

## F.4 População Final

A estatística da *fitness* da população final considera um mapeamento dos autovalores e suas sensibilidades nas variáveis escalares  $\lambda_i$  e  $s_i$ . Desenvolveu-se um algoritmo genético, (Abreu, 2008b) que nos forneceu a estatística da *fitness* da População Final, sendo que foram realçados alguns resultados. Observou-se que o *posto* do melhor indivíduo é 6.06; percebeu-se que dos cinquenta (50) indivíduos, sete (7) satisfizeram as restrições de sensibilidades, quatro (4) as restrições de autovalores e quatro (4) indivíduos satisfizeram as restrições de autovalores e sensibilidades, segundo a tabela F.7.



Tabela F.7: *Fitness* da População Final

<i>Individuos</i>	<i>Fitness<sub>A</sub></i>	<i>Fitness<sub>S</sub></i>	<i>Fitness<sub>T</sub></i>
1	5.000	0.060	5.060
3	6.000	0.060	6.060
4	6.000	0.060	6.060
5	6.000	0.060	6.060
8	6.000	0.060	6.060
11	5.000	0.060	5.060
19	5.000	0.060	5.060

*Fitness* total, as *fitness* de autovalor e sensibilidade são apresentadas na Fig. F.6 para cada indivíduo da população geração 20.

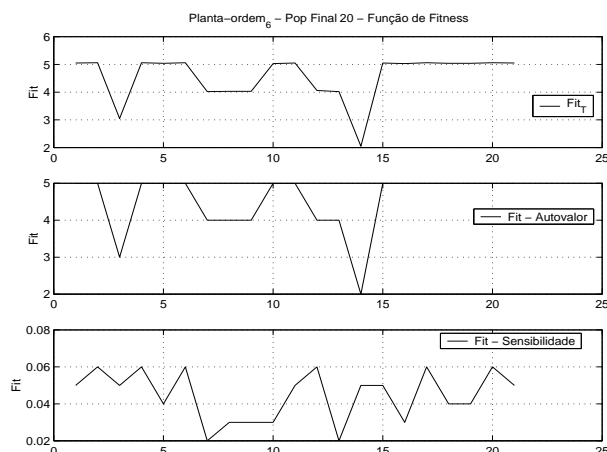


Figura F.6: Função de *Fitness* Mapeada - População Final-Sistema de Ordem 6.

Os histogramas da Fig. F.7 representam o agrupamento das *fitness* de autovalor e sensibilidade, *fitness* total, respectivamente, e a frequência dos indivíduos. Este agrupamento é construído em torno de um valor central.

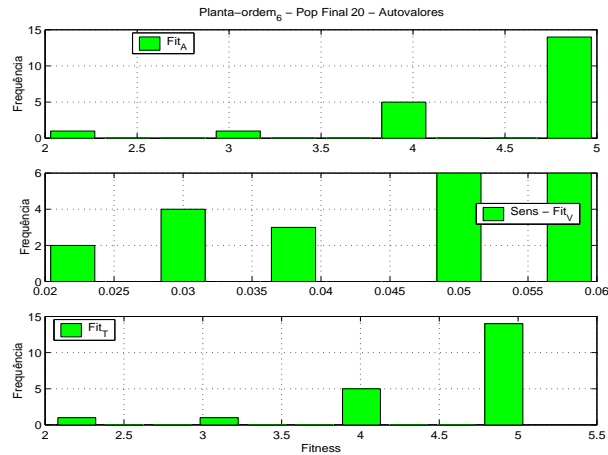


Figura F.7: Frequência da Função de *Fitness*-População Final-Sistema de Ordem 6.

## F.5 Resposta ao Impulso

Agora apresenta-se os resultados de desempenho do método de sintonia de ganhos de controladores *LQR*, para alocação da autoestrutura (AAE) sob o ponto de vista de controle. Os testes de desempenho da resposta ao impulso são ações tomadas para avaliar o desempenho do método de busca. Considera-se que as matrizes de ponderação são obtidas conforme os modelos genéticos do Capítulo 4 e os ganhos são obtidos com base na implementação destes modelos, veja (Abreu, 2007).

A resposta ao sinal impulso do sistema é uma forma para analisar, ao longo do tempo, o comportamento do sistema dinâmico para o controlador obtido pelo *AG*. O desempenho do controlador é verificado por meio de comparação com o do controlador base, (Teixeira Ferreira Março, 2008) e (Davis e Clarke, 1995), que é utilizado para fins de comparação segundo as Tabelas F.2 e F.3. As Figuras F.8 e F.10 ilustram a resposta ao impulso para os controladores base e o gerado pelo *AG* para suas saídas.

Em seguida, compara-se a resposta ao impulso de um controlador de referência (Davis e Clarke, 1995) com a do controlador produzido pelo *AG*, para analisar

o comportamento transitório das variáveis de saída. Baseado na análise destas figuras, constata-se na Fig. F.8, saída  $y_1$ , que os tempos  $t_s$  de acomodação são aproximadamente iguais, enquanto o controlador gerado pelo *AG* atingiu melhor desempenho para o instante  $t_p$  de pico em relação ao controlador base e o valor máximo  $M_p$  de ultrapassagem do controlador base é aproximadamente igual ao controlador gerado pelo *AG*, e entraram em regime permanente praticamente ao mesmo tempo.

Para a saída  $y_2$ , o controlador gerado pelo *AG* obteve desempenho um pouco melhor em relação ao controlador de referência, conforme as especificações do tempo de resposta  $t_p$ ,  $M_p$  e  $t_r$ , enquanto que  $t_s$  foram praticamente aproximados.

Já para a saída  $y_3$ , os valores  $t_p$  e  $t_r$  do controlador projetado pelo *AG* foram também um pouco menores em relação ao de referência, o valor da  $M_p$  do controlador *AG* é menor, ao passo que os  $t_s$  são aproximadamente iguais.

Finalmente, para a saída  $y_4$ , o controlador desenvolvido pelo *AG* obteve um desempenho relativamente melhor. As figuras de mérito dos tempos  $t_p$  e  $t_r$  são menores, o valor de  $M_p$  também é menor, enquanto que os tempos  $t_s$  são aproximadamente iguais. Considerando que o objetivo é obter um desempenho igual ou superior ao controle base, pode-se afirmar que o controlador determinado pela *AG* proposto, atingiu a alocação de autovalores e autovetores especificada pelo projetista.

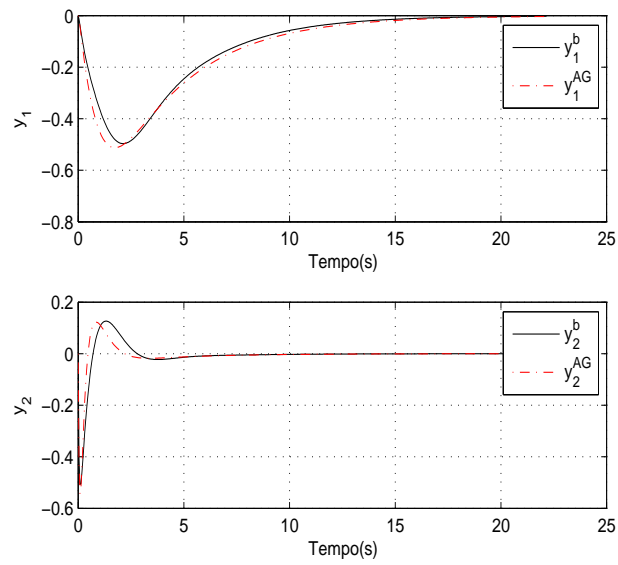


Figura F.8: Curvas de resposta ao impulso para as saídas 1 e 2.

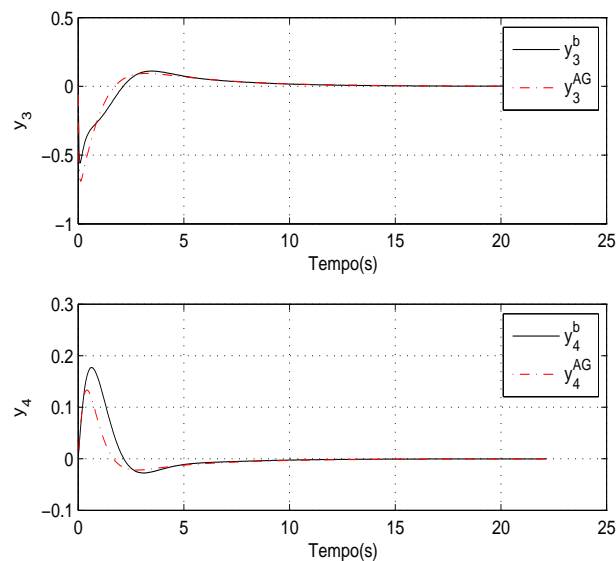


Figura F.9: Curvas de resposta ao impulso do controlador base e controlador usando *AG* para as saídas 3 e 4.

## F.6 Decomposição em Valores Singulares-*SVD*

Como o sistema em questão é um sistema MIMO, a análise via diagrama de *Bode* se torna inviável. Para contornar esse problema é feito a decomposição em valores singulares, de modo que possamos avaliar o comportamento do sistema no domínio da frequências. Os limites para garantir a robustez de sistemas MIMO, que são dados em termos do mínimo valor singular do ganho de malha que dever ser grande nas baixas frequências para garantir a robustez de desempenho, e o máximo valor singular de ganho de malha deve ser baixo nas altas frequências para garantir a robustez de estabilidade, (Lewis e Syrmos ,1995).

A Fig. F.10 exhibe os valores singulares do sistema teste com a abordagem neuronal. Percebe-se que em baixas frequências as condições de desempenho robusto são satisfeitas, visto que os mínimos valores singulares tiveram um ligeiro ganho quando comparado com o sistema base. Nas altas frequências, as condições de robustez de estabilidade também estão garantidas, pois os máximos

valores singulares com o controlador *LQR* tiveram uma atenuação em relação ao sistema não compensado. Portanto, o controlador projetado satisfaz as condições de desempenho e estabilidade no domínio de faixa da frequência estabelecida.

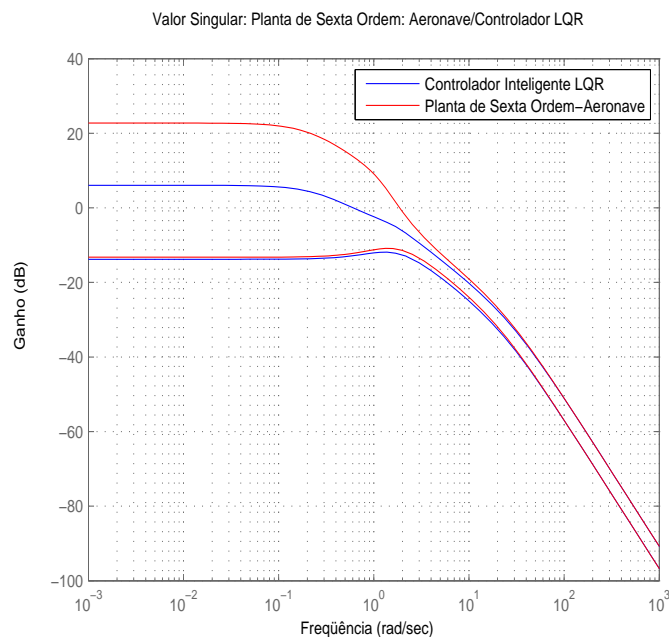


Figura F.10: Gráfico da magnitude do valor singular em função da frequência do diagrama de Bode para sistemas MIMO.

## F.7 Análise de Convergência da RNA

A seguir apresenta-se a análise de convergência e estabilidade da Rede Neural Artificial para a solução da equação algébrica de *Riccati*, sendo que analisa-se o comportamento numérico da solução. As matrizes de ponderação  $Q$  e  $R$  necessárias para a aplicação da RNA, foram escolhidas por um *AG* segundo (Neto, 2000).

A rede neuronal utilizada para solucionar a *EAR* é representada pelas Eqs. (4.33-4.36) para resolver a estrutura de otimização (4.31). Faz-se uma análise sobre a escolha dos parâmetros  $\eta_z$  e  $\eta_v$  que nos leva a uma melhor solução para

*EAR* e também nos mostra o quanto as variações desses parâmetros influenciam na busca da solução, (Abreu ,2006). A análise de convergência da solução neuronal da equação de *Riccati* está fundamentada em dois aspectos. O primeiro está direcionado a existência da solução e o segundo, na velocidade para se alcançar uma solução numericamente estável. Para a rede neural recorrente, temos alguns problemas relacionados ao ajuste dos dos parâmetros , sendo estes responsáveis pela velocidade de convergência da rede neuronal, qualidade da solução de *Riccati*. Deste modo, a sua escolha é de extrema importância para a sintonia da rede neuronal, sendo que a determinação dos mesmos é realizada por meio da avaliação do comportamento da RNA para uma faixa de variação de dois parâmetros.

Para a determinação dos parâmetros de convergência da RNR, para resolução do problema de otimização associado a *EAR* e fatoração de *Cholesky*, varia-se  $\eta_z$  e  $\eta_v$  de maneira que possamos verificar a influência dos mesmos em relação em relação a busca do mínimo da função energia e por conseguinte a solução de *Riccati*. Assim, no experimento podemos verificar o quanto esses parâmetros influenciam no comportamento da solução, sendo que apresenta-se uma variação para os mesmos em relação a norma do infinito da energia e da solução da *EAR*.

A análise de convergência é feita baseando-se principalmente na avaliação dos resultados a partir das variações dos parâmetros  $\eta_z$  e  $\eta_v$ . Estamos interessados em escolher esses parâmetros para obtermos a melhor solução, bem como a escolha das condições iniciais.

O papel das constantes  $(\eta_z, \eta_v)$  diz respeito às taxas de convergência entre as camadas ocultas e de saída do sistema. Quanto maior o valor dos parâmetros  $(\eta_z, \eta_v)$ , mais rápida será a nossa taxa de convergência, visto que tem-se uma rede neuronal baseada no método gradiente. Logo a convergência da rede neural varia para cada escolha dos parâmetros  $(\eta_z, \eta_v)$ . Além disso, afim de assegurar que o estado permanente de  $P(t)$  seja definida positiva, é necessário que o cálculo da camada oculta  $L$  convirja mais rápido que a camada de saída  $P$ , o que pode ser solucionado, escolhendo-se  $\eta_z \geq \eta_v$  suficientemente grandes.

## F.8 Análise de Estabilidade da RNA

A rede neural apresentada é utilizada para representar um sistema dinâmico de tempo contínuo e não-linear. A estabilidade dos sistemas dinâmicos não lineares está intrinsicamente relacionado com o Método Direto de *Lyapunov*.

Supondo que a função de ativação seja contínua, pelo menos diferenciável por partes e monotonicamente decrescente,

$$\begin{aligned} \frac{df_{ij}(\xi)}{d\xi} &\geq 0 & -\infty < \xi < +\infty \\ f_{ij}(\xi) &= 0, & \text{se } \xi = 0. \end{aligned}$$

A rede neural será assintoticamente estável se

$$\begin{aligned} \forall P(0), L(0) &\rightarrow \exists \bar{L}, \bar{P} \\ \text{de modo que,} \\ \bar{L} &= \lim_{t \rightarrow \infty} L(t) \quad \text{e} \quad \bar{P} = \lim_{t \rightarrow \infty} P(t) \end{aligned}$$

A análise de estabilidade indica que a transição do estado de ativação forma um fluxo de estado gradiente que minimiza a função energia  $E$  convexa e limitada, (Wuang e Wu ,1998) e (Abreu ,2006).

## F.9 Simulação Computacional da RNA

Para a simulação computacional, utiliza-se o sistema dinâmico aeronave de 6ª ordem. Apresenta-se os passos necessários para aplicação da abordagem apresentada referente à RNA. Formula-se a inicialização com as camadas de entrada, ocultas e saída; em seguida a integração de uma camada oculta relacionada com a dinâmica da fatoração de *Cholesky*, Eq. (4.38) e da solução da EAR, Eq. (4.37). Para integração usa-se o método de *Runge-Kutta* de quarta ordem sendo que têm-se como retorno as matrizes referentes a camada de saída representando a solução  $P$  da equação algébrica de *Riccati*, Eq. (3.5) e ao Fator de *Cholesky*, Eq. (4.30). O número de amostras  $N_{sample} = 1600$  e tempo de processamento  $T_p = 0.0003$



segundo. Uma vez que a solução retornada pela RNR é uma matriz, deve-se especificar uma métrica, pois assim pode-se comparar as várias soluções que o algoritmo nos retorna a cada iteração. Para as soluções apresentadas dentre os vários valores dos parâmetros  $\eta_z$  e  $\eta_v$ , utiliza-se a *Norma do Máximo*, aplicado duas vezes na matriz  $P$  retornada pela rede neuronal. As figuras apresentadas na subseção a seguir mostram o comportamento da solução obtida pela rede neuronal para as variações dos parâmetros  $\eta_z$  e  $\eta_v$ . Tem-se como condições iniciais  $f_{ij}(\xi) = \xi$ , e as matrizes  $Y$ ,  $L$  e  $U$  iguais à matriz Identidade.

## F.10 Variações nos Parâmetros $\eta_z$ e $\eta_v$

Com o aumento do número de variáveis e da ordem do sistema, maior a complexidade e mais difícil torna-se a análise tanto do comportamento do algoritmo quanto na compreensão da dinâmica do sistema, que está direta e indiretamente associada às variáveis responsáveis pelo ajuste do algoritmo. A variação dos parâmetros  $\eta_z$  e  $\eta_v$  é apresentada como um das responsáveis no desempenho e na exatidão do funcionamento da rede neuronal.

A Fig. F.11 mostra o quanto uma matriz simétrica e definida positiva para a equação de *Riccati* pode ser satisfatória. Pode-se perguntar: Qual matriz  $P$  associada a estrutura de otimização irrestrita, resolve a EAR e como avaliar seus elementos? Para isso faz-se uso de uma métrica, a norma do máximo, que avalia a maior soma das colunas da matriz  $P$  para cada par de parâmetros  $(\eta_v, \eta_z)$ . Conforme pode-se observar, para cada variação nesses parâmetros, a RNR gera uma nova solução para a equação de *Riccati*. Observa-se que a solução pode permanecer em três regiões distintas que são estabelecidas pela norma do erro como função da faixa de variação dos parâmetros  $(\eta_v, \eta_z)$ . Na faixa de  $0 \leq \eta_v \leq 400$  e  $500 \leq \eta_z \leq 2500$  a norma do máximo é mínima. Para faixa de valores,  $400 \leq \eta_v \leq 550$ , ocorreu um crescimento na taxa de variação da norma do máximo, acarretando desta maneira uma solução inadequada para a RNR, enquanto que para faixa de valores de  $\eta_v > 550$  a inclinação aumenta, significando que para essa faixa a energia aumenta bruscamente. Pode-se dizer que à medida que a taxa de variação aumenta, a RNR fornece uma solução, porém não é a ótima.

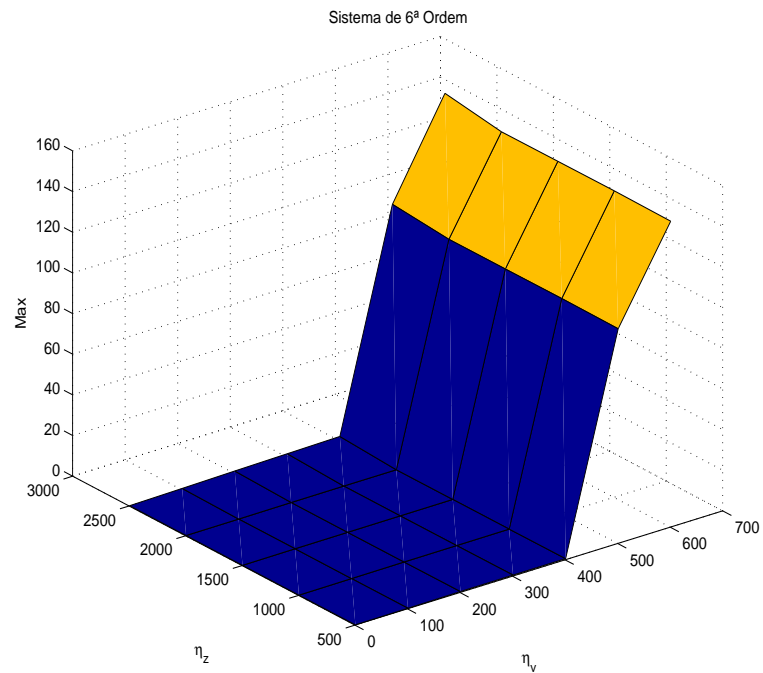


Figura F.11:  $n_{\max}$  da solução  $P$  para a variação  $\eta_v$ : 1-600 e  $\eta_z$ :500-2500 para um sistema de 6ª ordem.

A Fig. F.12 mostra a melhor faixa para os parâmetros de ajuste da norma do máximo da solução simétrica e definida positiva para a Eq.(4.31). Para a primeira, para  $900 \leq \eta_z \leq 2500$  a taxa de variação é ligeiramente positiva, para  $600 \leq \eta_z \leq 900$  a inclinação é bem maior, enquanto que para faixa de valores de  $\eta_v > 550$  a inclinação aumenta, significando que para essa faixa a energia aumenta bruscamente. Faz-se uma análise similar para a Fig. F.13

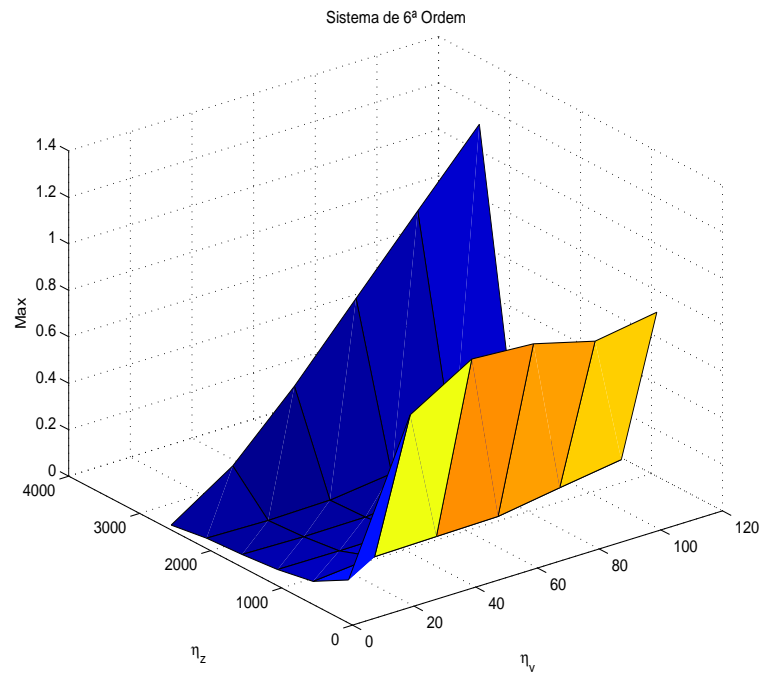


Figura F.12:  $n_{\max}$  da solução  $P$  para a variação  $\eta_v$ : 1-120 e  $\eta_z$ :500-2500 para um sistema de 6ª ordem.

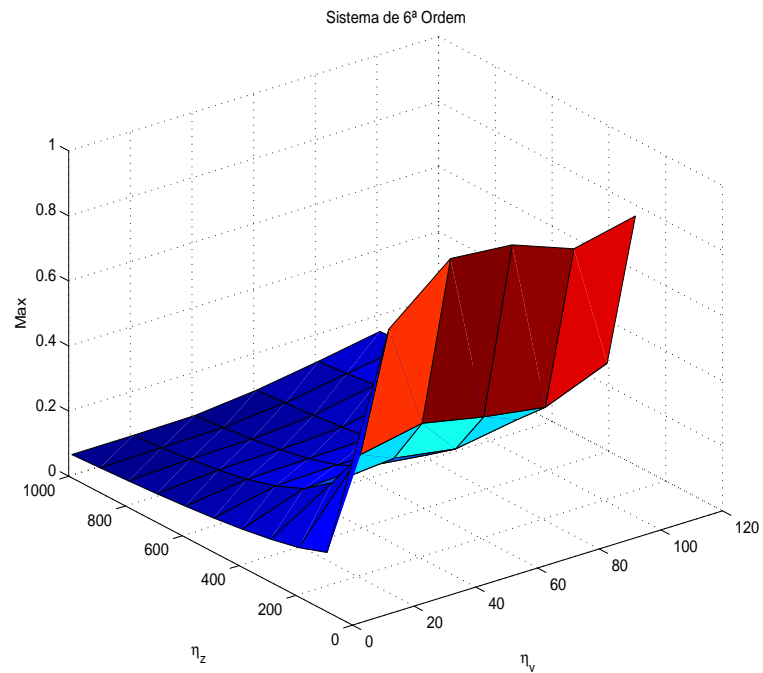


Figura F.13:  $n_{\max}$  da solução  $P$  para a variação  $\eta_v$ : 1-120 e  $\eta_z$ :500-2500 para um sistema de 6ª ordem.

A Fig. F.14 mostra a superfície da função energia da estrutura de otimização irrestrita, sendo que esta é gerada por pontos da função que são obtidos para cada par de parâmetros  $(\eta_v, \eta_z)$ . Os parâmetros de ajustes que minimizam a função energia são direcionados para a RNR que exibem a melhor solução  $(P, L)$  para o problema. Nota-se que para a variação desses parâmetros nessa faixa, o problema de otimização irrestrito é satisfatório.

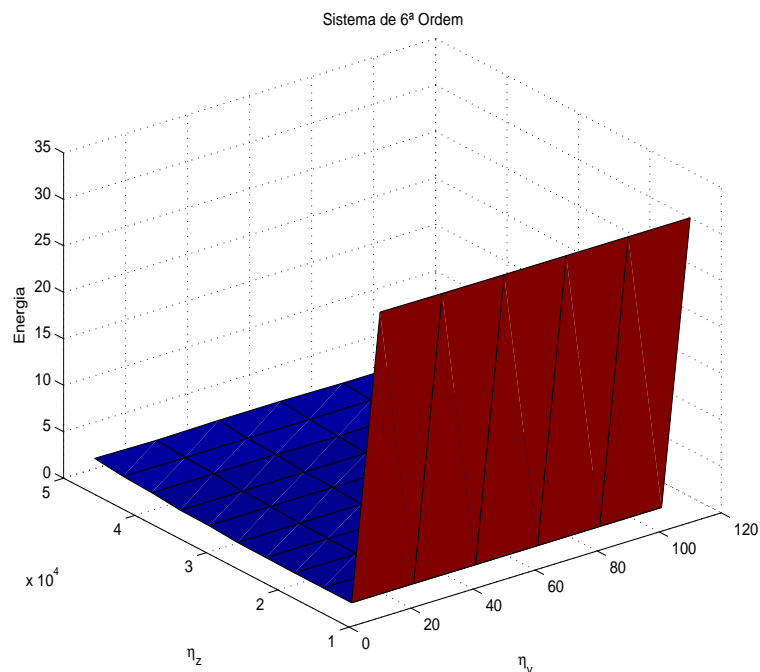


Figura F.14: Superfície da função energia para a variação  $\eta_v$  : 10-110 e  $\eta_z$ : 14000-32000 para aeronave.

## F.11 Discussão do Modelo de Fusão

Em uma primeira instância, a convergência do algoritmo genético e da rede neural recorrente foi avaliada de maneira desacoplada, como pode ser notado nas últimas subseções. A convergência do AG apresentou um bom desempenho para impor a especificada autoestrutura, porém a velocidade de convergência pode ser melhorada pelo seu paralelismo intrínseco. (Neto ,2000) e (M. Alba ,2002), realizam a seleção das matrizes de ponderação em tempo real. Do mesmo modo, a RNR pode ser melhorada pelo paralelismo natural e a implementação em tempo real foi desenvolvida por (Rocco Furferi e Daou ,2007) e outros.

O desempenho desacoplado satisfatório do modelo de fusão proposto, conduz a uma discussão sobre algumas melhorias para acelerar a convergência em atender as requisições em tempo real, em que o motor da estratégia de controle deve exe-

cutar ações e processar informações em uma magnitude de tempo menor que as reações da planta controlada, (Mehdi Amirijoo e Son ,2008) e (Mehdi Amirijoo e Gunnarsson ,2007), ou seja, mudanças operacionais ou perturbações. Os desenvolvimentos adicionais da metodologia vão na direção de busca das matrizes *QR* e a solução da EAR via RNR em atender os prazos limites dos sistemas dinâmicos.

## F.12 Conclusão

Apresentou-se uma metodologia para realizar a sintonia e síntese de sistemas de controle ótimo que tem por base AG e RNR. Uma planta de quarta ordem e duas de sexta ordem foram usadas para apresentar o controle inteligente *LQR* neuro-genético em sistemas dinâmicos multivariáveis. O procedimento proposto, mostrou que soluções computacionais exibidas por modelos computacionais podem ser uma opção próspera para o projeto e controle de sistemas.

# Referências Bibliográficas

(n.d. *a*). PhD thesis.

(n.d. *b*). PhD thesis.

Abreu, Ivanildo Silva; Neto, João Viana da Fonseca; e Silva Fábio Nogueira. (,2008*a*). Neuro-genetic state space controllers synthesis based on linear quadratic regulator design for eigenstructure assignment. *Transactions on Systems, Man, and Cybernetics, Part B*.

Abreu, Ivanildo Silva; Neto, João Viana da Fonseca; Rêgo Patrícia Helena Moraes; Wolff Marlon Paulo de Melo; Silva Orlando Fonseca; (,2008*b*). Modelos e Convergência de um Algoritmo Genético para Alocação de Autoestrutura via LQR. *Revista IEEE Latina Americana*.

Abreu, Ivanildo Silva; Neto, João Viana da Fonseca; Silva Fábio Nogueira da; Teles José Carlos; Pinto Vandilberto Pereira (,2007). Controle LQR Inteligente para Geradores Eólicos DFIG. *Simpósio Brasileiro de Automação inteligente, SBAI*.

Abreu, Ivanildo Silva; Silva, Fábio Nogueira da; Neto João Viana da Fonseca; Barros Allan K. e Silva Orlando Fonseca (,2006). Solução da Equação Algébrica de Riccati via Rede Neural Recorrente para Projeto do Regulador Linear Quadrático. *Congresso Brasileiro de Automática, Salvador-Ba, Brasil*.

Arnold, W. F. e A. J. Laub (,1984). Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations. *Proceedings of the IEEE*.

- Athans, Michael e L. Peter Falb (,1966). *Optimal Control - An Introduction to the Theory and Its Applications*. McGRAW-Hill Book Company . United States of America.
- Athans, Michael e L. Peter Falb (,2007). *OPTIMAL CONTROL- An Introduction to the Theory and Its Applications*. McGRAW-Hill Book Company. United States of America.
- Bazaraa, Mokhtar S., Hanif D. Sherali e C. M. Shetty (,1993). *Nonlinear Programming: Theory and Algorithms*. second ed.. John Wiley and Sons, Inc.. Canada.
- Bäck, T., Fogel D. B. e Z. Michalewicz (,1997). *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press.
- Bottura, C. P. e 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.
- Bryson, Arthur E. e Yu-Chi Ho Jr (,1975). *Applied Optimal Control*. Oxford-Press . Stanford University, California.
- Bryson, Arthur E., Ho Yu-Chi (,1969). *Applied optimal control: optimization, estimation and control*. Waltham, Mass. Ginn.
- Bunse-Gerstner A., Fassbender H. (,1997). A Jacobi-like method for solving algebraic equations on parallel computers. *IEEE Transactions on Automatic Control* **42**, 1071–1084.
- C. Duval, G. Clercb e Y. Le Gorrecc (,2006). Induction machine control using robust eigenstructure assignment. *Control Engineering Practice* **14**, 29–43.
- Chen, C. T (,1999). *Linear System Theory and Design*. Saunders College Publishing. Oxford University Press.
- Chipperfield, Andrew e Fleming Peter (,1996). Evolutionary Algorithms for Control Engineering. *Proceedings 13th Triennial IFAC World Congress, San Francisco-USA* **1**, 181–186.



- Davis, L. (,1971). *Handbook of genetic Algorithms*. Van Nostrand Reinhold. New York.
- Davis, R. e T. Clarke (,1995). Parallel Implementation of a Genetic Algorithm. *Control Eng. Practice* **3**(1), 11–19.
- Dazzo, John Joachim, Houpis Constantine H. (,1984). *Análise e projeto de sistemas de controladores lineares*. Guanabara. Rio de Janeiro.
- Doyle, J. C. e G. Stein (,1979). Robustness with Observers. *IEEE Transactions on Automatic Control* **24**(4), 607–611.
- Doyle, J. C. e G. Stein (,1981). Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis. *IEEE Transactions on Automatic Control* **26**, 4–16.
- Feng, G. e R. Lozano (,1999). *Adaptive Control Systems*. Newnes.
- Ferreira, Carlos César Teixeira e João Viana da Fonseca Neto (,2003). Projeto do Filtro de Kalman via Alocação de Autoestrutura e Algoritmo Genético. *VI Simpósio Brasileiro de Automação Inteligente*.
- Ferreira, Carlos César Teixeira; Neto, João Viana da Fonseca; A. Fernando Torrico. (,2003). Alocação de Autoestrutura via Controle LQG/LTR e Computação Evolutiva. *VI Simpósio Brasileiro de Automação Inteligente*.
- Fletcher, L.R. (,1981a). On Pole Assignment in Linear Systems with direct feedthrough - I: theoretical considerations . *International Journal of Control* **33**, 739–749.
- Fletcher, L.R. (,1981b). On Pole Assignment in Linear Systems with direct feedthrough - II: computational considerations . *International Journal of Control* **33**, 1147–1154.
- 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.

- Fraile-Ardanuy, Jesús e P.J. Zufiria (,2007). Design and comparison of adaptive power system stabilizers based on neural fuzzy networks and genetic algorithms. *Neurocomputing* **70**, 2902–2912.
- Garcia, José Paulo F;Bortolo Edson S.; Ribeiro, Jean Marcos S.; e Garcia Lizete Maria C. F. (,2004). Active Noise attenuation Using LQG/LTR Control. *Control Systems Magazine, IEEE* pp. 1 – 4.
- Gardiner, J. D. e Alan Laub (,1991*a*). Parallel algorithms for algebraic riccati equations. *Int. J. Control* **54**(6), 1317–1333.
- Gardiner, Judith D. e Alan Laub (,1991*b*). Parallel alghorithms for algebraic riccati equations. *Int. J. Control* **54**(6), 115–131.
- Gasem, C. Z., J. W. E. M. K. (,1995). Application of enhanced LQG/LTR for distillation control. *Control Systems Magazine, IEEE* **15**(4), 56–63.
- Goldberg, David Edward (,1985). Dynamic System Control using Rule Learnig and Genetic Algorithms. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* pp. 588–592.
- Goldberg, Jack L. (,1991). *Matrix theory with applications* . McGraw-Hill . Uiversity of Michigan.
- Golub, Gene H., Loan Charles F. Van (first edition 1983, second edition 1989 and third edition 1996). *Matrix Computations*. Johns Hopkins. .
- Graupe, D. (,1972). Derivation of weighting matrices towards satisfying eigenvalue requirements . *Int. J. Control* **16**(5), 881–888.
- H. Ko, K. Y. Lee e H. Kim (,2004). An intelligent based LQR controller design to power system stabilization. *Electric Power System Research-Elsevier* **71**, 1–9.
- Hansen, A. D., Jauch C. Sorensen P. Iov F. Blaabjerg (,2003). *Dynamic wind turbine models in power system simulation tool*. Riso national Laboratory.
- Hansen, A. Helle L. Blaabjerg (,2001). Conceptual survey of generators and power eletrronics for wind turbines. *Riso national Laboratory* **8**, 5–91.

- Harvey, Charles A. e Gunter Stein (,1978). Quadratic Weights for asymptotic Regulator. *IEEE Transactions on Automatic Control* **23**(3), 378–387.
- Ilke Turkmena, Kerim Guneyb e Dervis Karabogac (,2006). Genetic tracker with neural network for single and multiple target tracking. *wwwelsevier.com/locate/neucom* (69), 2309–2319.
- Jain, L. C. e N. M. Martin (,1999). *Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms Industrial Applications*. New York CRC Press. New York.
- Jih-Gau Juanga, Hou-Kai Chioub e Li-Hsiang Chiena (,2008). Analysis and comparison of aircraft landing control using recurrent neural networks and genetic algorithms approaches. *wwwelsevier.com/locate/neucom*.
- Johnson, M. A. e M. J. Grimble (,1987). Recent Trends in Linear Optimal Quadratic Multivariable Control Systems Design. *IEE-Review* **134**, 53–71.
- K. Zhou, J. C. Doyle e K. Glover (,1996). *Robust and Optimal Control*. Upper Addle River-NJ-USA: Prentice Hall. USA.
- Kailath, Thomas (,1980). *Linear Systems*. first ed.. Prentice-Hall Inc.. Englewood Cliffs, N.J.,USA.
- Kalman, R. (,1960a). On the General Theory of Control Systems. *Automatic Control, IRE Transactions on* **4**, 110–110. Butter Worth, London.
- Kalman, R. E. (,1960b). Contribution to the Theory of Optimal Control. *Bol* **5**, 102–119.
- Kalman, R.E; Falb, Peter; Arbib M.A (,1969). *Topics in Mathematical System Theory*. McGraw-Hill. New York, USA.
- Kawasaki, Naoya e Etsujíro Shimemura (,1983). Determining Quadratic Weighing matrices to Locate Poles in a Specified Region. *Automática* **19**(5), 555–560.
- Kaynak, O. (,2001). The fusion of computationally intelligent methodologies and sliding modes control a survey. *IEEE Transactions on Industrial Electronics* **48**(1), 4–17.

- King, R. E. (,1999). *Computational Intelligence in Control Engineering*. Marcel Dekker. New York.
- Kirk, Donald E. (,1970). *Optimal Control Theory - An Introduction*. Prentice-Hall. New Jersey.
- Kleinman, D.L. (,1968). On a iterative technique for riccati equation computations. *IEEE Transactions on Automatic Control* **13**, 114–115.
- Ksouri, A. Zaafour; A. Kochbati; M. (,2002). Lqg/ltr control of a direct current motor. *Systems, Man and Cybernetics, IEEE International Conference on* **Volume: 5**, Pages:5 pp. vol.5.
- Kwakernaak, H. Bosgra, O. H. (,2000). *Design Methods for Control systems: Notes for a course of the Dutch institute of Systems and Control Winter term. . .*
- Kwakernaak, H. e R. Sivan (,1972). *Linear Optimal Control Systems*. Wiley-Interscience. New York.
- Lancaster, P. e L. Rodman (,1995). *Algebraic Riccati Equations*. Clarendon Press-Oxford. New York-uSA.
- Laub, Alan J. (,1972). Canonical forms for  $\sigma$ -symplectic matrices. PhD thesis. M. S. Thesis, School of Math., University of Minnesota.
- Laub, Alan J. (,1979). A Schur Method for Solving Algebraic Riccati Equations. *IEEE Transactions on Automatic Control* **24**(6), 913–921.
- Lee, Myoen-Song Choi; Jeong-Woo Lee; K. Y. (,1996). An Auxiliar LQG/LTR robust controller design for cogeneration plants young-moon park. *Energy Conversion, IEEE Transactions on.* **1**, 407–413.
- Lewis, Frank L. e Vassilis L. Syrmos (,1995). *Optimal Control*. John Wiley and Sons, Inc.. USA.
- Liu, G.P. e R.J Patton (,1998). *Eigenstructure Assignment for Control System Design*. John Willey & Sons.
- Luenberger, David G. (,1984). *Linear and Nonlinear Programming*. Addison-Wesley Publ. Co.. USA.

- M. Alba, E., Tomassini (,2002). Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* **6**(5), 1026–1032.
- M. Nasser, K. Asghari e M. J. Abedini (,2008). Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network. *Expert Systems with Applications* **35**(1), 1415–1421.
- Marques, J. Pinheiro, H. Gründling H. A Pinheiro J. R. Hey H. L. (,2003). A survey on variable-speed wind turbine system. *Congresso Brasileiro de Eletrônica de Potência (COBEP) Fortaleza CE*.
- Medanic, J., H. S. Tharp e W. R. Perkins (,1988). Pole Placement by Performance Criterion Modification. *IEEE Transactions on Automatic Control* **33**(5), 469–472.
- Mehdi Amirijoo, Jörgen Hansson, Sang H. Son e Svante Gunnarsson (,2007). Experimental evaluation of linear time-invariant: Models for feedback performance control ,issn:0922-6443. in *Real-Time Systems-Springer-Verlag* **35**(3), 209–238.
- Mehdi Amirijoo, Jörgen Hansson, Svante Gunnarsson e Sang H. Son (,2008). Quantifying and suppressing the measurement disturbance in feedback controlled real-time systems. *Real-Time Systems-Publisher Springer Netherlands, ISSN:0922-6443* **40**(2), 44–76.
- Meystel, A. (,1985). Intelligent control: Issues and perspectives. *IEEE Workshop on Intelligent Control*.
- M.Maciejowski, J. (,1989). *Multivariable Feedback Design*. Addison - Wiley.
- Moore, B.C. (,1976). On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment. *IEEE Transaction on Automatic Control* **21**(6), 689–692.
- Murnaghan, F. D., Winter A. (,1991). A canonical form for real matrices under orthogonal transformations . *Proc. Nat. Acad. Sci.* **17**, 417–420.
- Neto, João Viana da Fonseca (,2000). Alocação Computacional Inteligente de Autoestruturas para Controle Multivariável. PhD thesis. UNICAMP.

- Neto, João Viana da Fonseca; Abreu, Ivanildo Silva; Silva Fábio Nogueira; Filho; Joaquim Gomes e Patrícia Helena Moraes. Rêgo (,2008). Hierarchical model of Genetic Algorithms for LQR Loop Transfer Recovery in LQG Controllers. *Asian Journal of Control*.
- Nichols, N.K. e P. Van Dooren (,1984). Robust Pole Assignment and Optimal Stability Margins. *Electronic Letters* **20**, 660–61.
- Nunes, M.A (,2003). Avaliação do comportamento de aerogeradores de velocidade fixa e variável integrados em redes elétricas fracas. *Tese de Doutorado Pós Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina*.
- 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* **3**, 3278–3282.
- Passino, Kevin M. (,2004). *Biomimicry for Optimization, Control, and Automation*. first ed.. Springer.
- Pedrycz, W. (,2007). Genetic tolerance fuzzy neural network: From data to fuzzy hyperboxes. *Neurocomputing* **70**(6), 1403–1413.
- Pedrycz, Witold (,1998). *Computational Intelligence Introduction*. CRC Press, Boca Raton, New York. Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Canada.
- Petkov, P. H., Christov N. D. e M. M. Konstantinov (,1991). *Computational Methods for Linear control Systems*. UK: Prentice Hall International.
- Ping Zhang, Brijesh Verma e Kuldeep Kumar (,2004). A Neural-genetic Algorithm for Feature Selection and Breast Abnormality Classification in Digital Mammography. *IEEE*.
- Pinheiro, E. L. R. (,2004). Análise do Comportamento Dinâmico de Usinas Eólicas de Velocidade variável Utilizando ATP Draw. PhD thesis. Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal de Minas Gerais.

- Pinto, Vandilberto Pereira (Março, 2007). *Modelagem e Simulação de uma Planta Eólica Controlada por um Regulador Linear Quadrático Conectada ao Sistema de Distribuição de Energia Elétrica*. Dissertação de Mestrado, Universidade federal do Ceará.
- Porter, B. (,1969a). Assignment of closed-loop eigenvalues by the direct method of Lyapunov. *International Journal of Control* **10**(2), 153–157.
- Porter, B. (,1969b). Eigenvalue sensitivity of Modal control to Loop gain Variations. *International Journal of Control* **10**(2), 159–162.
- Preeti Bajaj, Prashant Sharma, Amol Deshmukh (,2007). Vehicle Classification for Loop Detector with Neural Genetic Controller: A Design Approach. *Intelligent Transportation Systems Conference*.
- 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* **1**(50), 211–218.
- Rawlins, S. J. Louis (,1993). *Predicting Convergence Time for Genetic Algorithms*. IEEE Transactions on Neural Networks and Cybernetics.
- Rêgo, Patrícia Helena Moraes (,2007). Convergência de Algoritmo Genético Hierárquico para Recuperação da Malha LQR por Controladores LQG/LTR. *Dissertação de Mestrado, Universidade Federal do Maranhão, UFMA*.
- Rissanen, J. (,1960). Control Systems Synthesis by Analogue Computer Based on the Generalized Linear Feedback Concept. In: *Proceedings of the Symposium on Analog Computation Applied to the Study of Chemical Processes. International Seminar. Brussels, Belgium*. Vol. 1. pp. 1–13.
- Rocco Furferi, M. e M. Daou (,2007). Artificial neural network software for real-time estimation of olive oil qualitative parameters during continuous extraction. *Publisher Elsevier Science* **55**(2), 115–131.
- Rudolph, Günter (,1994a). Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks, and Cybernetics*.

- Rudolph, Gunter (,1994*b*). Convergence Analysis of Canonical Genetics Algorithms. *IEEE Transactions on Neural Networks, and Cybernetics*.
- Sandel, J. (,1970). On Newton's method for Riccati equation solutions. *IEEE Transactions on Automatic Control* **19**, 870–876.
- Sarangapani, J. (,2006). *Neural Network Control of Nonlinear Discrete-time Systems*. CRC Press. New York.
- Silva, Fábio Nogueira (Junho, 2008). *Métodos neuronais para a solução e o LQR*. Dissertação de Mestrado, Universidade Federal do Maranhão, UFMA. .
- Sobel, K. M. e E. Y. Shapiro (,1985). Eigenstructure Assignment: A Tutorial Part I Theory. In: *Proceedings of American Control Conference 5<sup>th</sup>* . Vol. 1. Saint-Nazaire, USA. pp. 456–460.
- Stewart, G. W. (,1979). *Introduction to Matrix Computations*. New York: Academic. .
- Suzuki, Joe (,1995). A Markov Chain Analysis on Simple Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*.
- Teixeira Ferreira, Carlos Cesar (Março, 2008). *Alocação de auto-estrutura Utilizando Controle Robusto LQG/LTR e Computação Evolutiva*. Dissertação de Mestrado, Universidade Federal do Maranhão, UFMA. São Luís-MA.
- V. R. Adineh, C. Aghanajafi, G. H. Dehghan e S. Jelvani (,2008). Optimization of the operational parameters in a fast axial flow CW  $CO_2$  laser using artificial neural networks and genetic algorithms. *journal homepage:wwwelsevier.com/locate/optlastec* **40**, 1000–1007.
- Valdivieso, P.A.; Merelo, J.J.; Prieto A.; Rojas I.; Romero G. (,2002). Statistical Analysis of the Parameters of a Neuro-Genetic Algorithm. *IEEE Transactions on Neural Networks* **13**(6), 1374–1394.
- Wang, J. e Wu Guang. (,1993). Recurrent neural Nnetwork for LU decomposition and Cholesky factorization. *Mathematical and Computer modeling* **18**, 1–8.



- Wang, Li Xin. e Mendel Jerry (,1992). Parallel Structured networks for solving a wide variety of matrix algebra problems. *Journal of Parallel and Distributed Computing* **14**, 1337–1345.
- Weerasooriya, S. Phan, D. (,1995). Discrete-time LQG/LTR design and modeling of a disk drive actuator tracking servo system. *Industrial Electronics, IEEE Transactions.* **42**, 214–247.
- 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.
- White, D. A. e D. A. Sofge (,1992). *Handbook of intelligent Control Neural, Fuzzy, and Adaptive Approach.* Van Nostrand Reinhold. New York.
- White, David A. (,1992). *Optimal ontrol: A Foundation for Intelligent Control.* Van Nostrand Reinhold: New York. .
- Wolff, Marlon Paulo Melo (,2006). *Análise de Convergência de Algoritmo Genético Dedicado a Alocação de Autoestrutura via Controle Ótimo.* Dissertação de Mestrado, Universidade Federal do Maranhão, UFMA. .
- Wonham, M. (,1986). Linear Multivariable Control: A geometric approach . *Springer Verlag, Berlin-Germany* **19**(6), 870–876.
- Wonham, W. M. (,1967). On Pole Assignment in Multi-Input Controllable Linear System. *IEEE Transaction on Automatic Control* **12**(6), 660–665.
- Wu Yongjian, Tan Minghao, Zhou Xiaojie Chai Tianyou (,2007). The Design and Application of Hybrid Intelligent Control System for Rotary Kiln Process in Alumina Production. *26th Chinese Control Conference Zhangjiajie, Hunan, China.*
- Wuang, Jun e Guang Wu (,1998). A multylayer recurrent neural network for solving continuous-time algebraic Riccati equations. *IEEE transactions on Computers.*
- Xia, Youshen e Ju Wuang (,2001). *Recurrent neural networks for Optimization: The State of the Art.*. CRC press. .

- Yu, Y. Nan (1983). *Electric Power Systems Dynamics*. Academic Press Inc.. New York-USA.
- Yuehjen E. Shao, Chien-Ho Wu Bin-Yih Ho e Bo-Sheng Hsu (,2007). Integration of Multivariate Control Charts and Neural Networks to Determine the Faults of Quality Characteristic(s) in a Multivariate Process. *11th International Conference on Computer Supported Cooperative Work in Design* pp. 1028–1031.
- Zhaoquan Cai, Guosheng Wang (,2008). Applications of Generalized Sylvester Matrix Equations in the Design of Eigenstructure Assignment. *Proceedings of the 7th World Congress on Intelligent Control and Automation , 2008, Chongqing, China* pp. 7317–7320.
- Zicouchian, Ali Mo Jamshidi (,2005). *Intelligent Control Systems Using Soft Computing Methodologies*. first ed.. CRC Press Boca Raton London, New York Washington, D. C.