

Universidade Federal Rural do Semi-Árido  
Universidade do Estado do Rio Grande do Norte  
Mestrado em Ciência da Computação

**Algoritmos Genéticos Aplicados ao Projeto de Controle  
com Realimentação de Estados**

**Phelipe Sena Oliveira**

Orientadores:

Prof. Dsc. Luciano Sales Barros

Prof. Dsc. Luiz Gonzaga de Q. Silveira Júnior

Mossoró - RN, dezembro de 2010.

Universidade Federal Rural do Semi-Árido  
Universidade do Estado do Rio Grande do Norte  
Mestrado em Ciência da Computação

**Algoritmos Genéticos Aplicados ao Projeto de Controle  
com Realimentação de Estados**

**Phelipe Sena Oliveira**

Orientadores:

Prof. Dr. Luciano Sales Barros

Prof. Dr. Luiz Gonzaga de Q. Silveira Júnior

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Ciência da Computação UERN/UFERSA (área de concentração: Infra-estrutura de Comunicação), como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Mossoró - RN, dezembro de 2010.

# **Algoritmos Genéticos Aplicados a Projeto de Controle com Realimentação de Estados**

**Phelipe Sena Oliveira**

Dissertação de Mestrado aprovada em 20 de dezembro de 2010 pela banca examinadora composta pelos seguintes membros:

---

Prof. Dsc. Luciano Sales Barros (Orientador) ..... UFERSA

---

Prof. Dsc. Luiz Gonzaga de Q. Silveira Júnior (Orientador) ..... UFERSA

---

Prof. Dsc. Iguatemi Eduardo da Fonseca ..... UFERSA

---

Prof. Dsc. Fabiano Fragoso Costa ..... UFRN

*Aos meus pais, irmãos e tios.*

# Agradecimentos

A Deus, por ter concedido boas oportunidades em minha vida.

Ao professor Luciano Sales Barros, pela ótima orientação durante todo o trabalho e pelos bons conselhos.

Ao professor Luiz Gonzaga de Queiroz Silveira Júnior, pelas boas discussões promovidas em suas orientações de alta qualidade.

Aos amigos Cleone, Fernando zambrotta, Gianna, Mailson Baia, Welliana, Aislânia e Ticiania pelos bons momentos compartilhados e ajuda nos momentos difíceis.

À minha namorada Ana Lúcia, pelo companheirismo e incentivos durante o trabalho.

A todos os professores do mestrado.

Ao programa de Mestrado em Ciência da Computação - UFERSA/UERN.

A toda a minha família, especialmente para minha mãe Carminha, meu pai Djalma e meus irmãos Tarcísio e Caruza.

À Capes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro prestado através do Programa PRÓ-ENGENHARIA.

# Resumo

Os sistemas de controle têm participação fundamental nos avanços da engenharia e da ciência. A demanda por realização de tarefas cada vez mais complicadas e de alta precisão faz com que os sistemas de controle se tornem cada vez mais complexos. Uma abordagem no Espaço de Estados é necessária para representar sistemas *MIMO*, e requer uma realimentação de estados como estratégia de controle, onde as técnicas mais utilizadas são *LQR* e *LQG*. Apesar de bons resultados obtidos por esses métodos, o projeto de controle não é uma tarefa simples devido ao método de tentativa e erro envolvido na definição das matrizes peso. Nesse caso, pode ser difícil o ajuste de parâmetros do controlador a fim de obter o comportamento ideal do sistema. Propõe-se uma estratégia de controle baseada na realimentação de estados com a utilização das técnicas de Algoritmos Genéticos em que não há o processo de tentativa e erro envolvido. O Algoritmo Genético busca os parâmetros para o projeto de controle que satisfaçam especificações de máximo sobre-sinal e tempo de acomodação. Os resultados obtidos mostram que é possível projetar controladores que cumprem especificações de projeto.

**Palavras-Chave:** Algoritmo Genético; Espaço de Estados; LQR; Máximo Sobre-Sinal; Tempo de Acomodação.

# Abstract

Control systems have an important role in engineering advancements. Currently, the requirement for complex tasks fulfilment makes the control systems become more sophisticated. The state space modelling is necessary to MIMO systems, and it requires the states feedback as control strategy, where the most used techniques are LQR and LQG. Although the good results obtained from these methods, the control design is not a trivial task due to the trial and error process involved in the weight matrices definition. In these cases, may be hard the controller parameters design in order to achieve the optimal system behavior. It proposes a states feedback control design technique based on genetic algorithms in which there are no trial and processes. The genetic algorithm searches the controller parameters which fulfil the design specifications of maximum overshoot and accommodation time. The obtained results show that is possible to design controllers which fulfil design specifications.

**Keywords:** Genetic Algorithm; State Space; LQR; Overshoot; Accommodation Time.

# Lista de Figuras

2.1	Sistema de Controle com Realimentação . . . . .	19
3.1	Processo de crossover . . . . .	32
3.2	Genes e cromossomo . . . . .	33
3.3	Esquema do Algoritmo Genético . . . . .	34
3.4	Cromossomo - Representação Binária . . . . .	35
3.5	Cromossomo - Problema do Caixeiro Viajante. . . . .	35
3.6	Cromossomo - Representação por Valor. . . . .	36
3.7	Roleta Viciada. . . . .	38
3.8	Roleta viciada em segmentos . . . . .	39
3.9	Roleta Viciada e Roleta com <i>Ranking</i> . . . . .	40
3.10	<i>Crossover</i> com um ponto de corte. . . . .	42
3.11	<i>Crossover</i> com $n$ pontos de corte. . . . .	42
3.12	Operador de mutação. . . . .	43
4.1	Curva de resposta a uma entrada degrau unitário que mostra $M$ e $t_s$ . . . . .	47
4.2	Representação da avaliação dos indivíduos . . . . .	49
5.1	Comparação dos resultados para $\phi_{11}$ , $\phi_{12}$ , $\phi_{21}$ e $\phi_{22}$ - Sistema 1. . . . .	54
5.2	Resposta livre para $x_1$ nas condições iniciais $x_1(0) = 1$ e $x_2(0) = 5$ - Sistema 1. . . . .	55
5.3	Resposta livre para $x_2$ nas condições iniciais $x_1(0) = 1$ e $x_2(0) = 5$ - Sistema 1. . . . .	55
5.4	Comparação dos resultados para $\phi_{11}$ , $\phi_{12}$ , $\phi_{21}$ e $\phi_{22}$ - Sistema 2. . . . .	56
5.5	Resposta livre para $x_1$ nas condições iniciais $x_1(0) = 2$ e $x_2(0) = 3$ - Sistema 2. . . . .	57
5.6	Resposta livre para $x_2$ nas condições iniciais $x_1(0) = 2$ e $x_2(0) = 3$ - Sistema 2. . . . .	57
5.7	Comparação dos resultados de módulo e fase para $\phi_{11}$ - Sistema 3. . . . .	59



5.8	Comparação dos resultados de módulo e fase para $\phi_{12}$ - Sistema 3. . . . .	59
5.9	Comparação dos resultados de módulo e fase para $\phi_{13}$ - Sistema 3. . . . .	59
5.10	Comparação dos resultados de módulo e fase para $\phi_{21}$ - Sistema 3. . . . .	60
5.11	Comparação dos resultados de módulo e fase para $\phi_{22}$ - Sistema 3. . . . .	60
5.12	Comparação dos resultados de módulo e fase para $\phi_{23}$ - Sistema 3. . . . .	60
5.13	Comparação dos resultados de módulo e fase para $\phi_{31}$ - Sistema 3. . . . .	61
5.14	Comparação dos resultados de módulo e fase para $\phi_{32}$ - Sistema 3. . . . .	61
5.15	Comparação dos resultados de módulo e fase para $\phi_{33}$ - Sistema 3. . . . .	61
5.16	Resposta livre para $x_1$ nas condições iniciais $x_1(0) = 1, x_2(0) = 1$ e $x_3(0) = 1$ - Sistema 3. . . . .	62
5.17	Resposta livre para $x_2$ nas condições iniciais $x_1(0) = 1, x_2(0) = 1$ e $x_3(0) = 1$ - Sistema 3. . . . .	62
5.18	Resposta livre para $x_3$ nas condições iniciais $x_1(0) = 1, x_2(0) = 1$ e $x_3(0) = 1$ - Sistema 3. . . . .	62
5.19	Comparação dos resultados de módulo e fase para $\phi_{11}$ - Sistema 4. . . . .	64
5.20	Comparação dos resultados de módulo e fase para $\phi_{12}$ - Sistema 4. . . . .	64
5.21	Comparação dos resultados de módulo e fase para $\phi_{13}$ - Sistema 4. . . . .	64
5.22	Comparação dos resultados de módulo e fase para $\phi_{21}$ - Sistema 4. . . . .	65
5.23	Comparação dos resultados de módulo e fase para $\phi_{22}$ - Sistema 4. . . . .	65
5.24	Comparação dos resultados de módulo e fase para $\phi_{23}$ - Sistema 4. . . . .	65
5.25	Comparação dos resultados de módulo e fase para $\phi_{31}$ - Sistema 4. . . . .	66
5.26	Comparação dos resultados de módulo e fase para $\phi_{32}$ - Sistema 4. . . . .	66
5.27	Comparação dos resultados de módulo e fase para $\phi_{33}$ - Sistema 4. . . . .	66
5.28	Resposta livre para $x_1$ nas condições iniciais $x_1(0) = 1, x_2(0) = 3$ e $x_3(0) = 2$ - Sistema 4. . . . .	67
5.29	Resposta livre para $x_2$ nas condições iniciais $x_1(0) = 1, x_2(0) = 3$ e $x_3(0) = 2$ - Sistema 4. . . . .	67
5.30	Resposta livre para $x_3$ nas condições iniciais $x_1(0) = 1, x_2(0) = 3$ e $x_3(0) = 2$ - Sistema 4. . . . .	67

# Lista de Tabelas

3.1	Selecionando indivíduos por roleta viciada . . . . .	39
5.1	Dados do projeto de controle do Sistema 1. . . . .	53
5.2	Autovalores do Sistema 1. . . . .	53
5.3	Dados do projeto de controle do Sistema 2. . . . .	55
5.4	Autovalores do sistema 2. . . . .	56
5.5	Dados do projeto de controle do Sistema 3. . . . .	58
5.6	Autovalores do Sistema 3. . . . .	58
5.7	Dados do projeto de controle do Sistema 4. . . . .	63
5.8	Autovalores do sistema 4. . . . .	63

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Motivação . . . . .	14
1.2	Objetivos . . . . .	15
1.3	Contribuições . . . . .	15
1.4	Metodologia . . . . .	16
1.5	Organização do Texto . . . . .	16
<b>2</b>	<b>Introdução aos Sistemas de Controle</b>	<b>18</b>
2.1	Introdução . . . . .	18
2.2	Conceitos Relacionados a Controle . . . . .	18
2.3	Modelagem de Sistemas de Controle . . . . .	19
2.4	Função de Transferência . . . . .	20
2.5	Modelagem no Espaço de Estados . . . . .	21
2.6	Análise de Sistemas de Controle no Espaço de Estados . . . . .	22
2.6.1	Correlação entre Função de Transferência e Equações no Espaço de Estados . . . . .	23
2.6.2	Autovalores de uma Matriz . . . . .	24
2.6.3	Solução da Equação de Estado . . . . .	25
2.6.4	Matriz de Transição de Estado . . . . .	26
2.7	Regulador Linear-Quadrático . . . . .	27
2.8	Considerações Finais . . . . .	29
<b>3</b>	<b>Fundamentos de Algoritmos Genéticos</b>	<b>30</b>
3.1	Origem . . . . .	30

3.2	Conceitos Básicos de Genética . . . . .	31
3.3	Terminologia . . . . .	33
3.4	Metodologia . . . . .	33
3.4.1	Codificação . . . . .	34
3.4.2	Inicialização . . . . .	36
3.4.3	Função de Avaliação . . . . .	37
3.4.4	Seleção . . . . .	37
3.4.5	Reprodução . . . . .	41
3.5	Considerações Finais . . . . .	44
<b>4</b>	<b>Sintonia de Controladores por Realimentação de Estados Baseada em Algoritmo Genético</b>	<b>45</b>
4.1	Estratégia de Controle Proposta . . . . .	45
4.2	Algoritmo Genético Desenvolvido . . . . .	48
4.3	Considerações Finais . . . . .	51
<b>5</b>	<b>Resultados</b>	<b>52</b>
5.1	Sistemas de ordem 2 . . . . .	52
5.1.1	Resultados do sistema 1 . . . . .	52
5.1.2	Resultados do sistema 2 . . . . .	55
5.2	Sistemas de ordem 3 . . . . .	57
5.2.1	Resultados do sistema 3 . . . . .	57
5.2.2	Resultados do sistema 4 . . . . .	63
5.3	Considerações Finais . . . . .	68
<b>6</b>	<b>Conclusões</b>	<b>69</b>
6.1	Conclusões . . . . .	69
6.2	Trabalhos Futuros . . . . .	70
<b>A</b>	<b>Cálculo da Matriz Exponencial <math>e^{At}</math></b>	<b>75</b>
A.1	Forma Canônica Diagonal . . . . .	75
A.2	Forma Canônica de Jordan . . . . .	76
A.3	Cálculo de $e^{At}$ : método 1. . . . .	77

A.4 Cálculo de $e^{At}$ : método 2. . . . .	78
<b>B Cálculo das Funções Fitness</b>	<b>79</b>

# Capítulo 1

## Introdução

### 1.1 Motivação

Sistemas de controle geralmente são utilizados para proporcionar maior produtividade e melhor desempenho de um dispositivo ou sistema. É possível notar aplicações destes sistemas nas mais diversas áreas, tais como, energia alternativa, robótica, biologia, indústria automobilística, dentre outras. Com o uso em larga escala, a exigência em relação aos sistemas de controle cresce devido à necessidade de realizar tarefas mais complexas e de alta precisão (DORF; BISHOP, 2009).

A finalidade do projeto de sistemas de controle é obter parâmetros do controlador, calculados por meio de diferentes técnicas, que garantam, principalmente, a estabilidade do sistema, e que minimizem índices de desempenho para fornecer o funcionamento desejado do sistema. A teoria de controle moderno trata de sistemas com múltiplas entradas e múltiplas saídas que podem se relacionar de maneira complexa. Para isso, utiliza uma representação baseada no espaço de estados e princípios de teoria de realimentação de estados.

As técnicas mais usadas para projeto de controle com realimentação de estados são *Linear Quadratic Regulator (LQR)* e *Linear Quadratic Gaussian (LQG)* (OGATA, 2003). Apesar de resultados satisfatórios obtidos a partir destes métodos, suas inicializações não são tarefas fáceis devido ao método de tentativa e erro envolvido na definição de matrizes peso. Neste caso, pode ser difícil o ajuste de parâmetros do controlador a fim de obter o comportamento ideal do sistema. Esta dificuldade é oriunda do fato de que o projeto utilizando tanto *LQR* quanto *LQG*, se baseia apenas no deslocamento dos pólos do sistema, sem atentar

para os índices de desempenho da resposta temporal do sistema.

Uma possibilidade para melhorar a busca por parâmetros do controlador é a eliminação do método de tentativa e erro utilizado pelo projetista, na definição das matrizes peso. Para viabilizar o processo é preciso utilizar um método que tenha capacidade de lidar com problemas contendo vasta área de busca, com isso, algoritmos de buscas exaustivas não poderão ser utilizados.

No entanto, Algoritmos Genéticos são uma técnica com características e potencial para ser aplicada na busca dos parâmetros de controle, pois, tendem a encontrar boas soluções em problemas combinatórios.

## 1.2 Objetivos

A partir das dificuldades observadas em técnicas de projeto de sistemas de controle por realimentação de estados, na utilização de métodos de tentativa e erro para cálculo de parâmetros do controlador, surgiu o principal objetivo desse trabalho: Construir uma estratégia de controle com utilização das técnicas de algoritmo genético para facilitar o projeto de sistemas de controle. Para isso foi necessário:

- Implementar um algoritmo genético que encontre parâmetros de projeto que satisfaçam as especificações, ou seja, índices de desempenho da resposta temporal do sistema.
- Realizar simulações com a finalidade de validar o novo método.

## 1.3 Contribuições

Podemos citar como contribuições do trabalho:

- Uma estratégia alternativa de projeto de controle, baseada em algoritmos genéticos.
- Garantia de um projeto de controle que satisfaçam as especificações de resposta do sistema.

## 1.4 Metodologia

O presente trabalho foi realizado de acordo com a seguinte metodologia:

- Foram feitas revisões bibliográficas das teorias de sistemas de controle. E algoritmos genéticos, e em trabalhos relevantes que correlacione os dois temas. Foi observado que:
  - Existem dificuldades na obtenção de parâmetros na fase de projetos de controle com realimentação de estados.
  - Algoritmos genéticos são considerados boas soluções em problemas combinatorios.
- Para diminuir a dificuldade na busca de parâmetros no projeto de controle, foi proposta uma estratégia de controle baseada nas técnicas de algoritmos genéticos.
- Por fim, foram realizadas simulações comparando a estratégia proposta com o método LQR, a fim de validar a técnica proposta.

## 1.5 Organização do Texto

A organização do texto dessa dissertação apresenta a seguinte estrutura:

- No capítulo 2 é feita uma breve fundamentação teórica sobre sistemas de controle, abordando conceitos, modelagem e análise de sistemas de controle, como também o método *Linear Quadratic Regulator*.
- No capítulo 3 é feita a fundamentação teórica sobre algoritmos genéticos, que vai desde a sua origem até a descrição de cada componente detalhadamente.
- No capítulo 4 é sugerido uma metodologia que aplica as teorias de algoritmos genéticos ao projeto de controle.
- No capítulo 5 são apresentados os resultados utilizando a estratégia de controle proposta.



- Por fim, no capítulo 6 são apresentadas as conclusões e sugeridas algumas possibilidades de trabalhos futuros.

# Capítulo 2

## Introdução aos Sistemas de Controle

Este capítulo mostra de maneira sucinta alguns conceitos sobre a Teoria de Controle, como a modelagem matemática que se utiliza em controle e tópicos em análise de sistemas de controle, além de abordar o projeto por *LQR*.

### 2.1 Introdução

Os sistemas de controle têm participação direta no desenvolvimento tecnológico, proporcionando avanços na engenharia e na ciência, como também, impactos em áreas tais como biologia e social (YUE et al., 2010), (CHAMAKEN; LITZ, 2010). Sua utilização aplica-se desde sistemas simples como controle de temperatura a sistemas digitais das mais sofisticadas aeronaves (RAYMOND, 1994).

Na seção seguinte, são apresentados conceitos sobre controle que são referenciados ao longo do trabalho.

### 2.2 Conceitos Relacionados a Controle

A seguir serão fornecidas algumas definições utilizadas em Teoria de Controle a fim de facilitar o entendimento de assuntos explanados posteriormente. A Figura 2.1 auxilia as definições.

**Planta:** algum objeto físico a ser controlado, ou conjunto de componentes integrados. Exemplos: estufa, espaçonave, robô, automóvel (controle de velocidade), etc.

**Entrada:** sinal de referência utilizado, normalmente um valor desejado.

**Saída:** grandeza ou condição que é medida e controlada.

**Sistema de Controle de Malha fechada:** também denominado de sistema de controle com realimentação, compara o sinal de entrada com o sinal de realimentação (sinal de saída) e determina o sinal de erro atuante, o objetivo é diminuir este sinal de erro para chegar a um valor desejado na saída, vide a Figura 2.1.

**Erro:** diferença entre o sinal de referência e o sinal de saída.

**Distúrbios:** sinais que afetam o sistema de maneira adversa interferindo nas variáveis de saída.

**Sensor:** dispositivo para medição, converte a variável de saída para viabilizar uma comparação entre a saída e o sinal de entrada de referência.

**Controlador:** dispositivo ou conjunto de dispositivos que detecta o sinal de erro atuante, e através de atuadores diminui ou zera o sinal de erro.

**Sistema de Controle de Malha aberta:** nesse sistema não existe comparação entre o sinal de saída e o sinal de referência, não utiliza-se controle por realimentação. É melhor aplicado onde não há distúrbios externos ou internos. Na Figura 2.1, retirando-se o ramo de realimentação juntamente com o sensor, o sistema passa a operar em malha aberta.

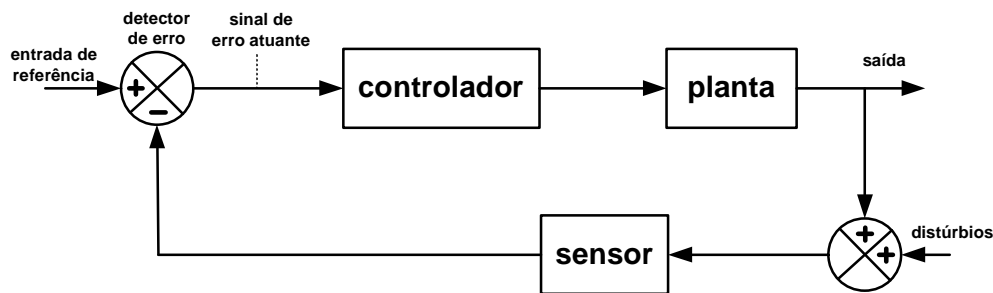


Figura 2.1: Sistema de Controle com Realimentação

## 2.3 Modelagem de Sistemas de Controle

Os sistemas de controle têm suas características denotadas por modelos matemáticos, que em grande parte são determinadas através de equações diferenciais. A partir dessas equações é possível obter representações para o sistema, por meio de Funções de Transferência ou

Equações de Estado. A escolha de qual modelo deverá ser utilizado para determinado sistema depende de suas circunstâncias particulares.

Na modelagem matemática do sistema, deve-se estabelecer um equilíbrio entre a simplicidade do modelo criado e a precisão dos resultados produzidos, pois, quanto maior a precisão, em termos de variáveis e características consideradas, maior será a complexidade do modelo obtido. Caso seja conveniente ou necessário a utilização de um modelo simplificado, escolhe-se algumas propriedades que serão ignoradas, de preferência as que causam poucos efeitos na resposta do sistema.

Um modelo simplificado pode facilitar o processo de modelagem e implementação do controlador, como também otimizar o tempo de resposta.

A seguir serão apresentadas de maneira mais detalhada duas abordagens de modelagem, Função de Transferência e Espaço de Estados.

## 2.4 Função de Transferência

Na Teoria de Controle, as relações entre a entrada e a saída dos sistemas (ou componentes) são, comumente, caracterizadas através de Funções de Transferência (FTs), isso quando o sistema pode ser descrito por meio de equações diferenciais lineares invariantes no tempo, ou seja, sistemas que são representados por equações diferenciais, cujos coeficientes são constantes no tempo.

A FT de um sistema é a relação entre a transformada de Laplace da saída e a transformada de Laplace da entrada, quando as condições iniciais são nulas e o sistema é representado por uma equação linear invariante no tempo. Essa relação apresenta-se da seguinte forma:

$$G(s) = \frac{\mathcal{L}[\text{SAÍDA}]}{\mathcal{L}[\text{ENTRADA}]} = \frac{Y(s)}{U(s)} = \frac{b_0s^n + b_1s^{n-1} + \dots + b_{n-1}s + b_n}{s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n} \quad (2.1)$$

Em que a saída do sistema é representada pelo numerador  $Y(s)$ , e a entrada representada pelo denominador  $U(s)$ . A *ordem* do sistema é determinada pelo expoente do termo de maior grau do denominador, nesse caso,  $n$ . O operador  $\mathcal{L}$  denota a transformada de Laplace. A utilização da transformada de Laplace na FT possibilita transformar uma equação diferencial linear em uma equação algébrica em  $s$ , com isso, é possível representar a dinâmica do sistema

através de simulações (OGATA, 2003).

A despeito de representarem toda a dinâmica do sistema, as FTs não exprimem nenhuma informação relacionada à estrutura ou composição do sistema descrito.

## 2.5 Modelagem no Espaço de Estados

É fácil constatar que a tendência dos sistemas de engenharia existentes na atualidade está relacionada ao aumento de sua complexidade, principalmente devido à necessidade de executar tarefas complexas e de maior precisão. Em conformidade às rigorosas exigências de desempenho dos sistemas de controle, a teoria de controle moderno e o subsequente projeto de sistemas sofisticados de controle vêm sendo desenvolvidos desde 1960.

A teoria de controle moderno é baseada na descrição de um sistema de  $n$  equações diferenciais de primeira ordem, onde o expoente do termo de maior grau é um. Diferentemente da FT, apresentada na seção anterior, que considera as equações de grau  $n$  na descrição do sistema. Então, apesar de aumentar o número de variáveis de estado, essa abordagem simplifica o processo de análise e modelagem de sistemas complexos, principalmente os sistemas Multiple-Input and Multiple-Output (*MIMO*) (FRANKLIN; EMAMI-NAEINI, 2006).

A abordagem no espaço de estados possibilita a descrição do comportamento interno do sistema e tem como base o conceito de estado. A seguir são explanados conceitos de estado, variáveis de estado, vetor de estado, espaço de estados e equações no espaço de estados (DORF; BISHOP, 2009).

**Estado:** pode ser definido como o menor conjunto de variáveis em  $t = t_0$  conhecidas em um tempo  $t \geq t_0$ , que determina completamente o comportamento do sistema para algum tempo  $t \geq t_0$ .

**Variáveis de estado:** são as variáveis  $x_1, x_2, \dots, x_n$  que juntas determinam o estado do sistema em um determinado tempo,  $t$ .

**Vetor de estado:** é o conjunto de  $n$  variáveis de estado que descreve completamente o comportamento do sistema.

**Espaço de estados:** é o espaço  $n$ -dimensional, cujos eixos coordenados são  $x_1, x_2, \dots, x_n$ .

**Equações no espaço de estados:** envolve três tipos de variáveis presentes na modelagem

de sistemas dinâmicos: variáveis de entrada, variáveis de saída e variáveis de estado. As Equações 2.2 e 2.3 representam as equações no espaço de estado.

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (2.2)$$

$$y = \mathbf{Cx} + \mathbf{Du} \quad (2.3)$$

onde:

$\mathbf{x}$  = vetor de estado (vetor de ordem  $n$ )

$y$  = sinal de saída (vetor de ordem  $m$ )

$\mathbf{u}$  = sinal de controle (vetor de ordem  $r$ )

$\mathbf{A}$  = matriz de estado, de ordem  $n \times n$

$\mathbf{B}$  = matriz de entrada, de ordem  $n \times r$

$\mathbf{C}$  = matriz de saída, de ordem  $m \times n$

$\mathbf{D}$  = matriz de transmissão direta, de ordem  $m \times r$

O leitor interessado em obter maiores informações poderá consultar (OGATA, 2003), (DORF; BISHOP, 2009), (LANDAU I. D.; M'SAAD, 1998) e (NISE, 2009).

O método de Espaço de Estados é utilizado no presente trabalho, pois, apresenta vantagens em relação às FTs. A representação por espaço de estados oferece uma maneira compacta e conveniente na modelagem e análise de sistemas com  $n$  entradas e  $m$  saídas. Ao contrário da modelagem e análise de um sistema *MIMO* por FTs, em que seria necessário realizar um número  $n \times m$  de transformadas de Laplace para processar todas as características do sistema (BROGAN, 1991).

## 2.6 Análise de Sistemas de Controle no Espaço de Estados

A teoria clássica de controle analisa sistemas que possuem única entrada e única saída, por meio da FT, mas os sistemas de controle atuais são mais complexos, no sentido de que precisam lidar com muitas entradas e muitas saídas, as quais podem se relacionar de maneira que seja difícil de compreender. Para superar as dificuldades e facilitar a análise, é comum expressar os sistemas no Espaço de Estados, pois o aumento do número de variáveis de es-

tado, do número de entradas ou do número de saídas não aumenta a dificuldade na resolução das equações.

Apesar das diferenças existentes entre as duas representações, elas podem ser convertidas mutuamente. Existem muitas técnicas (DORF; BISHOP, 2009) matemáticas e software especializados (MATLAB, 2008) que auxiliam no processo de análise e projeto de Sistemas de Controle.

### 2.6.1 Correlação entre Função de Transferência e Equações no Espaço de Estados

Enquanto que as equações no espaço de estados representam melhor os sistemas dinâmicos, a função de transferência é mais conveniente para sistemas lineares invariantes no tempo.

#### Espaço de Estados Representado por Funções de Transferência

É interessante observar que a FT,  $G(s)$ , pode ser obtida através das equações no espaço de estados como apresentado a seguir. Para tanto, considere o sistema cuja FT é dada por:

$$\frac{Y(s)}{U(s)} = G(s) \quad (2.4)$$

A representação desse sistema no espaço de estados é definida pela transformada de Laplace das Equações (2.2) e (2.3). Que resulta em:

$$s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}U(s) \quad (2.5)$$

$$Y(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}U(s), \quad (2.6)$$

Sendo as condições iniciais nulas, ou seja,  $x(0)$  igual a zero, é possível escrever

$$s\mathbf{X}(s) - \mathbf{A}\mathbf{X}(s) = \mathbf{B}U(s) \quad (2.7)$$

ou,

$$(s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{B}U(s) \quad (2.8)$$

onde  $\mathbf{I}$  é a matriz identidade. Multiplicando ambos os lados por  $(s\mathbf{I} - \mathbf{A})^{-1}$ , tem-se

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) \quad (2.9)$$

Substituindo a Equação (2.9) na Equação (2.6), fica:

$$Y(s) = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}]\mathbf{U}(s) \quad (2.10)$$

Comparando a Equação (2.10) com a Equação (2.4), observamos que:

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (2.11)$$

Essa é a expressão da FT que podemos obter a partir de um sistema representado pelas equações no espaço de estados em termos de  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  e  $\mathbf{D}$ .

É importante destacar que ao se transformar uma representação no Espaço de Estados para Funções de Transferências pode ocasionar perdas de informações internas inerentes ao sistema.

### Representação de Funções de Transferência no Espaço de Estados

Poderá ser necessário a transformação de FTs de um dado sistema para o Espaço de Estados, com a intenção de observar características internas do sistema. É possível fazer a conversão de FTs para as formas canônicas, a fim de obter informações sobre controlabilidade, observabilidade e ainda, fornecer meios para o cálculo de  $e^{\mathbf{A}t}$ .

#### 2.6.2 Autovalores de uma Matriz

A análise dos autovalores de um sistema propicia o conhecimento completo desse sistema. É possível observar a partir dos autovalores se o sistema é, por exemplo, estável, ou mesmo se um sistema antes instável foi estabilizado, ou não, com a aplicação de técnicas de controle. Para que um sistema seja considerado estável é necessário que os autovalores da matriz de estado tenham parte real negativa (KUNDUR, 1994).

Os autovalores de uma matriz  $\mathbf{A}$   $n \times n$  são as raízes da equação característica

$$|\lambda\mathbf{I} - \mathbf{A}| = 0$$

A partir da seguinte matriz  $A$ :



$$A = \begin{bmatrix} 2 & 1 & 4 \\ 0 & 2 & 0 \\ 0 & 3 & 1 \end{bmatrix}$$

Calcula-se o polinômio característico, dado por:

$$|\lambda \mathbf{I} - \mathbf{A}| = \begin{vmatrix} \lambda - 2 & -1 & -4 \\ 0 & \lambda - 2 & 0 \\ 0 & -3 & \lambda - 1 \end{vmatrix} = (\lambda - 2)^2(\lambda - 1)$$

Os autovalores de  $\mathbf{A}$  são as raízes da equação característica, ou seja, 2, 2 e 1.

### 2.6.3 Solução da Equação de Estado

A abordagem pela Transformada de Laplace na solução da Equação de Estado para casos homogêneos é mostrado abaixo.

$$\dot{x} = ax \quad (2.12)$$

Considerando a Transformada de Laplace da Equação (2.12), obtemos:

$$sX(s) - x(0) = aX(s) \quad (2.13)$$

Em que,  $X(s)$  é a transformada de Laplace de  $x$  e  $x(0)$  é o estado inicial. Resolvendo a Equação (2.13) para  $X(s)$ , tem-se

$$X(s) = \frac{x(0)}{s - a} = (s - a)^{-1}x(0) \quad (2.14)$$

A Transformada Inversa de Laplace desta última equação fornece a solução:

$$x(t) = e^{at}x(0) \quad (2.15)$$

Esta abordagem pode ser estendida para o caso vetorial:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \quad (2.16)$$

Considerando a Transformada de Laplace nos dois lados da Equação (2.16), obtemos:

$$s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s)$$

Portanto,

$$(s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{x}(0) \quad (2.17)$$

Onde  $\mathbf{I}$  é a matriz identidade. Pré-multiplicando ambos os lados da Equação (2.17) por  $(s\mathbf{I} - \mathbf{A})^{-1}$

obtemos:

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0)$$

A Transformada Inversa de Laplace de  $\mathbf{X}(s)$  fornece a solução  $\mathbf{x}(t)$ . Então,

$$\mathbf{x}(t) = \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A})^{-1}]\mathbf{x}(0) \quad (2.18)$$

Resulta em

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) \quad (2.19)$$

Existem várias maneiras para o cálculo de  $e^{\mathbf{A}t}$ , para maiores detalhes ver Apêndice A.

### 2.6.4 Matriz de Transição de Estado

Podemos escrever a solução da Equação de Estado Homogênea como:

$$\mathbf{x}(t) = \Phi(t)\mathbf{x}(0) \quad (2.20)$$

Em que  $\Phi(t)$  é uma matriz  $n \times n$ , que é dada por:

$$\Phi(t) = e^{\mathbf{A}t}$$

Com as seguintes características:

$$\dot{\Phi}(t) = \mathbf{A}\Phi(t), \quad \Phi(0) = \mathbf{I}$$

Para verificar isso, note que:

$$\mathbf{x}(0) = \Phi(0)\mathbf{x}(0) = \mathbf{x}(0)$$

$$\dot{\mathbf{x}}(t) = \dot{\Phi}(t)\mathbf{x}(0) = \mathbf{A}\Phi(t)\mathbf{x}(0) = \mathbf{A}\mathbf{x}(t)$$

Confirmamos, portanto, que a Equação (2.15) é a solução da Equação (2.12).

A partir das Equações (2.18), (2.19) e (2.20), obtemos:

$$\Phi(t) = e^{\mathbf{A}t} = \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A})^{-1}]$$

Note que

$$\Phi^{-1}(t) = e^{-\mathbf{A}t} = \Phi(-t)$$

A matriz  $\Phi(t)$  é denominada matriz de transição de estados. Ela contém toda a informação a respeito da resposta livre do sistema definido pela Equação (2.19) (OGATA, 2003).

Se os autovalores  $\lambda_1, \lambda_2, \dots, \lambda_n$  da matriz  $\mathbf{A}$  são distintos, então  $\Phi(t)$  contém as  $n$  exponenciais

$$e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t}$$

Em um caso particular, se  $A$  é uma matriz diagonal, então

$$\Phi(t) = e^{\mathbf{A}t} = \begin{bmatrix} e^{\lambda_1 t} & & & & 0 \\ & e^{\lambda_2 t} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ 0 & & & & & e^{\lambda_n t} \end{bmatrix}$$

## 2.7 Regulador Linear-Quadrático

O Regulador Linear-Quadrático, *LQR*, é um método de controle baseado na realimentação de estados. O controle *LQR* utiliza a seguinte estratégia de controle:

$$u = -Kx \tag{2.21}$$

Em que  $K$  é a matriz de realimentação de estados. Substituindo a Equação (2.21) na Equação de Estado (2.2), o sistema passa a ser:

$$\dot{x} = (A - BK)x \quad (2.22)$$

A definição da matriz  $K$  para que se consiga um equilíbrio entre a velocidade de resposta e elevação do sinal do sistema é um problema de otimização.

A lei de controle ótimo ou  $LQR$  é computada pela minimização do índice de desempenho quadrático:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.23)$$

Sendo  $Q$  a matriz hermitiana definida ou semidefinida positiva (ou real simétrica) que determina a importância do erro e  $R$  uma matriz hermitiana definida positiva ou real simétrica que representa o consumo de energia.

A quantidade  $(x^T Q x)$  é uma medida da extensão do desvio do estado num tempo  $t$  para o estado inicial em  $t = 0$ . O termo  $(u^T R u)$  penaliza o esforço do controlador no processo de minimização de  $J$ .

A minimização de índices de desempenho quadráticos, via solução da equação de Riccati, conduz naturalmente a leis de realimentação lineares (OGATA, 2003), e a matriz ganho  $K$  em (2.21) é dada por:

$$K = \bar{R}^{-1} \bar{B}^T P$$

Em que  $P$  é a matriz solução (única, simétrica e positivo-semidefinida) da equação de Riccati (KWAKERNAAK; SIVAN, 1972).

$$P\bar{A}' + \bar{A}'^T P - P\bar{B}'\bar{R}^{-1}\bar{B}'^T P + \bar{Q} = 0$$

A solução desta equação, para o cálculo de  $K$ , pode ser obtida através de métodos numéricos como o Método Iterativo de Newton-Raphson.

## 2.8 Considerações Finais

Neste capítulo foram apresentados conceitos sobre a Teoria de Controle. Verificou-se a importância e vantagens em representar um sistema de controle pela modelagem no Espaço de Estados. Na análise de sistemas foi constatado que a matriz de transição de estados contém toda informação da resposta livre de um sistema. É através de modificações nessa matriz que se baseia a nova estratégia de controle desenvolvida nesse trabalho. O método *LQR* mostrado na seção anterior é utilizado para comparações com a estratégia desenvolvida, e assim, possibilitar averiguações.

O próximo capítulo trata de fundamentos de algoritmos genéticos. Cada etapa do algoritmo é detalhada mostrando possibilidades de implementação de acordo com o problema abordado.

# Capítulo 3

## Fundamentos de Algoritmos Genéticos

Algoritmos Genéticos (AG), são inspirados no processo de evolução natural, e visa buscar soluções sub-ótimas para problemas cuja complexidade combinatória é elevada. Assim, suas bases surgiram com as teorias de Charles Darwin. Posteriormente, a estas teorias foi-se adicionando o formalismo matemático, o que permitiu escrever soluções na forma de um algoritmo computacional, o *algoritmo genético* (MELANIE, 1999), (HAUPT; HAUPT, 1998).

Os AGs são heurísticas<sup>1</sup> que não têm a garantia de encontrar a solução ótima, porém, são capazes de encontrar boas soluções. A utilização de AG é mais apropriada em problemas intratáveis. Os problemas intratáveis têm o espaço de solução tão amplo que, precisariam de um tempo extraordinariamente longo para percorrê-lo, sendo, portanto considerados sem solução para métodos de busca exaustiva (LINDEN, 2006).

### 3.1 Origem

O naturalista Charles Robert Darwin (1809 - 1882) despontou seu trabalho em uma viagem a bordo do navio H.M.S Beagle para as Ilhas Galápagos, uma viagem que durou quase cinco anos e teve início no ano de 1831. Antes de Darwin, outros estudiosos já estavam tentando decifrar o evolucionismo, mas foi ele em seu livro “A Origem das Espécies”, publicado em 1859, quem definiu as teorias que são aceitas como corretas até os dias de hoje. Darwin observou nas Ilhas, que animais da mesma espécie tinham pequenas diferenças devido a

---

<sup>1</sup>Heurísticas são algoritmos que tendem a encontrar a solução sub-ótima em tempo de execução aceitável na maioria de suas execuções.

algumas adaptações aos diferentes ecossistemas em que viviam (HAUPT; HAUPT, 1998).

Sua obra defende duas proposições importantes: organismos descendem de ancestrais em comum com algumas modificações e que o principal fator de modificação é a **Seleção Natural**. Numa definição mais detalhada, a teoria da evolução descreve a disputa dos indivíduos dentro de um mesmo ecossistema devido às limitações de recurso na natureza. Aqueles que forem classificados como mais fortes são os que terão maior chance de proliferar, o que garante a formação de descendentes mais adaptados àqueles ecossistemas.

Baseando-se na teoria da seleção natural, por volta de 1940, inicia-se a história dos Algoritmos Genéticos. Os pesquisadores buscavam soluções para problemas complexos e tomavam como inspiração a teoria da evolução, visto que, essa teoria foi concebida a partir da análise de um complexo problema na natureza: a seleção natural (MELANIE, 1999).

Na década de 60, John Holland começa a investigar os Algoritmos genéticos. Ele representou os cromossomos por cadeias de 0's e 1's e propôs um modelo heurístico na tentativa de chegar à solução de problemas complexos. Holland em 1975 publicou "*Adaptation in Natural and Artificial Systems*", o primeiro trabalho em AG, considerado na época um grande incentivo para pesquisas sobre o assunto, que vem evoluindo até os dias de hoje com uma grande dimensão de aplicações nas mais diversas áreas (HAUPT; HAUPT, 1998).

## 3.2 Conceitos Básicos de Genética

As particularidades de cada indivíduo estão armazenadas no núcleo de suas células em estruturas denominadas cromossomos, que em conjunto denominam-se genoma. Os cromossomos são os responsáveis pelas características físicas: cor dos pêlos, tamanho, cor dos olhos e também por algumas características psicológicas, dentre outras. São formados em pares, sendo que a quantidade varia de espécie para espécie, nos humanos por exemplo, são 23 pares em cada célula, ou seja, 46 cromossomos. A unidade básica dos cromossomos são os genes, cada gene é responsável por apenas uma característica física do indivíduo, eles são blocos de sequência de ácido desoxirribonucleico ou *Deoxyribonucleic Acid DNA*, e têm sua posição determinada no cromossomo chamada de locus (OTTO, 2004).

Na natureza existem duas maneiras dos seres vivos se reproduzirem: assexuada e sexuada. A primeira, ocorre em organismos inferiores como os vírus e as bactérias, os quais

fazem cópias de si próprios e não apresentam variações, exceto, no caso de mutação. Já na reprodução sexuada, dois organismos formam um novo organismo com a junção de seus materiais genéticos. O novo se difere de seus pais por ter sido criado a partir de metade dos genes de cada um e ainda por alguns eventos durante a meiose (processo de divisão celular): *crossover* e mutação (OTTO, 2004).

O processo de gerar um novo indivíduo tem como primeira etapa a duplicação dos cromossomos, seguida do *crossover* ou *crossing-over*, que é a troca de parte do cromossomo por outra parte do cromossomo homólogo<sup>2</sup>. Esse processo é ilustrado na Figura 3.1.

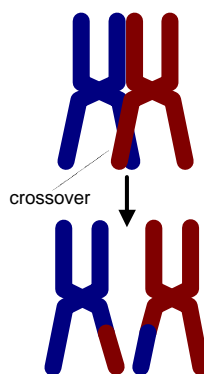


Figura 3.1: Processo de crossover

O nome *crossover* é devido ao cruzamento de um cromossomo sobre o outro para troca de genes, sendo que poderá ocorrer em vários pontos. A Figura 3.1 ilustra o *crossover* de apenas um ponto.

A divisão celular (processo de replicação do DNA) tem complexidade elevada, portanto, nesse processo podem ocorrer as mutações. As mutações são pequenos erros que geram modificações no conjunto de genes da população. Existem mutações desfavoráveis que são eliminadas por meio da seleção natural, mutações vantajosas que são proliferadas sendo aceitas como mudanças adaptativas e também as mutações neutras, que não influenciam na aptidão dos indivíduos (MELANIE, 1999).

<sup>2</sup>Cromossomo homólogo é o cromossomo que forma o par para troca de material genético



### 3.3 Terminologia

Os AGs utilizam muitas terminologias da biologia. A seguir temos uma descrição sucinta dos significados que serão explorados pelo algoritmo genético.

**Gene:** variável de otimização, ou seja, uma codificação dos parâmetros de otimização, ilustrado na Figura 3.2 .

**Cromossomo:** conjunto de genes, é também chamado de indivíduo, candidato a solução de otimização, ilustrado na Figura 3.2.

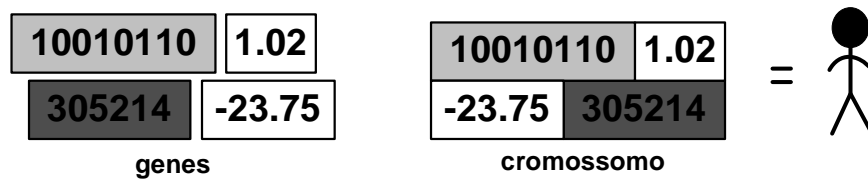


Figura 3.2: Genes e cromossomo

**População inicial:** conjunto de todos os cromossomos (indivíduos) criados aleatoriamente, onde será feita a busca pela solução.

**Gerações:** populações criadas a partir da população inicial através da aplicação de *crossover* e mutação.

**Pai:** indivíduo selecionado para o cruzamento.

**Filho:** indivíduo criado no cruzamento entre dois pais.

Além desses conceitos, existem outros que são relevantes para o algoritmo genético.

**Função de Avaliação:** função que avalia cada indivíduo da população atribuindo um valor que mostra quão distante aquele indivíduo está da solução. É também chamada de função de aptidão ou função *fitness*.

**Critério de Parada:** exigência para que o algoritmo encerre. Pode ser número de gerações, tempo de execução ou solução encontrada.

### 3.4 Metodologia

Os Algoritmos Genéticos são inspirados na teoria da evolução Darwiniana e sua metodologia é baseada na seleção natural. A primeira etapa do AG é transformar a representação real do indivíduo em uma representação computacional, necessária para que o

algoritmo possa manipulá-la. Posteriormente, a população inicial é criada, em seguida, uma avaliação é feita em todos os indivíduos, verificando se algum atende o critério de parada. Em caso negativo, uma nova reprodução é feita originando uma nova população que será avaliada, iniciando um novo ciclo. Na Figura 3.3, é ilustrado um esquema de como o AG funciona. A seguir serão apresentados as etapas do AG com um nível maior de detalhes de funcionamento.

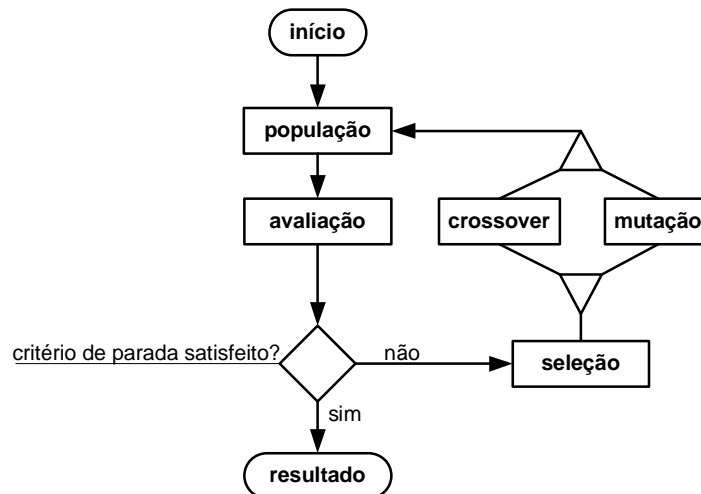


Figura 3.3: Esquema do Algoritmo Genético

### 3.4.1 Codificação

Nesta etapa, o projetista do AG terá que definir, antes mesmo de implementar, o esquema de codificação dos parâmetros a serem manipulados pelo algoritmo. Cada indivíduo da população é uma solução em potencial e responsável pela conexão entre a busca atual, o problema de otimização, e o AG. Em outras palavras, uma maneira de traduzir a informação do problema em uma forma viável de ser tratada pelo computador. Isso é possível devido a uma codificação do indivíduo real, transformando-o em uma representação computacional. A representação computacional permite uma manipulação fácil e eficiente dos operadores genéticos (definidos a seguir) sobre os indivíduos, ela pode ser binária, por permutação, por valor ou poderá ser criada uma outra representação a critério do programador. Em seguida, serão apresentados alguns tipos de codificação existentes na literatura (JIAN-CHENG et al., 2009), (ROTHLAUF, 2002).

### Binária

A codificação binária foi utilizada por Holland, no surgimento do AG, e é a preferida entre os desenvolvedores devido à simplicidade, pois codifica um indivíduo apenas como uma cadeia de bits, como visto na Figura 3.4. É uma codificação de fácil manipulação pelos métodos de reprodução e adaptável para diversos problemas, uma vez que, os indivíduos serão os mesmos: um vetor de bits. A desvantagem dessa codificação é a necessidade de conversão de dados para valores binários e possíveis valores inconsistentes depois da decodificação (LIANG et al., 2006).

1	0	0	1	0	1
---	---	---	---	---	---

Figura 3.4: Cromossomo - Representação Binária

### Por Permutação

Existem problemas denominados combinatórios, neles a ordem dos genes é relevante para encontrar uma solução. O Problema do Caixeiro Viajante (PCV) é um exemplo (KAUR; MURUGAPPAN, 2008). Na Figura 3.5, cada número representa um gene e a sequência dos números significa a ordem das cidades que o Caixeiro Viajante irá percorrer. A diferença entre os indivíduos é apenas a ordem dos genes. Nos problemas combinatórios, é essencial a definição de algumas restrições na criação dos indivíduos, para que o algoritmo não perca tempo trabalhando com indivíduos que não estão no espaço de busca de uma solução (KROMER et al., 2009).

0	7	4	6	1	8	3	9	5	2
---	---	---	---	---	---	---	---	---	---

Figura 3.5: Cromossomo - Problema do Caixeiro Viajante.

### Valor

A codificação por valor não utiliza conversão de dados, por isso, também é chamada de representação direta. Na figura 3.6, percebemos que os genes são valores utilizados no

problema real. Essa codificação é aplicada quando se encontra dificuldade na conversão entre a representação da situação real e a computacional (JIAN-CHENG et al., 2009).

3.12	-0.098	1024.54	0.76
------	--------	---------	------

Figura 3.6: Cromossomo - Representação por Valor.

### 3.4.2 Inicialização

O início do AG é através da criação da população inicial, mas antes, é preciso definir alguns parâmetros, tais como, tamanho da população e intervalo de valores possíveis para os genes. O tamanho da população interfere na diversidade e no tempo em que o algoritmo encontra uma resposta. Quanto maior a população, maior a diversidade desta e mais tempo de processamento será gasto e vice-versa. O mesmo ocorre com o intervalo de valores possíveis para os genes, quando muito extenso o algoritmo torna-se mais lento e, se muito pequeno, a diversidade da população é afetada. É notável, em alguns trabalhos, que o tamanho da população varia na faixa entre 20 e 200 indivíduos, e o intervalo de valores, até onde são considerados valores admissíveis para os genes. Se o desenvolvedor tem algum conhecimento sobre o problema a ser tratado pelo AG, este poderá reduzir o intervalo de valores de acordo com suas informações, para otimizar o tempo de busca. A inicialização dos indivíduos é feita de maneira aleatória, atendendo a especificação de valores possíveis e a quantidade de indivíduos da população.

#### Variações na população

Existem pesquisas feitas apenas para determinar o tamanho ideal da população e o intervalo de valores possíveis para os genes. Em Zhang e Liu (2007) é proposta a divisão do intervalo de valores para descartar partes que não contenham o máximo global. Com isso ganha-se em tempo de processamento e rapidez na convergência para uma solução. O trabalho desenvolvido por Junhua e Ming (2008) utiliza duas populações para o AG. As taxas de crossover e mutação são atribuídas de acordo com a diversidade de cada uma das populações. Uma população em que os indivíduos têm alta similaridade recebe taxa de crossover

alta, para diminuir a similaridade entre os indivíduos. Em uma população com baixa similaridade entre os indivíduos, a mutação tem mais probabilidade de acontecer do que o normal.

### 3.4.3 Função de Avaliação

A função de avaliação é a única inerente ao problema. Ela tem o propósito de relacionar o problema ao AG, e de avaliar o indivíduo atribuindo a este um valor que mostre o quão distante está da solução ideal. Muito cuidado deve-se ter para implementação desta função, pois ela contém todas as informações sobre o problema, como restrições e objetivos (GUO-SHENG et al., 2010).

### 3.4.4 Seleção

A perpetuação dos indivíduos mais aptos depende da maneira como é feita a seleção. A seleção dos indivíduos para reprodução é uma tarefa que pode afetar diretamente no resultado final do algoritmo. Dependendo da forma com que esta for realizada, a convergência para uma solução desejada poderá ficar mais rápida, ou mais lenta. Se a seleção escolhe apenas os indivíduos bons, então a nova geração poderá perder algumas características favoráveis que, possivelmente, existem em indivíduos ruins. Já no caso de se escolherem muitos indivíduos ruins, as características ruins desses indivíduos tendem a proliferar por várias gerações. O método da roleta é o que imita a seleção que acontece na natureza, mas existem outros que também são utilizados, como veremos a seguir (LINDEN, 2006), (AL-HAJRI; ABIDO, 2009).

#### Seleção por Roleta Viciada

Em uma roleta, cada indivíduo é representado por uma fatia diretamente proporcional à avaliação recebida. Após girar a roleta, o indivíduo escolhido é aquele que corresponde à fatia selecionada.

A relevância desse modelo está no favorecimento aos indivíduos mais aptos, mas sem, necessariamente, ter que eliminar os menos aptos. Dessa forma é possível aproveitar as características favoráveis que possam existir em indivíduos ruins e manter a diversidade da população no decorrer das gerações. A seguir, é mostrado uma versão pseudo-algorítmica da roleta:

1. Some todas as avaliações armazenando em *soma*
2. Ordene os indivíduos em ordem crescente de avaliação
3. Selecione um número  $s$  entre 0 e *soma*
4.  $i=1$
5.  $temp = \text{avaliação do indivíduo } 1$
6. enquanto  $temp < s$
7.      $i = i + 1$
8.      $temp = temp + \text{avaliação do indivíduo } i$
9. fim enquanto

Na Figura 3.7, notamos que o melhor indivíduo, *A*, em termos de avaliação, recebe a maior fatia na roleta por ter a maior avaliação e, provavelmente, será escolhido com maior frequência. O pior indivíduo, *F*, recebe a menor fatia, mesmo assim, ainda tem chance de ser escolhido.

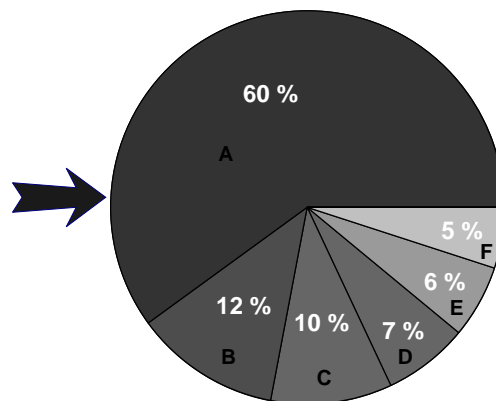


Figura 3.7: Roleta Viciada.

A roleta pode ser vista como um segmento, onde os indivíduos recebem segmentos contíguos proporcionalmente ao seu *score*. Em seguida é gerado um número aleatório, e o primeiro indivíduo cujo segmento ultrapassa o valor desse número é escolhido para reprodução. Vejamos o seguinte exemplo para mostrar na prática como é feita a escolha dos indivíduos. A Tabela 3.1 apresenta a descrição de uma população com 6 indivíduos.

Tabela 3.1: Selecionando indivíduos por roleta viciada

Descrição dos Indivíduos						
Identificação	A	B	C	D	E	F
Score= $f(x)$	48	9.6	8	5.6	4.8	4
Probabilidade $f(x)/\sum f(x)$	0.6	0.12	0.10	0.07	0.06	0.05

A Figura 3.8 ilustra o método da roleta viciada em segmentos construídos a partir da Tabela 3.1.

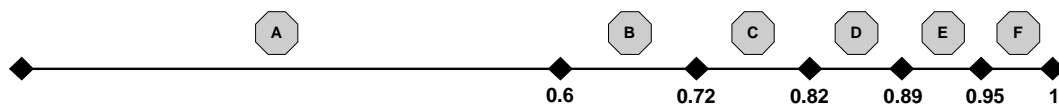


Figura 3.8: Roleta viciada em segmentos

Suponha-se que seja necessário a escolha de 3 pares de indivíduos para reprodução. Então gera-se 6 números aleatórios, como mostrado abaixo

0.49 e 0.84    0.77 e 0.23    0.58 e 0.69

Sendo assim, os indivíduos que vão gerar descendentes são:

A e D    C e A    A e B

É conveniente a realização de ajustes na escolha do aleatório para que um par de indivíduos sempre seja formado por indivíduos distintos.

### Seleção por Torneio

Nesse tipo de seleção escolhe-se aleatoriamente uma quantidade  $k$  de indivíduos que vão competir entre si usando suas avaliações, o vencedor desse grupo participará da reprodução. O valor mínimo para  $k$  é 2, tornando sempre possível uma competição. Quanto maior o tamanho de  $k$  maior a dominância do indivíduo mais forte, se  $k$  for igual ao tamanho da população o vencedor sempre será o mesmo. De acordo com Zhong et al. (2005), a estratégia de seleção por torneio converge mais rápido do que a seleção por roleta viciada, o que é comprovado por experimentos numéricos (CHAKRABORTY; CHAKRABORTY, 1997).

### Seleção por *Ranking*

Este método de seleção foi descrito por Mitchell (1996), e consiste em evitar a rápida convergência do AG para um máximo local, e perda de diversidade da população causada pela dominância de um superindivíduo. Na seleção por *ranking*, a população é ordenada de acordo com a função de avaliação, criando-se assim, um *ranking*. Após estabelecido o *rank*, é feito um mapeamento para uma função de avaliação. A nova avaliação de cada indivíduo é calculada por:

$$E(i,t) = Min + (Max - Min) \cdot \frac{rank(i,t) - 1}{N - 1}$$

$E(i,t)$ : avaliação do indivíduo  $i$  na geração  $t$ .

$Min$ : avaliação atribuída ao indivíduo da última colocação.

$Max$ : avaliação atribuída ao indivíduo melhor colocado.

$N$ : quantidade de indivíduos na população.

$rank(i,t)$ : é a posição do indivíduo  $i$  na geração  $t$

Este método de seleção favorece os melhores indivíduos, controla a convergência prematura e evita o domínio de um superindivíduo. Depois de calculado as novas avaliações escolhe-se outro método de seleção para escolher os pais a partir do ranking. A Figura 3.9 ilustra a comparação de uma roleta viciada com outra roleta que utiliza o *ranking* (LINDEN, 2006), (LIU et al., 2003), (ABDULAL et al., 2009).

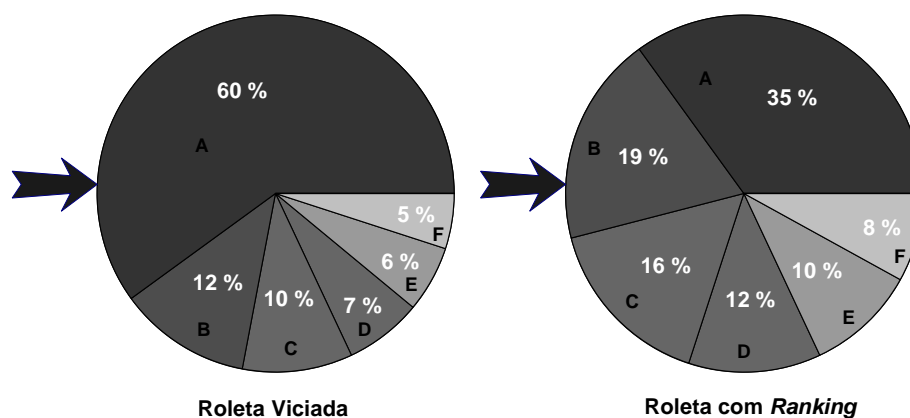


Figura 3.9: Roleta Viciada e Roleta com *Ranking*.



### Seleção Elitista

Ao passar de uma geração para outra, a população vai mudando, gerando novos indivíduos, na busca por uma solução sub-ótima. No AG padrão, é possível que uma geração  $k$  tenha o seu melhor indivíduo com mais qualidade do que o melhor indivíduo de uma geração  $k + 1$ , isso significa dizer que um indivíduo mais próximo do ideal pode ser perdido no decorrer das gerações, tornando o algoritmo mais lento na capacidade de convergir os indivíduos para boas soluções. Para evitar tal situação é utilizada a técnica do elitismo.

O elitismo garante a permanência de  $n$  melhores indivíduos de uma população para a próxima geração, isso faz com que as avaliações dos melhores indivíduos no decorrer das gerações sejam sempre crescentes, pois os  $n$  melhores indivíduos em uma população são melhores, ou pelo menos iguais, aos melhores indivíduos de uma geração anterior (JONG, 1975), (RAMAKRISHNA, 2003).

### 3.4.5 Reprodução

A reprodução no AG tem o propósito de combinar características dos indivíduos selecionados em busca de melhorá-los em relação à sua avaliação, até que se chegue à solução do problema. O *crossover* é o principal operador do AG, pois é responsável pela reprodução. Depois de criados os filhos com o *crossover*, podem vir a acontecer mutações, que desempenham um papel importante no AG. A seguir serão apresentados o *crossover* e a mutação (QI-YI; SHU-CHUN, 2009).

#### *Crossover*

O operador de *crossover* ou recombinação, cria os filhos através da troca de material genético dos pais selecionados. O *crossover* de um ponto é o cruzamento mais simples entre dois indivíduos, outros operadores de *crossover* mais sofisticados também são utilizados.

#### ***Crossover com um ponto de corte***

Nesse operador de *crossover*, os progenitores são divididos em duas partes pelo ponto de corte, que normalmente é escolhido aleatoriamente, com isso, o novo indivíduo é formado por partes não semelhantes dos indivíduos pais, como ilustrado na Figura 3.10.

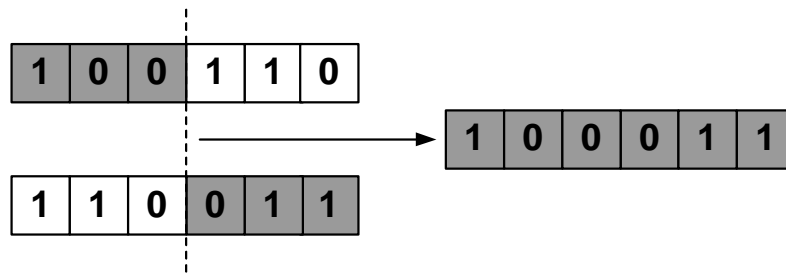


Figura 3.10: *Crossover* com um ponto de corte.

### ***Crossover* com $n$ pontos de corte**

O funcionamento desse *crossover* é semelhante ao anterior. Agora serão escolhidos  $n$  pontos de corte para troca do material genético. A Figura 3.11 ilustra um *crossover* com 3 pontos de corte.

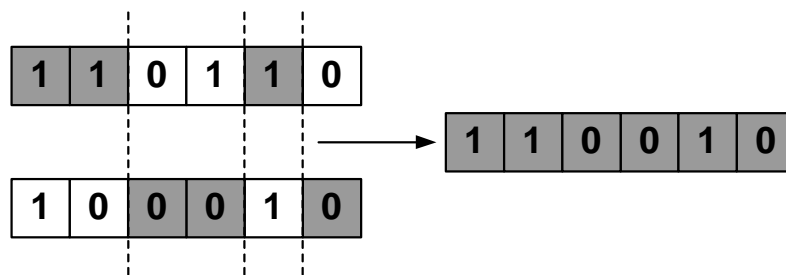


Figura 3.11: *Crossover* com  $n$  pontos de corte.

### ***Crossover* Aritmético**

Além dos operadores tradicionais de *crossover*, foram implementados outros modelos para serem utilizados em algoritmos com representação real, como por exemplo, o *crossover* aritmético. O *crossover* aritmético não utiliza pontos de corte, ele utiliza dois números aleatórios e cria os filhos de acordo com os cálculos mostrados pelas Equações 3.1 e 3.2 (MICHALEWICZ, 1999).

$$newgene1 = prob1 \cdot gene1 + (1 - prob1) \cdot gene2 \quad (3.1)$$

$$newgene2 = (1 - prob2) \cdot gene1 + prob2 \cdot gene2 \quad (3.2)$$

*newgene1* e *newgene2*: são os genes criados do 1º e do 2º filho

*prob1* e *prob2*: são duas probabilidades entre 0 e 1

*gene1* e *gene2*: são os genes correspondentes do 1º e do 2º pai

Esse calculo é feito para cada par de genes que correspondem às mesmas características dos indivíduos selecionados.

### Mutação

Quando se trata de operadores de reprodução, o *crossover* é considerado relevante, mas a mutação também desempenha um papel importante no algoritmo. A mutação é responsável por não deixar que o algoritmo fique preso em um máximo ou mínimo local, permitindo a busca em outras áreas que provavelmente não seriam visitadas pelo algoritmo. A probabilidade de acontecer a mutação é muito pequena, abaixo de 1% normalmente, e quando ocorre, consiste em alterar o valor de um ou mais genes. A Figura 3.12 ilustra um exemplo de mutação em um indivíduo binário.

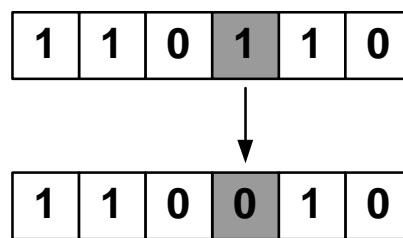


Figura 3.12: Operador de mutação.

O operador de mutação é aleatório e geralmente tem uma probabilidade fixa ao longo de toda a execução do algoritmo. Existem algoritmos que mudam esta característica da mutação (BARCELLOS, 2000). O novo operador de mutação recebe probabilidades diferentes no decorrer das gerações, como visto em Srinivas e Patnaik (1994). Isto com a finalidade de além de sair de máximos ou mínimos locais, aumentar a diversidade da população quando se percebe uma queda nessa medida. Outra modificação nesse operador foi denominada de mutação dirigida, que consiste em direcionar em quais indivíduos o operador irá atuar, essa inovação pode trazer um tempo de convergência menor para o algoritmo e também evitar altas taxas de similaridade na população, no decorrer das gerações (LI; YE, 2006).

## **3.5 Considerações Finais**

Neste capítulo foram apresentados os Algoritmos Genéticos. Abordou-se desde a sua origem, conceitos de genética, até a descrição de forma mais detalhada de cada componente que faz parte do algoritmo genético.

No próximo capítulo, mostramos a estratégia de controle proposta juntamente com o Algoritmo Genético Desenvolvido.

# Capítulo 4

## Sintonia de Controladores por Realimentação de Estados Baseada em Algoritmo Genético

Este capítulo mostra a estratégia de controle proposta neste trabalho, feita através da modificação da matriz de transição de estados, e apresenta o algoritmo genético desenvolvido.

### 4.1 Estratégia de Controle Proposta

Para controlar um sistema que está representado pelas equações no espaço de estados

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (4.1)$$

$$y = \mathbf{Cx} + Du \quad (4.2)$$

o método *LQR* pode ser utilizado a fim de obter uma resposta satisfatória. Esse método tem como estratégia de controle a realimentação de estados, dada por

$$u = -Kx \quad (4.3)$$

A minimização do índice de desempenho da Equação 2.23 poderá proporcionar ao sistema o desempenho desejado. Entretanto, as matrizes *Q* e *R* são matrizes peso determinadas

pelo usuário com o método de tentativa e erro. Neste caso, pode ser difícil o ajuste de parâmetros do controlador a fim de obter o comportamento ideal do sistema, visto que uma simples observação dos elementos das matrizes  $Q$  e  $R$  não permite obter relações com a saída do sistema.

O presente trabalho propõe uma técnica baseada em uma modificação feita na matriz de transição de estado,  $\Phi$ , com o propósito de evitar o método de tentativa e erro na definição das matrizes  $Q$  e  $R$ .

Na estratégia de controle proposta, o sistema é realimentado similarmente ao método  $LQR$ , realizado pela Equação 4.3. Essa regra de realimentação faz com que a equação de estado se transforme em:

$$\dot{x} = (A - BK)x$$

Com isso, a solução dessa nova equação de estado pode ser dada pela nova matriz de transição de estado:

$$\Phi(t) = e^{(A-BK)t} = \mathcal{L}^{-1}[(sI - A + BK)^{-1}] \quad (4.4)$$

Agora, o comportamento temporal das variáveis de estado depende da matriz de ganho de realimentação  $K$ . É através da matriz  $K$  que os autovalores de  $A - BK$  são forçados a terem os valores desejados, ou seja, com a escolha correta da matriz  $K$  o sistema possivelmente será estável com os autovalores da matriz  $A - BK$  no lado esquerdo do plano  $s$  e, se vier a acontecer distúrbios, o sistema será estabilizado à medida que  $t$  tender ao infinito.

Para um sistema de ordem  $n$  temos:

$$\Phi(t) = \begin{bmatrix} \phi_{11}(t) & \phi_{12}(t) & \cdots & \phi_{1n}(t) \\ \phi_{21}(t) & \phi_{22}(t) & \cdots & \phi_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1}(t) & \phi_{n2}(t) & \cdots & \phi_{nn}(t) \end{bmatrix}$$

Onde todos os elementos de  $\Phi$  são funções de elementos de  $K$ . Para maiores detalhes a respeito do cálculo das funções  $\Phi$  vide Apêndice B. Para um sistema com  $r$  entradas, a matriz  $K$  fica:

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n}(t) \\ k_{21} & k_{22} & \cdots & k_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ k_{r1} & k_{r2} & \cdots & k_{rn}(t) \end{bmatrix}$$

Depois de encontrada uma estratégia para o controle, seria necessário encontrar meios para a busca da matriz  $K$  ideal. O espaço de busca para elementos da matriz  $K$  é muito grande, não se tornando aplicável a utilização de algoritmos de busca exaustiva, como a busca em profundidade ou em largura, ou até técnicas informadas como  $A^*$ , pois, estes são algoritmos que percorrem todo o espaço de busca à procura da melhor resposta. Devido a estas características do problema, foi imprescindível a aplicação de um método de busca baseado em aproximação.

Para solucionar o problema de busca foram aplicados os algoritmos genéticos. AG é uma técnica evolucionária de busca global por soluções ótimas, inspirado no processo biológico de evolução natural. Geralmente a utilização de AGs se torna vantajosa em problemas com vasto espaço de busca, pois, eles não precisam percorrer todas as possíveis soluções. No AG, cada combinação de dados que se torna uma possível solução, é denominada de indivíduo. E o processo de busca é feito mediante competições entre indivíduos, sobrevivendo os mais aptos em direção à solução.

Para o controle, os bons indivíduos são aqueles que obedecem especificações do sistema, de tempo de acomodação e máximo sobre-sinal, exigidas pelo projetista. A Figura 4.1 ilustra estas especificações, a letra  $M$  mostra o máximo sobre-sinal, enquanto que  $t_s$  representa o tempo de acomodação.

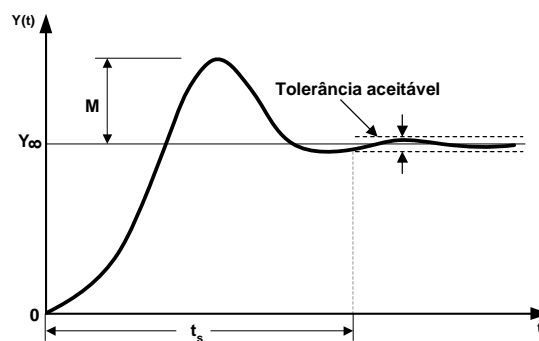


Figura 4.1: Curva de resposta a uma entrada degrau unitário que mostra  $M$  e  $t_s$

O máximo sobre-sinal é o valor máximo de pico da curva de resposta, medido a partir do valor  $Y_{\infty}$ , que é o valor final da resposta em regime permanente.

O tempo de acomodação é o tempo necessário para que a curva de resposta alcance valores de mais ou menos 2% (Tolerância aceitável), e permaneça nessa faixa indefinidamente.

## 4.2 Algoritmo Genético Desenvolvido

O AG tem a tarefa de buscar indivíduos da matriz  $k$  que satisfaçam as especificações de máximo sobre-sinal e tempo de acomodação especificados pelo projetista do sistema.

A primeira tarefa a ser realizada na implementação do algoritmo é a codificação do cromossomo, ou seja, transformar a representação real do indivíduo em uma representação computacional. Cada cromossomo do Algoritmo Genético Desenvolvido (AGD) é uma matriz  $K$ , matriz de ganho de realimentação de estado, que é uma possível solução para o sistema. Para um sistema com ordem  $n$  e um número  $r$  de entradas, cada indivíduo será uma matriz  $K$  de ordem  $r \times n$ .

Dentre as codificações possíveis a codificação por valor atende melhor as particularidades do problema, pois, os genes (valores de  $k_{rn}$ ) são formados apenas por valores reais. Optando por essa codificação algumas vantagens são adquiridas, tais como, realização de forma simples e a inexistência de conversão de dados eliminando a produção de erros que surgem nas etapas de reprodução e possíveis correções.

Após escolhido o método de codificação dos cromossomos, a população é iniciada com alguns valores previamente estabelecidos: o tamanho da população e um intervalo de valores possíveis para os genes. Uma população pequena acarreta em pouca diversidade entre os indivíduos dificultando a busca por uma boa solução. Já uma população muito extensa exige mais processamento tornando o algoritmo lento. A escolha de qual tamanho deverá ter a população exige certo equilíbrio o que também acontece para o intervalo de valores dos indivíduos.

No AGD, a população tem 30 indivíduos e o intervalo de valores para os genes é de -20 a 20. O intervalo para os genes foi escolhido com base no projeto de controle que com valores altos para matriz  $K$  torna o sistema superamortecido, mas na prática, não é conveniente a implementação.



Com a população inicial definida, o próximo passo é avaliar cada indivíduo da população e atribuir a este uma nota (*score*). A Figura 4.2 faz uma representação do funcionamento da avaliação. O ponto de partida para o AGD são as equações de estado do sistema. A função de avaliação passa as matrizes  $A, B, C$  e  $D$  para o Matlab, que com estas, calcula a matriz  $\Phi$  devolvendo-a para a função de avaliação.

Com acesso aos elementos de  $\Phi$  a função de avaliação calcula o *score* de cada indivíduo.

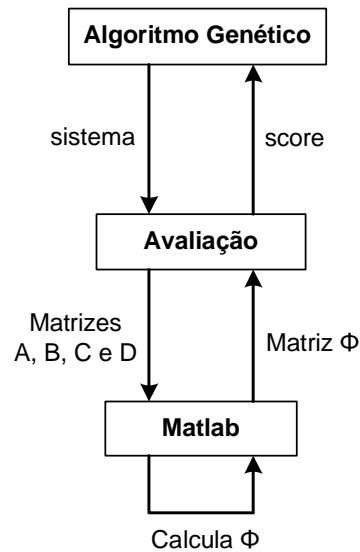


Figura 4.2: Representação da avaliação dos indivíduos

A matriz  $\Phi$ , matriz de transição de estado, gera  $n^2$  expressões para um sistema que tenha ordem  $n$ , todas em função dos elementos da matriz de ganho  $K$ . O *score* de um indivíduo é obtido a partir dos valores de tempo de acomodação e máximo sobre-sinal das expressões, após substituído os valores de seus genes (valores da matriz  $K$ ). Em cada uma das expressões o indivíduo pode receber até dois pontos. Um ponto se o valor do tempo de acomodação calculado é menor ou igual ao tempo de acomodação especificado pelo projetista, e o outro ponto na comparação entre o máximo sobre-sinal calculado e o máximo sobre-sinal especificado. Com isso, um indivíduo poderá receber até 8 pontos em um sistema de ordem 2.

Para o cálculo dessa matriz em sistemas de ordem dois a ferramenta *Matlab*<sup>®</sup> MATLAB (2008) foi utilizada. Em sistemas com ordens superiores a ferramenta apresentou os elementos da matriz  $\Phi$  de uma forma não legível pelo algoritmo genético, com isso, a ferramenta Maple 12.0 (MAPLE, 2008) foi empregada.

O próximo passo após a avaliação é a verificação feita pela função *parada* para saber se

na geração corrente existe uma solução, em outras palavras, um indivíduo que obteve a máxima pontuação. Caso encontre uma solução na população inicial, ou em qualquer geração, esta será salva em um arquivo e o algoritmo finalizado. Caso não encontre uma solução na população atual o algoritmo encaminha a população para os métodos de reprodução.

Antes de submetida aos métodos de reprodução, a população é ordenada para facilitar a escolha dos pais que serão utilizados na reprodução. Na função *ordenação* terá a melhor colocação o indivíduo que mais pontuou. Os valores individuais de tempo de acomodação e máximo sobre-sinal são utilizados para critério de desempate.

A escolha dos indivíduos para reprodução é feita utilizando dois métodos de seleção. A seleção por roleta viciada juntamente com a seleção por *ranking*. Na roleta viciada, os melhores indivíduos recebem as maiores partes da roleta, respectivamente com maiores probabilidades de serem escolhidos. O uso da roleta viciada favorece os melhores indivíduos mas não elimina os piores completamente, isso é de grande importância porque alguns indivíduos que não são tão bons podem ter características genéticas boas.

A seleção por *ranking* tem a finalidade de evitar problemas encontrados na roleta viciada, que afetam diretamente na performance do algoritmo. No caso em que uma população possui um superindivíduo, este será frequentemente escolhido para reprodução. Como resultado, grande parte da população se transforma em descendentes desse superindivíduo, diminuindo drasticamente a diversidade. Outro problema é a divisão proporcional da roleta em relação aos *scores* dos indivíduos com a existência de *scores* negativos. Para evitar que tais situações venham afetar a qualidade dos resultados do AGD, a seleção por *ranking* monta um *rank* de acordo com a ordenação dos indivíduos. O melhor indivíduo recebe uma faixa da roleta, o segundo melhor recebe uma faixa um pouco menor do que a do primeiro, e assim sucessivamente até o último indivíduo receber a menor faixa da roleta. Isso acaba com o problema do superindivíduo e com a divisão proporcional com valores negativos.

Após a seleção dos pais para a reprodução, inicia-se a função *crossover* para gerar os filhos. O *crossover* no AGD não utiliza pontos de corte, ele faz o Cruzamento Aritmético entre os pares de genes correspondentes dos dois indivíduos selecionados.

Depois do operador de *crossover* é feita a mutação. Nesta etapa, cada indivíduo tem a mesma chance de sofrer mutação, isso com uma probabilidade bem pequena em relação à probabilidade do *crossover*. A mutação no AGD segue a mesma lógica da mutação aleatória

apresentada em Linden (2006), com algumas modificações. Na mutação aleatória, o gene a sofrer mutação recebe um valor aleatório entre zero e um limite estabelecido de acordo com o problema. No AGD o operador de mutação escolhe um valor aleatório, que é somado ou diminuído do gene, entre 0 e um limite que depende da grandeza do próprio gene a sofrer mutação. Por exemplo, um gene com valor 0.3 (ou -0.3) é somado ou diminuído de um aleatório entre 0 e 1, já um gene com valor de 4.5 é somado ou diminuído de um aleatório entre 0 e 10 e assim por diante. Para saber se o valor encontrado irá ser somado ou subtraído do gene, é atribuída a probabilidade de 50% a cada uma das operações.

No AGD, foi utilizada a técnica de elitismo, com a finalidade de garantir a permanência dos  $n$  melhores indivíduos de uma população para a geração seguinte. Caso os operadores de reprodução consigam criar indivíduos melhores, os indivíduos que ocupam as  $n$  posições perdem suas posições para estes.

Após a execução das funções cruzamento e mutação os indivíduos são avaliados novamente e acontece uma verificação do critério de parada, se satisfeito, o algoritmo encerra, senão, a população é submetida novamente a reprodução e assim continua o ciclo até que se encontre um indivíduo vencedor ou que o critério de parada seja satisfeito. No AGD, o critério de parada só é satisfeito quando o algoritmo encontra uma solução. Número de gerações e tempo de execução são outros exemplos para o critério de parada.

### 4.3 Considerações Finais

Neste capítulo foi apresentada a estratégia de controle utilizada no presente trabalho, que tem a finalidade de evitar o método de tentativa e erro feita pelo usuário na definição das matrizes peso quando se utiliza o método *LQR*. Mostrou-se que para a nova estratégia é necessário encontrar uma matriz  $K$  ideal para satisfazer as especificações do projetista. E por fim o Algoritmo Genético Desenvolvido que tem como principal objetivo, encontrar a matriz  $K$  para o sistema desejado.

O próximo capítulo mostra os resultados obtidos para vários sistemas com o método AGD, compara-os com o método *LQR* e com o sistema sem controle.

# Capítulo 5

## Resultados

Para mostrar os resultados do presente trabalho, simulações digitais foram feitas utilizando a ferramenta Matlab a fim de averiguar e comparar as melhorias de desempenho obtidas com o Algoritmo Genético Proposto e o método *LQR*. Nas simulações e na execução do AG foram utilizados dois computadores com sistema operacional Windows XP 32 bits com processador Intel® Core™ 2 Duo 1.80GHz. Este capítulo apresenta os resultados de desempenho obtidos para vários sistemas.

### 5.1 Sistemas de ordem 2

Dois sistemas de ordem dois foram analisados.

**Sistema 1:**

$$\dot{x} = \begin{bmatrix} -1 & -1 \\ 6.5 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

**Sistema 2:**

$$\dot{x} = \begin{bmatrix} 0 & 0.9 \\ -1 & -0.03 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

#### 5.1.1 Resultados do sistema 1

Os sistemas apresentados estão na forma de equação de estado, representados pelas matrizes A, B, C e D.

Ver Apêndice B, a obtenção das funções Fitness para este sistema. Para os demais sistemas aqui avaliados, o método de obtenção das funções de Fitness é o mesmo.

$$A = \begin{bmatrix} -1 & -1 \\ 6.5 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad e \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

A partir destas matrizes, o Algoritmo Genético Desenvolvido, através da Equação (4.4), encontrou valores da matriz  $K$  que satisfizeram as especificações de máximo sobre-sinal e tempo de acomodação. Estes valores, juntamente com os valores produzidos pelo método  $LQR$ , são apresentados na Tabela 5.1.

Tabela 5.1: Dados do projeto de controle do Sistema 1.

<b>Parâmetros do controlador</b>	
AGD	$K = \begin{bmatrix} -6.228714 & 8.673320 \end{bmatrix}$
$LQR$	$K = \begin{bmatrix} -3.4198 & 8.0259 \end{bmatrix}$

As especificações utilizadas para o AGD foram  $M = 5\%$  e  $Ts = 2seg$ . A solução foi encontrada na geração 11. Para calcular a matriz  $K$  pelo método  $LQR$  foram utilizadas as matrizes peso:

$$Q = \begin{bmatrix} 22 & 0 \\ 0 & 5.7 \end{bmatrix} \quad e \quad R = \begin{bmatrix} 0.099 \end{bmatrix}$$

Como o método  $LQR$  consiste em um processo de tentativa e erro, a busca pelo ganho  $K$  foi feita até os autovalores da matriz  $A - BK$  serem o mais próximo possível dos autovalores obtidos pelo AGD. Os autovalores são apresentados na Tabela 5.2.

Tabela 5.2: Autovalores do Sistema 1.

<b>Autovalores</b>	
AGD	$\lambda_1 = -3.4255$ e $\lambda_2 = -6.2478$
$LQR$	$\lambda_1 = -2.9570$ e $\lambda_2 = -6.0689$

A Figura 5.1 ilustra a comparação entre o AGD e o método  $LQR$  para as funções custo (elementos da matriz  $\Phi$ ). Além disto, a resposta livre do sistema original, sem controle, também é mostrada.

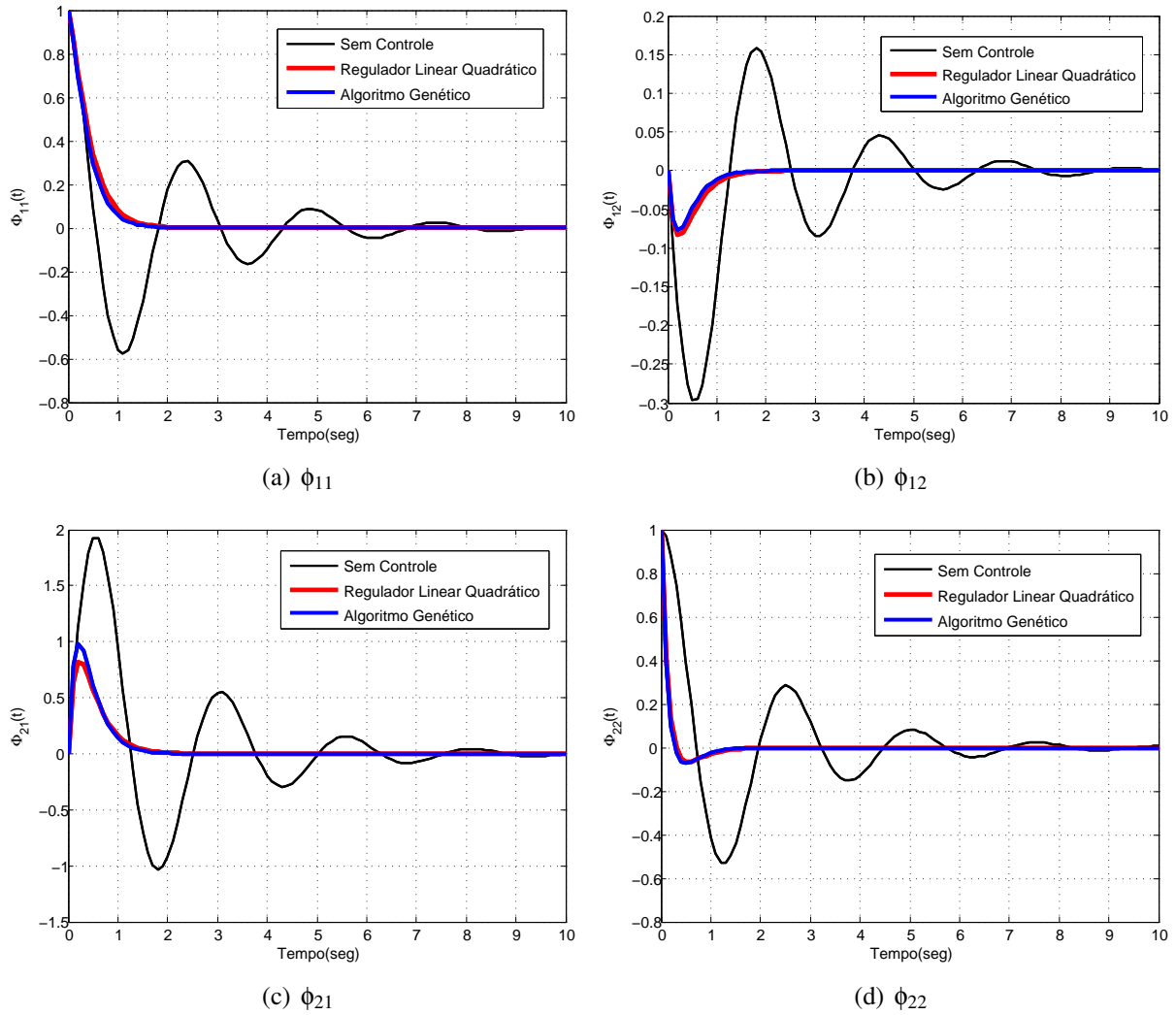


Figura 5.1: Comparação dos resultados para  $\phi_{11}$ ,  $\phi_{12}$ ,  $\phi_{21}$  e  $\phi_{22}$  - Sistema 1.

Estes resultados mostram que o AGD apresenta amortecimento satisfatório das oscilações. Uma vez que para este sistema, temos

$$x_1(t) = \phi_{11}(t)x_1(0) + \phi_{12}(t)x_2(0)$$

$$x_2(t) = \phi_{21}(t)x_1(0) + \phi_{22}(t)x_2(0)$$

Em que  $x_1(0)$  e  $x_2(0)$  são condições iniciais para as variáveis de estado, as respostas das variáveis de estado dependem do comportamento das funções  $\phi$ . O comportamento das variáveis de estado para o AGD, para o *LQR* e para o sistema sem controle são mostrados nas Figuras 5.2 e 5.3.

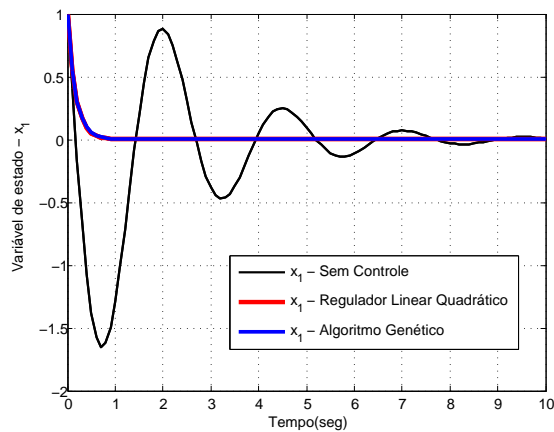


Figura 5.2: Resposta livre para  $x_1$  nas condições iniciais  $x_1(0) = 1$  e  $x_2(0) = 5$  - Sistema 1.

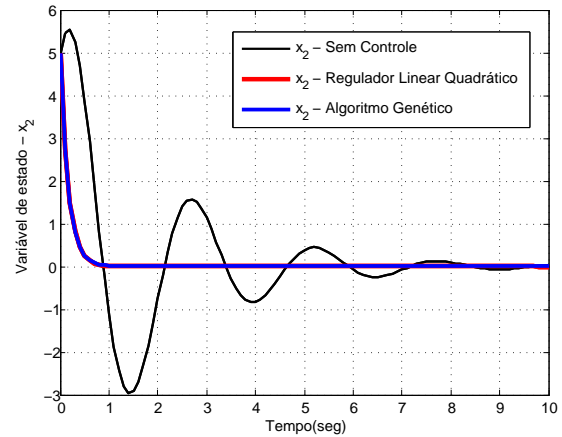


Figura 5.3: Resposta livre para  $x_2$  nas condições iniciais  $x_1(0) = 1$  e  $x_2(0) = 5$  - Sistema 1.

### 5.1.2 Resultados do sistema 2

As matrizes da Equação de Estado para o Sistema 2 são:

$$A = \begin{bmatrix} 0 & 0.9 \\ -1 & -0.03 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad e \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

Para o Sistema 2, os elementos de  $K$  obtidos pelo AGD e pelo método  $LQR$  são mostrados na Tabela 5.3

Tabela 5.3: Dados do projeto de controle do Sistema 2.

Parâmetros do controlador	
AGD	$K = \begin{bmatrix} 8.623016 & 10.824402 \end{bmatrix}$
$LQR$	$K = \begin{bmatrix} 9.0499 & 10.7538 \end{bmatrix}$

As especificações utilizadas para o AGD foram  $M = 10\%$ , e  $T_s = 5\text{seg}$ , a solução foi encontrada na geração 34. Para encontrar os elementos da matriz  $K$  pelo método  $LQR$  foram utilizadas as matrizes peso:

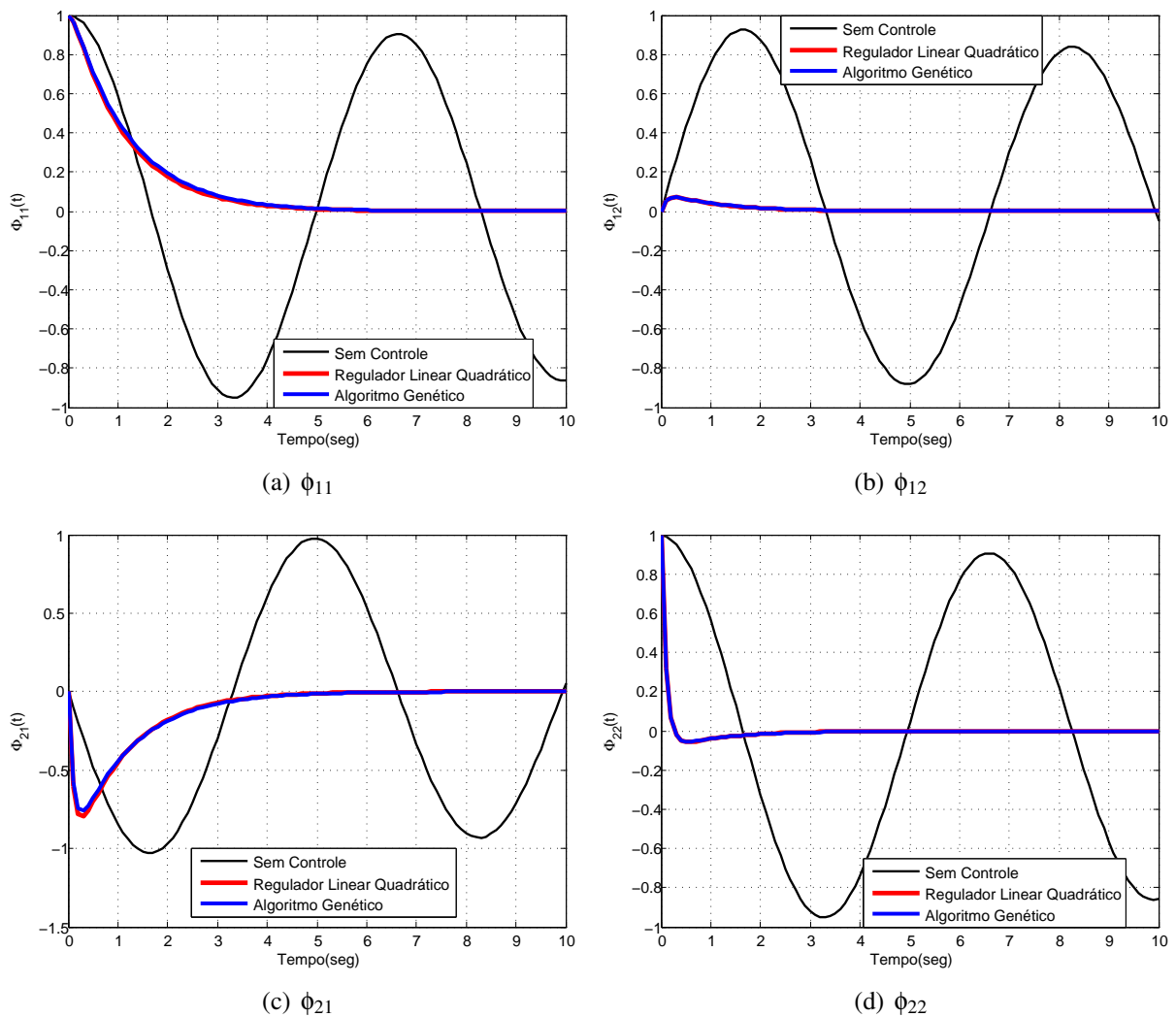
$$Q = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad e \quad R = \begin{bmatrix} 0.05 \end{bmatrix}$$

Os autovalores são mostrados da Tabela 5.4.

Tabela 5.4: Autovalores do sistema 2.

Autovalores	
AGD	$\lambda_1 = -0.8672$ e $\lambda_2 = -9.9872$
LQR	$\lambda_1 = -0.9167$ e $\lambda_2 = -9.8671$

A Figura 5.4 ilustra a comparação entre AGD e LQR para as funções custo. Além disto, o resultado para o sistema original, sem controle, é mostrado.

Figura 5.4: Comparação dos resultados para  $\phi_{11}$ ,  $\phi_{12}$ ,  $\phi_{21}$  e  $\phi_{22}$  - Sistema 2.

Novamente, o comportamento das funções custo é melhor para ambas as técnicas de controle. Os efeitos da ação de controle sobre o comportamento de variáveis de estado no tempo é mostrado nas Figuras 5.5 e 5.6.



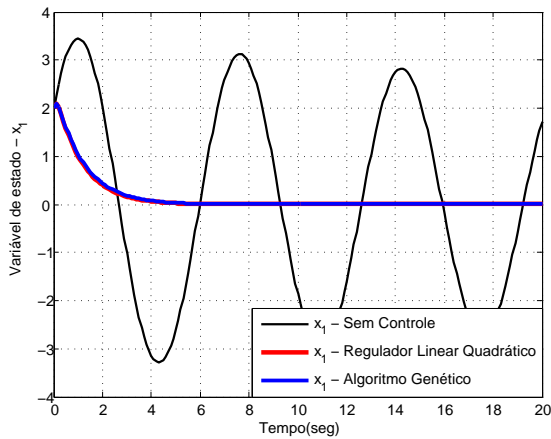


Figura 5.5: Resposta livre para  $x_1$  nas condições iniciais  $x_1(0) = 2$  e  $x_2(0) = 3$  - Sistema 2.

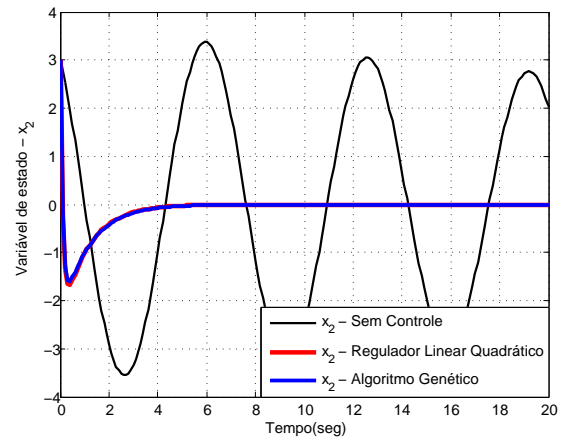


Figura 5.6: Resposta livre para  $x_2$  nas condições iniciais  $x_1(0) = 2$  e  $x_2(0) = 3$  - Sistema 2.

## 5.2 Sistemas de ordem 3

Dois sistemas de ordem três foram analisados.

### Sistema 3:

$$\dot{x} = \begin{bmatrix} -5 & 10 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x$$

### Sistema 4:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -10 & -5 & -6 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x$$

### 5.2.1 Resultados do sistema 3

Analisando o Sistema 3, podemos notar que as matrizes da Equação de Estado são:

$$A = \begin{bmatrix} -5 & 10 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad e \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

Para o Sistema 3, os elementos de  $K$  obtidos pelo AGD e pelo método  $LQR$  são mostrados na Tabela 5.5

Tabela 5.5: Dados do projeto de controle do Sistema 3.

<b>Parâmetros do controlador</b>	
AGD	$K = \begin{bmatrix} 1.415916 & 12.962830 & 0.082893 \end{bmatrix}$
$LQR$	$K = \begin{bmatrix} 4.9447 & 12.5088 & 0.1199 \end{bmatrix}$

As especificações utilizadas para o AGD foram  $M = 20\%$  e  $T_s = 8\text{seg}$ , e a solução foi encontrada na geração 170. Para encontrar os elementos da matriz  $K$  pelo método  $LQR$  foram utilizadas as matrizes peso:

$$Q = \begin{bmatrix} 11 & 3 & 2 \\ -3 & 5.7 & 1 \\ -2 & 2 & 1 \end{bmatrix} \quad e \quad R = \begin{bmatrix} 0.099 \end{bmatrix}$$

Os autovalores são mostrados da Tabela 5.6.

Tabela 5.6: Autovalores do Sistema 3.

<b>Autovalores</b>		
AGD	$\lambda_1 = -10.6588$	$\lambda_2 = -7.1209$
		$\lambda_3 = -1.1832$
$LQR$	$\lambda_1 = -8.6946 + 5.8681i$	$\lambda_2 = -8.6946 - 5.8681i$
		$\lambda_3 = -1.1196$

As Figuras 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 e 5.15 ilustram a comparação entre AGD e  $LQR$  para as funções custo. O resultado para o sistema sem controle também é apresentado. Para cada elemento  $\phi_{ij}$  é ilustrada uma figura para o módulo desta função, e outra figura para a fase. A divisão em módulo e fase é devido ao aparecimento de coeficientes e/ou expoentes complexos nestas funções. Os comportamentos das funções custo são melhores para ambas as técnicas de controle. Os efeitos da ação de controle sobre o comportamento de variáveis de estado, no tempo, são mostrados nas Figuras 5.16, 5.17 e 5.18

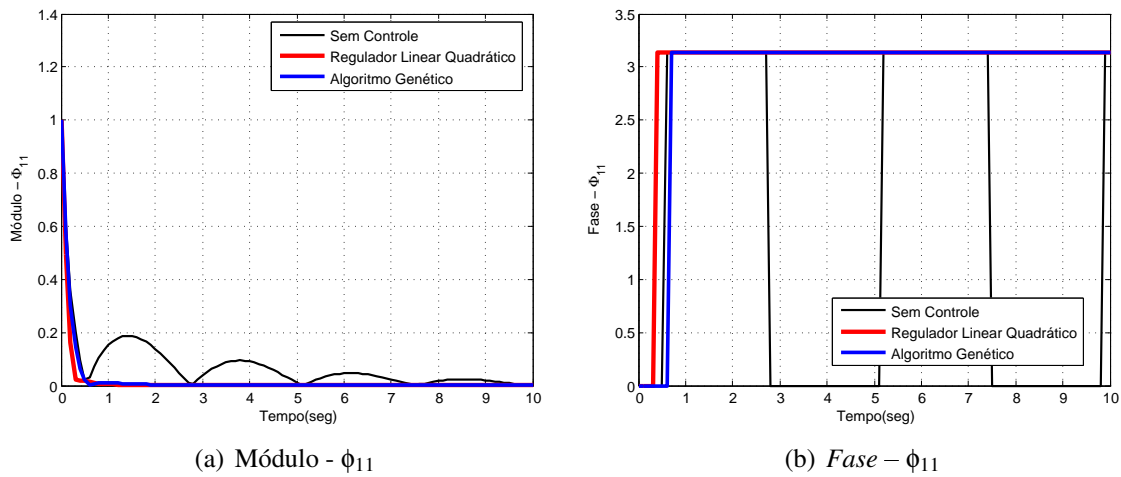


Figura 5.7: Comparação dos resultados de módulo e fase para  $\phi_{11}$  - Sistema 3.

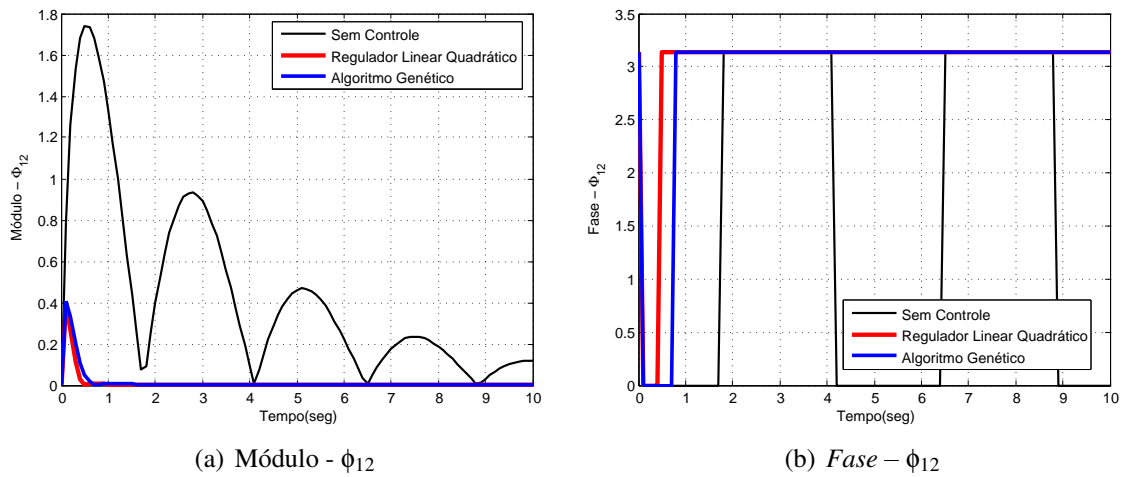


Figura 5.8: Comparação dos resultados de módulo e fase para  $\phi_{12}$  - Sistema 3.

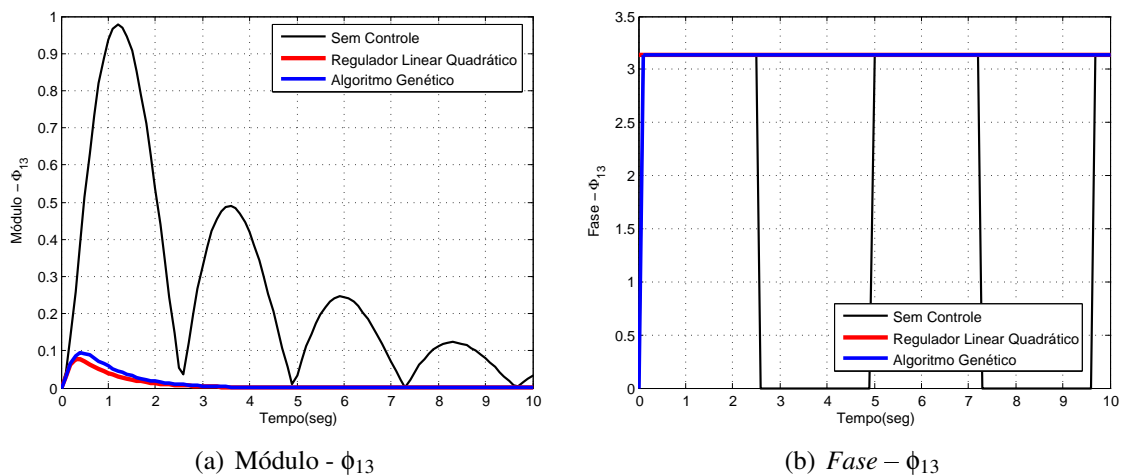


Figura 5.9: Comparação dos resultados de módulo e fase para  $\phi_{13}$  - Sistema 3.

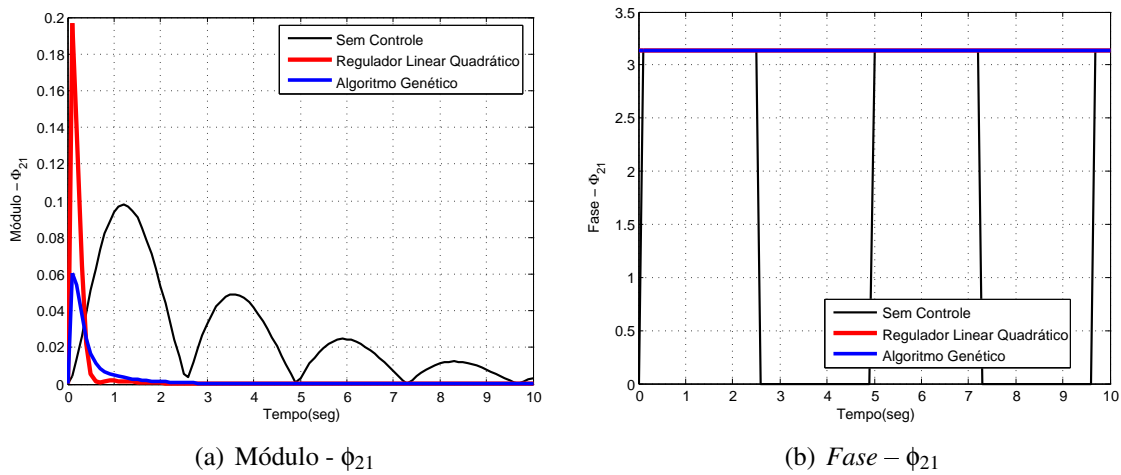


Figura 5.10: Comparação dos resultados de módulo e fase para  $\phi_{21}$  - Sistema 3.

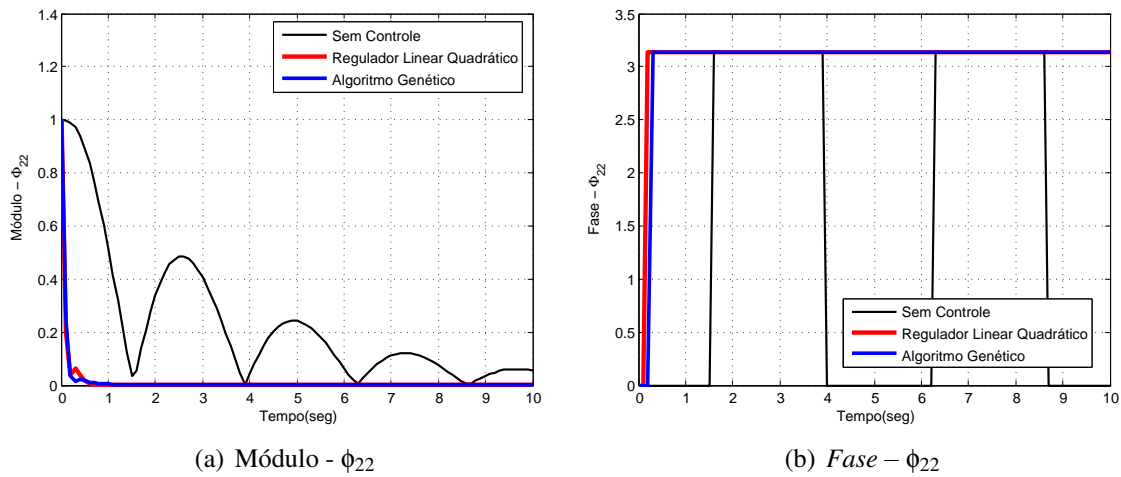


Figura 5.11: Comparação dos resultados de módulo e fase para  $\phi_{22}$  - Sistema 3.

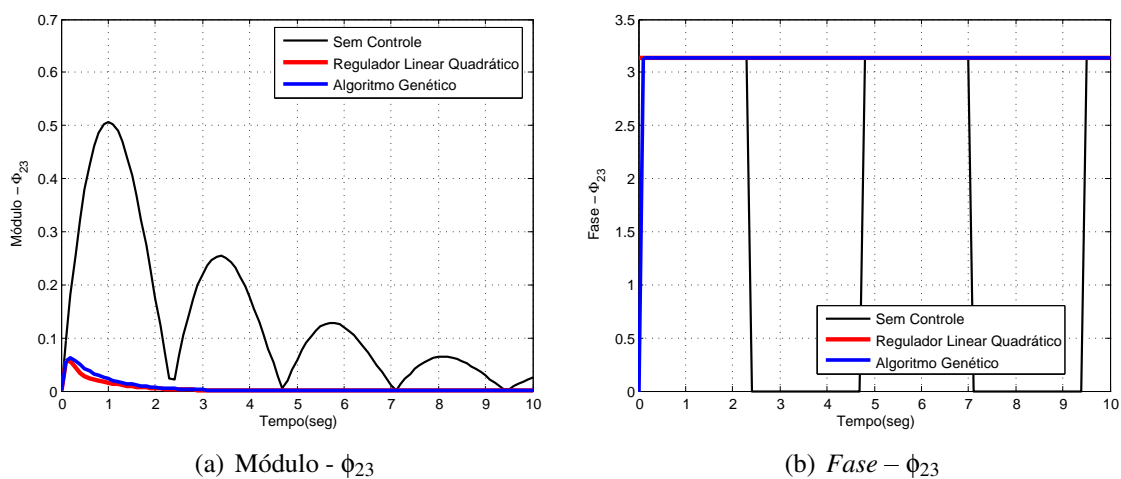
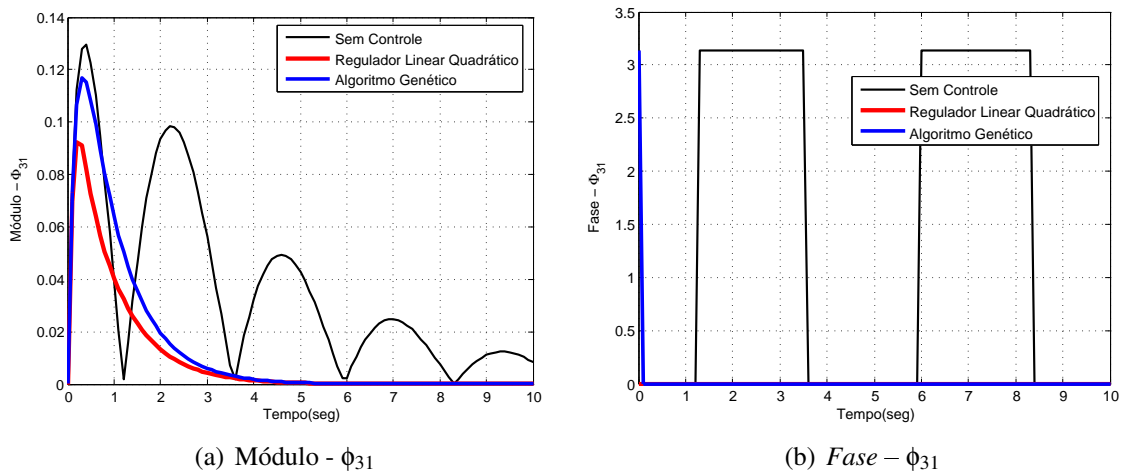
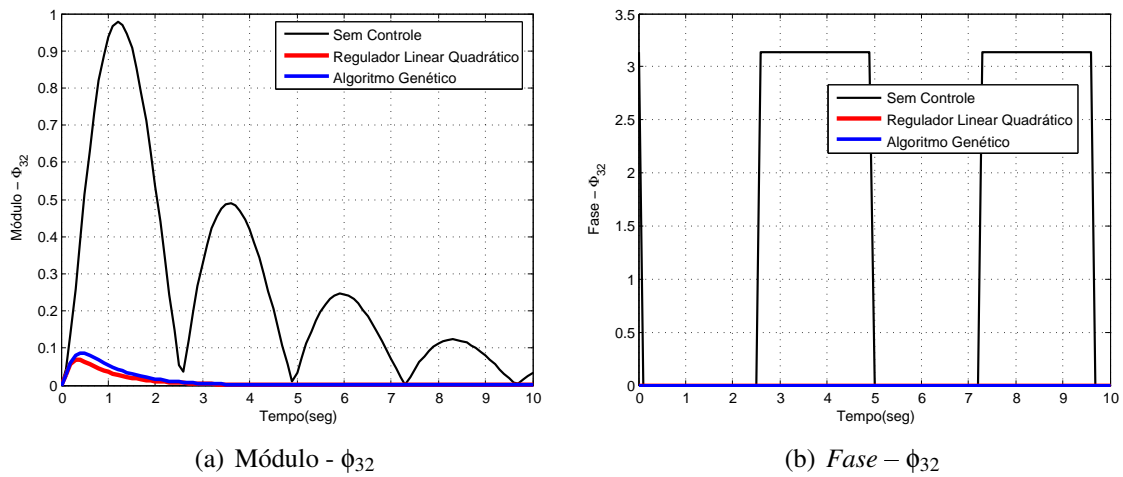
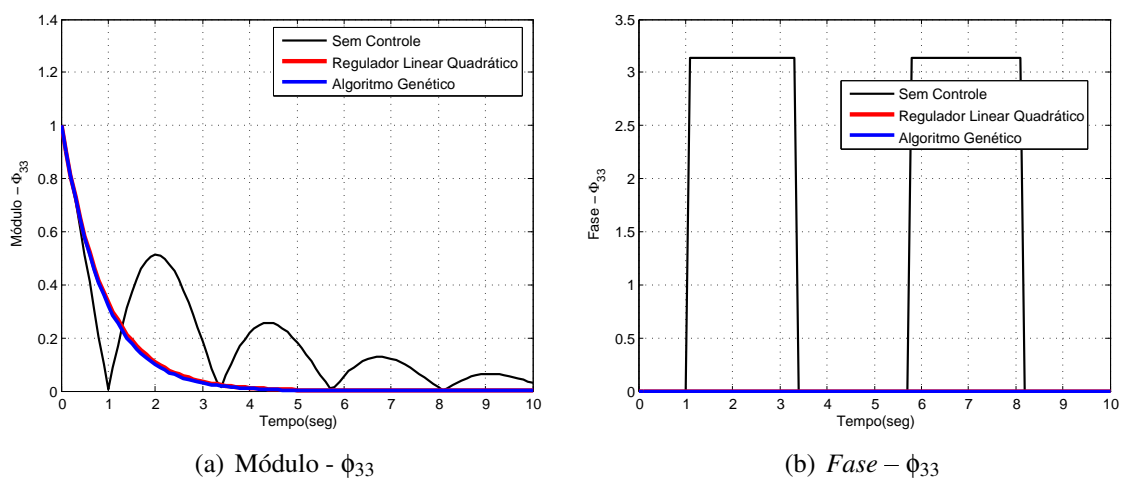


Figura 5.12: Comparação dos resultados de módulo e fase para  $\phi_{23}$  - Sistema 3.

Figura 5.13: Comparação dos resultados de módulo e fase para  $\phi_{31}$  - Sistema 3.Figura 5.14: Comparação dos resultados de módulo e fase para  $\phi_{32}$  - Sistema 3.Figura 5.15: Comparação dos resultados de módulo e fase para  $\phi_{33}$  - Sistema 3.

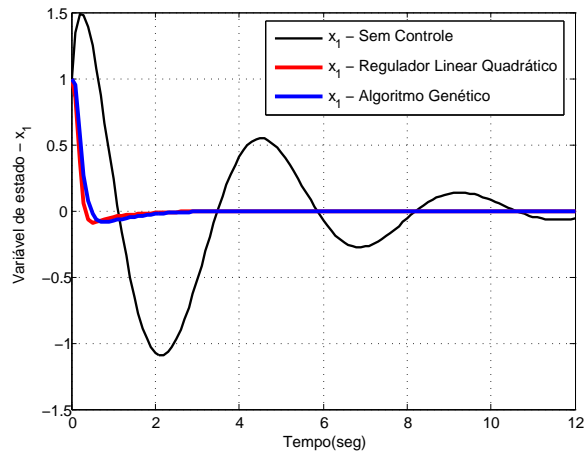


Figura 5.16: Resposta livre para  $x_1$  nas condições iniciais  $x_1(0) = 1$ ,  $x_2(0) = 1$  e  $x_3(0) = 1$  - Sistema 3.

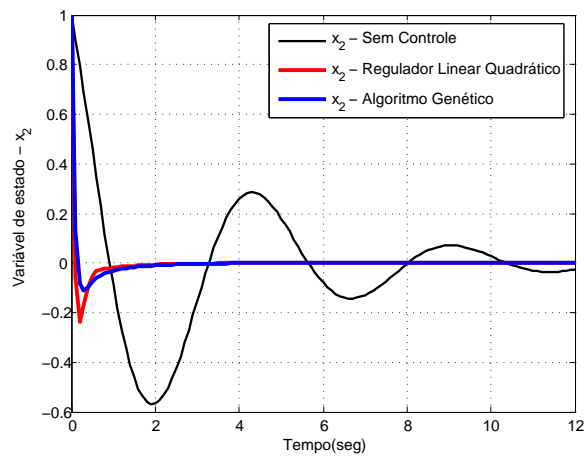


Figura 5.17: Resposta livre para  $x_2$  nas condições iniciais  $x_1(0) = 1$ ,  $x_2(0) = 1$  e  $x_3(0) = 1$  - Sistema 3.

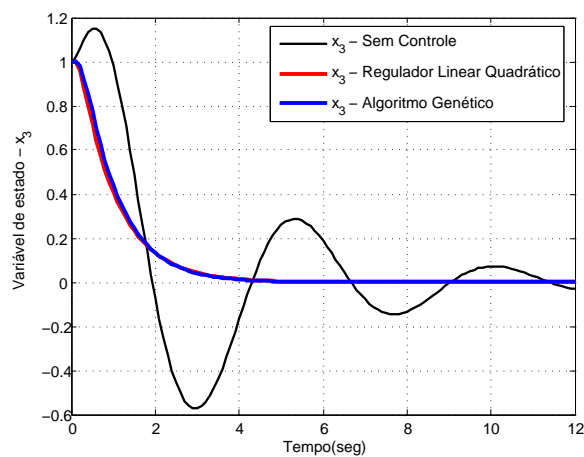


Figura 5.18: Resposta livre para  $x_3$  nas condições iniciais  $x_1(0) = 1$ ,  $x_2(0) = 1$  e  $x_3(0) = 1$  - Sistema 3.

### 5.2.2 Resultados do sistema 4

Analisando o Sistema 4, podemos notar que as matrizes da Equação de Estado são:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -10 & -5 & -6 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = [1 \ 0 \ 0] \quad e \quad D = [0]$$

Para o Sistema 4, os elementos de  $K$  obtidos pelo AGD e pelo método  $LQR$  são mostrados na Tabela 5.7.

Tabela 5.7: Dados do projeto de controle do Sistema 4.

<b>Parâmetros do controlador</b>	
AGD	$K = [1.763673 \ 8.486381 \ 6.559724]$
$LQR$	$K = [1.6316 \ 8.7066 \ 5.9016]$

As especificações utilizadas para o AGD foram  $M = 20\%$  e  $T_s = 8\text{seg}$ , e a solução foi encontrada na geração 13. Para encontrar os elementos da matriz  $K$  pelo método  $LQR$  foram utilizadas as matrizes peso:

$$Q = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 0.02 & 0 \\ -1 & 0 & 15 \end{bmatrix} \quad e \quad R = [0.17]$$

Os autovalores são mostrados da Tabela 5.8.

Tabela 5.8: Autovalores do sistema 4.

<b>Autovalores</b>		
AGD	$\lambda_1 = -11.4737$	$\lambda_2 = -0.5430 + 0.8546i$ $\lambda_3 = -0.5430 - 0.8546i$
$LQR$	$\lambda_1 = -10.7247$	$\lambda_2 = -0.5885 + 0.8592i$ $\lambda_3 = -0.5885 - 0.8592i$

As Figuras 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.25, 5.26 e 5.27 ilustram a comparação entre o AGD e  $LQR$  para as funções custo. O resultado para o sistema sem controle também é apresentado. O comportamento das funções custo é melhor para ambas as técnicas de controle. Os efeitos da ação de controle sobre o comportamento de variáveis de estado no tempo é mostrado nas Figuras 5.28, 5.29 e 5.30.

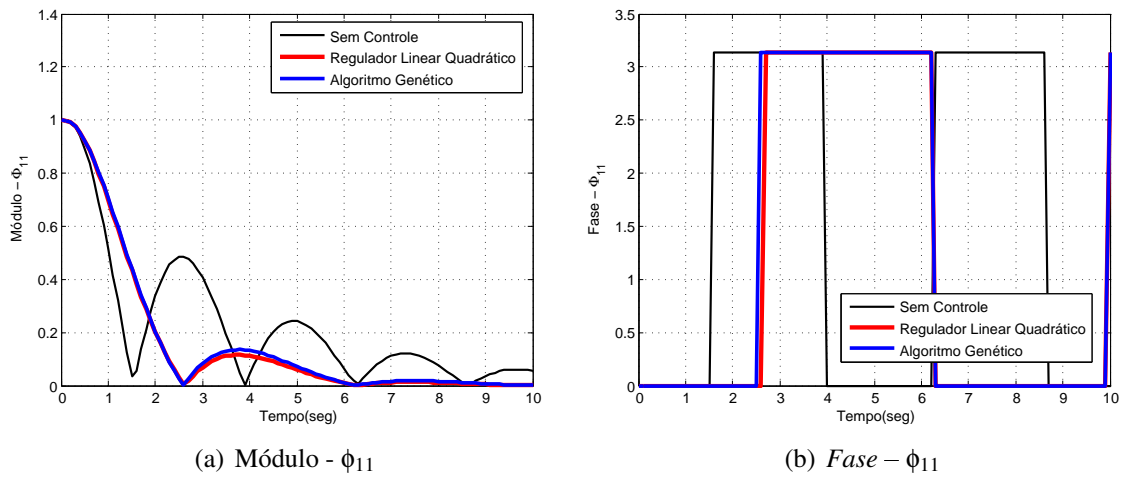


Figura 5.19: Comparação dos resultados de módulo e fase para  $\phi_{11}$  - Sistema 4.

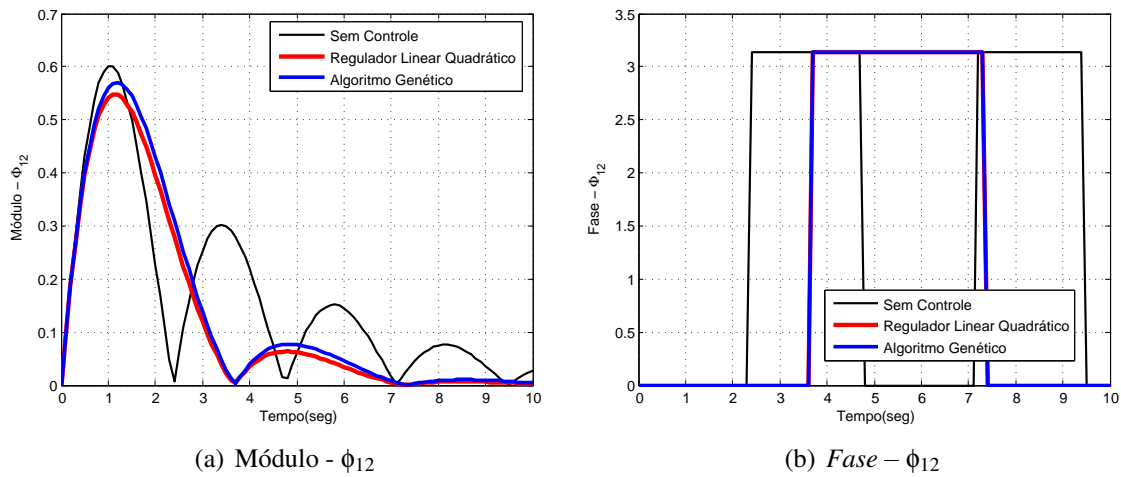


Figura 5.20: Comparação dos resultados de módulo e fase para  $\phi_{12}$  - Sistema 4.

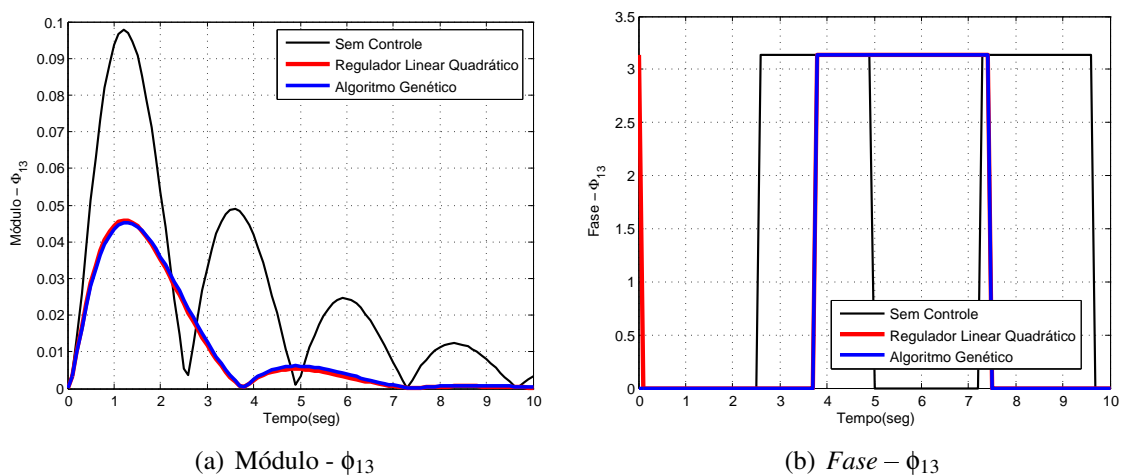


Figura 5.21: Comparação dos resultados de módulo e fase para  $\phi_{13}$  - Sistema 4.



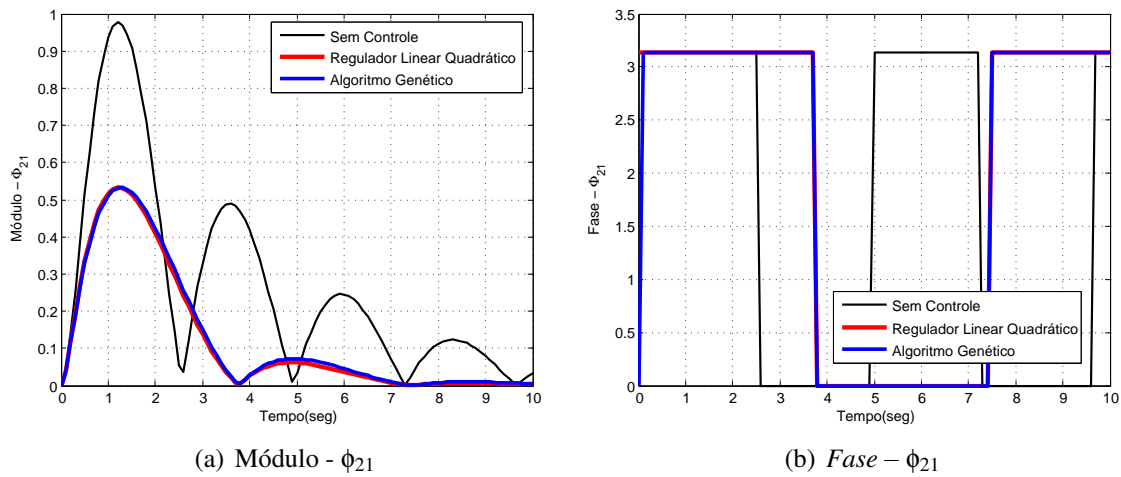


Figura 5.22: Comparação dos resultados de módulo e fase para  $\phi_{21}$  - Sistema 4.

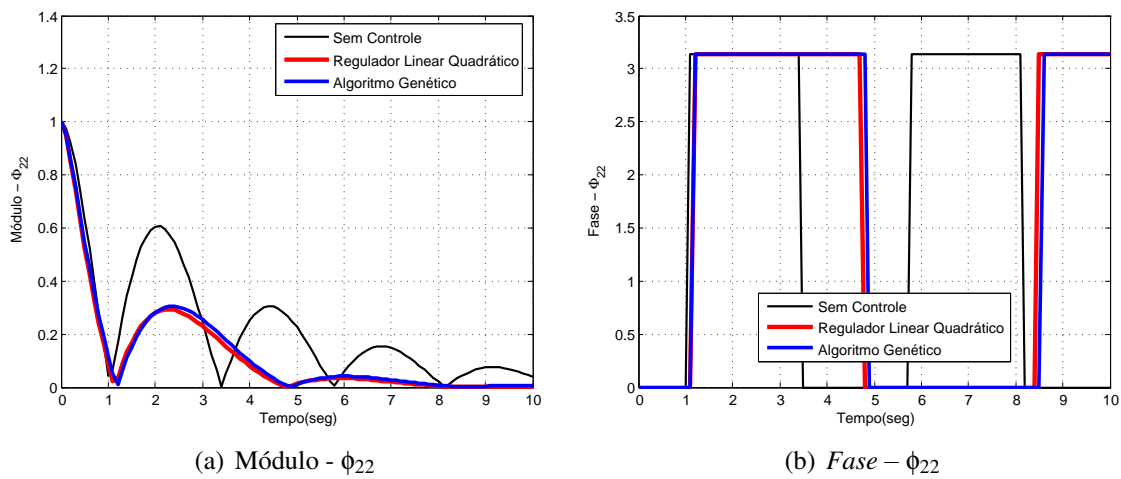


Figura 5.23: Comparação dos resultados de módulo e fase para  $\phi_{22}$  - Sistema 4.

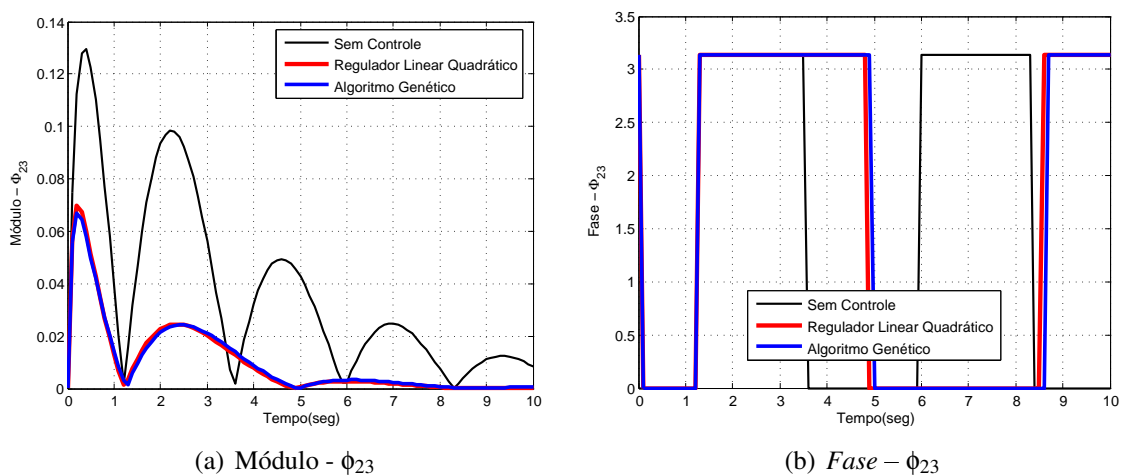


Figura 5.24: Comparação dos resultados de módulo e fase para  $\phi_{23}$  - Sistema 4.

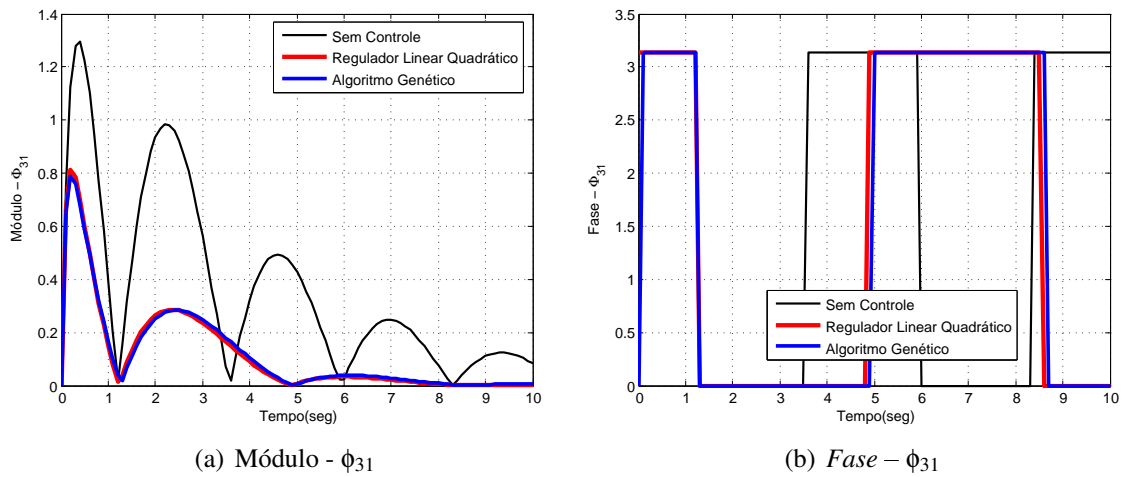


Figura 5.25: Comparação dos resultados de módulo e fase para  $\phi_{31}$  - Sistema 4.

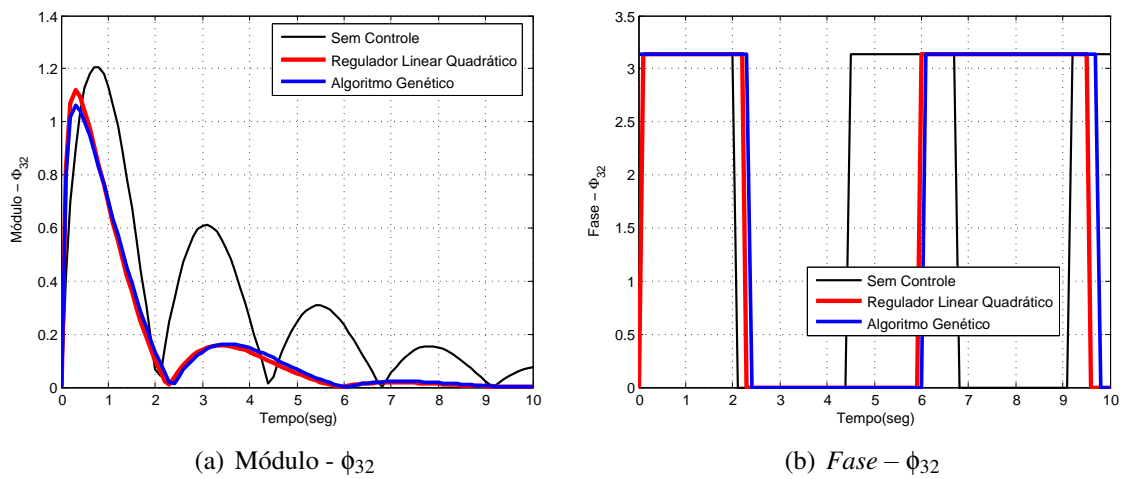


Figura 5.26: Comparação dos resultados de módulo e fase para  $\phi_{32}$  - Sistema 4.

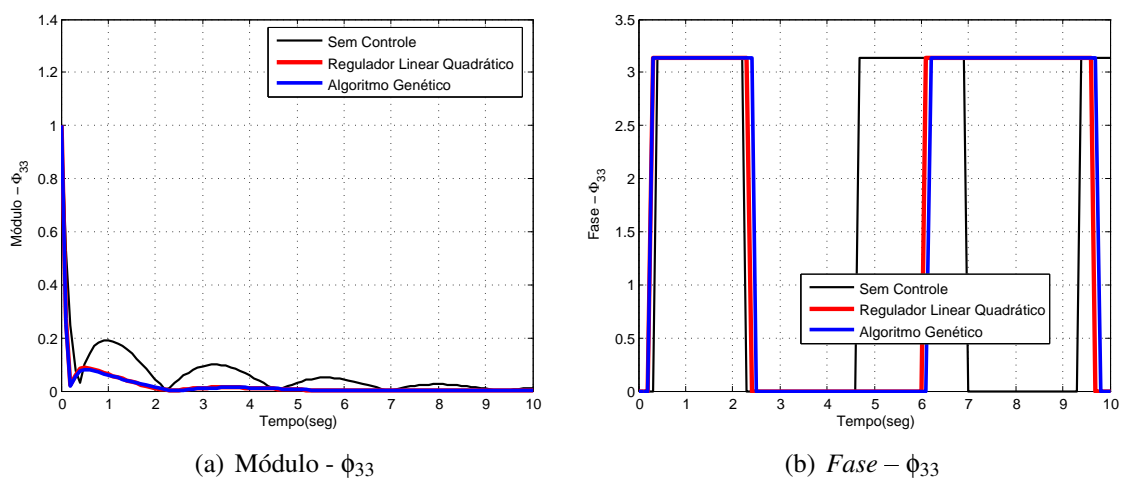


Figura 5.27: Comparação dos resultados de módulo e fase para  $\phi_{33}$  - Sistema 4.

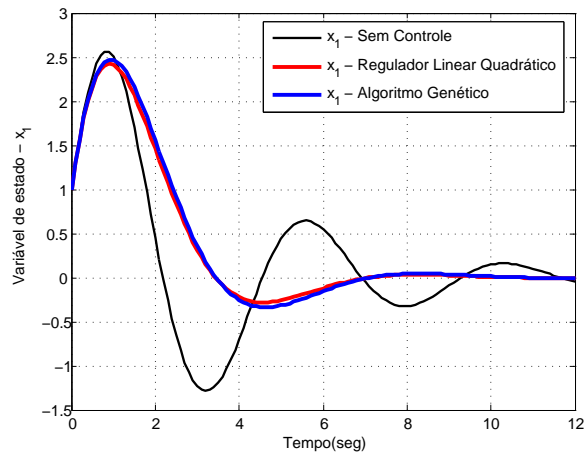


Figura 5.28: Resposta livre para  $x_1$  nas condições iniciais  $x_1(0) = 1$ ,  $x_2(0) = 3$  e  $x_3(0) = 2$  - Sistema 4.

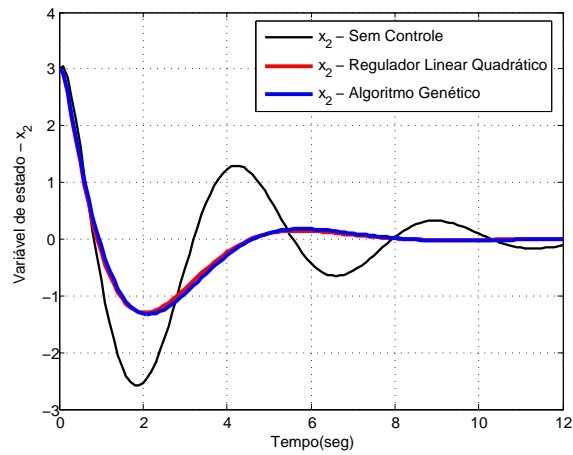


Figura 5.29: Resposta livre para  $x_2$  nas condições iniciais  $x_1(0) = 1$ ,  $x_2(0) = 3$  e  $x_3(0) = 2$  - Sistema 4.

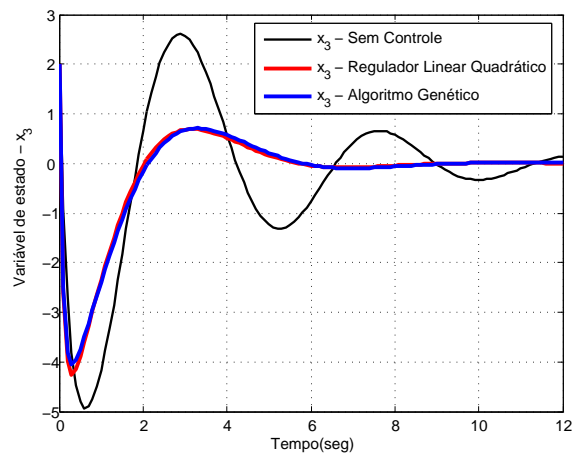


Figura 5.30: Resposta livre para  $x_3$  nas condições iniciais  $x_1(0) = 1$ ,  $x_2(0) = 3$  e  $x_3(0) = 2$  - Sistema 4.

## 5.3 Considerações Finais

Neste capítulo, foram apresentados os resultados obtidos pelo AGD em comparação com os resultados do método *LQR* e do sistema sem controle para dois sistemas de ordem 2 e dois sistemas de ordem 3. Em todos os sistemas apresentados foi possível observar que as respostas do AGD melhora o sistema original, com a garantia de atender especificações de máximo sobre-sinal e tempo de acomodação exigidas pelo projetista do sistema.

No próximo capítulo, serão expostas as conclusões a respeito da nova estratégia de controle proposta e sugeridas algumas possibilidades de trabalhos futuros.

# Capítulo 6

## Conclusões

### 6.1 Conclusões

Neste trabalho, é apresentado um método que utiliza a técnica de algoritmo genético aplicado ao projeto de controle por realimentação de estados. A estratégia de controle consiste na modificação da matriz de transição de estado para o sistema realimentado. Este método foi proposto para contornar a dificuldade no processo de projeto de controle quando o método *LQR* é usado. Esta dificuldade consiste na definição de matrizes de peso, uma vez que estas matrizes não apresentam uma relação bem definida com a resposta do sistema controlado. Ambas as técnicas foram aplicadas em vários sistemas e seus desempenhos comparados.

De acordo com os resultados obtidos, observa-se que o método proposto apresentou desempenho satisfatório, equivalente ao *LQR*. Entretanto, apresentou a garantia da resposta satisfazendo especificações de máximo sobre-sinal e tempo de acomodação, uma vez que o método proposto tem o projeto baseado nestas especificações da resposta no tempo. O método *LQR* apresentou desempenho semelhante ao AGD porque utilizou os resultados do AGD para se parametrizar, pois, sozinho, sem uma referência, o *LQR* não apresentaria o mesmo desempenho.

No AGD o projetista pode alterar as especificações a fim de alterar os requisitos, para ajustar o tempo que o sistema retorna ao seu estado inicial e o máximo sobre-sinal tolerado em casos de distúrbios. É importante ressaltar que para um tempo de amortecimento mais rápido, a energia requerida pelo controlador é maior.

## 6.2 **Trabalhos Futuros**

Como proposta para continuação do presente trabalho, seguem algumas alternativas.

- Diagonalização da matriz de transição de estado,  $\Phi$ , para diminuir o processamento realizado pelo algoritmo. Para um sistema de ordem  $n$  é necessário otimizar  $n^2$  funções  $\phi$ , já com a diagonalização seria necessário otimizar apenas  $n$  funções  $\phi$ .
- Generalizar o código do AG para que seja possível encontrar os parâmetros de otimização para sistemas de várias ordens com o mesmo algoritmo.
- Aplicação das técnicas de Algoritmos Genéticos Adaptativos para que o Algoritmo utilize seus parâmetros de acordo com cada sistema especificado.

# Referências Bibliográficas

ABDULAL, W. et al. An improved rank-based genetic algorithm with limited iterations for grid scheduling. In: *Industrial Electronics Applications, 2009. ISIEA 2009. IEEE Symposium on*. [S.l.: s.n.], 2009. v. 1, p. 215 –220.

AL-HAJRI, M.; ABIDO, M. Assessment of genetic algorithm selection, crossover and mutation techniques in reactive power optimization. In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*. [S.l.: s.n.], 2009. p. 1005 –1011.

BARCELLOS, J. C. H. de. *Algoritmos Genético Adaptativos: Um estudo comparativo*. Dissertação (Mestrado) — Universidade de São Paulo, São Paulo, 2000.

BROGAN, W. L. *Modern Control Theory*. 3ª edition. ed. Englewood Cliffs, New Jersey: Prentice Hall, 1991.

CHAKRABORTY, M.; CHAKRABORTY, U. An analysis of linear ranking and binary tournament selection in genetic algorithms. In: *Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on*. [S.l.: s.n.], 1997. v. 1, p. 407 –411 vol.1.

CHAMAKEN, A.; LITZ, L. Joint design of control and communication in wireless networked control systems: A case study. In: *American Control Conference (ACC), 2010*. [S.l.: s.n.], 2010.

DORF, R. C.; BISHOP, R. H. *Sistemas de Controle Modernos*. 11. ed. [S.l.]: LTC Editora, 2009. ISBN 9788521617143.

FRANKLIN, J. D. P. G. F.; EMAMI-NAEINI, A. *Feedback Control of Dynamic Systems*. 5. ed. New Jersey: Addison-Wesley, 2006.

- GUO-SHENG, H. et al. Parameters selection of fitness scaling in genetic algorithm and its application. In: *Control and Decision Conference (CCDC), 2010 Chinese*. [S.l.: s.n.], 2010. p. 2475 –2480.
- HAUPT, R. L.; HAUPT, S. E. *Practical Genetic Algorithms*. 2<sup>a</sup>. ed. Hoboken, New Jersey: Wiley, 1998.
- JIAN-CHENG, A.; HAI-JUAN, J.; CHAOHUI, L. Improved real-coding genetic algorithm. In: *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on*. [S.l.: s.n.], 2009. v. 2, p. 696 –698.
- JONG, K. A. D. *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Dissertação (Mestrado) — University of Michigan, Michigan, USA, 1975.
- JUNHUA, L.; MING, L. Genetic algorithm with dual species. In: *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*. [S.l.: s.n.], 2008. p. 2572 –2575.
- KAUR, D.; MURUGAPPAN, M. Performance enhancement in solving traveling salesman problem using hybrid genetic algorithm. In: *Fuzzy Information Processing Society, 2008. NAFIPS 2008. Annual Meeting of the North American*. [S.l.: s.n.], 2008. p. 1 –6.
- KROMER, P.; PLATOS, J.; SNASEL, V. Modeling permutations for genetic algorithms. In: *Soft Computing and Pattern Recognition, 2009. SOCPAR '09. International Conference of*. [S.l.: s.n.], 2009. p. 100 –105.
- KUNDUR, P. *Power System Stability and Control*. Mc.Graw Hill: Book, 1994.
- KWAKERNAAK, H.; SIVAN, R. *Linear Optimal Control Systems*. United States of America: John Wiley Sons, Inc, 1972.
- LANDAU I. D., L. R.; M'SAAD, M. *Adaptive Control*. London, Great Britain: Springer - Verlag, 1998.
- LI, N.; YE, F. Optimal design of discrete structure with directed mutation genetic algorithms. In: *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*. [S.l.: s.n.], 2006.



LIANG, Y.; LEUNG, K.-S.; LEE, K.-H. A novel binary variable representation for genetic and evolutionary algorithms. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. [S.l.: s.n.], 2006.

LINDEN, R. *Algoritmos Genéticos: Uma Importante Ferramenta da Inteligência Computacional*. Rio de Janeiro: Brasport, 2006.

LIU, Z.; ZHOU, J.; LAI, S. New adaptive genetic algorithm based on ranking. In: *Machine Learning and Cybernetics, 2003 International Conference on*. [S.l.: s.n.], 2003. v. 3, p. 1841 – 1844 Vol.3.

MAPLE. *The Essential Tool for Mathematics and Modeling*. [S.l.]: MapleSoft, 2008.

MATLAB. *The Language of Technical Computing*. [S.l.]: The Mathworks, 2008.

MELANIE, M. *An Introduction to Genetic Algorithm*. Cambridge, Massachusetts: Bradford Book, 1999.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Third, revised and extended edition,. [S.l.]: Springer-Verlag, 1999.

MITCHELL, M. *An Introduction to Genetic Algorithms*. [S.l.]: MIT Press, 1996.

NISE, N. S. *Engenharia de Sistemas de Controle*. 5ª edição. ed. [S.l.]: LTC Editora, 2009. ISBN 8521617046.

OGATA, K. *Engenharia de Controle Moderno*. 4ª edição. ed. São Paulo: Pearson Prentice Hall, 2003.

OTTO, P. *Genética humana e clínica*. [S.l.]: Editora Roca, 2004. ISBN 9788572414944.

QI-YI, Z.; SHU-CHUN, C. An improved crossover operator of genetic algorithm. In: *Computational Intelligence and Design, 2009. ISCID '09. Second International Symposium on*. [S.l.: s.n.], 2009. v. 2, p. 82 –86.

RAMAKRISHNA, R. C. W. A. Elitism-based compact genetic algorithms. In: . [S.l.: s.n.], 2003. v. 7, p. 367–385.

RAYMOND, G. J. *Modern Digital Control Sysetms*. [S.l.]: Marcel Dekker Inc., 1994.

ROTHLAUF, F. *Representations for Genetic and Evolutionary Algorithms*. New York: Physica-Verl.: Heildelberg, 2002.

SRINIVAS, M.; PATNAIK, L. Adaptive probabilities of crossover and mutation in genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, v. 24, n. 4, p. 656 –667, abr. 1994. ISSN 0018-9472.

YUE, W. et al. Design of new intelligent street light control system. In: *Control and Automation (ICCA), 2010 8th IEEE International Conference on*. [S.l.: s.n.], 2010. p. 1423 –1427. ISSN 1948-3449.

ZHANG, X.; LIU, S. A new interval-genetic algorithm. In: *Natural Computation, 2007. ICNC 2007. Third International Conference on*. [S.l.: s.n.], 2007. v. 4, p. 193 –197.

ZHONG, J. et al. Comparison of performance between different selection strategies on simple genetic algorithms. In: . [S.l.: s.n.], 2005. v. 2, p. 1115 –1121.

# Apêndice A

## Cálculo da Matriz Exponencial $e^{At}$

Na engenharia de controle, para solucionar alguns problemas normalmente é necessário calcular  $e^{At}$ . O objetivo do presente trabalho é a solução das equações de estado, são vários os métodos analíticos para o cálculo de  $e^{At}$ . Apresentaremos três métodos a seguir, mas antes é importante verificar a representação de duas formas canônicas no espaço de estados.

### A.1 Forma Canônica Diagonal

Considere um sistema definido por:

$$\frac{Y(s)}{U(s)} = \frac{b_0s^n + b_1s^{n-1} + \dots + b_{n-1}s + b_n}{s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n} \quad (\text{A.1})$$

Considere a função de transferência definida pela Equação (A.1), no caso em que o polinômio do denominador tem apenas raízes distintas. Para este caso, a Equação (A.1) pode ser escrita como:

$$\begin{aligned} \frac{Y(s)}{U(s)} &= \frac{b_0s^n + b_1s^{n-1} + \dots + b_{n-1}s + b_n}{(s + p_1)(s + p_2) \dots (s + p_n)} \\ &= b_0 + \frac{c_1}{s + p_1} + \frac{c_2}{s + p_2} + \dots + \frac{c_n}{s + p_n} \end{aligned}$$

A forma **canônica diagonal** da representação no espaço de estados para esse sistema é:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} -p_1 & & & & 0 \\ & -p_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ 0 & & & & -p_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + b_0 u$$

A forma canônica diagonal pode auxiliar o cálculo de  $e^{At}$  no método 1.

## A.2 Forma Canônica de Jordan

No caso em que o polinômio do denominador da Equação (A.1) envolve múltiplas raízes, a forma canônica diagonal sofre algumas alterações se transformando na forma **canônica de Jordan**. Considere que os pólos são diferentes entre si, exceto os três primeiros que são idênticas,  $p_1 = p_2 = p_3$ . A Equação (A.1) pode ser escrita como:

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{(s + p_1)^3 (s + p_4) (s + p_5) \cdots (s + p_n)} \quad (\text{A.2})$$

A expansão em frações parciais da Equação (A.2) resulta em:

$$= b_0 + \frac{c_1}{(s + p_1)^3} + \frac{c_2}{(s + p_1)^2} + \frac{c_3}{s + p_1} + \frac{c_4}{s + p_4} + \cdots + \frac{c_n}{s + p_n}$$

A forma canônica de Jordan pode vir a ajudar no cálculo de  $e^{At}$ , e sua representação no espaço de estados é dada por:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} -p_1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -p_1 & 1 & \vdots & & \vdots \\ 0 & 0 & -p_1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -p_4 & & 0 \\ \cdot & & \cdot & & \cdot & \\ \cdot & & \cdot & & \cdot & \\ \cdot & & \cdot & & \cdot & \\ 0 & \cdots & 0 & 0 & & -p_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + b_0 u$$

### A.3 Cálculo de $e^{At}$ : método 1.

Caso a matriz  $\mathbf{A}$  possa ser transformada na forma canônica diagonal, então  $e^{At}$  pode ser dada por:

$$e^{At} = \mathbf{P} e^{\mathbf{D}t} \mathbf{P}^{-1} = \mathbf{P} \begin{bmatrix} e^{\lambda_1 t} & & & 0 \\ & e^{\lambda_2 t} & & \\ & & \cdot & \\ & & & \cdot \\ 0 & & & e^{\lambda_n t} \end{bmatrix} \mathbf{P}^{-1}$$

em que  $\mathbf{P}$  é uma matriz de transformação diagonalizante, que transforma  $\mathbf{A}$  em diagonal.

Caso a matriz  $\mathbf{A}$  possa ser transformada na forma canônica de Jordan, então  $e^{At}$  pode ser dada por:

$$e^{At} = \mathbf{S}e^{\mathbf{J}t}\mathbf{S}^{-1}$$

onde  $\mathbf{S}$  é a matriz de transformação para tornar  $\mathbf{A}$  na forma canônica de Jordan.

## A.4 Cálculo de $e^{At}$ : método 2.

Esse método utiliza a transformada de Laplace para o cálculo de  $e^{At}$ . Pela resolução da equação de estado,  $e^{At}$  pode ser dada como:

$$e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}]$$

Para calcular  $e^{At}$ , inverte-se a matriz  $(sI - A)$ , obtém-se uma matriz em que os elementos são funções em  $s$ . Posteriormente, aplica-se a transformada inversa de Laplace em cada elemento da matriz.

## Apêndice B

### Cálculo das Funções Fitness

A partir do Sistema 1, apresentado no Capítulo de Resultados, obtemos as matrizes A, B, C e D.

$$A = \begin{bmatrix} -1 & -1 \\ 6.5 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad e \quad D = \begin{bmatrix} 0 \end{bmatrix} \quad (\text{B.1})$$

A Função Fitness, utilizada para avaliação dos indivíduos do AGD, é obtida através dos elementos da matriz de transição de estados, que é mostrada na Equação B.2.

$$\Phi(t) = e^{(A-BK)t} = \mathcal{L}^{-1}[(sI - A + BK)^{-1}] \quad (\text{B.2})$$

Substituindo as matrizes do sistema na Equação B.2, tem-se

$$\Phi(t) = \mathcal{L}^{-1} \left[ \left( \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -1 & -1 \\ 6.5 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \right)^{-1} \right] \quad (\text{B.3})$$

Fazendo-se as operações entre as matrizes o resultado fica

$$\Phi(t) = \mathcal{L}^{-1} \left[ \begin{bmatrix} s+1 & 1 \\ \frac{-13}{2+k_1} & s+k_2 \end{bmatrix}^{-1} \right]$$

Calculando a inversa da matriz, obtemos

$$\Phi(t) = \mathcal{L}^{-1} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

em que, os valores de  $p_{11}$ ,  $p_{12}$ ,  $p_{21}$  e  $p_{22}$ , são mostrados abaixo:

$$p_{11} = 2(s + k_2)/(2s^2 + 2sk_2 + 2s + 2k_2 + 13 - 2k_1)$$

$$p_{12} = -2/(2s^2 + 2sk_2 + 2s + 2k_2 + 13 - 2k_1)$$

$$p_{21} = -(-13 + 2k_1)/(2s^2 + 2sk_2 + 2s + 2k_2 + 13 - 2k_1)$$

$$p_{22} = 2(s + 1)/(2s^2 + 2sk_2 + 2s + 2k_2 + 13 - 2k_1)$$

Aplicando-se a transformada inversa de Laplace, obtemos os elementos da matriz  $\Phi$ .

$$\Phi(t) = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}$$

em que, os valores de  $\phi_{11}$ ,  $\phi_{12}$ ,  $\phi_{21}$  e  $\phi_{22}$ , são mostrados abaixo:

$$\phi_{11} = \frac{(\frac{1}{2} - \frac{k_2}{2} + \frac{w^{\frac{1}{2}}}{2})z + (\frac{w^{\frac{1}{2}}}{2} - \frac{-1}{2} + \frac{k_2}{2})y}{w^{\frac{1}{2}}}$$

$$\phi_{12} = \frac{z-y}{w^{\frac{1}{2}}}$$

$$\phi_{21} = \frac{(\frac{-13}{2} + k_1)(z-y)}{w^{\frac{1}{2}}}$$

$$\phi_{22} = \frac{(\frac{w^{\frac{1}{2}}}{2} - \frac{1}{2} + \frac{k_2}{2})z + (\frac{1}{2} - \frac{k_2}{2} + \frac{w^{\frac{1}{2}}}{2})y}{w^{\frac{1}{2}}}$$

Os valores de  $w$ ,  $y$  e  $z$  são utilizados com a finalidade de tornar os valores de  $\phi$  mais legíveis. Estes são destacados abaixo:

$$w = -25 - 2k_2 + k_2^2 + 4k_1$$

$$y = e^{(\frac{-1}{2}(-w^{\frac{1}{2}} + 1 + k_2)t)}$$

$$z = e^{(\frac{-1}{2}(w^{\frac{1}{2}} + 1 + k_2)t)}$$