

**FÁBIO ALESSANDRO GUERRA**

**ANÁLISE DE MÉTODOS DE AGRUPAMENTO  
PARA O TREINAMENTO DE REDES NEURAIIS  
DE BASE RADIAL APLICADAS À  
IDENTIFICAÇÃO DE SISTEMAS**

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

**CURITIBA**

**2006**

**FÁBIO ALESSANDRO GUERRA**

**ANÁLISE DE MÉTODOS DE AGRUPAMENTO  
PARA O TREINAMENTO DE REDES NEURAIIS  
DE BASE RADIAL APLICADAS À  
IDENTIFICAÇÃO DE SISTEMAS**

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

Área de Concentração: Automação e Controle de Processos.

Orientador: Prof. Dr. Leandro dos Santos Coelho

Co-orientador: Dr. Milton Pires Ramos – TECPAR  
(Instituto de Tecnologia do Paraná)

**CURITIBA**

**2006**

G934a  
2006 Guerra, Fábio Alessandro  
Análise de métodos de agrupamento para o treinamento de redes neurais de base radial aplicadas à identificação de sistemas / Fábio Alessandro Guerra ; orientador, Leandro dos Santos Coelho ; co-orientador, Milton Pires Ramos. – 2006.  
xi, 131 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2006  
Inclui bibliografia

1. Sistemas não-lineares - Identificação. 2. Redes neurais (Computação). 3. Algoritmos. I. Coelho, Leandro dos Santos. II. Ramos, Milton Pires. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia de Produção e Sistemas. IV. Título.

CDD 21. ed. – 629.836  
006.32

Guerra, Fábio Alessandro

ANÁLISE DE MÉTODOS DE AGRUPAMENTO PARA O TREINAMENTO DE REDES NEURAIS DE BASE RADIAL APLICADAS À IDENTIFICAÇÃO DE SISTEMAS. Curitiba, 2006. 131 p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

1. identificação de sistemas não-lineares 2. rede neural 3. algoritmos de agrupamento

Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

# Agradecimentos

Ao professor Leandro dos Santos Coelho, pela orientação, paciência e incentivos em todos os aspectos referentes ao desenvolvimento desta dissertação. E principalmente pela sua amizade e dedicação fazendo com que este trabalho se tornasse realidade.

Aos srs. Milton Pires Ramos e Julio Cezar Zanoni pelo apoio dado na parte de metodologia e por sempre acreditarem em meu trabalho.

Aos professores Myrian Regattieri de Biase da Silva Delgado e Júlio Cezar Nievola, membros da banca examinadora, que com suas argumentações, opiniões e conhecimento contribuíram para que este trabalho fosse concluído com êxito.

Aos meus pais e irmão, Benedito, Sonia e Michael, pelo amor e por sempre acreditar em minha capacidade, me incentivando a cada obstáculo encontrado neste trabalho.

Em especial, à minha noiva Fernanda que sempre esteve ao meu lado me auxiliando nos momentos difíceis com paciência e dedicação, mas principalmente pelo seu amor que me renovava de energia para que minha caminhada se tornasse possível. Sou um homem de sorte, pois tenho uma grande mulher ao meu lado.

A Pontifícia Universidade Católica do Paraná (PUC-PR) pela sua estrutura e seu apoio financeiro.

*“Aprender é a única coisa que a mente nunca se cansa,  
nunca tem medo e nunca se arrepende”*

*Leonardo da Vinci*

# Resumo

Para sistemas complexos, a modelagem baseada em leis elementares para determinar o comportamento de sistemas dinâmicos nem sempre é possível. Uma alternativa para solucionar este problema é a utilização de conceitos de identificação de sistemas. Através da identificação de sistemas é possível determinar um modelo matemático baseado em dados experimentais das entradas e saídas do sistema. Quando existe pouco conhecimento *a priori* do sistema, é comum utilizar-se do modelo matemático do tipo caixa-preta para representar diferentes sistemas não-lineares. As redes neurais têm provado serem estruturas não-lineares caixa-preta em muitas aplicações. Uma concepção de rede neural que pode ser aplicada à identificação de sistemas é a de base radial. Esta rede é um aproximador local e é composta por uma camada oculta definida por um conjunto de funções de base radial. As unidades ocultas fornecem um conjunto de funções de base radial que constituem uma base arbitrária para os padrões de entrada. Para treinamento de uma rede neural de base radial, pode-se aplicar os algoritmos de agrupamento para determinação dos centros de cada função de base radial, visando descobrir padrões nos dados de entrada. O objetivo desta dissertação é a análise e comparação dos métodos de agrupamento  $k$ -médias,  $c$ -médias nebuloso, Gustafson-Kessel e Gath-Gheva nas redes neurais de base radial, aplicadas em identificação dos sistemas não-lineares de um trocador de calor, sistema de Mackey-Glass, sistema caótico de Rössler e da fornalha de gás de Box-Jenkins.

**Palavras-Chave:** identificação de sistemas não-lineares, rede neural, algoritmos de agrupamento.

# Abstract

For complex systems, modeling using basic laws to determine their dynamic behavior is not always possible. An alternative to solve these problems is use concepts of systems identification. Trough system identification it is possible to determine a mathematical model based on input and output data of the system. When little prior knowledge is available, it is common to use a black-box mathematical model to represent different nonlinear systems. The neural network models have proven to be successful nonlinear black-box model structures in many applications. A conception of neural network that can be applied to the systems identification is the radial basis function. This network follows a local approximation approach and is composed by a hidden layer that is defined by a set of radial basis functions. The hidden units supply a set of radial basis functions that constitute an arbitrary base for the patterns of input. In order to train a radial basis neural network, the clustering algorithms are applied for determination of the centers of each radial basis function, aiming to discover patterns in the input data. The objective of this dissertation is the analysis and comparison of  $k$ -means, fuzzy  $c$ -means, Gustafson-Kessel and Gath-Gheva clustering methods in the radial basis functions neural networks, applied in nonlinear identification of a heat exchanger, Mackey-Glass system, Rössler chaotic system and Box-Jenkins gas furnace.

**Keywords:** non-linear system identification, neural network, clustering algorithms.

# Sumário

<b>Lista de Figuras .....</b>	<b>iv</b>
<b>Lista de Tabelas .....</b>	<b>ix</b>
<b>Lista de Abreviaturas .....</b>	<b>x</b>
<b>1 Introdução.....</b>	<b>1</b>
1.1 Identificação de Sistemas Não-Lineares e Redes Neurais Artificiais.....	1
1.2 Motivação e Justificativa.....	2
1.3 Proposta de Dissertação .....	3
1.4 Organização da Dissertação .....	3
<b>2 Revisão da Literatura .....</b>	<b>5</b>
2.1 Literatura de Redes Neurais .....	5
2.2 Comentários Finais .....	8
<b>3 Identificação de Sistemas Não-Lineares.....</b>	<b>9</b>
3.1 Introdução .....	9
3.2 Etapas para Identificação de Sistemas .....	12
3.3 Representações de Sistemas Não-Lineares.....	15
3.3.1 Séries de Volterra.....	15
3.3.2 Modelos de Hammerstein e de Wiener.....	16
3.3.3 Classificação de Modelos Não-Lineares.....	17
3.3.3.1 Modelos Caixa-Preta .....	18
3.3.3.2 Estruturas Não-Lineares Caixa-Preta.....	18
3.3.3.3 Regressores .....	20
3.3.3.3.1 Uma Revisão de Modelos Lineares Caixa-Preta.....	20
3.3.3.3.2 Regressores para Modelos Dinâmicos Não-Lineares Caixa-Preta .....	21
3.3.3.4 Representações de Modelos Baseados em Redes Neurais .....	23
3.3.3.4.1 Modelo de Redes Neurais para Identificação de Sistemas Não-Lineares .....	26
<b>4 Redes Neurais Artificiais .....</b>	<b>29</b>
4.1 Introdução .....	29



4.2 Breve Histórico.....	29
4.3 Neurônio Biológico .....	30
4.4 Redes Neurais Artificiais .....	32
4.4.1 Rede Neural <i>Perceptron</i> Multicamadas.....	33
4.4.2 Rede Neural de Função de Base Radial (RN-RBF).....	35
4.4.2.1 Teorema de Cover .....	36
4.4.2.2 Problema de Interpolação.....	38
4.4.2.3 Teorema de Micchelli .....	41
4.4.2.4 Problema do Ou Exclusivo (XOR) .....	43
4.4.2.5 Estratégias de Aprendizado.....	49
4.4.2.5.1 Centros Fixos Seleccionados Aleatoriamente .....	50
4.4.2.5.2 Seleção Auto-Organizada de Centros.....	50
4.4.2.5.3 Seleção Supervisionada de Centros .....	51
4.4.3 RN-RBF versus RN-MLP.....	51
4.4.4 Aplicações .....	54
<b>5 Algoritmos.....</b>	<b>56</b>
5.1 Introdução .....	56
5.2 Algoritmos de Agrupamento .....	57
5.2.1 Algoritmo <i>k</i> -médias Convergente ( <i>k-means</i> ) .....	57
5.2.1.2 Etapas do Algoritmo <i>k</i> -médias Convergente.....	57
5.2.2 Algoritmo <i>c</i> -médias Nebuloso ( <i>c-means</i> ) .....	58
5.2.2.1 Etapas do Algoritmo <i>c</i> -médias nebuloso.....	59
5.2.3 Algoritmo de Gustafson-Kessel (GK) .....	59
5.2.3.1 Etapas do Algoritmo GK .....	61
5.2.4 Algoritmo de Gath-Gheva (GG) .....	62
5.2.4.1 Etapas do algoritmo GG.....	62
5.3 Algoritmos de Otimização.....	63
5.3.1 Mínimos Quadrados em batelada (Pseudo-Inversa de Penrose-Moore) .....	63
5.4 Validação de modelos .....	64
5.4.1 Coeficiente de correlação múltipla e erro médio quadrático.....	64
<b>6 Casos Estudados.....</b>	<b>66</b>
6.1 Identificação.....	66
6.2 Casos Estudados .....	68
6.2.1 Trocador de Calor .....	68
6.2.2 Sistema de Mackey-Glass.....	70
6.2.3 Sistema Caótico de Rössler.....	71
6.2.4 Fornalha de Gás de Box e Jenkins.....	73

<b>7 Resultados de Simulação .....</b>	<b>75</b>
7.1 Resultados para o Trocador de calor .....	77
7.1.1 Comentários dos resultados para o Trocador de Calor .....	83
7.2 Resultados para o Sistema de Mackey-Glass .....	84
7.2.1 Comentários dos resultados para o Sistema de Mackey-Glass.....	90
7.3 Resultados para o Sistema Caótico de Rössler .....	91
7.3.1 Resultados para a coordenada $x$ .....	91
7.3.1.1 Comentários dos resultados para a coordenada $x$ .....	97
7.3.2 Resultados para a coordenada $y$ .....	98
7.3.2.1 Comentários para os resultados para a coordenada $y$ .....	104
7.3.3 Resultados para a coordenada $z$ .....	105
7.3.3.1 Resultados para a coordenada $z$ .....	111
7.4 Resultados para a Fornalha de Box e Jenkins.....	112
7.4.1 Comentários dos resultados para a fornalha de Box-Jenkins .....	118
<b>8 Conclusão e Futura Pesquisa.....</b>	<b>121</b>
<b>Referências Bibliográficas .....</b>	<b>122</b>

## Lista de Figuras

Figura 3.1. Princípios para a construção de um modelo matemático (modelagem).....	10
Figura 3.2. Diagrama de blocos para uma identificação de sistemas. ....	14
Figura 3.3. Representação gráfica da série de Volterra.....	16
Figura 3.4. Representação 1 de redes neurais. ....	24
Figura 3.5. Representação 2 de redes neurais. ....	24
Figura 3.6. Representação 3 de redes neurais. ....	25
Figura 3.7. Representação 4 de redes neurais. ....	25
Figura 3.8. Modelo paralelo. ....	27
Figura 3.9. Modelo série-paralelo.....	28
Figura 4.1. Neurônio do sistema nervoso central de um vertebrado. ....	31
Figura 4.2. Estrutura geral de uma RN-MLP. ....	35
Figura 4.3. (a) dicotomia linearmente separável; (b) dicotomia esfericamente separável; (c) dicotomia quadraticamente separável. ....	38
Figura 4.4. Funções de base radial cobertas pelo teorema de Micchelli. ....	43
Figura 4.5. Rede RBF para resolver o problema do XOR. ....	44
Figura 4.6. Quatro padrões de entrada para o problema do XOR. ....	45
Figura 4.7. Diagrama de decisão. ....	47
Figura 4.8. Particionamento dos dados de entrada realizada por uma RN-RBF com quatro neurônios na camada oculta. ....	52
Figura 4.9. Particionamento dos dados de entrada realizada por uma RN-RBF com três neurônios na camada oculta. ....	53
Figura 6.1. Entrada do trocador de calor.....	69
Figura 6.2. Saída do trocador de calor. ....	69
Figura 6.3. Comportamento da variável de saída $y(t)$ .....	70
Figura 6.4. Atrator do sistema de Rössler para $(a, b, c) = (0,36; 0,40; 4,50)$ . ....	71
Figura 6.5. Coordenada $x$ do sistema de Rössler. ....	72

Figura 6.6. Coordenada $y$ do sistema de Rössler.....	72
Figura 6.7. Coordenada $z$ do sistema de Rössler. ....	73
Figura 6.8. Sinal de entrada de gás metano.....	74
Figura 6.9. Sinal de saída da $\%CO_2$ .....	74
Figura 7.1. Arquitetura geral da RN-RBF para os sistemas analisados.....	75
Figura 7.2. Saída do melhor resultado para identificação do trocador de calor com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 5). ....	79
Figura 7.3. Erro do melhor resultado para identificação do trocador de calor com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 5). ....	79
Figura 7.4. Saída do melhor resultado para identificação do trocador de calor com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 4). ....	80
Figura 7.5. Erro do melhor resultado para identificação do trocador de calor com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 4). ....	80
Figura 7.6. Saída do melhor resultado para identificação do trocador de calor com agrupamento GK e otimização pela pseudo-inversa (simulação 2). ....	81
Figura 7.7. Erro do melhor resultado para identificação do trocador de calor com agrupamento GK e otimização pela pseudo-inversa (simulação 2). ....	81
Figura 7.8. Saída do melhor resultado para identificação do trocador de calor com agrupamento GG e otimização pela pseudo-inversa (simulação 2). ....	82
Figura 7.9. Erro do melhor resultado para identificação do trocador de calor com agrupamento GG e otimização pela pseudo-inversa (simulação 2). ....	82
Figura 7.10. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 5). ....	86
Figura 7.11. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 5). ....	86
Figura 7.12. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 5). ....	87
Figura 7.13. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 5). ....	87
Figura 7.14. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GK e otimização pela pseudo-inversa (simulação 2). ....	88

Figura 7.15. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GK e otimização pela pseudo-inversa (simulação 2). .....	88
Figura 7.16. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GG e otimização pela pseudo-inversa (simulação 2). .....	89
Figura 7.17. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GG e otimização pela pseudo-inversa (simulação 2). .....	89
Figura 7.18. Saída do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 1). .....	93
Figura 7.19. Erro do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 1). .....	93
Figura 7.20. Saída do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 1). .....	94
Figura 7.21. Erro do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 1). .....	94
Figura 7.22. Saída do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5). .....	95
Figura 7.23. Erro do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5). .....	95
Figura 7.24. Saída do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 3). .....	96
Figura 7.25. Erro do melhor resultado para identificação da coordenada $x$ do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 3). .....	96
Figura 7.26. Saída do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 5). .....	100
Figura 7.27. Erro do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 5). .....	100
Figura 7.28. Saída do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 5). .....	101
Figura 7.29. Erro do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 5). .....	101
Figura 7.30. Saída do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 3). .....	102

Figura 7.31. Erro do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 3). .....	102
Figura 7.32. Saída do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 4). .....	103
Figura 7.33. Erro do melhor resultado para identificação da coordenada $y$ do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 4). .....	103
Figura 7.34. Saída do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 2). .....	107
Figura 7.35. Erro do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 2). .....	107
Figura 7.36. Saída do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 2). .....	108
Figura 7.37. Erro do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 2). .....	108
Figura 7.38. Saída do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5). .....	109
Figura 7.39. Erro do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5). .....	109
Figura 7.40. Saída do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 1). .....	110
Figura 7.41. Erro do melhor resultado para identificação da coordenada $z$ do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 1). .....	110
Figura 7.42. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 4). .....	114
Figura 7.43. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento $k$ -médias e otimização pela pseudo-inversa (simulação 4). .....	114
Figura 7.44. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 4). .....	115
Figura 7.45. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento $c$ -médias e otimização pela pseudo-inversa (simulação 4). .....	115
Figura 7.46. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GK e otimização pela pseudo-inversa (simulação 4). .....	116

Figura 7.47. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GK e otimização pela pseudo-inversa (simulação 4). .....	116
Figura 7.48. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GG e otimização pela pseudo-inversa (simulação 4). .....	117
Figura 7.49. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GG e otimização pela pseudo-inversa (simulação 4). .....	117

## Lista de Tabelas

Tabela 4.1: Especificação para o problema do XOR.....	45
Tabela 4.2: Especificação das funções ocultas para o problema do XOR.....	46
Tabela 4.3: Transformação de entrada-saída calculada para o problema do XOR. ....	49
Tabela 4.4: Comparação entre uma rede neural RBF e uma rede neural MLP (convencionais). .....	53
Tabela 7.1: Resultados das simulações na fase de estimação para o trocador de calor.....	77
Tabela 7.2: Resultados das simulações na fase de validação para o trocador de calor. ....	78
Tabela 7.3: Resultados das simulações na fase de estimação para o sistema de Mackey-Glass. .....	84
Tabela 7.4: Resultados das simulações na fase de validação para o sistema de Mackey-Glass. .....	85
Tabela 7.5: Resultados das simulações na fase de estimação para a coordenada $x$ .....	91
Tabela 7.6: Resultados das simulações na fase de validação para a coordenada $x$ .....	92
Tabela 7.7: Resultados das simulações na fase de estimação para a coordenada $y$ .....	98
Tabela 7.8: Resultados das simulações na fase de validação para a coordenada $y$ . ....	99
Tabela 7.9: Resultados das simulações na fase de estimação para a coordenada $z$ .....	105
Tabela 7.10: Resultados das simulações na fase de validação para a coordenada $z$ .....	106
Tabela 7.11: Resultados das simulações na fase de estimação para a fornalha de Box-Jenkins. .....	112
Tabela 7.12: Resultados das simulações na fase de validação para a fornalha de Box-Jenkins. .....	113



## Lista de Abreviaturas

AFLS	<i>adaptive fuzzy logic system</i>
ANOVA	<i>analysis of the variance</i>
AR	<i>auto regressive</i>
ARIMA	<i>autoregressive integrated moving average</i>
ARMA	<i>autoregressive moving average model structure</i>
ARMAX	<i>autoregressive moving average model structure with exogenous inputs</i>
ARNN	<i>autoregressive recurrent neural network</i>
ARX	<i>autoregressive model structure with exogenous inputs</i>
ASE	<i>average square error</i>
BFGS	<i>Broyden-Fletcher-Goldfarb-Shanno</i>
BIBO	<i>bounded-input and bounded-output</i>
BJ	<i>Box-Jenkins model structure</i>
BP	<i>back-propagation</i>
CSTR	<i>continuous stirred tank reactor</i>
FCM	<i>fuzzy c-means</i>
FIR	<i>finite impulse response</i>
FMLE	<i>fuzzy maximum likelihood estimates</i>
GN	<i>Gauss-Newton</i>
HLNAM	<i>hybrid linear neural network adaptive model</i>
HMLP	<i>hybrid multilayered perceptron</i>
HSOL	<i>hierarchically self-organization learning</i>
LM	<i>Levenberg-Marquardt</i>
LMS	<i>least mean square</i>
LS	<i>least squares</i>
MIMO	<i>multi-inputs multi-outputs</i>
MISO	<i>multi-inputs single-output</i>

MSE	<i>mean square error</i>
NARMAX	<i>non-linear autoregressive moving average model structure with exogenous inputs</i>
NARX	<i>non-linear autoregressive model structure with exogenous inputs</i>
NBJ	<i>non-linear Box-Jenkins model structure</i>
NN	<i>neural network</i>
NNBJ	<i>neural network Box-Jenkins</i>
NFIR	<i>non-linear finite impulse response</i>
NNAR	<i>neural network autoregressive</i>
NNARMA	<i>neural network autoregressive moving average</i>
NNARMAX	<i>neural network autoregressive moving average with exogenous inputs</i>
NNARX	<i>neural network autoregressive with exogenous inputs</i>
NNFIR	<i>neural network finite impulse response</i>
NNOE	<i>neural network output error</i>
NOE	<i>non-linear output error model</i>
OE	<i>output error model</i>
OLS	<i>orthogonal least squares</i>
PLSNN	<i>partial least square neural network</i>
RBF	<i>radial basis functions</i>
RLS	<i>recursive least square</i>
RN	<i>rede neural</i>
RN-MLP	<i>rede neural perceptron multicamadas</i>
RN-RBF	<i>rede neural de base radial</i>
ROLS	<i>recursive orthogonal least square</i>
SISO	<i>single-input single-output</i>
SOM	<i>self-organization map</i>
SONARX	<i>self-organization map non-linear autoregressive moving average model structure with exogenous inputs</i>
XOR	<i>exclusive OR</i>

# Capítulo 1

## 1 Introdução

Nos últimos anos, é crescente o interesse no desenvolvimento de estratégias de identificação de sistemas não-lineares, muitas destas utilizando redes neurais artificiais. Tal fato é motivado por diversos fatores, tais como: (i) os avanços da teoria de sistemas não-lineares, que originaram metodologias de projeto aplicáveis a problemas de controle de processos e (ii) o desenvolvimento de métodos de identificação eficazes em lidar com sistemas não-lineares presentes no meio industrial.

### 1.1 Identificação de Sistemas Não-Lineares e Redes Neurais Artificiais

A área de identificação de sistemas está consolidada para sistemas lineares. Entretanto, o modelo linear é útil se o processo físico exibe um comportamento linear nas regiões do processo. Entretanto, na realidade, a grande maioria dos sistemas físicos são não-lineares em alguma região de operação. Assim, este tipo de comportamento não-linear motivou o desenvolvimento de ferramentas matemáticas que consigam modelar e analisar este comportamento [COE04], [LJU87].

Para os pesquisadores desta área [GOR00], [SJO95a], [LJU94], [SJO95b], [AGU04], a etapa mais difícil na identificação de um sistema não-linear é a maldição da dimensionalidade (*curse of dimensionality*). Esta é uma característica de modelos não-lineares, pois estes exibem uma variedade de comportamentos dinâmicos, que podem ser difíceis de lidar com métodos convencionais de modelagem e identificação linear. Em sistemas não-lineares, geralmente, não é possível incorporar nenhum conhecimento *a priori* da dinâmica do sistema

no procedimento de identificação, dificultando estimar o modelo não-linear de um processo físico [HAY01].

Os fundamentos das redes neurais artificiais são inspirados em sistemas neurais biológicos, com intenção de simular, mesmo que rudimentarmente, a forma como o cérebro aprende, recorda e processa as informações. As redes neurais artificiais podem ser projetadas como um sistema interconectado de elementos de processamento (neurônios), cada um dos quais com um número limitado de entradas e saídas. Entre as características que melhor explanam o potencial das redes neurais sobressaem-se a adaptabilidade, paralelismo inerente, multidisciplinaridade e tolerância à falhas. Em síntese, uma rede neural é uma ferramenta matemática promissora para aplicações práticas de reconhecimento de padrões e mapeamentos não-lineares [BIS95], [HAY01].

## 1.2 Motivação e Justificativa

A principal potencialidade na estrutura das redes neurais consiste na sua habilidade de adaptação e aprendizagem. A habilidade de adaptação e aprendizagem pelo ambiente significa que modelos de redes neurais podem lidar com dados imprecisos e situações complexas. Uma rede neural treinada adequadamente tem a habilidade de generalizar quando são apresentadas entradas que não estão presentes em dados já conhecidos por ela.

Uma rede neural possui neurônios conectados por pesos com capacidade de adaptação, que podem ser arranjados em uma estrutura paralela. Por causa deste paralelismo, a falha de alguns neurônios não causa efeitos significantes para o desempenho do sistema, o que é denominado tolerância à falhas [HAY01].

Uma característica significativa das redes neurais está em sua habilidade de aproximar qualquer função contínua não-linear com um grau de precisão determinado pelo projetista. Esta habilidade das redes neurais são úteis para modelar e identificar sistemas não-lineares ou mesmo em combinação com controladores não-lineares [COE00].

As redes neurais podem ter várias entradas e várias saídas e são facilmente aplicáveis à identificação de sistemas não-lineares. Com o avanço em tecnologias de *hardware*, existem componentes com funções voltadas a sistemas com implementações direcionadas para redes

neurais, o que traz uma velocidade adicional à computação de métodos de treinamento de diferentes redes neurais.

### **1.3 Proposta de Dissertação**

Nesta dissertação é apresentado um estudo comparativo de desempenho dos quatro métodos de agrupamento: (i)  $k$ -médias, (ii)  $c$ -médias nebuloso, (iii) Gustafson-Kessel e (iv) Gath-Gheva, para o treinamento de redes neurais de base radial. Estas redes neurais são então aplicadas em uma configuração apropriada de sinais de entrada e saída com intuito de identificação de sistemas não-lineares. Os sistemas não-lineares abordados são os seguintes: (i) um trocador de calor, (ii) sistema de Mackey-Glass, (iii) sistema caótico de Rössler e (iv) fornalha de gás de Box-Jenkins.

### **1.4 Organização da Dissertação**

Esta dissertação está organizada da seguinte maneira.

- O capítulo 2 apresenta uma revisão da literatura nas áreas de identificação de sistemas e redes neurais em geral ou mais especificamente de redes neurais de base radial utilizando algoritmos de agrupamento na identificação de sistemas. Neste capítulo é identificado uma grande quantidade de trabalhos onde cada um se encontra em uma das duas áreas ou na união entre elas e suas devidas aplicações;
- Uma discussão da teoria de identificação de sistemas não-lineares e sua intersecção com as redes neurais é detalhada no capítulo 3. Este capítulo engloba tópicos de identificação, mas o foco são os modelos dinâmicos não-lineares do tipo caixa-preta;
- O capítulo 4 fundamenta os conceitos e potencialidades das redes neurais. As redes neurais de base radial, sua estrutura e aplicação em identificação de sistemas não-lineares são também detalhadas;

- No capítulo 5 são abordados aspectos de projetos dos algoritmos de agrupamento e mecanismos de validação dos modelos matemáticos obtidos na identificação baseados em índices de desempenho do erro médio quadrático e o coeficiente de correlação múltipla;
- A descrição de quatro estudos de caso para identificação utilizando a rede neural de base radial é apresentada no capítulo 6. Os sistemas descritos são: (i) o trocador de calor, (ii) o sistema de Mackey-Glass, (iii) o sistema caótico de Rössler e (iv) a fornalha de gás de Box e Jenkins), estes abordados no procedimento de identificação;
- Os resultados das simulações para identificação de cada sistema são apresentados no capítulo 7. Neste contexto tabelas de resultados e figuras correspondentes à identificação dos casos estudados são mostradas.
- No capítulo 8, a conclusão e possíveis pesquisas futuras são mencionadas.

# Capítulo 2

## 2 Revisão da Literatura

Este capítulo apresenta uma revisão da literatura em duas áreas do conhecimento: (i) identificação de sistemas e (ii) redes neurais artificiais. Existe uma vasta gama de trabalhos publicados, por isso foram selecionadas referências bibliográficas que proporcionam um panorama geral de como essas áreas estão sendo abordadas. O foco desta revisão é descrever e identificar: a estrutura utilizada nas redes neurais e a estrutura de modelos matemáticos utilizados na identificação.

### 2.1 Literatura de Redes Neurais

As redes neurais são técnicas matemáticas inspiradas pelo avanço nos estudos do cérebro humano e seu sistema nervoso. As redes neurais podem ser consideradas sistemas dinâmicos não-lineares com elementos de processamento (neurônios), que se interconectam e possuem ponderações ajustáveis. Algumas das características relevantes podem ser citadas: aprendizado, processamento paralelo, memória associativa e distribuída. Essas características são inspiradas nas redes neurais biológicas, mesmo que rudimentarmente [LU92].

O desenvolvimento de processos industriais modernos acarretou o surgimento de sistemas tecnológicos mais complexos. Este desenvolvimento gerou a necessidade de pesquisas relativas a técnicas matemáticas que tenham a capacidade de identificá-los ou mesmo controlá-los. Neste contexto, a área de identificação de sistemas tem sido largamente estudada e aprimorada. Uma relevante descrição do sistema a partir de dados observados é o que define a identificação de sistemas e a descrição do sistema resultante é denominada

modelo [COE04]. Entretanto, o que é identificar um sistema dinâmico? Existem na literatura algumas definições para responder esta pergunta:

- Identificação é a determinação, baseada nos sinais de entrada e saída, de um sistema em que, para uma classe especificada de sistemas, o sistema sob teste é equivalente [ZAD65];
- Identificação de sistemas representa a interface entre modelos do mundo matemático e o mundo real [JOH93];
- A idéia de identificação de sistemas é permitir a elaboração do modelo matemático de um sistema dinâmico, baseado em medidas coletadas, pelo ajuste de parâmetros e/ou do modelo, até que a saída do sistema coincida, tão bem quanto possível, com as amostras das saídas medidas [LJU96];
- A identificação de sistemas se propõe a obter um modelo matemático que explique, pelo menos em partes, de forma aproximada, a relação de causa e efeito presente nos dados [AGU04].

Alguns estudos e aplicações referentes a redes neurais em identificação de sistemas não-lineares têm sido apresentados. Em [LJU01] foram propostas as estruturas de rede de *wavelets*, RBF (*radial basis function*), *B-spline* e *fuzzy* na identificação de vários modelos de entradas-saídas do tipo caixa-preta, sendo estes modelos um linear ARMAX (*autoregressive moving average model structure with exogenous inputs*), e de espaço de estados ARX (*autoregressive model structure with exogenous inputs*), OE (*output error model*), e BJ (*Box-Jenkins model structure*). Em [MAS04] utilizou-se as estruturas HMLP (*hybrid multilayered perceptron*), MLP (rede neural perceptron multicamadas) e RBF com os algoritmos BP (*back-propagation*), *k*-médias adaptativo para a identificação não-linear *on-line* de um modelo NARMAX (*non-linear autoregressive moving average model structure with exogenous inputs*) na aplicação de dois sistemas não-lineares. Na identificação de um sistema de míssil [YAN00] utilizou a estrutura da RBF com gaussiana crescente e Lyapunov, sendo a identificação multivariável e *on-line*.

Existem algumas aplicações que utilizam a simulação de equações para a geração de dados que formam o sistema a ser identificado como são apresentados em [YIN97], [TAN95] e [COH02]. Em [YIN97] foi utilizado a estrutura da RBF com algoritmos de LS (*least squares*), um esquema de rede neural de função mínima de base radial, aprendizagem HSOL



(*hierarchically self-organization learning*) para encontrar o menor número de neurônios sendo a identificação multivariável e *on-line* e utilizando um modelo NARMAX. Em [TAN95] foi utilizada a estrutura da RBF com os algoritmos *k*-médias e LS, em identificação não-linear multivariável (MIMO (*multi-inputs multi-outputs*) e BIBO (*bounded-input bounded-output*)) em tempo discreto e um-passo-à-frente e utilizando um modelo ARMA (*autoregressive moving average model structure*). Em [COH02] foi utilizada a RBF com os algoritmos LS e *k*-médias para regressão não-linear e classificação de padrões.

Existem também aplicações de identificação de sistemas caóticos utilizando redes neurais como apresentado em [DUD04], onde foi utilizado a estrutura da MLP e RBF, com os algoritmos LM (Levenberg-Marquardt), método de quasi-Newton, BP e Lyapunov, na identificação do sistema caótico de Lorenz utilizando modelos AR (*auto regressive*), ARMA, FIR (*finite impulse response*), ARX, NNAR (*neural network autoregressive*) e NNARMA (*neural network autoregressive moving average*) em identificação com 1, 10, 20 e 50 passos-à-frente. No mesmo sistema caótico [ROJ00] utilizou a MLP com os algoritmos ANOVA (*analysis of the variance*), multifatorial ANOVA e várias funções de ativação não-linear tais como triangular, multiquadrática e gaussiana.

[PEN03] aplicou a estrutura da RBF com os algoritmos de LM, LMS (*least mean square*), BFGS (*Broyden-Fletcher-Goldfarb-Shanno*), McLoone e GN (Gauss-Newton) para a identificação não-linear 6-passos-à-frente e utilizando os modelos AR e ARX para um sistema de decomposição de óxido de nitrogênio de uma planta química. Na identificação de um processo CSTR, [YU00] utilizou a RBF com os algoritmos *k*-médias e OLS (*orthogonal least squares*) para realizar a identificação MIMO e MISO (*multi-inputs and single-output*) com os modelos NARX (*non-linear autoregressive model structure with exogenous inputs*), NARMAX e ARX. E também [KLI98] utilizou para o processo CSTR (*continuous stirred tank reactor*) a HLNAM (*hybrid linear neural network adaptive model*) com uma abordagem de modelagem híbrida, combinada com um modelo de espaço de estados linear com uma correção e adaptação não-linear e PLSNN (*partial least square neural network*).

Em uma série temporal financeira, [KOD02] utilizou as redes MLP, RBF, AFLS (*adaptive fuzzy logic system*), ARNN (*autoregressive recurrent neural network*) e Elman com os algoritmos BP, OLS, ROLS (*recursive orthogonal least square*), RLS (*recursive least square*), *c*-médias e SOM (*self-organization map*) para sua identificação não-linear multi-passos-à-frente com modelos ARIMA (*autoregressive integrated moving average*) e AR. Para

um processo químico [YU02] utilizou a RBF com os algoritmos ROLS, LS e uma configuração série e série-paralelo sendo a identificação multivariável (MIMO, MISO) e SISO (*single-input single-output*) e modelos NARX e ARX.

Finalizando esta gama de aplicações que estão sendo comentadas [SHI04] utilizou a RBF com os algoritmos ASE (*average square error*) e LS com aprendizado supervisionado, Tikhonov e máxima entropia para identificação de perfis de temperatura e radiação da atmosfera medida por um satélite. [SAC04] utilizou a RBF-SONARX (*self-organization map non-linear autoregressive moving average model structure with exogenous inputs*) e rede SONARX com o algoritmo SOM para previsão e identificação não-linear de um sistema hidráulico com modelos SONARX.

## 2.2 Comentários Finais

Recentemente, as aplicações da RN-RBF utilizando algoritmos de agrupamento em identificação de sistemas pode ser encontrada na literatura, como em [PAN06] que utilizou esta rede integrada com um sistema de ruído para identificação de *spoiled beef*. Em [FOL04] foram comparadas as redes CPNN (*constructive probabilistic neural network*), RN-RBF e RN-MLP na identificação de sons da respiração humana. Para a identificação de um sistema de colônia de bactérias [YAT05] utilizou a RN-RBF. Esta rede neural também foi utilizada em [JAN05] para previsão financeira.

Sejam as RN's-RBF ou outros tipos de redes neurais, pode-se observar que as pesquisas nessas áreas são de crescente interesse e estão sendo utilizadas em larga escala para identificação de sistemas de qualquer tipo e complexidade.

Através desta revisão pode-se observar que existe uma vasta quantidade de aplicações nas quais redes neurais podem ser utilizadas em identificação de sistemas. Pode-se observar, também, um acentuado número de algoritmos e derivações destes, que estão sendo criados ou modificados de acordo com cada aplicação.

# Capítulo 3

## 3 Identificação de Sistemas Não-Lineares

### 3.1 Introdução

A tentativa de representar um sistema dinâmico, ou simplesmente um fenômeno físico, através de uma equação ou de uma representação matemática é algo que desperta cada vez mais o interesse dos cientistas e engenheiros da área. Este fato remonta aos primórdios da ciência quando vários cientistas procuraram descrever fenômenos físicos através de leis matemáticas, como Isaac Newton com as leis do movimento e gravitação universal, Charles Coulomb com as leis da repulsão e atração de cargas elétricas, Maxwell e Boltzmann com a teoria cinética dos gases, dentre outros [COR01].

Com o desenvolvimento de processos industriais e a necessidade de controlá-los, surgiu também a necessidade do desenvolvimento de modelos que consigam reproduzir suas características estáticas e dinâmicas. Em virtude disso, surgiram os métodos de identificação de sistemas e a necessidade de representar comportamentos não-lineares. Com isso entende-se por modelagem e identificação a determinação do modelo matemático de um sistema, representando os seus aspectos essenciais para uma utilização particular (diagnóstico, supervisão, otimização e controle) [COE04], [LJU94], [NAR90]. Na interação de um indivíduo e um determinado sistema é necessário o conhecimento das variáveis deste, bem como a maneira com a qual estas se relacionam. Ou seja, é necessária a obtenção de um modelo para representar o sistema em questão [LJU87], [CAS99].

Existem diversos tipos de modelos. Pode-se citar os modelos físicos, os mentais, os gráficos e os matemáticos ou analíticos. A escolha do tipo de modelo para representar o sistema depende das suas características estáticas ou dinâmicas, e também do propósito para

qual o modelo está sendo determinado. Os modelos matemáticos, no entanto, permitem aplicações avançadas, por isso são largamente utilizados, seja na engenharia, biologia, medicina, economia ou em outras áreas [LJU87], [CAS99].

Existem diversas maneiras de modelar um sistema, tais como a modelagem física do processo e a modelagem através de técnicas de identificação, conforme ilustrado na figura 3.1 [COE04]. A modelagem pela física do processo é uma abordagem analítica realizada através de conhecimento de leis físicas do processo, exigindo um conhecimento detalhado do mesmo, o que torna em alguns casos difícil a sua aplicação. A identificação, por outro lado, é uma abordagem experimental, que consiste no desenvolvimento de técnicas para a estimação de modelos de sistemas dinâmicos através de dados experimentais, não exigindo conhecimento prévio do processo, embora tal conhecimento possa ser utilizado, caso esteja disponível [AGU04].

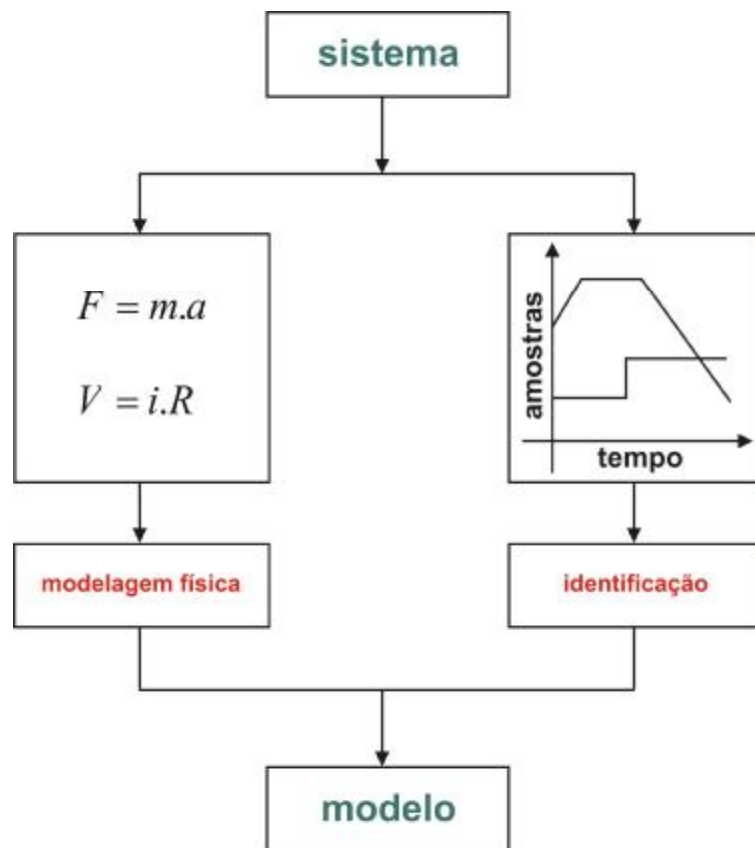


Figura 3.1. Princípios para a construção de um modelo matemático (modelagem).

Uma comparação pode ser realizada entre a modelagem através de leis físicas do processo e da identificação de sistemas. A vantagem da modelagem matemática é que ela permite visualizar características intrínsecas ao sistema, sendo que, na maioria dos casos, os parâmetros do modelo têm significado físico. A desvantagem, é que a determinação do modelo pode ser uma tarefa difícil ou mesmo viável somente em longo prazo. Em casos em que o processo a ser modelado é complexo, a obtenção de modelos usando apenas conhecimento físico do mesmo é uma tarefa árdua. Nestes casos, pode ser mais apropriado utilizar técnicas de identificação de sistemas. No entanto, a modelagem baseada na física pode ser combinada com métodos de identificação, para auxiliar na detecção da estrutura ou apenas para a determinação de coeficientes de um modelo, proporcionando desta forma, uma modelagem eficiente. Uma situação comum na prática é quando se tem a estrutura do modelo, mas parâmetros do mesmo são variáveis, ou quando alguns deles não são conhecidos, devendo ser estimados [CAS99], [COE04], [AGU04], [AST90].

Obtido o modelo (modelagem matemática), este é utilizado para responder às questões sobre o sistema sem a realização de experimentações práticas. Através deste modelo, pode-se calcular ou decidir como o sistema comporta-se sob determinadas condições operacionais.

A utilização do modelo para a simulação do sistema constitui-se de um procedimento de baixo custo e seguro para experimentar o sistema. Entretanto, a validade (adequação) dos resultados de simulação depende da qualidade do modelo matemático do sistema [COE04].

Alguns dos diferentes propósitos para a utilização de modelos matemáticos em automação industrial são [COE04]:

- Previsão: é a tentativa de prever o comportamento futuro do sistema, mas está diretamente ligada à precisão do modelo e aos efeitos de perturbações que podem atuar sobre o sistema;
- Análise e projeto de sistemas de controle: proporciona um vasto campo para a aplicação em modelagem e identificação na sintonia de controladores clássicos, síntese de algoritmos de controle adaptativos e preditivos e na estimação do estado de variáveis não-mensuráveis;
- Supervisão: através do modelo matemático, realizam-se simulações para testar e avaliar características operacionais do sistema, para o projeto de engenharia ou para o treinamento de operadores. Muitas vezes é também utilizado na detecção de erros e diagnósticos;

- Otimização: é empregada para o melhoramento do sistema como um todo, mas para isso a otimização de sistemas necessita de modelos matemáticos precisos.

É importante salientar que um modelo matemático para um dado sistema não é único. Deve-se observar se o modelo escolhido é capaz de explicar os dados medidos satisfatoriamente, podendo ser usado na compreensão de fenômenos observados no sistema real. Além disso, é fundamental que o modelo seja capaz de atender ao objetivo final para o qual foi obtido. Em alguns casos, o objetivo principal da modelagem de um sistema físico é auxiliar ou tornar eficiente o projeto de controladores. Em outros casos, o próprio modelo é o objetivo principal para, por exemplo, estudar e analisar um determinado processo ou para prever dados futuros [CAS99], [OGA93].

### 3.2 Etapas para Identificação de Sistemas

A identificação de sistemas é um procedimento alternativo. A motivação básica é simples. Supondo que estejam disponíveis os sinais de entrada  $u(t)$ , e de saída  $y(t)$ , de um sistema real qualquer onde  $t$  é o tempo, a identificação de sistemas se propõe a obter um modelo matemático que explique, pelo menos em parte e de forma aproximada, a relação de causa e efeito presente nos dados. Ou seja, tenta-se responder à pergunta: existe um modelo que ao ser excitado por  $u(t)$ , resulta em  $y(t)$ ? [CAS99], [LJU87], [AGU04].

Em projetos ou situações que utilizam apenas os dados de entrada  $u(t)$  e de saída  $y(t)$ , para a tentativa de se identificar um sistema, é denominada de identificação caixa-preta.

Existem vários aspectos importantes envolvidos em problemas reais de identificação. Entretanto, as principais etapas de um problema de identificação são [LJU87], [AGU04]:

1. Testes dinâmicos e coleta de dados: uma vez que a identificação se propõe a obter modelos a partir de dados, é necessário gerar tais dados. Muitas vezes, os únicos dados disponíveis são dados de condições de operação normal. Em outras situações, entretanto, é possível e desejável efetuar testes de maneira a extrair informação dinâmica do sistema. Os problemas importantes relacionados a esta etapa são a

escolha dos sinais de excitação, a execução do teste e a escolha do período de amostragem;

2. Escolha da representação matemática a ser utilizada: esta etapa não tem sido cumprida criteriosamente na identificação de sistemas dinâmicos não-lineares. Tal ausência pode ser em parte justificada pela falta de estudos mais consistentes a respeito das características de cada representação. Existe uma grande variedade de representações não-lineares que, ao menos em princípio, podem ser utilizadas na identificação de sistemas. Pode-se citar diversas representações para a identificação não-linear. Dentre elas, incluem-se as séries de Volterra, os modelos orientados por blocos, polinomiais, bilineares e redes neurais artificiais [GRE89], [WIE58], [BIL80]. Assim, a escolha de estrutura fica a critério do modelador, pois não existe um método ou uma representação que possa ser considerada “ótima”;
3. Determinação de estrutura do modelo: para modelos lineares, a escolha da sua estrutura se restringe, basicamente, à escolha do número de pólos e de zeros, bem como à determinação do atraso puro de tempo. Mas para modelos não-lineares, cada representação é melhor para identificar ou representar um tipo de sistema de acordo com o tipo de não linearidade presente nos dados. Não se pode dizer, de imediato, que uma das abordagens seja adequada, mesmo porque, para algumas representações, não existe uma maneira sistemática para determinar a estrutura ideal de um dado sistema. A escolha do tipo de modelo também depende da existência ou não de informação *a priori*, do propósito para o qual o modelo está sendo utilizado e da complexidade do sistema;
4. Estimação de parâmetros: esta etapa inicia-se com a escolha do algoritmo a ser utilizado. Desta maneira, sabem-se quais e como se estimar os parâmetros adequados para a identificação do sistema em questão;
5. Validação do modelo: após a obtenção do modelo, é necessário verificar se este incorpora ou não as características de interesse do sistema original. Além disso, é interessante poder comparar vários modelos entre si e decidir se há algum candidato

significativamente melhor que os demais. Esta etapa é certamente subjetiva, sendo que o resultado da validação depende fortemente da aplicação pretendida para o modelo matemático e da quantidade de informação disponível sobre o sistema original, se houver.

Pode-se também descrever um laço em diagrama de blocos para uma identificação de sistemas conforme mostrado na figura 3.2 [LJU87].

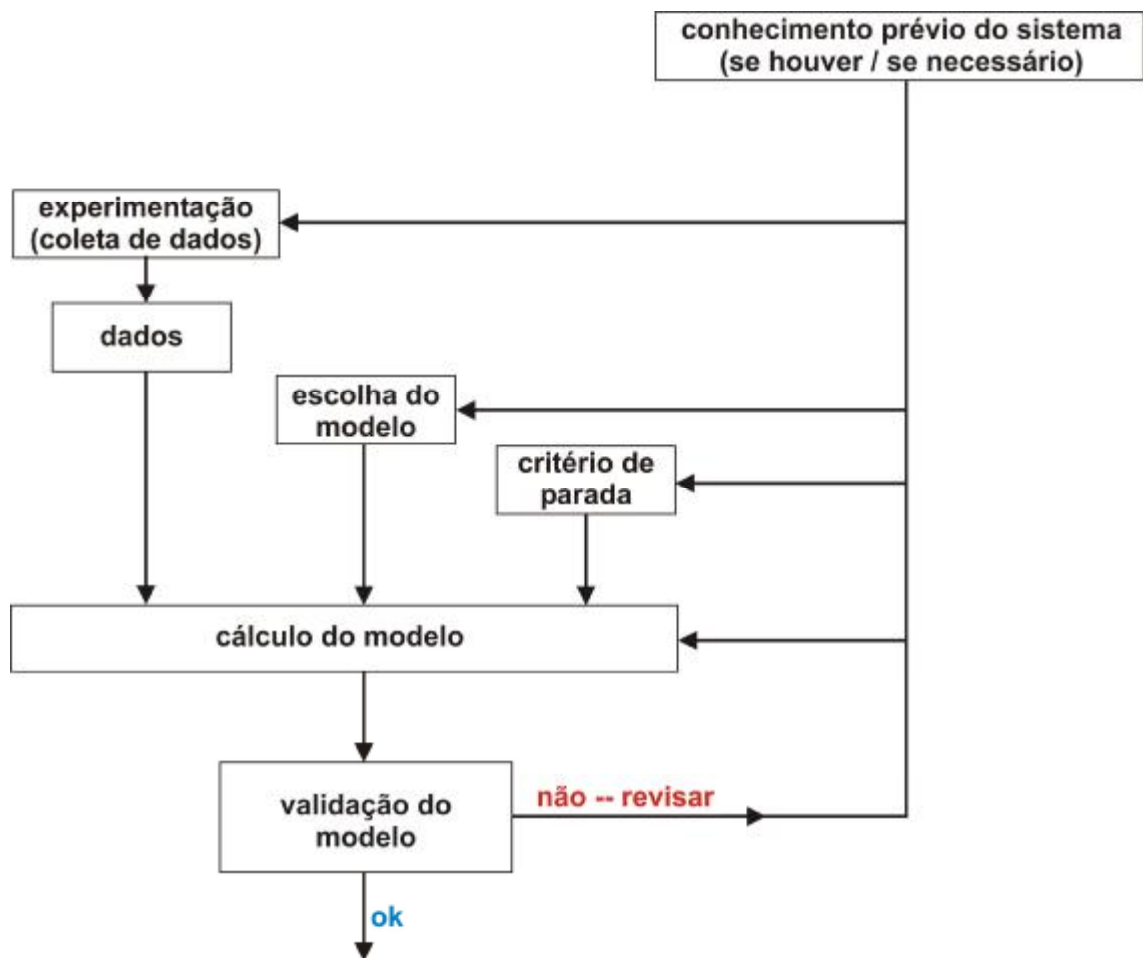


Figura 3.2. Diagrama de blocos para uma identificação de sistemas.



### 3.3 Representações de Sistemas Não-Lineares

O problema de identificação no domínio do tempo para o caso linear ou não-linear é determinar a relação entre as entradas e saídas passadas e as saídas futuras. Se um número finito de entradas  $u(t)$  e saídas  $y(t)$  passadas forem coletadas em um vetor  $\mathbf{j}(t)$ , tal que

$$\mathbf{j}(t) = [y(t-1)\mathbf{K}y(t-n_y) \quad u(t-1)\mathbf{K}u(t-n_u)]^T \quad (3.1)$$

onde  $n_y$  é o número de saídas atrasadas e  $n_u$  é o número de entradas atrasadas, então o problema deve ser compreendido como o relacionamento da função  $f$  entre a próxima saída  $y(t)$  e  $\mathbf{j}(t)$ .

$$y(t) \rightarrow f(\mathbf{j}(t)) \quad (3.2)$$

Para se obter essa compreensão, é necessário um conjunto de dados observados, o qual consiste em dados de entradas  $u(t)$  e saídas  $y(t)$  observadas do sistema real para que o vetor  $\mathbf{j}(t)$  possa ser construído. A função  $f$  pode ser qualquer função, linear ou não-linear, e definirá a estrutura do modelo [LJU94], [AGU04], [AHM00].

Existe uma grande variedade de representações não-lineares que, pelo menos em princípio, podem ser utilizadas na identificação de sistemas. Esta seção “Representações de Sistemas Não-Lineares” não objetiva listar um grande número de representações, nem mesmo detalhar qualquer uma delas, mas objetiva fornecer uma visão geral de algumas representações existentes.

#### 3.3.1 Séries de Volterra

A saída  $y(t)$  de um sistema não-linear com entrada  $u(t)$  pode ser representada pela chamada série de Volterra definida como:

$$y(t) = h_0 + \sum_{j=1}^{\infty} \int_{-\infty}^{\infty} \mathbf{L} \int_{-\infty}^{\infty} h_j(t_1, \mathbf{K}, t_j) \prod_{i=1}^j u(t-t_i) dt_i \quad (3.3)$$

sendo que as funções  $h_j$  são os núcleos (*kernels*) e claramente são generalizações não-lineares da resposta ao impulso  $h_1(t)$ . De fato, para um sistema linear  $j=1$ , a equação (3.3) se reduz à integral de convolução. Esta equação (3.3) representa a soma das saídas dos sub-

sistemas chamados de funcionalidades de Volterra que pode ser representado graficamente como ilustra a figura 3.3 [BIL80], [AGU04].

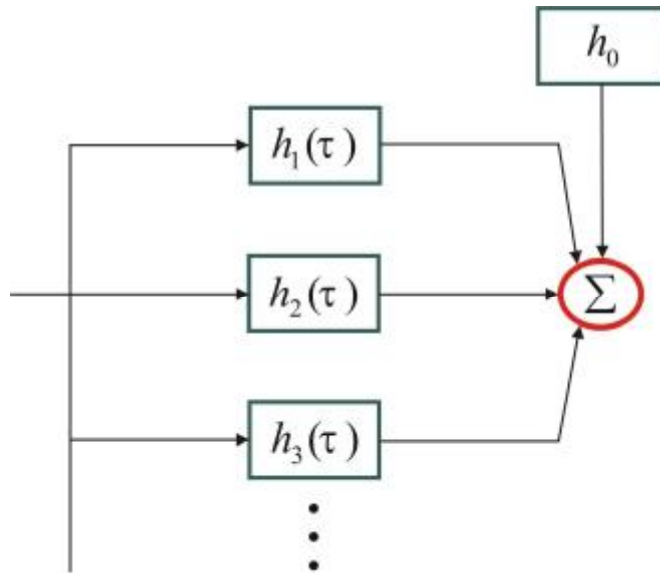


Figura 3.3. Representação gráfica da série de Volterra.

A dificuldade prática da aplicação da série de Volterra na identificação de sistemas é que, mesmo para sistemas “pouco” não-lineares, o número de parâmetros a serem determinados é grande. A necessidade de parâmetros é consequência da série de Volterra tentar explicar a saída  $y(t)$  apenas em função da entrada. Uma maneira de se reduzir o número de parâmetros é utilizar bases de funções ortonormais como forma de parametrizar os núcleos. Outra alternativa, para reduzir ainda mais o número de parâmetros, é utilizar valores da própria saída, além de valores da entrada, para determinar  $y(t)$ , ou seja, utilizar recorrência ou auto-regressão da saída [AGU04]. Os modelos NARX, têm essas características, e são discutidos adiante.

### 3.3.2 Modelos de Hammerstein e de Wiener

Dois tipos de modelos não-lineares que foram populares até duas ou três décadas atrás são os modelos de Hammerstein e de Wiener [GRE89], [WIE58], [NEP02]. Ambos são uma composição de um modelo dinâmico linear  $H(s)$  em cascata com uma função estática não-

linear  $f(\cdot)$ . No caso do modelo de Hammerstein, a não-linearidade estática precede o modelo dinâmico linear, ou seja:

$$U^*(s) = f(U(s)) \quad e \quad Y(s) = H(s)U^*(s) \quad (3.4)$$

onde  $U(s)$  e  $Y(s)$  são a entrada e a saída do sistema, respectivamente, no domínio da frequência.

No caso do modelo de Wiener, o modelo dinâmico linear precede a não-linearidade estática, isto é:

$$Y^*(s) = H(s)U(s) \quad e \quad Y(s) = f(Y^*(s)) \quad (3.05)$$

Em problemas de identificação, apenas tem-se acesso aos sinais  $y(t)$  e  $u(t)$  e a função  $f(\cdot)$  é desconhecida [AGU04], [CAS99].

### 3.3.3 Classificação de Modelos Não-Lineares

O principal problema na identificação de sistemas é determinar uma estrutura do modelo apropriada, dentro da qual um bom modelo deve ser encontrado. A determinação de um modelo dentro de uma estrutura dada (estimação dos parâmetros) é, na maioria dos casos, um problema fácil de ser resolvido. A regra básica na estimação é a seguinte: *não estimar o que você já conhece*. Em outras palavras, deve-se utilizar conhecimento prévio e características físicas sobre o sistema em questão ao selecionar a estrutura do modelo. É habitual classificar o conhecimento prévio de um sistema em três níveis, os quais recebem códigos baseados em cores, como seguem [SJO95b]:

- Modelos Caixa-Branca (*white box*): neste caso, o modelo é perfeitamente conhecido. Assim, é possível construí-lo inteiramente através de conhecimento prévio e características físicas do sistema;
- Modelos Caixa-Cinza (*gray box*): neste caso, algumas características físicas do sistema podem ser utilizadas, mas outros parâmetros devem ser determinados através de dados observados do sistema. É útil considerar dois sub-casos:

- Modelagem Física: uma estrutura do modelo pode ser construída através da fundamentação física, a qual terá parâmetros a serem estimados através dos dados observados. Isto pode ser um modelo de espaço de estados;
- Modelagem Semi-Física: características físicas são utilizadas para sugerir uma determinada combinação não-linear dos dados observados criando novos dados (sinais). Então esses novos dados estão sujeitos à estrutura do modelo do tipo caixa-preta.
- Modelos Caixa-Preta (*black box*): nenhuma característica física do sistema é utilizada, mas as escolhas da estrutura do modelo pertencem às famílias que são conhecidas, para se ter uma boa flexibilidade e por ter sido bem “sucedida no passado”.

Para esta dissertação o tipo de modelo utilizado é o modelo caixa-preta com representação utilizando a RN-RBF (rede neural de base radial).

### 3.3.3.1 Modelos Caixa-Preta

Para modelos lineares do tipo caixa-preta, o objetivo é descrever ou aproximar a resposta ao impulso, o qual é apenas mapeado através de  $\mathfrak{R} \rightarrow \mathfrak{R}^m$  (onde  $p$  é o número de saídas e  $m$  é o número de entradas). Tipicamente, para modelos lineares existem funções que dominam este tipo de aplicação, as quais são bem conhecidas e discutidas na literatura e obtêm com sucesso as estruturas lineares do tipo caixa-preta [AGU04].

Para modelos não-lineares do tipo caixa-preta, encontrar a estrutura do modelo se torna uma tarefa mais difícil do que para o caso linear. A principal razão para esta dificuldade é que nada é excluído, e existe um campo vasto de possibilidades de modelos para serem descritos e testados, tais como: redes neurais artificiais, *wavelets* e sistemas *fuzzy* [SJO95b].

### 3.3.3.2 Estruturas Não-Lineares Caixa-Preta

Neste caso, o problema de identificação consiste em obter as entradas  $u(t)$  e as saídas  $y(t)$  do sistema:

$$u = [u(1), u(2), \mathbf{K}, u(t)] \quad (3.6)$$

$$y = [y(1), y(2), \mathbf{K}, y(t)] \quad (3.7)$$

Procura-se uma relação entre os dados observados do passado  $[u(t-1), y(t-1)]$  e as saídas futuras  $y(t)$ :

$$y(t) = g(u(t-1), y(t-1)) + v(t) \quad (3.8)$$

O termo adicionado  $v(t)$  serve para o fato de que a próxima saída  $y(t)$  não é exatamente em função das saídas passadas. Entretanto, o objetivo é que  $v(t)$  seja o menor possível, assim  $g(u(t-1), y(t-1))$  contém uma boa previsão de  $y(t)$  dos dados passados.

A equação (3.8) modela genericamente, em tempo discreto, o sistema dinâmico. Então, como se pode encontrar a função  $g$  a partir da equação (3.8). Inicialmente, parametriza-se esta família de funções com um parâmetro finito dimensional denominado de vetor  $q$ :

$$y(t) = g(u(t-1), y(t-1), q) \quad (3.9)$$

A parametrização da função  $g$  com um vetor finito dimensional  $q$  é geralmente uma aproximação. Assim a qualidade do  $q$  pode naturalmente ser avaliada por meio de ajuste entre o modelo e os dados observados:

$$y(t) = \|y(t) - g(u(t-1), y(t-1), q)\|^2 \quad (3.10)$$

A norma ( $\|\cdot\|$ ) e a maneira real de se obter um  $q$  mínimo podem diferir, mas a maioria dos esquemas de identificação de sistemas segue este conceito [AGU04], [SJO95b].

Neste momento, a família da estrutura do modelo (equação (3.9)) é realmente genérica, e assim pode-se escrever  $g$  como uma concatenação de dois mapeamentos: (i) um que pegue um número grande de observações passadas  $u$  e  $y$  para mapeá-las dentro de um vetor finito dimensional  $j(t)$  de dimensão fixa; e (ii) outro que pegue este vetor e leve ao espaço de saída:

$$g(u(t-1), y(t-1), q) = g(j(t), q) \quad (3.11)$$

onde

$$j(t) = j(u(t-1), y(t-1)) \quad (3.12)$$

e  $j(t)$  é denominado de vetor de regressores e seus componentes são denominados regressores.

A escolha do mapeamento não-linear na equação (3.9) pode ser decomposta em dois problemas parciais para sistemas dinâmicos:

1. Como escolher o vetor de regressores  $j(t)$  das entradas e saídas passadas;
2. Como escolher o mapeamento não-linear  $g(j)$  de espaço do regressor para o espaço de saída.

Para esta dissertação o mapeamento não-linear escolhido foram as funções de base radial através da RN-RBF, esta será descrita no próximo capítulo. Assim, os próximos tópicos nesta etapa de identificação tratam apenas sobre os regressores e das representações destes baseados em redes neurais.

### 3.3.3.3 Regressores

Para o estudo sobre os regressores, deve-se inicialmente, apresentar uma revisão do caso linear.

#### 3.3.3.3.1 Uma Revisão de Modelos Lineares Caixa-Preta

O modelo dinâmico mais simples é o modelo finito da resposta ao impulso (FIR):

$$y(t) = B(q)u(t) + e(t) = b_1u(t-1) + \mathbf{K} + b_nu(t-n) + e(t) \quad (3.13)$$

onde  $e(t)$  é o ruído,  $q$  é o operador de deslocamento (*shift operator*), e  $B$  é um polinômio em  $q^{-1}$ . O preditor correspondente  $\hat{y}(t|q) = B(q)u(t)$  é baseado no vetor de regressão, desta maneira:

$$j(t) = [u(t-1), u(t-2), \mathbf{K}, u(t-n)] \quad (3.14)$$

com  $n \rightarrow \infty$ , pode-se então descrever a dinâmica de sistemas lineares.

As estruturas lineares do tipo caixa-preta utilizadas na prática são todas variantes da equação (3.13), apenas alteram-se na maneira de escolher os pólos do sistema e na maneira de descrever as características do ruído.

Os modelos mais comuns podem ser resumidos, como em [LJU87] através de:

$$A(q)y(t) = \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} e(t) \quad (3.15)$$

Os casos especiais desta equação (3.15) são conhecidos como modelos:

- Box-Jenkins (BJ): para  $A = 1$ ;
- ARMAX: para  $F = D = 1$ ;
- Erro da saída (OE: *Output-Error*): para  $A = C = D = 1$ ;
- ARX: para  $F = C = D = 1$ ;

O preditor associado com a equação (3.15) pode ser dado pela regressão pseudo-linear, como em:

$$\hat{y}(t | \mathbf{q}) = \mathbf{q}^T \mathbf{j}(t, \mathbf{q}) \quad (3.16)$$

Os regressores, isto é, os componentes de  $\mathbf{j}(t)$  nestes casos gerais, podem ser dados por:

1.  $u(t-k)$ : associado ao polinômio  $B$ ;
2.  $y(t-k)$ : associado ao polinômio  $A$ ;
3.  $\hat{y}_u(t-k | \mathbf{q})$ : saídas simuladas apenas através das entradas  $u$  passadas (associado ao polinômio  $F$ );
4.  $e(t-k) = y(t-k) - \hat{y}(t-k | \mathbf{q})$ : erros de predição (associado ao polinômio  $C$ );
5.  $e_u(t-k) = y(t-k) - \hat{y}_u(t-k | \mathbf{q})$ : erros de simulação (associado ao polinômio  $D$ ).

Um modelo de espaço de estados linear na forma de preditor é dado por:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + K(y(t) - Cx(t)) \\ y(t) &= Cx(t) + e(t) \end{aligned} \quad (3.17)$$

e pode ser descrito também como uma regressão pseudo-linear (equação (3.16)), com o preditor  $\hat{y}(t | \mathbf{q}) = Cx(t)$ , onde os estados  $x$  são os regressores.

### 3.3.3.3.2 Regressores para Modelos Dinâmicos Não-Lineares Caixa-Preta

Os regressores descritos dão toda liberdade necessária ao caso linear do tipo caixa-preta, e é natural utilizar também esta liberdade no caso não-linear. Assim pode-se trabalhar com uma estrutura do tipo:

$$\hat{y}(t|\mathbf{q}) = g(\mathbf{j}(t), \mathbf{q}) \quad (3.18)$$

onde  $g$  é alguma função não-linear parametrizada por  $\mathbf{q}$  e os componentes de  $\mathbf{j}(t)$  são similares aos regressores já descritos. Para os dois primeiros casos de regressores, entradas-saídas  $u(t-k)$  e  $y(t-k)$  são variáveis medidas e não causam nenhum problema por serem incluídas. Para os outros três casos baseados em saídas passadas do modelo caixa-preta  $\hat{y}(t-k|\mathbf{q})$ , deve-se escrever  $\mathbf{j}(t, \mathbf{q})$  em vez de  $\mathbf{j}(t)$  na equação (3.18). Uma questão a ser levantada é: como a saída  $\hat{y}_u(t-k|\mathbf{q})$  simulada é computada se a rede produz as saídas  $\hat{y}(t-k|\mathbf{q})$ ? A resposta a esta questão, é que a saída do modelo da equação (3.18)  $\hat{y}(t|\mathbf{q})$  é igual a  $\hat{y}_u(t|\mathbf{q})$  se todas as saídas medidas  $y(t-k)$  nos regressores forem substituídas pela última saída calculada  $\hat{y}_u(t-k|\mathbf{q})$ .

Seguindo a nomenclatura dos modelos lineares, os modelos não-lineares podem ser descritos da seguinte maneira:

- NFIR (*non-linear finite impulse response*): utiliza apenas  $u(t-k)$  como regressores;
- NARX: utiliza  $u(t-k)$  e  $y(t-k)$  como regressores;
- NOE (*non-linear output error model*): utiliza  $u(t-k)$  e  $\hat{y}_u(t-k|\mathbf{q})$  como regressores. Neste caso a saída do modelo é igual a  $\hat{y}(t|\mathbf{q})$ ;
- NARMAX: utiliza  $u(t-k)$ ,  $y(t-k)$  e  $e(t-k|\mathbf{q})$  como regressores;
- NBJ (*non-linear Box-Jenkins model structure*): utiliza  $u(t-k)$ ,  $\hat{y}(t-k|\mathbf{q})$ ,  $e(t-k|\mathbf{q})$  e  $e_u(t-k|\mathbf{q})$  como regressores. Neste caso a saída  $\hat{y}_u$  simulada é obtida como a saída da equação (3.18), utilizando a mesma estrutura e substituindo  $e$  e  $e_u$  por zero no vetor de regressão  $\mathbf{j}(t)$ ;
- Espaço de estados não-linear: utiliza componentes passadas de saídas virtuais, isto é, valores nos nós internos da rede que não correspondem à variável de saída.

Em [NAR90], outras notações são utilizadas para os mesmo modelos, quando são utilizados em conjunto com redes neurais, conforme descritas na próxima seção.



### 3.3.3.4 Representações de Modelos Baseados em Redes Neurais

A utilização de redes neurais na identificação de sistemas dinâmicos foi proposta em [NAR90]. Os referidos autores propuseram quatro representações diferentes para as redes neurais:

- Representação 1:

$$y(t) = \sum_{n_u}^{n_y} a_i y(t-i) + g(u(t-1), u(t-2), \mathbf{K}, u(t-n_u)) \quad (3.19)$$

- Representação 2:

$$y(t) = f(y(t-1), y(t-2), \mathbf{K}, y(t-n_y)) + \sum_{i=1}^{n_u} b u(t-i) \quad (3.20)$$

- Representação 3:

$$y(t) = f(y(t-1), y(t-2), \mathbf{K}, y(t-n_y)) + g(u(t-1), u(t-2), \mathbf{K}, u(t-n_u)) \quad (3.21)$$

- Representação 4:

$$y(t) = f(y(t-1), y(t-2), \mathbf{K}, y(t-n_y), u(t-1), u(t-2), \mathbf{K}, u(t-n_u)) \quad (3.22)$$

As figuras 3.4 a 3.7 ressaltam as diferenças básicas para cada representação proposta, através de diagrama de blocos. No primeiro caso (figura 3.4), toda a não-linearidade do sistema está em função da entrada  $u(t)$ . No segundo caso (figura 3.5), a não-linearidade está representada em função da saída  $y(t)$ . No terceiro caso (figura 3.6), aparecem funções não-lineares tanto para entrada  $u(t)$  quanto para a saída  $y(t)$ , podendo estas ser diferentes. No quarto caso (figura 3.7), a função não-linear  $f(\cdot)$  inclui tanto os sinais da entrada  $u(t)$  quanto da saída  $y(t)$  [NAR90].

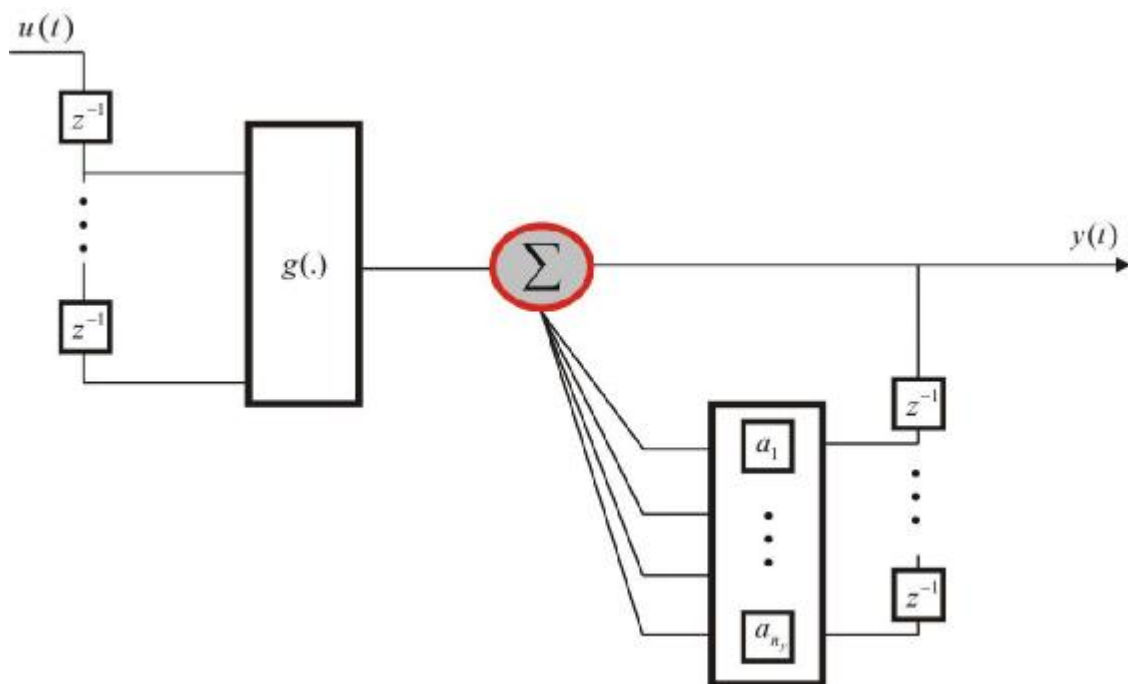


Figura 3.4. Representação 1 de redes neurais.

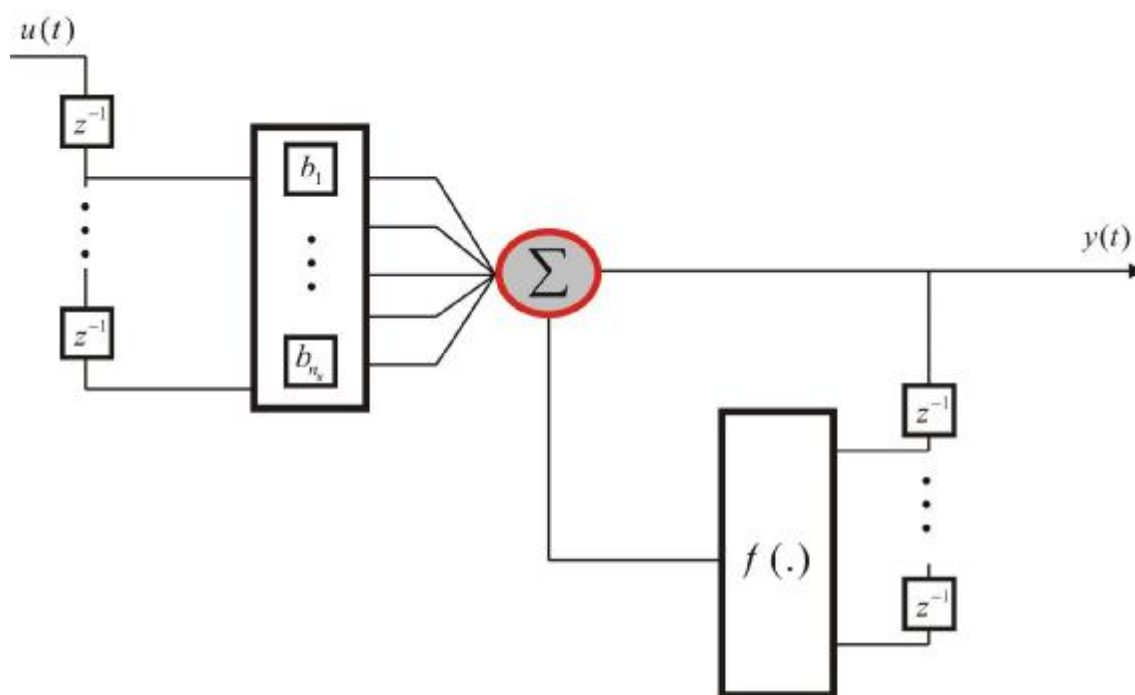


Figura 3.5. Representação 2 de redes neurais.

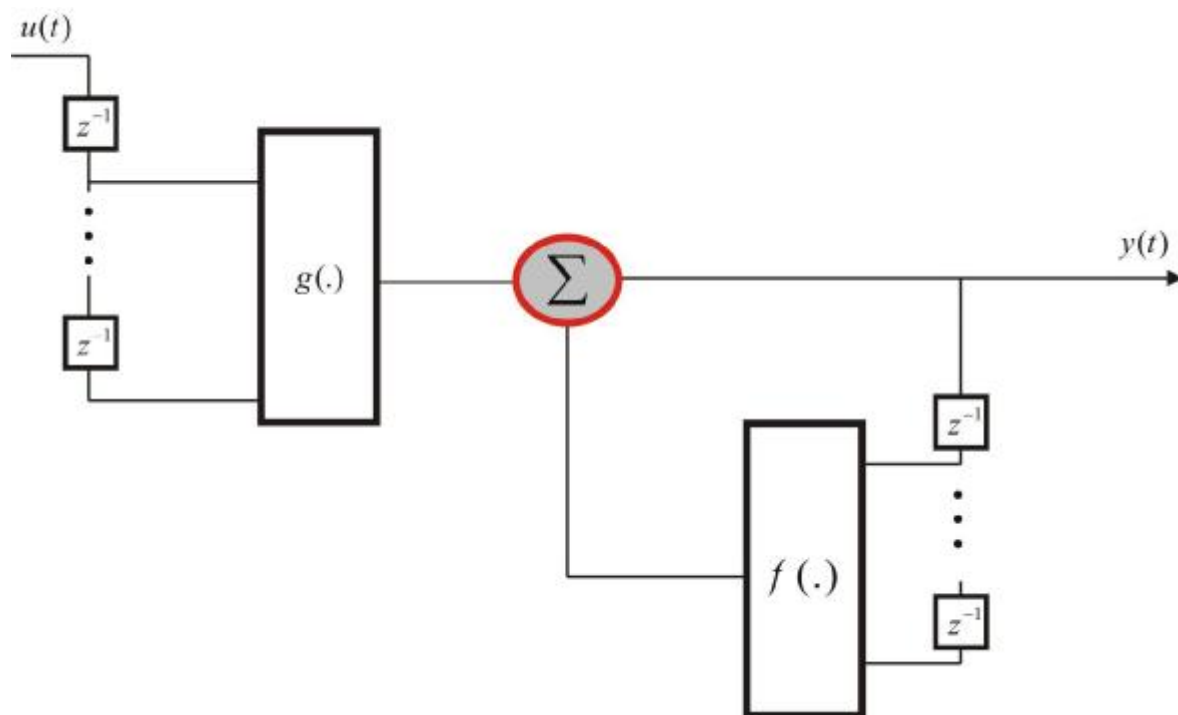


Figura 3.6. Representação 3 de redes neurais.

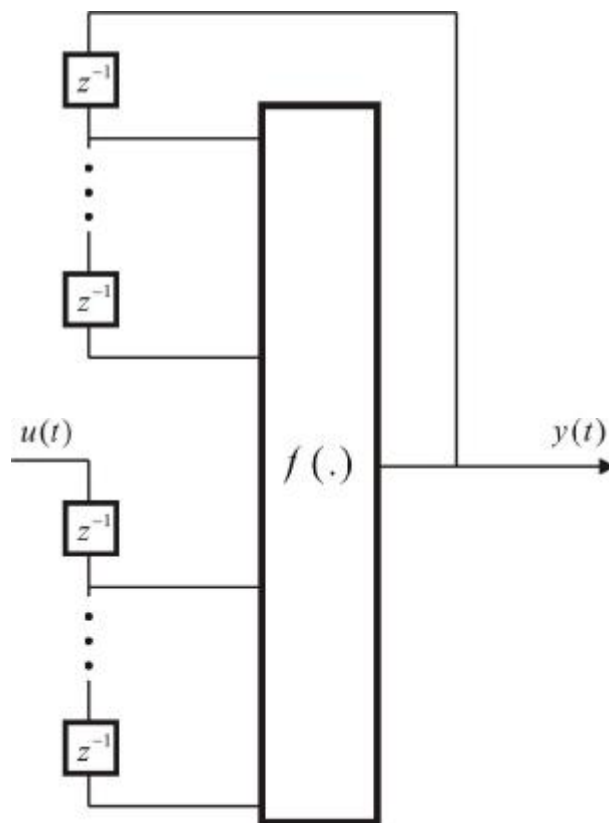


Figura 3.7. Representação 4 de redes neurais.

As redes neurais inicialmente foram concebidas para mapeamento de funções estáticas e reconhecimento de padrões. Com o desenvolvimento da teoria de redes neurais, o aparecimento de novas redes e o aprimoramento de algoritmos de treinamento, estas vêm sendo cada vez mais utilizadas na identificação de sistemas dinâmicos [HAY01].

#### 3.3.3.4.1 Modelo de Redes Neurais para Identificação de Sistemas Não-Lineares

Seguindo o mesmo tipo de escolha dos regressores, é possível defini-los para as estruturas dinâmicas dos modelos de redes neurais, como segue:

- Modelo NNFIR (*neural network finite impulse response*): utiliza apenas  $u(t-k)$  como regressores;
- Modelo NNARX (*neural network autoregressive with exogenous inputs*): utiliza  $u(t-k)$  e  $y(t-k)$  como regressores;
- Modelo NNOE (*neural network output error*): utiliza  $u(t-k)$  e  $\hat{y}_u(t-k|\mathbf{q})$  como regressores;
- Modelo NNARMAX (*neural network autoregressive moving average with exogenous inputs*): utiliza  $u(t-k)$ ,  $y(t-k)$  e  $\hat{y}(t-k|\mathbf{q})$  como regressores;
- Modelo NNBJ (*neural network Box-Jenkins*): utiliza  $u(t-k)$ ,  $\hat{y}(t-k|\mathbf{q})$ ,  $\hat{y}_u(t-k|\mathbf{q})$  e  $\hat{y}(t-k|\mathbf{q})$  como regressores.

Para representações de redes neurais, o modelo NARX é chamado de modelo série-paralelo, e o modelo NOE é chamado de modelo paralelo. Os modelos NOE, NBJ, NARMAX e espaço de estados não-linear, correspondem as estruturas recorrentes, pois o vetor de regressão consiste de saídas passadas (atrasadas) do modelo.

Para identificação de sistemas, considera-se o sistema sendo observável e onde medidas de entrada e saída estão disponíveis. Considera-se, também, um sistema dinâmico possuindo entrada exógena  $u(t) \in \mathfrak{R}^m$  e saída  $y(t) \in \mathfrak{R}^n$ . Dependendo do tipo de entradas utilizadas, existem dois tipos de modelos a serem estruturados: o modelo paralelo e o modelo série-paralelo [SJO95a].

- Modelo paralelo: a sua saída é utilizada para criar um tempo de atraso nas entradas. Este modelo pode ser considerado inteiramente recorrente, e é capaz de realizar

predições em um curto período de tempo. O modelo é dito ter dinâmica interna (*internal dynamics*);

- O modelo série-paralelo: as saídas do sistema real são utilizadas como entradas ao modelo. Apenas predições de um-passo-à-frente são possíveis. O modelo é dito ter dinâmica externa (*external dynamics*).

Os modelos paralelo e série-paralelo estão ilustrados nas figuras 3.8 e 3.9, respectivamente. Em ambos os casos, a predição do erro do modelo comparada com as saídas reais do sistema é utilizada como uma medida para otimizar os parâmetros do modelo.

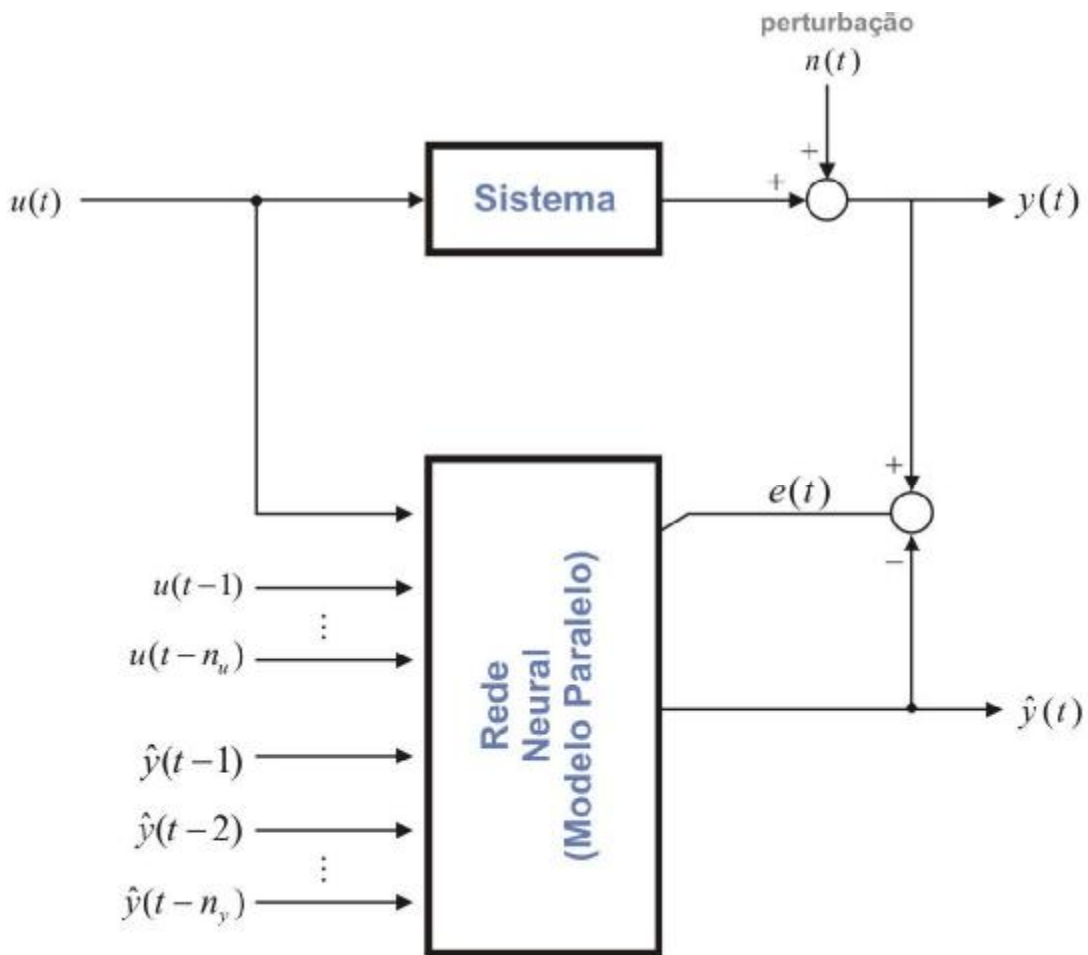


Figura 3.8. Modelo paralelo.

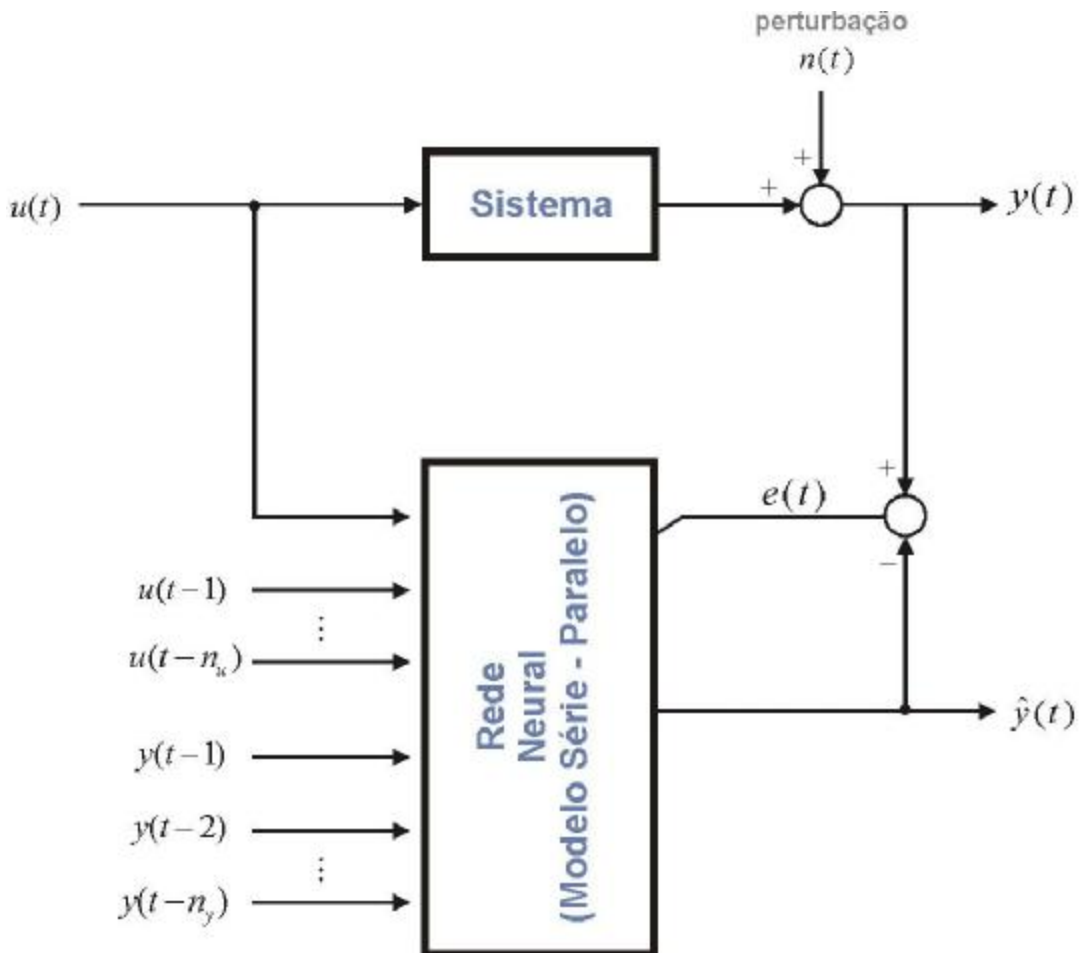


Figura 3.9. Modelo série-paralelo.

Nas figuras 3.8 e 3.9, o vetor  $\hat{y} \in \mathfrak{R}^m$  representa as saídas estimadas da rede neural,  $e(t)$  é o erro definido por  $e(t) = y(t) - \hat{y}(t)$ ,  $n(t) \in \mathfrak{R}^m$  é um ruído (ou perturbação),  $n_y$  e  $n_u$  são os números de saídas e entradas atrasadas (ordem do sistema), respectivamente.

Um detalhe importante desses modelos é que, inserindo na rede entradas e saídas atrasadas, a rede passa a ter informação sobre a dinâmica do sistema. Assim dependendo dos parâmetros de taxa de aprendizado e estabilidade da otimização, esta pode convergir para a saída desejada mais rapidamente.

# Capítulo 4

## 4 Redes Neurais Artificiais

### 4.1 Introdução

As redes neurais (RNs) ou redes neurais artificiais são sistemas paralelos e distribuídos, compostos por unidades de processamento simples (neurônios), que podem mapear funções matemáticas não-lineares. Tais unidades são dispostas em uma (ou mais) camada(s), interligadas por conexões. Na maioria dos modelos, estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. O funcionamento destas redes é inspirado em uma estrutura física concebida pela natureza: o cérebro humano [BRA00].

### 4.2 Breve Histórico

As primeiras RNs foram desenvolvidas em meados de 1940, baseadas no conhecimento sobre a arquitetura do cérebro humano. Esta pesquisa foi realizada por Warren McCulloch e Walter Pitts, os quais são considerados os inventores do modelo de uma RN, cujo título da publicação era *A Logical Calculus of the Ideas Immanent in Nervous Activity*. Este modelo inclui uma função não-linear na ativação do neurônio e um ponto inicial. Assim, o neurônio disparava somente se a entrada fosse maior que o ponto inicial. No final da década de 40, o biólogo Donald Hebb publicou *The Organization of Behavior*, onde propunha um princípio de aprendizagem. Este modelo mais tarde foi nomeado de rede de Hebb [BRA00], [GOR00].

Por volta de 1950 e 1960, cresceu o interesse nas RNs, dando origem a um grande número de modelos. Frank Rosenblatt criou o modelo neural Perceptron de uma única camada, e introduziu o conceito de aprendizado em RNs. No mesmo período, Bernard Widrow desenvolveu um modelo linear que foi denominado de *ADALINE*. Posteriormente, tal modelo foi generalizado para várias camadas passando a ser denominado de *MADALINE*. A maioria dos modelos de RNs atualmente utilizados foram inventados nestes anos. A principal contribuição para estes modelos foi realizada pela área da psicologia [BRA00], [GOR00].

Em 1970 esta pesquisa foi deixada de lado. O desenvolvimento das RNs não conduziu a uma revolução no entendimento do cérebro humano ou na própria inteligência artificial. O fato das RNs não conseguirem resolver o problema do XOR (*exclusive OR*), a ausência de algoritmos de otimização e a falta de aplicações reais conduziram à descrença nas RNs.

Widrow e Hoff introduziram o modelo de RNs na engenharia elétrica. Algumas abordagens de RNs iniciaram-se em 1970, mas obtiveram sucesso em 1980. Também Kohonen, Anderson, Carpenter, e principalmente Grossberg que escreveram vários trabalhos sobre matemática e biologia. Em 1980, apareceram trabalhos sobre algoritmos de aprendizado, novas topologias de rede e aproximadores universais de RNs. Foram inventadas as redes de Hopfield e Neocognitron, e a máquina de Boltzmann e vários outros trabalhos foram realizados na área de combinação de RNs com funções de base radial [GOR00] [KHA90]. E em 1987, ocorreu a primeira conferência na área de RNs, organizada pela IEEE composta por mais de 200 pesquisadores.

### **4.3 Neurônio Biológico**

O cérebro pode ser visto como um computador (sistema de processamento de informação) altamente complexo, não-linear e paralelo. Este tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar certos processamentos (por exemplo, reconhecimento de padrões, percepção e controle motor) mais rapidamente que o melhor computador digital hoje existente.

O cérebro é composto por células altamente diferenciadas denominadas neurônios. Cada neurônio possui um corpo celular, ou soma, dentro do qual encontram-se a maioria dos



seus orgânulos. Do corpo celular de cada neurônio partem prolongamentos que podem ser funcionalmente divididos em dendritos e axônios [HAY01], [NOR04].

O neurônio é uma célula altamente especializada, capaz de receber impulsos elétricos a partir de seus dendritos, processá-los no corpo celular e, finalmente, transmiti-los através de seu axônio (geralmente único) para os dendritos (ou mesmo para o corpo celular) de outros neurônios.

A conexão entre um axônio de um neurônio e um dendrito de outro é denominada sinapse. A sinapse é a unidade funcional básica envolvendo as membranas plasmáticas de dois neurônios, de modo a formar uma junção pontual (o tamanho de uma junção sináptica é menor do que 1 mm) e orientada do neurônio pré-sináptico para o pós-sináptico [KHA90].

A Figura 4.1 ilustra um neurônio do sistema nervoso central de um vertebrado, onde são também ilustrados e descritos os componentes de um neurônio [BRA00], [HAY01].

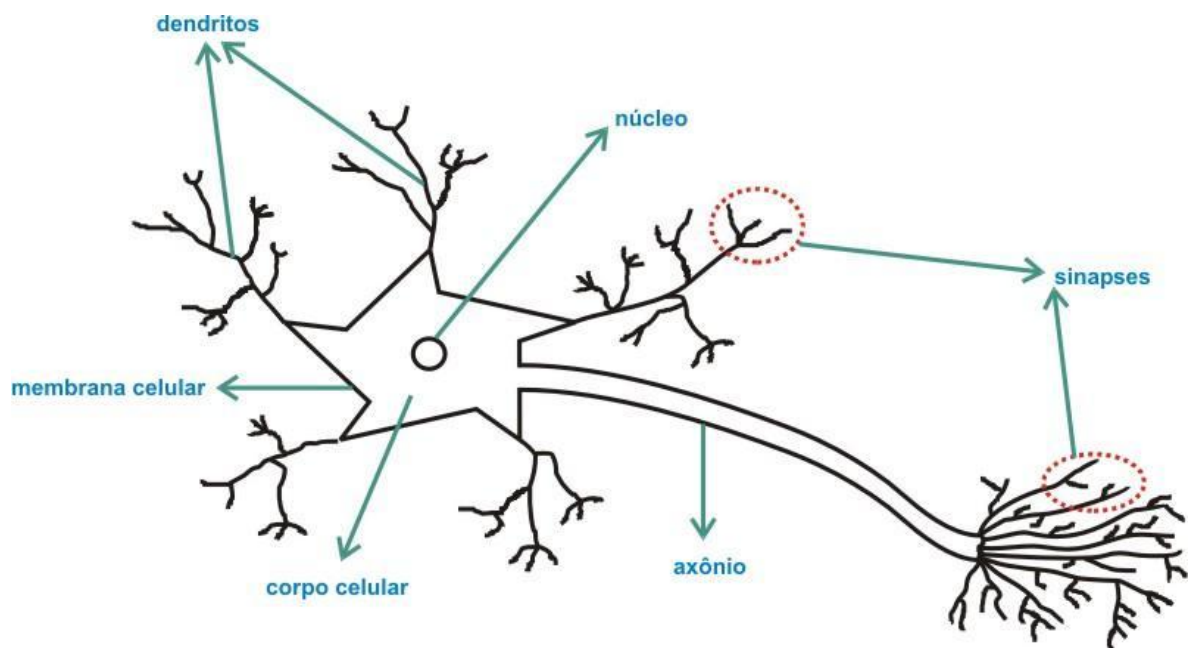


Figura 4.1. Neurônio do sistema nervoso central de um vertebrado.

- Corpo Celular (ou soma): nele ocorrem os processos metabólicos da célula, e é onde se encontra o núcleo que contém as informações sobre hereditariedade;
- Membrana Celular: desenvolve as funções biológicas normais de sustentação e de proteção de uma célula e permite que, através do diferencial de potencial entre sua

parte interior e exterior, exista a formação do potencial de ação axonal. Isto leva a membrana a uma despolarização, o que faz com que ela ultrapasse um valor pré-determinado denominado limiar (*threshold*) do neurônio (tipicamente um valor próximo de 40 mV). Esta diferença de potencial é sustentada por um processo de difusão de íons de sódio e de potássio através da membrana, sendo conhecido como bomba de sódio;

- Dendritos: são filamentos por onde o neurônio recebe informações. Frequentemente, os dendritos cobrem uma área maior do que o próprio corpo celular e formam uma árvore dendrital;
- Axônio: pode ser considerado como a projeção do corpo celular e às vezes é referido como fibra nervosa servindo para conectar a célula nervosa a outras células do sistema nervoso. Geralmente, a célula possui um único axônio;
- Sinapse: região de conexão entre um dendrito e um axônio.

#### 4.4 Redes Neurais Artificiais

As RNs são especificadas pela topologia, características dos neurônios e regras de treinamento. Estas regras de treinamento especificam como o conjunto de pesos iniciais deve ser adaptado para a rede aprender o comportamento de um sistema. Os algoritmos de treinamento (ou aprendizado) de uma RN ajustam iterativamente os pesos das conexões entre os neurônios até que os pares desejados de informações de entrada(s) e saída(s) sejam obtidos e as relações de causa e efeito possam ser estabelecidas. Se as condições mudarem de tal modo que o desempenho do modelo não seja mais adequado, pode-se submeter à RN a mais um treinamento sob estas novas condições de entrada(s) e saída(s) para corrigir seu desempenho. Assim sendo, podem-se apresentar dados a RN para uma atualização periódica, resultando num modelo auto-ajustável em linha [SCH97a], [REZ03], [CAS01].

Entre as aplicações usuais de RNs têm-se: reconhecimento e classificação de padrões, agrupamento (*clustering*), previsão de séries temporais, aproximação de funções, predição, otimização, setor militar (processamento de sinais para identificação de alvos e análise de imagens), sistemas especialistas, processamento de sinais (imagens, sensores, voz, caracteres,

visão, compressão de dados, filtragem de sinais), telecomunicações, manufatura, monitoramento de processos e robótica [VEE95], [REZ03], [COR01].

#### 4.4.1 Rede Neural *Perceptron* Multicamadas

Tipicamente, este tipo de rede neural consiste de um conjunto de unidades sensoriais (nós de fonte) que compõe a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída de nós computacionais. O sinal de entrada se propaga para frente através da rede, camada por camada. Esta rede é normalmente chamada rede neural perceptron de múltiplas camadas (RN-MLP), as quais representam uma generalização de perceptron de camada única [HAY01], [KRO96].

As RN-MLP têm sido aplicadas com sucesso para resolver diversos problemas, através do seu treinamento de forma supervisionada com o algoritmo de retropropagação do erro (*error back-propagation*), muito popular. Este algoritmo é baseado na regra de aprendizagem por correção de erro. Como tal, pode ser visto como uma generalização de um algoritmo de filtragem adaptativa igualmente popular: o algoritmo de mínimos quadrados.

Basicamente, a aprendizagem por retropropagação do erro consiste de dois passos através das diferentes camadas da rede: um passo à frente, a propagação, e um passo para trás, a retropropagação. No passo para frente, um padrão de atividade (vetor de entrada) é aplicado aos nós sensoriais da rede e seu efeito se propaga através da rede, camada por camada. Finalmente, um conjunto de saídas é produzido como resposta real ou estimada da rede. Durante o passo de propagação, os pesos sinápticos da rede são todos fixos. Durante o passo para trás, por outro lado, os pesos sinápticos são todos ajustados de acordo com uma regra de correção do erro. Especificamente, a resposta real da rede é subtraída de uma resposta desejada para produzir um sinal de erro. Este sinal é propagado para trás através da rede, contra a direção das conexões sinápticas, por isso o nome de retropropagação do erro. Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se mova para mais perto da resposta desejada, em um sentido estatístico. O processo de aprendizagem realizado com o algoritmo é denominado aprendizagem por retropropagação [HAY01], [HAG96].

Uma RN-MLP possui três características distintas:

1. O modelo de cada neurônio da rede inclui uma função de ativação não-linear. Um aspecto importante a se enfatizar aqui é que a não-linearidade é suave (diferenciável

em qualquer ponto). A forma normalmente utilizada de não-linearidade que satisfaz esta exigência é a função sigmóide definida por:

$$y_i = \frac{1}{1 + \exp(-v_i)} \quad (4.1)$$

onde  $v_i$  é o campo local induzido (soma ponderada de todas as entradas sinápticas acrescidas do bias) do neurônio  $i$ , e  $y_i$  é a saída do neurônio. A presença de não-linearidades é importante porque, do contrário, a relação de entrada-saída da rede poderia ser reduzida a um perceptron de camada única. Além disso, a utilização da função logística tem motivação biológica, pois procura levar em conta a fase refratária de neurônios reais;

2. A rede contém uma ou mais camadas de neurônios ocultos, que não são parte da entrada ou da saída da rede. Estes neurônios ocultos capacitam a rede a aprender tarefas complexas, extraíndo progressivamente as características mais significativas dos padrões (vetores) de entrada;
3. A rede exibe um alto grau de conectividade, determinado pelas sinapses da rede. A modificação na conectividade da rede requer a mudança no conjunto das conexões sinápticas ou de seus pesos.

É através da combinação destas características, juntamente com a habilidade de aprender da experiência através de treinamento, que o perceptron de múltiplas camadas deriva seu poder computacional. Estas mesmas características, entretanto, são também responsáveis pelas deficiências sobre o comportamento da rede. Primeiramente, a presença de uma forma distribuída de não-linearidade e a alta conectividade da rede torna difícil a análise teórica de uma RN-MLP. Também, a utilização de neurônios ocultos torna o processo de aprendizagem mais difícil de ser visualizado. Em sentido implícito, o processo de aprendizagem deve decidir quais características do padrão de entrada devem ser representadas pelos neurônios da camada oculta (intermediária). O processo de aprendizagem, conseqüentemente, torna-se complexo porque a busca deve ser conduzida em um espaço maior de funções possíveis, e deve ser feita a escolha entre representações alternativas do padrão de entrada. A figura 4.2, ilustra a estrutura geral de uma RN-MLP [HAY01], [BIS95], [JAN97].

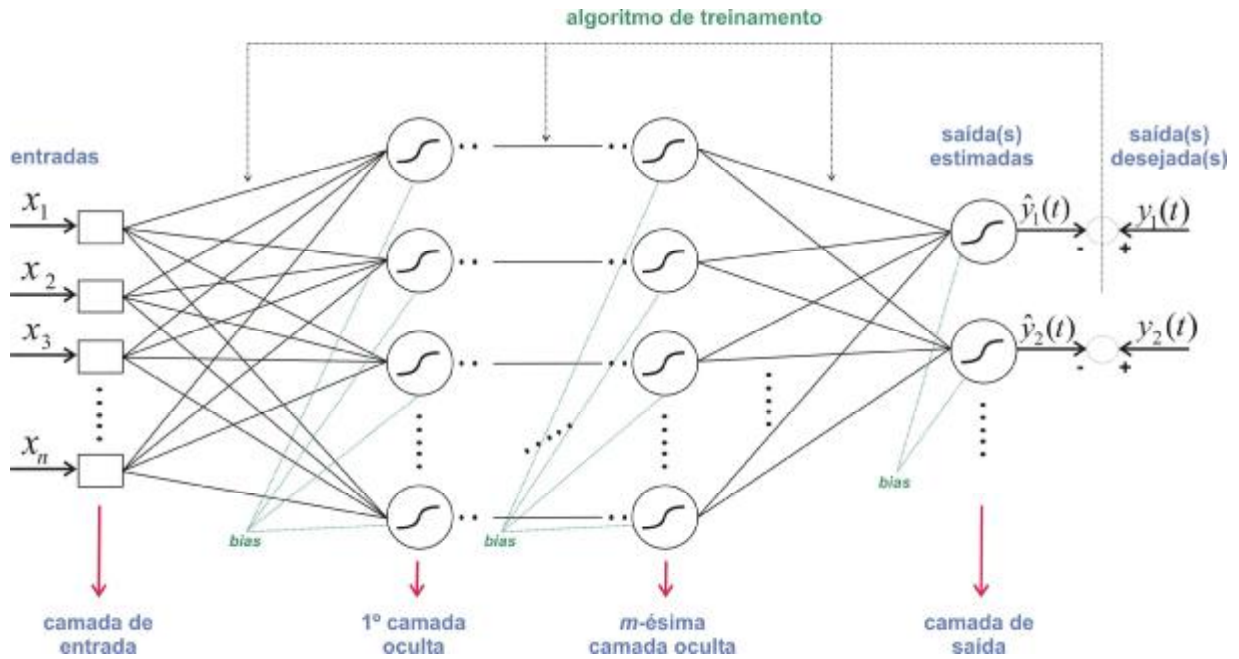


Figura 4.2. Estrutura geral de uma RN-MLP.

#### 4.4.2 Rede Neural de Função de Base Radial (RN-RBF)

A Rede Neural de Função de Base Radial (RN-RBF) do termo em inglês *Radial Basis Function Neural Network*, pode ser considerada como uma rede neural para um problema de ajuste de curva (aproximação) em um espaço de alta dimensionalidade. Desta maneira, aprender é equivalente a encontrar uma hiper-superfície em um espaço multidimensional que forneça o melhor ajuste para os dados de treinamento, com o critério de “melhor ajuste” sendo medido em um sentido estatístico. Então o processo de generalização equivale a utilizar esta superfície multidimensional para interpolar outros pontos que não pertençam aos dados de treinamento, mas estejam em sua vizinhança [HAY01], [ZHA96].

Para uma RN-RBF, as unidades ocultas fornecem um conjunto de “funções” que consistem em uma “base” arbitrária para os padrões (vetores) de entrada, quando estes são expandidos sobre o espaço oculto: estas funções são denominadas funções de base radial, das quais a rede deriva seu nome. Cada uma dessas funções é centrada em uma coordenada particular do espaço multidimensional dos pontos que compõem o espaço de dados de entrada. Cada uma destas coordenadas particulares caracteriza-se por definir o centro de uma

(entre várias possíveis) região de maior aglomeração de pontos ou grupos (*clusters*), do espaço de dados de entrada [HAY01], [BOR01].

A arquitetura da RN-RBF, em sua estrutura mais básica, é constituída de três camadas, com propósitos totalmente diferentes:

- Camada de entrada: é composta por nós de fonte (unidades sensoriais) que conectam a rede ao seu ambiente;
- Camada oculta: é única, aplica uma transformação não-linear do espaço de entrada para o espaço oculto (na maioria das aplicações, o espaço oculto é de alta dimensionalidade);
- Camada de saída: é linear, fornecendo a resposta da rede ao padrão (sinal) de ativação aplicada à camada de entrada.

A utilização de uma transformação não-linear seguida de uma transformação linear originou-se através do Teorema de Cover [COV65]. De acordo com este teorema, um problema de classificação de padrões disposto em um espaço de alta dimensionalidade tem maior probabilidade de ser linearmente separável do que em um espaço de baixa dimensionalidade.

#### 4.4.2.1 Teorema de Cover

Na RN-RBF, a capacidade de separação deste modelo é fundamentada no teorema conhecido como Teorema de Cover sobre a separabilidade de padrões. Quando uma RN-RBF é utilizada para realizar uma tarefa complexa de classificação de padrões, o problema é resolvido basicamente pela sua transformação para um espaço de alta dimensionalidade, de uma forma não-linear. Em termos qualitativos o Teorema de Cover pode ser descrito da seguinte maneira:

*“Um problema complexo de classificação de padrões disposto não-linearmente em um espaço de alta dimensionalidade tem maior probabilidade de ser linearmente separável do que em um espaço de baixa dimensionalidade” [COV65].*

Considere uma família de superfícies, onde cada uma divide naturalmente um espaço de entrada em duas regiões. Considere que  $X$  represente um conjunto de  $N$  padrões (vetores)  $x_1, x_2, \dots, x_N$  cada um dos quais é atribuído a uma de duas classes  $X_1$  e  $X_2$ . Esta dicotomia (partição binária) dos pontos é separável em relação à família de superfícies, se existir uma superfície da família que separe os pontos da classe  $X_1$  daqueles da classe  $X_2$ . Para cada padrão  $x \in X$ , defina um vetor constituído de um conjunto de funções de valor real  $\{j_i(x) \mid i = 1, 2, \dots, M\}$ , como mostrado por:

$$\mathbf{j}(x) = [j_1(x), j_2(x), \dots, j_M(x)]^T \quad (4.2)$$

Suponha que o padrão  $x$  é um vetor em um espaço de entrada de dimensão  $r$ . O vetor  $\mathbf{j}(x)$ , então, mapeia pontos no espaço de entrada de dimensão  $r$  para os pontos em um novo espaço de dimensão  $M$ . Refere-se a  $j_i(x)$  como uma função oculta, porque esta desempenha um papel similar ao de uma camada oculta em uma rede neural alimentada adiante (*feedforward*). Assim, o espaço abrangido pelo conjunto de funções ocultas  $\{j_i(x)\}_{i=1}^M$  é referido como o espaço oculto ou espaço de características [HAY01], [BIS95].

Uma dicotomia  $\{X_1, X_2\}$  de  $X$  é dita ser linearmente separável por  $\mathbf{j}$ , se existir um vetor  $w$  de dimensão  $M$  para o qual pode-se escrever:

$$w^T \mathbf{j}(x) > 0, x \in X_1 \quad (4.3)$$

e

$$w^T \mathbf{j}(x) < 0, x \in X_2$$

O hiperplano definido pela equação:

$$w^T \mathbf{j}(x) = 0 \quad (4.4)$$

descreve a superfície de separação no espaço  $\mathbf{j}$  (espaço oculto). A imagem inversa deste hiperplano:

$$x : w^T \mathbf{j}(x) = 0 \quad (4.5)$$

define a superfície de separação no espaço de entrada.

Resumidamente, o teorema de Cover sobre a separabilidade de padrões engloba dois itens básicos:

1. A formulação não-linear da função oculta definida por  $j_i(x)$ , onde  $x$  é o vetor de entradas e  $i = 1, 2, \dots, M$ ;
2. A alta dimensionalidade do espaço oculto, comparando com o espaço de entrada. Esta dimensionalidade é determinada pelo valor atribuído a  $M$ , onde  $M$  é o número de unidades ocultas.

A figura 4.3, ilustra três exemplos de dicotomias separáveis por  $j$  de diferentes conjuntos de cinco pontos em duas dimensões.

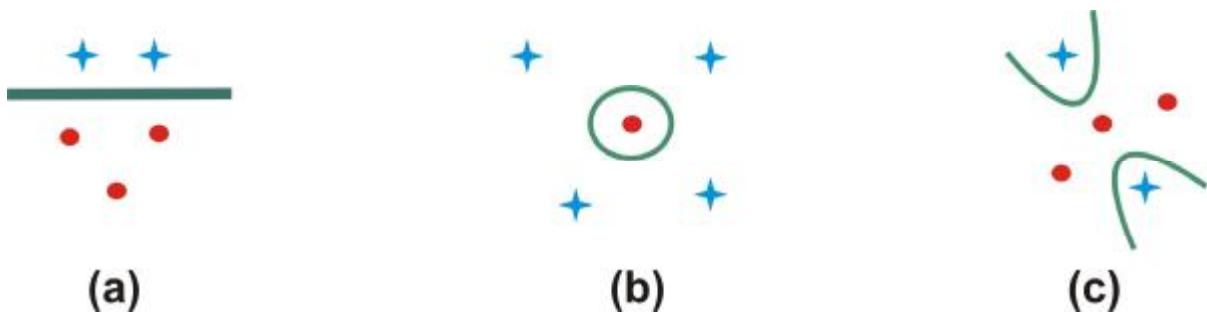


Figura 4.3. (a) dicotomia linearmente separável; (b) dicotomia esfericamente separável; (c) dicotomia quadraticamente separável.

#### 4.4.2.2 Problema de Interpolação

As funções de base radial são funções em geral não-lineares, cujo valor cresce ou decresce monotonicamente à medida que a distância a um ponto central aumenta. A esse ponto costuma-se denominar “centro” da função de base radial. A utilização destas funções, no contexto de aproximação de funções, tem sua origem na teoria da interpolação multivariada. Para a demonstração deste problema, considere uma rede com uma camada de entrada, uma única camada oculta e uma camada de saída consistindo de uma única unidade. A escolha de uma unidade de saída é proposital para simplificar a exposição sem perda de generalidade. A rede é projetada para realizar um mapeamento não-linear do espaço de entrada para o espaço oculto, seguido de um mapeamento linear do espaço oculto para o espaço de saída. Considere que  $r$  represente a dimensão do espaço de entrada. Então, de uma



maneira global a rede representa um mapa do espaço de entrada de dimensionalidade  $r$  em um espaço de saída unidimensional, escrito como [HAY01], [SCH97a], [BIS95]:

$$s : \mathfrak{R}^r \rightarrow \mathfrak{R}^1 \quad (4.6)$$

Considera-se o mapa  $s$  como uma hipersuperfície (gráfico)  $\Gamma \subset \mathfrak{R}^{r+1}$ . A superfície  $\Gamma$  é um gráfico multidimensional da saída como função da entrada. Em uma situação prática, a superfície  $\Gamma$  é desconhecida e os dados de treinamento estão normalmente contaminados com ruído. A fase de treinamento e a fase de generalização do processo de aprendizagem podem ser entendidas, respectivamente, como:

- A fase de treinamento constitui a otimização de um procedimento de ajuste para a superfície  $\Gamma$ , baseada nos pontos dos dados conhecidos apresentados à rede na forma de exemplos (padrões) de entrada-saída;
- A fase de generalização é sinônimo de interpolação entre os pontos de dados, com a interpolação sendo formada ao longo da superfície restrita gerada pelo procedimento de ajuste, como a aproximação ótima à superfície verdadeira  $\Gamma$ .

Desta maneira leva-se à teoria da interpolação multivariada em um espaço de alta dimensionalidade. O problema de interpolação, no seu sentido estrito, pode ser formulado como:

*“Dados um conjunto de  $N$  pontos diferentes  $\{d_i \in \mathfrak{R}^r \mid i=1,2,\mathbf{K},N\}$  e um conjunto correspondente de  $N$  números reais  $\{d_i \in \mathfrak{R}^1 \mid i=1,2,\mathbf{K},N\}$  encontre uma função  $F : \mathfrak{R}^N \rightarrow \mathfrak{R}^1$  que satisfaça a condição de interpolação:”*

$$F(x_i) = d_i, \quad i=1,2,\mathbf{K},N \quad (4.7)$$

A superfície de interpolação (isto é, a função  $F$ ) é obrigada a passar por todos os pontos de dados de treinamento. A técnica de funções de base radial consiste em escolher uma função  $F$  que tem a seguinte forma:

$$F(x) = \sum_{i=1}^N w_j j(\|x - x_i\|) \quad (4.8)$$

onde  $j(\|x - x_i\|) | i=1,2,\mathbf{K},N$  é o conjunto de  $N$  funções (geralmente não-lineares) arbitrárias, conhecidas como funções de base radial, e  $\|\cdot\|$  representa uma norma, geralmente Euclidiana. Os pontos de dados conhecidos por  $x_i \in \mathfrak{R}^r, i=1,2,\mathbf{K},N$  são tomados como *centros* das funções de base radial.

Inserindo as condições de interpolação da equação (4.7) em (4.8), obtêm-se o seguinte conjunto de equações lineares simultâneas para coeficientes (pesos) desconhecidos da expansão  $\{w_i\}$ :

$$\begin{bmatrix} j_{11} & j_{12} & \mathbf{L} & j_{1N} \\ j_{21} & j_{22} & \mathbf{L} & j_{2N} \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ j_{N1} & j_{N2} & \mathbf{L} & j_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \mathbf{M} \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \mathbf{M} \\ d_N \end{bmatrix} \quad (4.9)$$

onde

$$j_{ij} = j(\|x_j - x_i\|), (j,i) = 1,2,\mathbf{K},N \quad (4.10)$$

Considere

$$d = [d_1, d_2, \mathbf{K}, d_N]^T$$

$$w = [w_1, w_2, \mathbf{K}, w_N]^T$$

Os vetores  $d$  e  $w$  representam o *vetor resposta desejada* e o *vetor de peso linear*, respectivamente, onde  $N$  é o tamanho da amostra de treinamento. Considere que  $\Phi$  represente uma matriz  $N \times N$  com elementos  $j_{ji}$ :

$$\Phi = \{j_{ji} | (j,i) = 1,2,\mathbf{K},N\} \quad (4.11)$$

A matriz  $\Phi$  é chamada de matriz de interpolação. Pode-se então reescrever a equação (4.9) na forma compacta:

$$\Phi w = x \quad (4.12)$$

Assume-se que  $\Phi$  seja não-singular e, portanto, que exista a matriz inversa  $\Phi^{-1}$ , pode-se então revolver a equação (4.12) para o vetor de peso  $w$  como mostrado por:

$$w = \Phi^{-1}x \quad (4.13)$$

Um problema neste ponto é como se pode ter certeza que a matriz de interpolação  $\Phi$  seja não-singular. Contata-se que para uma grande classe de funções de base radial e sob certas condições, a solução deste problema é dada pelo teorema de Micchelli mostrado a seguir [HAY01], [HAG96].

### 4.4.2.3 Teorema de Micchelli

Em [MIC86], o seguinte teorema é provado:

“Considere que  $\{x_i\}_{i=1}^N$  seja um conjunto de pontos distintos em  $\mathbb{R}^r$ . Então, a matriz de interpolação  $\Phi$ ,  $N \times N$ , cujo elemento  $j_i$  é  $j_{ij} = j(\|x_j - x_i\|)$ , é não-singular”.

Existe uma grande classe de funções de base radial que é coberta pelo teorema de Micchelli, que inclui as seguintes funções, de particular interesse no estudo de RNs-RBF:

- Multiquadráticas:

$$j(x) = \sqrt{\|x - c_i\|^2 + s_i^2} \quad (4.14)$$

- Multiquadráticas Inversas:

$$j(x) = \frac{1}{\sqrt{\|x - c_i\|^2 + s_i^2}} \quad (4.15)$$

- Funções Gaussianas:

$$j(x) = e^{-\frac{\|x - c_i\|^2}{s_i^2}} \quad (4.16)$$

onde, em todos os casos,  $c_i$  representa o centro da função de base radial e  $s_i$  é a sua largura. O parâmetro  $s_i$  pode ser interpretado como um fator de escala para a distância  $\|x - c_i\|^2$ . No caso da função Gaussiana, por exemplo, o valor de  $j(x)$  decresce mais rapidamente quando  $s_i \rightarrow 0$ . A definição das larguras  $s_i$  tem um forte impacto sobre as características da função de aproximação. As funções multiquadráticas e as multiquadráticas inversas foram propostas por Hardy em 1971 [HAY01].

Para que as funções de base radial listadas nas equações (4.14) a (4.16) sejam não-singulares, os pontos  $\{x_i\}_{i=1}^N$  devem ser todos diferentes (distintos). Isto é tudo que é exigido

para a não-singularidade da matriz de interpolação  $\Phi$ , independentemente dos valores de tamanho  $N$  dos pontos de dados ou da dimensão  $r$  dos vetores (pontos)  $x_i$ .

A função Gaussiana e a multiquadrática inversa são funções locais, ou seja, fornecem uma resposta significativa apenas na vizinhança do centro  $c_i$ . A função multiquadrática, por sua vez, é global, uma vez que o seu valor  $j(x)$  torna-se ilimitado quando a distância ao centro tende ao infinito. A figura 4.4 ilustra essa diferença para o caso unidimensional com centro na origem e  $s_i = 1$ .

É notável que as funções de base radial que crescem ao infinito, como as multiquadráticas, podem ser usadas para aproximar um mapeamento de entrada-saída suave com maior precisão que aquelas que produzem uma matriz de interpolação definida positivamente [HAY01], [BIS95].

A função de base radial do tipo Gaussiana é a mais comumente utilizada em aplicações práticas. Neste tipo de função de base radial, o parâmetro  $s$  corresponde ao desvio padrão da função Gaussiana. Assim,  $s$  define a distância Euclidiana média (raio médio) que mede o espalhamento dos dados representados pela função de base radial em torno de seu centro. Os raios de cada uma das funções de base radial de uma mesma RN-RBF podem assumir diferentes valores, no entanto, para as RN-RBF usuais, o mesmo raio utilizado para cada neurônio não-linear já permite que a rede aproxime uniformemente qualquer função contínua, desde que exista número suficiente de funções de base radial. Em síntese, o valor do raio das funções de base radial afeta as propriedades numéricas dos algoritmos de aprendizado, mas não afeta a capacidade geral de aproximação das RN-RBF [HAY01], [BIL95], [COR01].

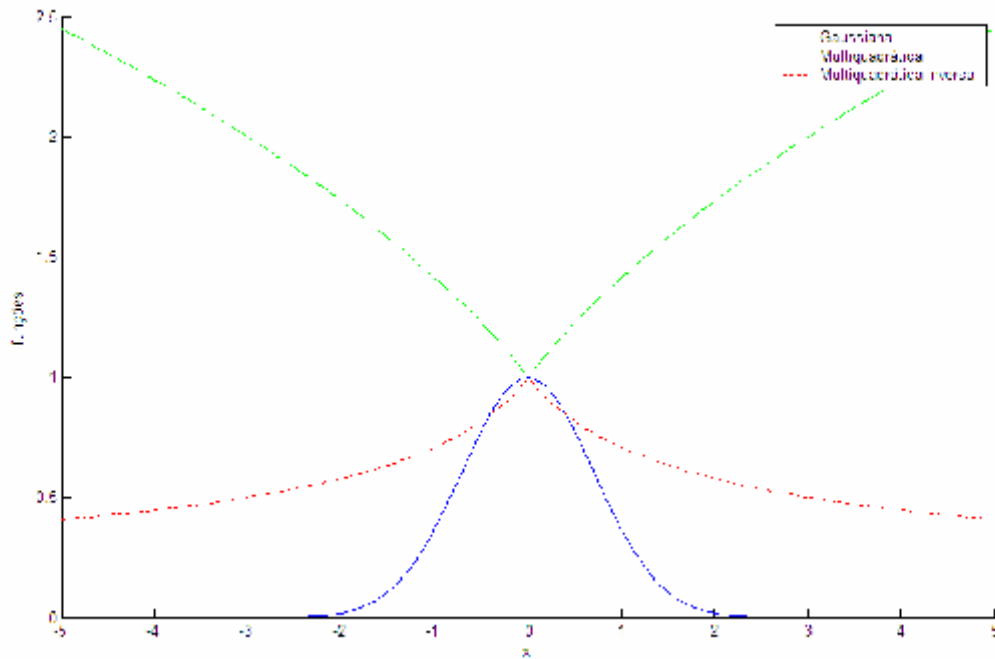


Figura 4.4. Funções de base radial cobertas pelo teorema de Micchelli.

#### 4.4.2.4 Problema do Ou Exclusivo (XOR)

O problema do XOR é um dos exemplos clássicos de um problema não-linear de classificação. Neste problema, existem quatro pontos (padrões),  $\{(1,1), (0,1), (0,0), (1,0)\}$ , em um espaço de entrada de dimensão dois (bidimensional). A exigência é construir um classificador que, com a saída binária ‘0’, corresponda aos padrões de entrada (1,1) ou (0,0), e para a saída binária ‘1’, corresponda aos padrões de entrada (0,1) e (1,0). Como discutido anteriormente, as RBFs transformam um problema não-linearmente separável em um problema linearmente separável, convertendo basicamente, o espaço de entrada em um espaço de alta dimensão.

Haykin define a rede RBF a ser investigada em um par de funções Gaussianas, para realizar a transformação, definidas como:

$$\begin{aligned}
 j_1 &= e^{-\|x-t_1\|}, & t_1 &= [1,1]^T \\
 j_2 &= e^{-\|x-t_2\|}, & t_2 &= [0,0]^T \\
 G(\|x-t_i\|) &= e^{-\|x-t_i\|^2}, & i &= 1,2
 \end{aligned}
 \tag{4.18}$$

onde  $t_1$  e  $t_2$  são os centros das funções gaussianas.

Para a caracterização da unidade de saída, assume-se que:

1. A unidade de saída utiliza compartilhamento de peso, que é justificável em virtude da simetria do problema. Esta é uma maneira de incorporar informação prévia no projeto da rede. Com apenas duas unidades ocultas é necessário determinar uma única ponderação  $w$ ;
2. A unidade de saída inclui um *bias*  $b$  (isto é, uma variável independente dos dados). O significado deste *bias* é que os valores de saída desejados da função XOR têm média diferente de zero.

Assim, a estrutura da rede RBF proposta para resolver o problema do XOR é ilustrada na figura 4.7.

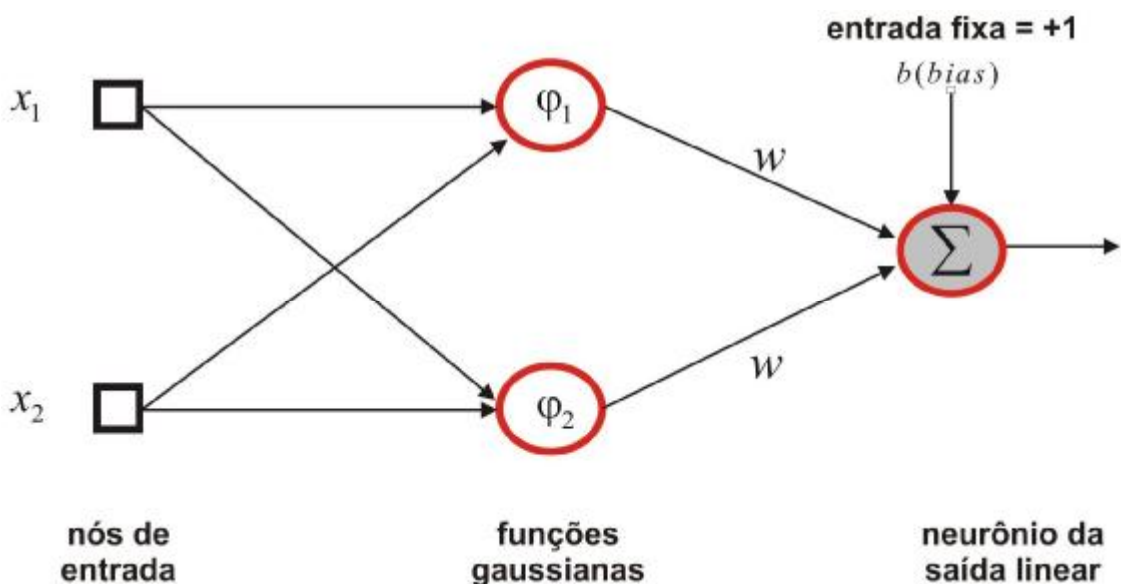


Figura 4.5. Rede RBF para resolver o problema do XOR.

A relação de entrada-saída da rede é definida por:

$$y(x) = \sum_{i=1}^2 wG(\|x - t_i\|) + b \quad (4.19)$$

Para ajustar os dados de treinamento da tabela 4.1, exige-se que:

$$y(x) = d_j, \quad j = 1, 2, 3, 4 \quad (4.20)$$

onde  $x_j$  é um vetor de entrada e  $d_j$  é o valor correspondente da saída desejada.

Tabela 4.1: Especificação para o problema do XOR

Ponto de dado, $j$	Padrão de entrada, $x_j$	Saída desejada, $d_j$
1	(1,1)	0
2	(0,1)	1
3	(0,0)	0
4	(1,0)	1

Através da tabela 4.1 pode-se construir o gráfico com os quatro padrões de entrada para o problema do XOR, ilustrado na figura 4.8.

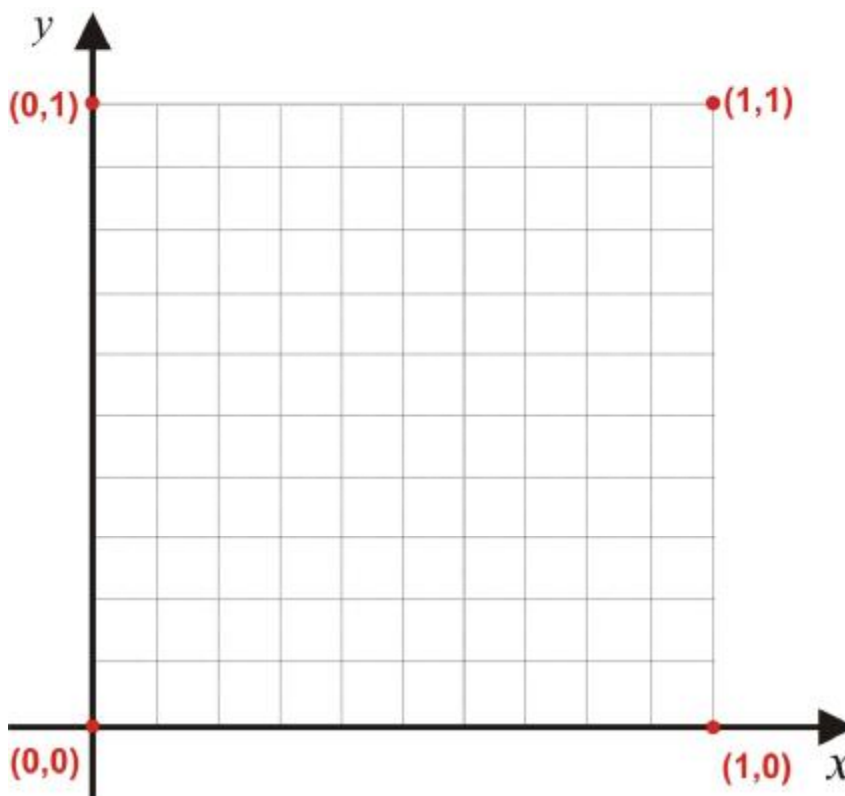


Figura 4.6. Quatro padrões de entrada para o problema do XOR.

Seja:

$$g_{ji} = G(\|x_j - t_i\|), \quad j = 1,2,3,4; \quad i = 1,2 \quad (4.21)$$

Então, usando os valores da tabela 4.1 na equação (4.21), obteve-se o seguinte conjunto de equações escritas na forma matricial:

$$Gw = d \quad (4.22)$$

onde

$$G = \begin{bmatrix} g_{11} & g_{12} & 1 \\ g_{21} & g_{22} & 1 \\ g_{31} & g_{32} & 1 \\ g_{41} & g_{42} & 1 \end{bmatrix}$$

$$x_1 = (1,1), x_2 = (0,1), x_3 = (0,0), x_4 = (1,0)$$

$$t_1 = (1,1), t_2 = (0,0)$$

$$g_{11} = e^{-\sqrt{(1-1)^2+(1-1)^2}} = 1$$

$$g_{12} = e^{-\sqrt{(0-1)^2+(0-1)^2}} = 0,1353$$

$$g_{21} = e^{-\sqrt{(1-0)^2+(1-1)^2}} = 0,3678$$

$$g_{22} = e^{-\sqrt{(0-0)^2+(0-1)^2}} = 0,3678$$

$$g_{31} = e^{-\sqrt{(1-0)^2+(1-0)^2}} = 0,1353$$

$$g_{32} = e^{-\sqrt{(0-0)^2+(0-0)^2}} = 1$$

$$g_{41} = e^{-\sqrt{(1-1)^2+(1-0)^2}} = 0,3678$$

$$g_{42} = e^{-\sqrt{(0-1)^2+(0-0)^2}} = 0,3678$$

$$G = \begin{bmatrix} 1 & 0,1353 & 1 \\ 0,3678 & 0,3678 & 1 \\ 0,1353 & 1 & 1 \\ 0,3678 & 0,3678 & 1 \end{bmatrix} \quad (4.23)$$

$$d = [1 \ 0 \ 1 \ 0]^T \quad (4.24)$$

$$w = [w \ w \ b]^T \quad (4.25)$$

Com os resultados de  $g_{ji}$ , pode-se montar tabela 4.2.

Tabela 4.2: Especificação das funções ocultas para o problema do XOR.

<b>Padrão de entrada,</b> $x_j$	<b>Primeira função oculta,</b> $j_1(x)$	<b>Segunda função oculta,</b> $j_2(x)$
(1,1)	1	0,1353
(0,1)	0,3678	0,3678
(0,0)	0,1353	1
(1,0)	0,3678	0,3678

Através da tabela 4.2 pode-se montar o diagrama de decisão, ilustrado na figura 4.7.



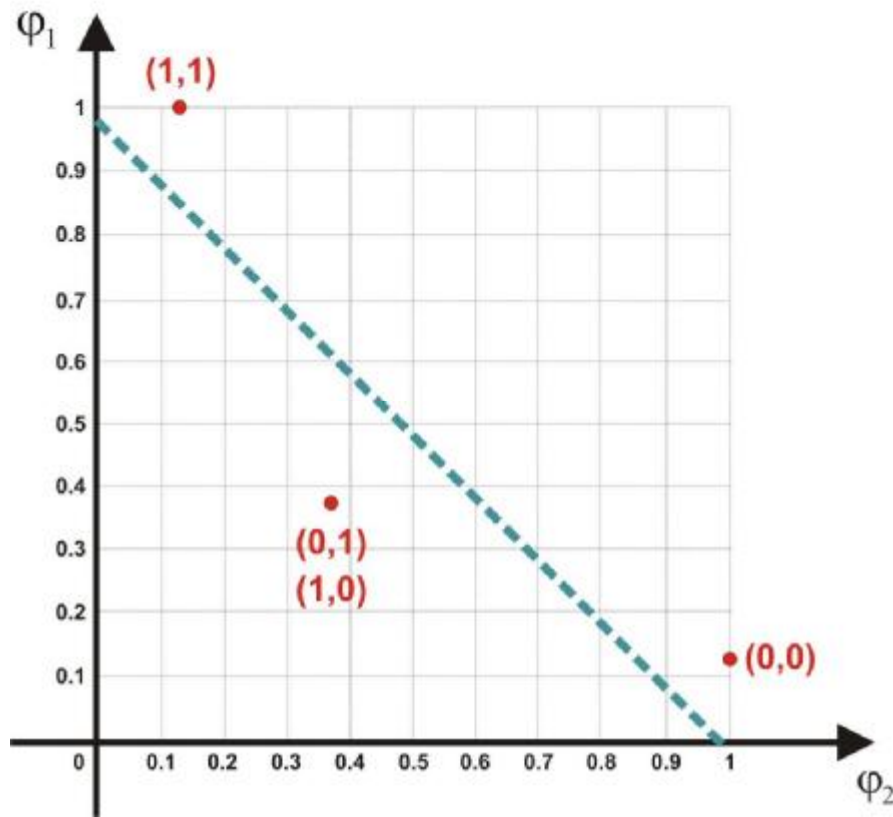


Figura 4.7. Diagrama de decisão.

Este problema é denominado superdeterminado, no sentido de que se têm mais pontos de dados que parâmetros livres. Isto explica por que a matriz  $G$  não é quadrada. Conseqüentemente, não existe uma inversa única para a matriz  $G$ . Para isso, usa-se a solução de norma mínima da equação (ver [HAY01]), e assim:

$$w = G^+ d = (G^T G)^{-1} G^T d \quad (4.26)$$

Note-se que  $G^T G$  é uma matriz quadrada com uma inversa única. Agora, substituindo a equação (4.23) em (4.26), obtém-se:

$$G^+ = \left( \begin{bmatrix} 1 & 0,1353 & 1 \\ 0,3678 & 0,3678 & 1 \\ 0,1353 & 1 & 1 \\ 0,3678 & 0,3678 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0,1353 & 1 \\ 0,3678 & 0,3678 & 1 \\ 0,1353 & 1 & 1 \\ 0,3678 & 0,3678 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 0,1353 & 1 \\ 0,3678 & 0,3678 & 1 \\ 0,1353 & 1 & 1 \\ 0,3678 & 0,3678 & 1 \end{bmatrix}^T =$$

$$G^+ = \begin{bmatrix} 1,8292 & -1,2509 & 0,6727 & -1,2509 \\ 0,6727 & -1,2509 & 1,8292 & -1,2509 \\ -0,9202 & 1,4202 & -0,9202 & 1,4202 \end{bmatrix} \quad (4.27)$$

Finalmente, substituem as equações (4.70) e (4.73) em (4.72), obtêm-se:

$$w = G^+ d = \begin{bmatrix} 1,8292 & -1,2509 & 0,6727 & -1,2509 \\ 0,6727 & -1,2509 & 1,8292 & -1,2509 \\ -0,9202 & 1,4202 & -0,9202 & 1,4202 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} =$$

$$w = \begin{bmatrix} -2,5019 \\ -2,5019 \\ 2,8404 \end{bmatrix} \quad (4.28)$$

que finaliza e completa a especificação da rede RBF.

Tirando a prova real dos resultados, pode-se calcular as saídas desejadas com os respectivos parâmetros que foram calculados anteriormente. Assim têm-se:

Para  $x_1 = (1,1)$  :

$$y(x) = \sum_{i=1}^2 wG(\|x - t_i\|) + b = (-2,5019 * 1) + (-2,5019 * 0,1353) + 2,8404 = 0$$

Para  $x_2 = (0,1)$  :

$$y(x) = \sum_{i=1}^2 wG(\|x - t_i\|) + b = (-2,5019 * 0,3678) + (-2,5019 * 0,3678) + 2,8404 = 1$$

Para  $x_3 = (0,0)$  :

$$y(x) = \sum_{i=1}^2 wG(\|x - t_i\|) + b = (-2,5019 * 0,1353) + (-2,5019 * 1) + 2,8404 = 0$$

Para  $x_4 = (1,0)$  :

$$y(x) = \sum_{i=1}^2 wG(\|x - t_i\|) + b = (-2,5019 * 0,3678) + (-2,5019 * 0,3678) + 2,8404 = 1$$

Através destes resultados, pode-se montar a tabela 4.3, que ilustra a transformação de entrada-saída para o problema do XOR. Nesta tabela, a última coluna mostra que a rede RBF conseguiu obter resultados ótimos para as saídas desejadas. Estes resultados indicam que a rede RBF conseguiu, com sucesso, separar os estados de polaridade oposta e resolver o problema.

Tabela 4.3: Transformação de entrada-saída calculada para o problema do XOR.

<b>Ponto de dado, <math>j</math></b>	<b>Padrão de entrada, <math>x_j</math></b>	<b>Saída desejada, <math>d_j</math></b>	<b>Saída real, <math>y_i</math></b>
1	(1,1)	<b>0</b>	<b>0</b>
2	(0,1)	<b>1</b>	<b>1</b>
3	(0,0)	<b>0</b>	<b>0</b>
4	(1,0)	<b>1</b>	<b>1</b>

#### 4.4.2.5 Estratégias de Aprendizado

O processo de aprendizagem de uma RN-RBF pode ser descrito da seguinte maneira. Os pesos associados com as unidades de saída da rede tendem a evoluir em uma escala de tempo diferente, comparada às funções de ativação não-lineares das unidades ocultas. Assim, como as funções de ativação da camada oculta evoluem lentamente de acordo com alguma estratégia de otimização não-linear, os pesos da camada de saída se ajustam rapidamente através de uma estratégia de otimização linear. O detalhe mais importante, é que as diferentes camadas de uma RN-RBF realizam tarefas diferentes, e assim, é razoável separar a otimização das camadas ocultas e de saída da rede usando técnicas diferentes, e talvez operando em diferentes escalas de tempo [LOW91], [HAY01].

Existem diferentes estratégias de aprendizagem que se pode seguir em um projeto de uma RN-RBF, dependendo de como os centros das funções de base radial da rede são especificados. Existem três abordagens principais para se determinar os centros, conforme explicitado a seguir.

#### 4.4.2.5.1 Centros Fixos Selecionados Aleatoriamente

A abordagem mais simples é assumir funções de base radial fixas, definindo as funções de ativação das unidades ocultas. A localização dos centros pode ser escolhida aleatoriamente do conjunto de dados de treinamento. Lowe define que isto é considerado uma abordagem “sensata”, desde que os dados de treinamento estejam distribuídos de uma forma representativa para o problema considerado [LOW89], [HAY01].

Nesta abordagem de aprendizado, os únicos parâmetros que devem ser aprendidos são os pesos lineares na camada de saída da rede. O procedimento direto para se calcular os pesos é utilizar o método da pseudo-inversa.

É interessante observar que a experiência com a seleção aleatória de centros mostra que este método é relativamente insensível ao uso de regularização. Este tipo de desempenho sugere que a seleção aleatória de centros como um método para o projeto de redes RBF a partir de um grande conjunto de treinamento fixo talvez seja, à sua própria maneira, um método de regularização (ver [HAY01] para a teoria de regularização).

#### 4.4.2.5.2 Seleção Auto-Organizada de Centros

O principal problema do método de centros fixos descritos anteriormente, é o fato deste requerer um grande conjunto de treinamento para obter um nível de desempenho satisfatório. Para superar esta limitação, pode-se utilizar um processo de aprendizagem híbrido, o qual possui de dois diferentes estágios [CHE92], [HAY01]:

1. Estágio de aprendizagem auto-organizada, cujo propósito é estimar localizações adequadas para os centros das funções de base radial na camada oculta;
2. Estágio de aprendizagem supervisionada, que completa o projeto da rede estimando os pesos lineares da camada de saída.

Apesar de se poder utilizar um procedimento por lote para implementar estes dois estágios de aprendizagem, é preferível adotar uma abordagem adaptativa (iterativa).

Para o processo de aprendizagem auto-organizado, necessita-se de um algoritmo de agrupamento que particione o conjunto fornecido de pontos de dados em sub-grupos, cada um dos quais sendo tão homogêneo quanto possível. Um desses algoritmos é o algoritmo de

agrupamento  $k$ -médias [DUD73a], que coloca os centros das funções de base radial apenas naquelas regiões do espaço de entrada  $X$  onde dados significativos estão presentes.

Tendo identificado os centros individuais das funções de base radial gaussianas e sua largura comum usando o algoritmo de agrupamento  $k$ -médias, o próximo estágio do processo de aprendizagem híbrido é estimar os pesos da camada de saída. Para isto, existe um método simples para esta estimação é o algoritmo do mínimo quadrado. Um detalhe importante, é que o algoritmo de agrupamento para a unidade oculta e o algoritmo de estimação dos pesos para a unidade de saída, podem realizar computações individuais de uma maneira concorrente, acelerando dessa forma o procedimento de treinamento [HAY01].

#### 4.4.2.5.3 Seleção Supervisionada de Centros

Nesta abordagem, os centros das funções de base radial e todos os outros parâmetros livres da rede sofrem um processo de aprendizagem supervisionada. Em outras palavras, Haykin diz que a rede RBF assume a sua forma mais generalizada. Haykin afirma também que o processo de aprendizagem por correção de erro é o mais conveniente, utilizando um processo de máxima descida que representa uma generalização do algoritmo LMS [HAY01].

### 4.4.3 RN-RBF versus RN-MLP

As RN-RBF e RN-MLP são consideradas aproximadores universais de funções, portanto são redes teoricamente equivalentes. Existem, no entanto, várias diferenças entre estes dois modelos. A primeira diferença diz respeito à partição do espaço de padrões de entrada realizada pela camada oculta de cada modelo.

Cada neurônio da camada oculta de um RN-RBF define uma hiperelipsóide no espaço de padrões de entradas. Desta maneira, a RN-RBF constrói aproximadores locais, isto é, apenas as regiões do espaço de entrada que apresentam dados de treinamento terão resposta da rede. A resposta de uma função de base radial diminui conforme os padrões se distanciam do centro da função de base radial. A figura 4.8 ilustra a partição dos dados de entrada realizada por uma RN-RBF com quatro neurônios na camada oculta.

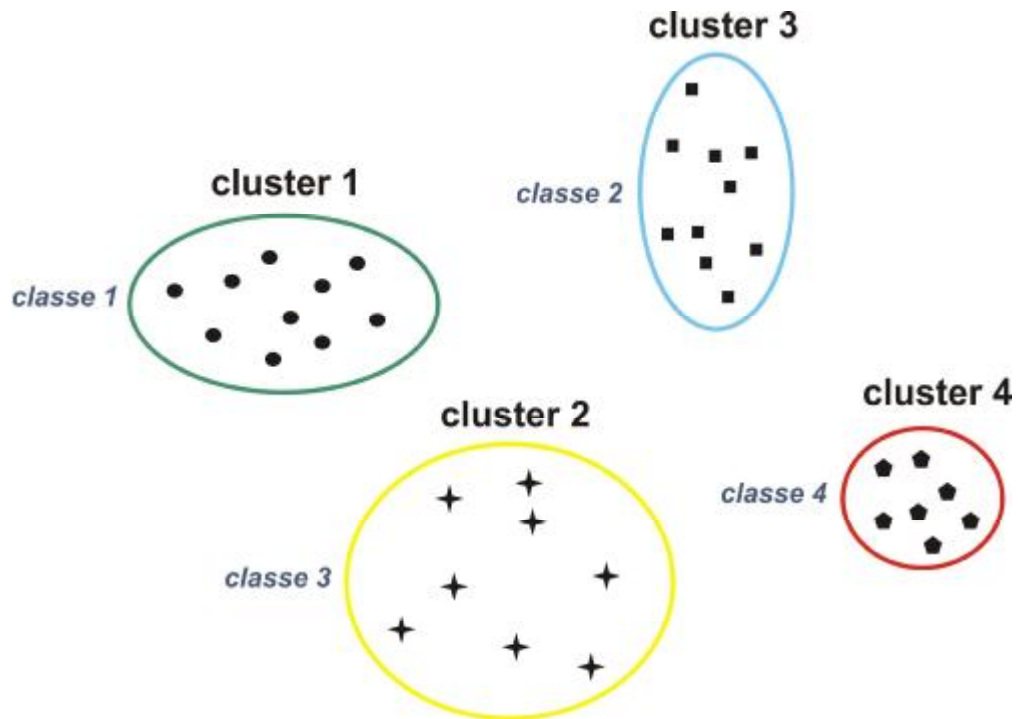


Figura 4.8. Particionamento dos dados de entrada realizado por uma RN-RBF com quatro neurônios na camada oculta.

As RN-MLP, por outro lado, particionam o espaço de entradas através de hiperplanos. Esta rede constrói aproximadores globais com maior capacidade de generalização para regiões onde não há dados de treinamento. Entretanto, isto permite que padrões completamente diferentes daqueles utilizados no treinamento sejam classificados em uma das classes existentes, o que é conhecido como problema de “falsos padrões”. A figura 4.9 ilustra a partição dos dados de entrada realizada por uma RN-MLP com dois neurônios na camada oculta.

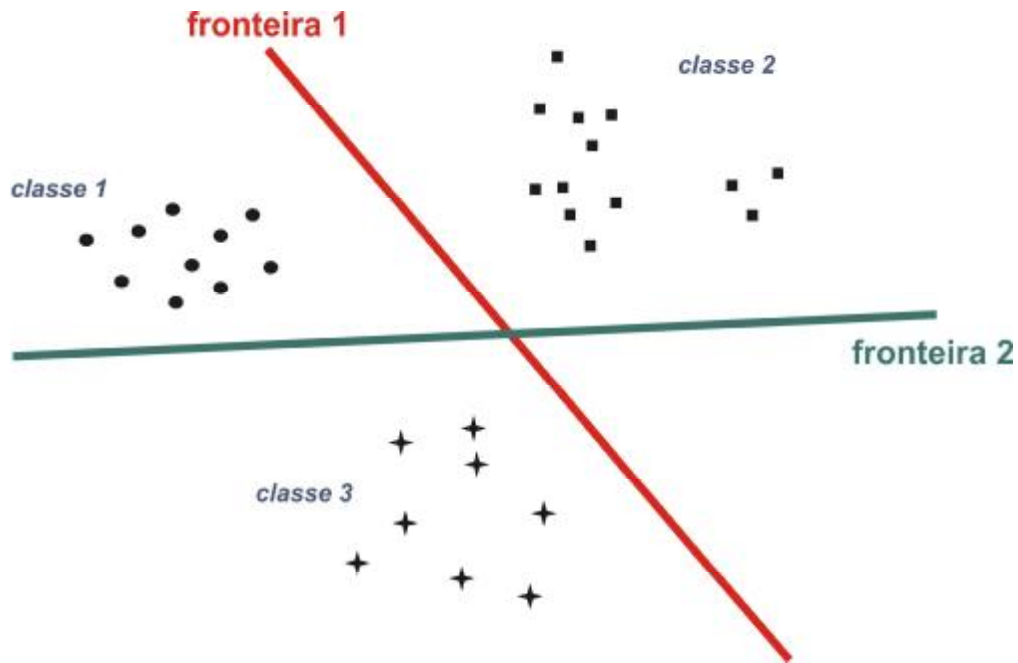


Figura 4.9. Particionamento dos dados de entrada realizada por uma RN-MLP com dois neurônios na camada oculta.

Outra diferença entre as RN-RBF e RN-MLP é que enquanto na RN-MLP o valor de ativação de uma unidade da camada oculta é uma função do produto escalar entre o vetor de entrada e o vetor de pesos da unidade. Na RN-RBF o valor de ativação é dado em função da distância euclidiana entre o vetor de entrada e o vetor de centro da unidade. Além disso, o treinamento de uma RN-MLP é geralmente mais lento (dependendo do método de treinamento adotado) que o da RN-RBF. A tabela 4.4 ilustra as principais diferenças entre essas redes neurais.

Tabela 4.4: Comparação entre uma rede neural RBF e uma rede neural MLP (convencionais).

<i>RN-MLP</i>	<i>RN-RBF</i>
Pode possuir uma ou mais camadas ocultas	Possui apenas uma camada oculta
Nós computacionais da camada oculta e de saída compartilham um modelo neuronal comum	Nós computacionais da camada oculta possuem funções diferentes e servem a um propósito diferente dos nós da camada de saída
Entrada da função de ativação é o produto inverso dos vetores de entrada pelos pesos	Entrada da função de ativação é a norma (distância) euclidiana entre os vetores de

	entrada e de pesos
As camadas ocultas e de saída são normalmente todas não-lineares	A camada oculta é não-linear e a camada de saída é linear
Separa padrões de entrada com hiperplanos	Separa padrões de entrada com hiperelipsóides
Melhor em problemas complexos	Melhor em problemas bem-definidos
Constrói aproximadores globais de um mapeamento de entrada-saída não-linear	Utiliza não-linearidades localizadas com decaimento exponencial construindo aproximadores locais para mapeamentos de entrada-saída não-lineares

#### 4.4.4 Aplicações

As RNs são capazes de resolver, basicamente, problemas de aproximação, predição, classificação e otimização. As aplicações conhecidas de RNs como reconhecimento de caracteres, reconhecimento de voz, previsão de séries temporais, modelagem de processos, controle de processos, entre outras, são na verdade novas formas de apresentação destas classes de problemas [HAY01], [REZ03].

Os problemas de aproximação de funções ou regressão são caracterizados pela interpolação, ou seja, dados são fornecidos dentro de determinados limites, onde a função definida e o modelo neural são ajustados para dar uma boa aproximação dentro dos mesmos [HER03].

O problema de previsão visa à previsão de estados seguintes de um determinado sistema, baseado nos seus estados anteriores, podendo ser caracterizado, portanto, como um problema de extrapolação. Alguns exemplos típicos são: previsão financeira, previsão de tempo e modelagem de processos dinâmicos industriais. O problema de predição não requer uma estrutura de rede neural específica para a sua solução, podendo o mesmo ser tratado como um problema de aproximação de funções, considerando que a saída seguinte do sistema seja uma função das entradas e saídas anteriores.

Em problemas de classificação, o objetivo é atribuir um padrão de entrada a uma classe entre um conjunto de classes conhecidas. Os exemplos típicos de problemas de classificação são: reconhecimento de voz, de caracteres e de imagens faciais. Para estes casos,



as classes são bem definidas e representadas pelos sons a serem reconhecidos, pelo conjunto de caracteres a ser reconhecido por um grupo de pessoas que se deseja identificar por meio de suas imagens [REZ03].

Os problemas de classificação envolvem a descoberta de características estatisticamente relevantes de um determinado conjunto de dados, e como estes podem ser divididos em classes. Neste tipo de problema, somente dados de entrada são conhecidos, cabendo à rede encontrar classes presentes neste conjunto [REZ03].

# Capítulo 5

## 5 Algoritmos

### 5.1 Introdução

O principal objetivo desta dissertação é o desenvolvimento de algoritmos de agrupamento. A análise de grupos abrange técnicas para determinar conjuntos de padrões que correspondem às classes dos dados, dividindo todas as amostras (objetos) em subgrupos menores e classificando-os de acordo com as similaridades entre elas. A tentativa de construir grupos é tentar extrair informações encapsuladas nos dados para uma melhor compreensão do sistema dinâmico em estudo.

As técnicas de agrupamento visam essencialmente particionar conjuntos de dados em classes homogêneas, em relação a uma determinada métrica. Os elementos que pertencem a um grupo devem ser os mais semelhantes possíveis entre si, e os que pertencem a grupos diferentes devem apresentar a menor semelhança. É um processo não supervisionado de classificação de padrões, organizando-se em grupos por um critério de similaridade. Havendo pouca informação preliminar sobre a distribuição dos dados no espaço de atributos, os métodos de agrupamento são particularmente apropriados para explorar as relações entre tais dados e sua estrutura.

Formalmente, [EVE80] apresenta algumas definições para um grupo:

- Conjunto de entidades parecidas, e as entidades que estão em grupos diferentes não são parecidas;
- Agrupamento no espaço de testes, de forma que a distância entre quaisquer dois pontos em um grupo e um ponto de outro grupo;

- Região conectada em um espaço multidimensional, com uma alta densidade de pontos, separada de outras regiões de alta densidade por regiões de baixa densidade.

## 5.2 Algoritmos de Agrupamento

### 5.2.1 Algoritmo $k$ -médias Convergente ( $k$ -means)

O algoritmo de agrupamento  $k$ -médias ( $k$ -means clustering algorithm) envolve um processamento simples de estimação dos parâmetros. Suponha que existam  $N$  amostras (dados)  $x^n$  no total e deseja-se encontrar  $K$  vetores  $c_j$  onde  $j = 1, \mathbf{K}, K$ . O algoritmo procura particionar as amostras  $x^n$  em  $K$  subconjuntos  $S_j$  contendo  $N_j$  amostras, de tal maneira a minimizar a função da soma dos quadrados dada por [BIS95]:

$$J = \sum_{j=1}^K \sum_{n \in S_j} \|x^n - c_j\|^2 \quad (5.1)$$

onde  $c_j$  é o centro das amostras do conjunto  $S_j$  e é dado por:

$$c_j = \frac{1}{N_j} \sum_{n \in S_j} x^n \quad (5.2)$$

#### 5.2.1.2 Etapas do Algoritmo $k$ -médias Convergente

O  $k$ -médias convergente é composto pelas seguintes etapas:

- Inicializar os centros das funções.  
Ajuste os centros iniciais da função aos primeiros dados do treinamento.
- Agrupar todos os dados com o centro de cada função.  
Cada dado de entrada ( $x_i$ ) pertencerá a um determinado cluster  $j^*$ , onde:

$$\|x_i - c_j^*\| = \min_j \|x_i - c_j\|$$

iii. Encontrar os centros de cada função, para cada  $c_j$  :

$$c_j = \frac{1}{N_j} \sum_{x_i \in j} x_i$$

onde  $N_j$  é o número de dados do grupo  $j$ .

iv. Repetir a etapa 2, até não haver mais alteração de cada grupo.

### 5.2.2 Algoritmo $c$ -médias Nebuloso ( $c$ -means)

O algoritmo *fuzzy c*-médias (FCM – *fuzzy c-means algorithm*) foi desenvolvido por [DUN73b] e melhorado por [BEZ81]. Este algoritmo é frequentemente utilizado para reconhecimento de padrões e é baseado na minimização da função objetivo:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m < \infty \quad (5.3)$$

onde  $u_{ij}$  é o grau do conjunto de  $x_j$  no grupo  $j$ ,  $x_i$  é o elemento  $i$  dos dados medidos,  $c_j$  é o centro do grupo  $j$ , e  $\|\cdot\|$  é a norma entre os dados medidos e o centro. A atualização de  $u_{ij}$  e  $c_j$  é dada por:

$$m_{ij}(x) = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|^2}{\|x_i - c_k\|^2} \right)^{\frac{2}{m-1}}} \quad (5.4)$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N (u_{ij}^m)} \quad (5.5)$$

### 5.2.2.1 Etapas do Algoritmo $c$ -médias nebuloso

O  $c$ -médias nebuloso é composto pelas seguintes etapas:

- i. Inicializar a matriz  $U = [u_{ij}]$
- ii.  $k$ -passos: calcular os vetores de centros  $C^k = [c_j]$  com  $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N (u_{ij}^m)}$$

- iii. Atualizar  $U^{(k)}, U^{(k+1)}$

$$m_{ij}(x) = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|^2}{\|x_i - c_k\|^2} \right)^{\frac{1}{m-1}}}$$

- iv. Repetir até  $\|U^{(k+1)} - U^{(k)}\| \leq \epsilon$ , onde  $\epsilon$  é o limiar de convergência, senão voltar para a etapa 2.

### 5.2.3 Algoritmo de Gustafson-Kessel (GK)

O algoritmo proposto por Gustafson-Kessel é uma extensão do algoritmo  $c$ -médias nebuloso, empregando uma norma adaptável da distância, a fim de detectar grupos de formas geométricas diferentes na série de dados [GUS79]. Cada conjunto tem sua matriz de indução  $A_i$ , o qual segue a seguinte norma produto interno:

$$D_{ika}^2 = (x_k - v_i)^T A_i (x_k - v_i), \quad 1 \leq i \leq c, \quad 1 \leq k \leq N \quad (5.6)$$

onde  $x_k$  é o elemento dos dados medidos,  $v_i$  é o centro do grupo e  $N$  é o número de amostras dos dados. As matrizes  $A_i$  são utilizadas como variáveis de otimização no funcional do  $c$ -médias, assim permitindo que cada grupo adapte a norma da distância à estrutura topológica local dos dados. Sendo  $A = (A_1, A_2, \mathbf{K}, A_c)$ , a função objetiva do algoritmo GK é definida como:

$$J(X; U, V, A) = \sum_{i=1}^c \sum_{k=1}^N (m_{ik})^m D_{ikA_i}^2 \quad (5.7)$$

onde  $V \in \mathfrak{R}^{n \times c}$  e  $m > 1$ .

Para um  $A$  fixo, as seguintes condições:

$$U = [u_{ik}], \quad m_{ik} \in [0,1], \quad 1 \leq i \leq N, \quad 1 \leq k \leq c$$

$$\sum_{k=1}^c m_{ik} = 1, \quad 1 \leq i \leq N$$

$$0 < \sum_{i=1}^N m_{ik} < N, \quad 1 \leq k \leq c$$

podem ser aplicadas diretamente. Entretanto, a função objetivo (5.7) não pode ser minimizada diretamente em relação a  $A_i$ , desde que seja linear em  $A_i$ . Isto significa que  $J$  pode ser projetado tão pequeno quanto desejado simplesmente fazendo  $A_i$  ser menos definida positiva. Para se obter uma solução flexível,  $A_i$  deve ser confinado de alguma maneira. Usualmente, esta maneira pode ser um confinamento do determinante de  $A_i$ . Permitindo que a matriz  $A_i$  varie com seu determinante fixo corresponda para otimizar a forma do grupo, enquanto seu volume permaneça constante:

$$\|A_i\| = r_i, \quad r_i > 0 \quad (5.8)$$

onde  $r_i$  é fixo para cada grupo. Utilizando o método do multiplicador de Langrange, a seguinte expressão para o  $A_i$  é obtida:

$$A_i = [r_i \det(F_i)]^{1/n} F_i^{-1} \quad (5.9)$$

onde  $F_i$  é a matriz de covariância do grupo  $i$ -th definido por:

$$F_i = \frac{\sum_{k=1}^N (m_{ik})^m (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^N (m_{ik})^m} \quad (5.10)$$

Note que substituindo as equações (5.10) e (5.9) na equação (5.6) resulta em uma norma quadrada generalizada da distância de Mahalanobis entre o  $x_k$  e o seu grupo  $v_i$ , onde a covariância seja relacionada pelos graus em  $U$ .

### 5.2.3.1 Etapas do Algoritmo GK

Dado um conjunto de dados  $X$ , escolha um número de grupos  $1 < c < N$ , o expoente  $m > 1$ , determine da tolerância  $\epsilon > 0$  [BAB02].

O GK é composto pelas seguintes etapas:

- i. Iniciar a matriz  $U = [u_{ik}]$
- ii. Calcular os centros dos grupos:

$$v_i^{(l)} = \frac{\sum_{k=1}^N (m_{ik}^{(l-1)})^m x_k}{\sum_{k=1}^N (m_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- iii. Calcular as matrizes de covariância:

$$F_i^{(l)} = \frac{\sum_{k=1}^N (m_{ik}^{(l-1)})^m (x_k - v_i^{(l)})(x_k - v_i^{(l)})^T}{\sum_{k=1}^N (m_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

Adicione uma matriz identidade:

$$F_i := (1 - g)F_i + g(F_0)^{1/n} I$$

Extraia os autovalores  $l_{ij}$  e os autovetores  $f_{ij}$  e encontre  $l_{i,\max} = \max_j l_{ij}$  e iguale o:

$$l_{i,\max} = l_{ij} / b, \forall j \text{ para cada } l_{i,\max} / l_{ij} \geq b$$

Reconstrua  $F_i$ :

$$F_i = [f_{i,1}, \mathbf{K}, f_{i,n}] \text{diag}(l_{i,1}, \mathbf{K}, l_{i,n}) [f_{i,1}, \mathbf{K}, f_{i,n}]^{-1}$$

- iv. Calcular as distâncias:

$$D_{ikA_i}^2(x_k, v_i) = (x_k - v_i^{(l)})^T [(r_i \det(F_i)^{1/n}) F_i^{-1}] (x_k - v_i^{(l)})$$

- v. Atualizar a matriz:

$$m_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikA_i}(x_k, v_i) / D_{jk}(x_k, v_j))^{2/(m-1)}}, \quad 1 \leq i \leq c, \quad 1 \leq k \leq N$$

- vi. Repetir até  $\|U^{(l+1)} - U^{(l)}\| \leq \epsilon$ , senão voltar para a etapa 2.

### 5.2.4 Algoritmo de Gath-Geva (GG)

O algoritmo de Gath-Geva emprega uma distância da norma baseado em estimativas nebulosas de probabilidade máxima (FMLE – *fuzzy maximum likelihood estimates*), proposta por [BEZ75]:

$$D_{ik}(x_k, v_i) = \frac{\sqrt{\det(F_{wi})}}{a_i} \exp\left(\frac{1}{2}(x_k - v_i^{(l)})^T F_{wi}^{-1}(x_k - v_i^{(l)})\right) \quad (5.11)$$

Note que diferentemente do algoritmo GK, esta distância da norma envolve um termo exponencial e diminui mais rapidamente do que a norma do produto interno.  $F_{wi}$  é a matriz de covariância nebulosa do  $i$  –ésimo grupo, dada por:

$$F_{wi} = \frac{\sum_{k=1}^N (m_{ik})^w (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^N (m_{ik})^w}, \quad 1 \leq i \leq c \quad (5.12)$$

onde  $w = 1$  no algoritmo original FMLE. Neste caso utiliza-se  $w = 2$ , para que o agrupamento compense o termo exponencial da distância da norma. A probabilidade de seleção de cada grupo é dada por:

$$a_i = \frac{1}{N} \sum_{k=1}^N m_{ik} \quad (5.13)$$

Os graus do conjunto são interpretados como as probabilidades posteriores para selecionar o  $i$  –ésimo grupo dos dados, dado o  $x_k$  do conjunto de dados.

#### 5.2.4.1 Etapas do algoritmo GG

Dado um conjunto de dados  $X$ , escolha um número de grupos  $1 < c < N$ , o expoente  $w > 1$ , determine da tolerância  $e > 0$  [GAT89].

O GG é composto pelas seguintes etapas:

- i. Iniciar a matriz  $U = [u_{ik}]$
- ii. Calcular os centros dos grupos:



$$v_i^{(l)} = \frac{\sum_{k=1}^N (m_{ik}^{(l-1)})^w x_k}{\sum_{k=1}^N (m_{ik}^{(l-1)})^w}, \quad 1 \leq i \leq c$$

iii. Calcular as matrizes de covariância:

$$F_i^{(l)} = \frac{\sum_{k=1}^N (m_{ik}^{(l-1)})^w (x_k - v_i^{(l)})(x_k - v_i^{(l)})^T}{\sum_{k=1}^N (m_{ik}^{(l-1)})^w}, \quad 1 \leq i \leq c$$

iv. Calcular as distâncias:

$$D_{ik}^2(x_k, v_i) = \frac{(2p)^{n/2} \sqrt{\det(F_i)}}{a_i} \exp\left(\frac{1}{2}(x_k - v_i^{(l)})^T F_i^{-1}(x_k - v_i^{(l)})\right)$$

com a probabilidade *a priori*

$$a_i = \frac{1}{N} \sum_{k=1}^N m_{ik}$$

v. Atualizar a matriz:

$$m_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikA_j}(x_i, v_i) / D_{jk}(x_k, v_j))^{2/(w-1)}}, \quad 1 \leq i \leq c, \quad 1 \leq k \leq N$$

vi. Repetir até  $\|U^{(l+1)} - U^{(l)}\| \leq \epsilon$ , senão voltar para a etapa 2.

## 5.3 Algoritmos de Otimização

### 5.3.1 Mínimos Quadrados em batelada (Pseudo-Inversa de Penrose-Moore)

O método de ajuste linear para os parâmetros de saída da RN-RBF é a pseudo-inversa. A atualização de cada peso para o treinamento da RN-RBF utilizando esta derivação do método de mínimos quadrados, é realizado através de:

$$w_m = (\text{inv}(k^T k) k^T) y(t) \quad (5.8)$$

onde  $w_m$  são os pesos da rede e  $k_m$  é a saída da camada oculta da rede.

O estimador dos mínimos quadrados da equação (5.8) é uma transformação linear sobre  $y$  (função linear das medidas) e, assim, é denominado estimador linear [LJU87].

## 5.4 Validação de modelos

A validação é o procedimento final da identificação de sistemas, sendo efetuada após a determinação da estrutura e estimação de parâmetros, e tem como objetivo verificar se o modelo matemático obtido é capaz de representar a dinâmica do sistema em questão, e também se o modelo é não polarizado. Um modelo polarizado pode ser capaz de prever a resposta do sistema quando utilizando o mesmo conjunto de dados da estimação, mas isto pode não ocorrer para dados desconhecidos [BIL91].

Na procura de um modelo “ideal”, devem ser observadas as seguintes questões: se o modelo encontrado é flexível o suficiente e se o modelo é desnecessariamente complexo, ou em outras palavras, se o modelo é confiável para o propósito desejado [CAS99]. Existem diversas maneiras de se realizar a validação de um modelo, mas apenas dois índices de desempenho serão utilizados: (i) o coeficiente de correlação múltipla, também denominado de  $R^2$ , e (ii) o erro médio quadrático.

### 5.4.1 Coeficiente de correlação múltipla e erro médio quadrático

O Erro Médio Quadrático ( $MSE - Mean Square Error$ ) é um índice muito utilizado e pode ser calculado por:

$$MSE = \frac{\sum_{t=1}^N (y(t) - \hat{y}(t))^2}{N} \quad (5.6)$$

onde  $N$  é o número de amostras (dados) utilizado,  $y(t)$  é a saída real do sistema e  $\hat{y}(t)$  é a saída estimada. O problema do  $MSE$  é que este não proporciona uma resposta explícita em relação a real qualidade do modelo. Então, para isso utiliza-se outro índice de desempenho, o  $R^2$ .

O  $R^2$  é um índice de desempenho calculado por:

$$R^2 = \frac{\sum_{t=1}^N (y(t) - \hat{y}(t))^2}{\sum_{t=1}^N (y(t) - \bar{y}(t))^2} \quad (5.7)$$

onde  $\bar{y}(t)$  é a média das  $N$  amostras medidas (média da saída real). Quando o valor de  $R^2$  é igual a 1, indica uma exata adequação do modelo para os dados medidos do processo. O valor de  $R^2$  entre  $0,9 \leq R^2 \leq 1$  é considerado suficiente para aplicações práticas em identificação de sistemas [SCH97b].

Uma observação a ser feita é que existe um problema na utilização desses critérios, é que estes determinam o melhor modelo. Entretanto, não atuam diretamente no modelo para melhorá-los.

# Capítulo 6

## 6 Casos Estudados

### 6.1 Identificação

Como já foi dito, a literatura na área de identificação de sistemas está bem sedimentada para lidar com sistemas lineares [LJU87]. Entretanto, o modelo linear é somente útil se o processo físico exibir o comportamento dinâmico similar ao modelo linear operando na região de interesse. O fato é que todos os sistemas físicos são não-lineares a uma extensão, conduzindo à necessidade do desenvolvimento de ferramentas de aproximações de modelos não-lineares e de análise de sistemas não-lineares [JOH93].

A identificação de processos, é uma área relevante em muitos campos do conhecimento, é o procedimento de identificar um modelo de um processo desconhecido, para propósitos de previsão e/ou compreensão do comportamento do processo. A complexidade inerente a muitos processos reais (não-lineares e variantes no tempo) dificulta a aplicação de técnicas convencionais de identificação. Este aspecto tem motivado o desenvolvimento de técnicas de identificação avançadas, baseadas em inteligência computacional, da qual as redes neurais, os algoritmos evolutivos e os sistemas nebulosos fazem parte [LJU87].

A identificação de sistemas é tratada, muitas vezes, como um problema de otimização, que envolve algumas medidas para adequação de modelos candidatos a representar um processo real. A escolha de modelos matemáticos e o ajuste dos parâmetros adequados são influenciados por diversos fatores, dentre os quais: (i) conhecimento *a priori* do sistema (linearidade, grau de não-linearidade, atraso de transporte); (ii) propriedades do modelo do sistema identificado (complexidade); (iii) escolha da medida de erro a ser minimizado; (iv) presença de ruídos. A identificação de sistemas é um claro exercício que envolve múltiplos e

conflitantes objetivos, tipicamente complexidade do modelo, critérios de desempenho e validação que influenciam a seleção das estruturas de modelo matemático mais adequadas. A noção de um “bom modelo” é subjetiva e depende da preferência pessoal, levando a tentativa e erro a ser uma regra relevante em identificação de processos [LJU87], [AGU04], [NAR90].

A meta do algoritmo de otimização é a minimização de um critério de desempenho. Se todas as restrições e condições forem atendidas, o modelo encontrado pode ser aceito. Caso contrário, se uma das condições impostas é violada, todo ou mesmo alguma parte do procedimento de identificação, estimação de parâmetros e diagnóstico do modelo, têm de ser repetidos até que um modelo apropriado seja encontrado. Nas últimas décadas, diversas configurações de algoritmos em identificação de processos não-lineares têm sido elaborados e propostos na literatura, tais como: métodos freqüenciais, técnicas baseadas em estimativas de modelos Wiener-Hammerstein e Volterra, análise de correlação, métodos de regressão, transformações *wavelets* e algoritmos de identificação recursivos [WIE58], [BIL80], [HAY01].

Muitos dos métodos de identificação, tais como os baseados em mínimos quadrados ou estimativas de máxima verossimilhança são, em essência, técnicas de busca local guiada por gradiente, e necessitam de um espaço de busca regular ou um índice de desempenho diferenciável. Estes métodos convencionais podem facilmente falhar na obtenção de um ótimo global se o espaço de busca do modelo é não-diferenciável ou o índice de desempenho não é “bem comportado” [HAY01].

Adicionalmente, os métodos convencionais de identificação sofrem de algumas desvantagens, tais como: (i) alguma informação inicial dos parâmetros do sistema é necessária *a priori* para convergência do método; (ii) os parâmetros estimados podem ser tendenciosos se o ruído é correlacionado; (iii) há dificuldade na identificação do atraso de transporte; e (iv) os métodos convencionais não podem ser facilmente aplicados em sistemas não-lineares [AGU04].

A tentativa de explicar ou reproduzir os comportamentos dos sistemas físicos de processos é algo que há tempo desperta o interesse de pesquisadores. A identificação de processos tem relevância, pois permite prever o que acontece a um processo, conhecendo a(s) entrada(s) e a(s) saídas anteriores, nele disponíveis [LJU94], [LJU87].

O problema principal em identificação de processos (sistema) é a determinação de uma estrutura apropriada para o modelo do processo. A regra básica em estimação é “não

estimar o que se conhece”. Em outras palavras, pode-se utilizar o conhecimento *a priori* do processo e suas características físicas, quando se seleciona a estrutura e os parâmetros do modelo matemático.

## 6.2 Casos Estudados

Nesta seção são apresentadas descrições de quatro sistemas não-lineares ou caóticos. Estes são detalhados com número de entradas e saídas, tempo de amostragem, total de amostras e juntamente com sua descrição do que o sistema se denomina.

### 6.2.1 Trocador de Calor

O processo consiste de um trocador de calor com vapor saturado, onde a água é aquecida no sistema através de vapor saturado pressurizado, que circula por uma tubulação de cobre. As variáveis do processo são:

- variável de saída: temperatura da água na saída do trocador de calor;
- variável de entrada: vazão de água na entrada no trocador de calor.

Para este experimento, as temperaturas de vapor e do líquido na entrada do sistema foram mantidas constantes para seus valores nominais. Desta maneira tem-se:

- tempo de amostragem: 1 s;
- total de amostras: 4000;
- entrada: vazão de água na entrada do sistema;
- saída: temperatura da água na saída do sistema.

Os dados do trocador de calor estão disponíveis na base de dados: Daisy [MOO04a].

As figuras 6.1 e 6.2 ilustram a entrada e a saída do trocador de calor, respectivamente.

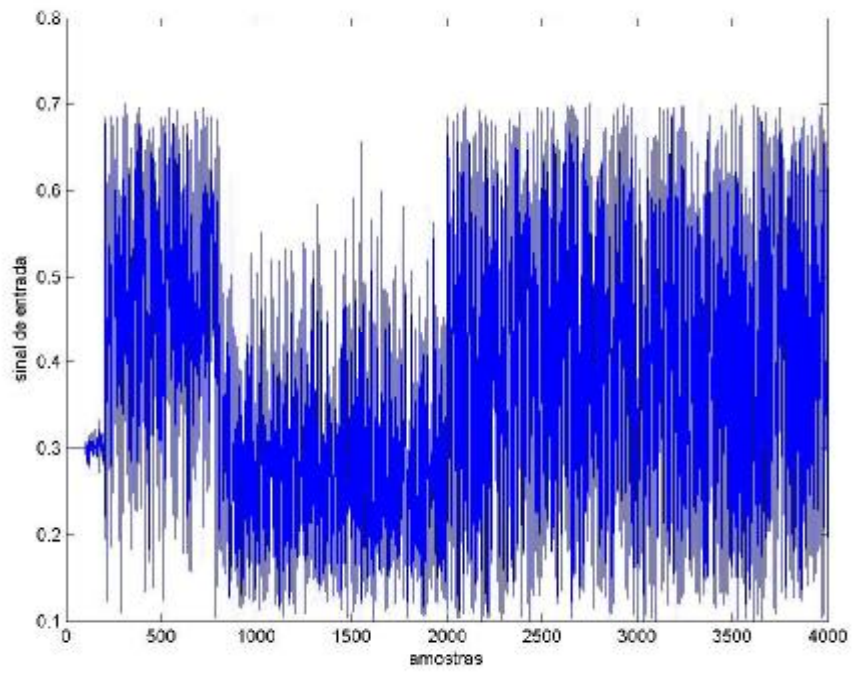


Figura 6.1. Entrada do trocador de calor.

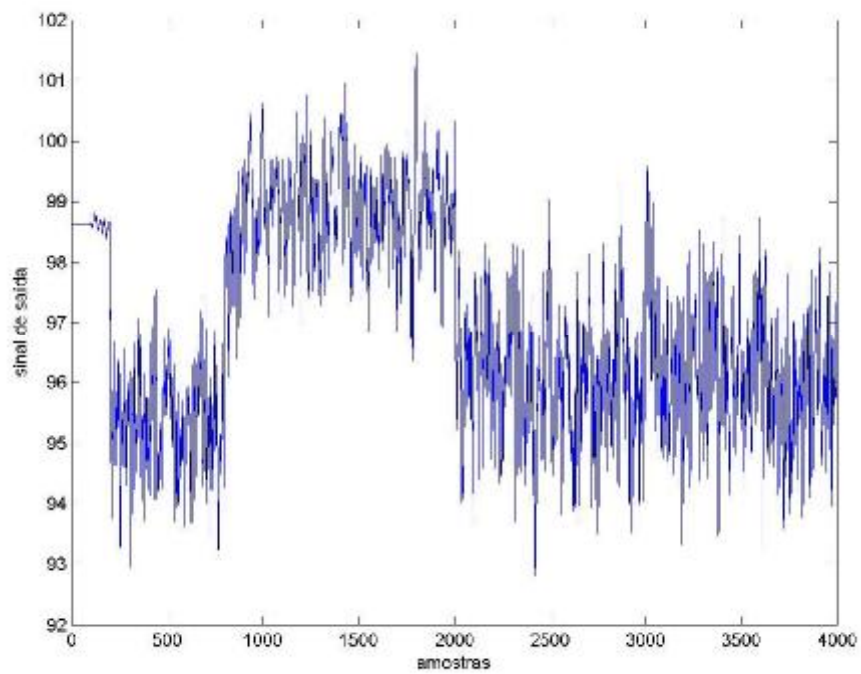


Figura 6.2. Saída do trocador de calor.

## 6.2.2 Sistema de Mackey-Glass

Neste caso utiliza-se a série denominada de equação diferencial de Mackey-Glass, que descreve um sistema de controle fisiológico [MAC77]. A equação de Mackey-Glass é uma série temporal obtida pela integração da equação diferencial:

$$\frac{dy(t)}{dt} = \frac{Ay(t-t)}{1+y^C(t-t)} - Bx(t) \quad (6.1)$$

onde:  $A=0,2$ ;  $B=0,1$ ;  $C=10$  e  $t=17$ . Para estes valores de  $A, B, C$  e  $t$  o sistema exibe um comportamento caótico. No experimento realizado os dados da série temporal de Mackey-Glass são obtidos através do método de Runge-Kutta de quarta ordem com condições iniciais  $y(0)=1,2$  e  $y(t-t)=0$ , para  $0 \leq t \leq t$  em intervalos de tempo de 1.

A variável do sistema é apenas o resultado da integração da equação diferencial, que exibe o comportamento caótico, com um total de amostras de 1200 [JAN06]. A figura 6.3 ilustra estes resultados.

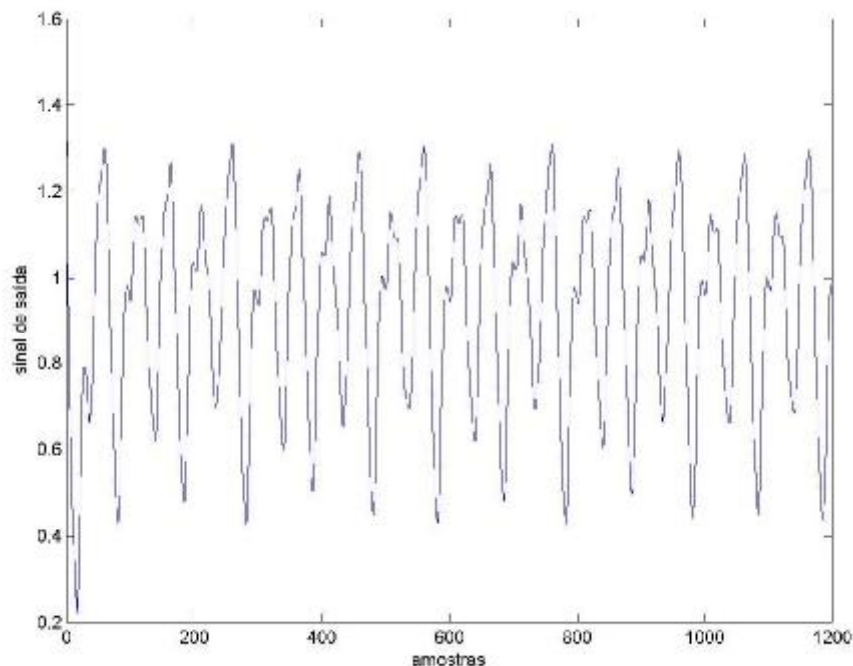


Figura 6.3. Comportamento da variável de saída  $y(t)$ .



### 6.2.3 Sistema Caótico de Rössler

Um dos estudos de caso de identificação abordado é o sistema dinâmico não-linear com comportamento caótico de Rössler [ROS76]. O cientista alemão Otto Rössler propôs um atrator caótico constituído de equações diferenciais não-lineares. As equações de Rössler são:

$$\frac{dx}{dt} = -y - z \quad (6.2)$$

$$\frac{dy}{dt} = x + ay \quad (6.3)$$

$$\frac{dz}{dt} = b + (x - c)z \quad (6.4)$$

que exibem comportamento caótico para  $(a, b, c) = (0,36; 0,40; 4,50)$ . Este comportamento está ilustrado na figura 6.4.

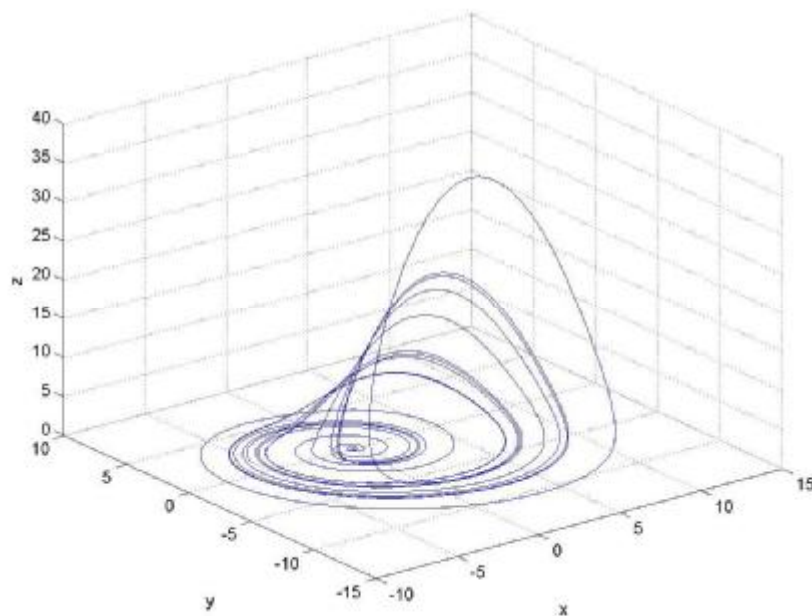


Figura 6.4. Atrator do sistema de Rössler para  $(a, b, c) = (0,36; 0,40; 4,50)$ .

As coordenadas  $x, y, z$  do atrator caótico de Rössler estão ilustradas nas figuras 6.5, 6.6 e 6.7, respectivamente.

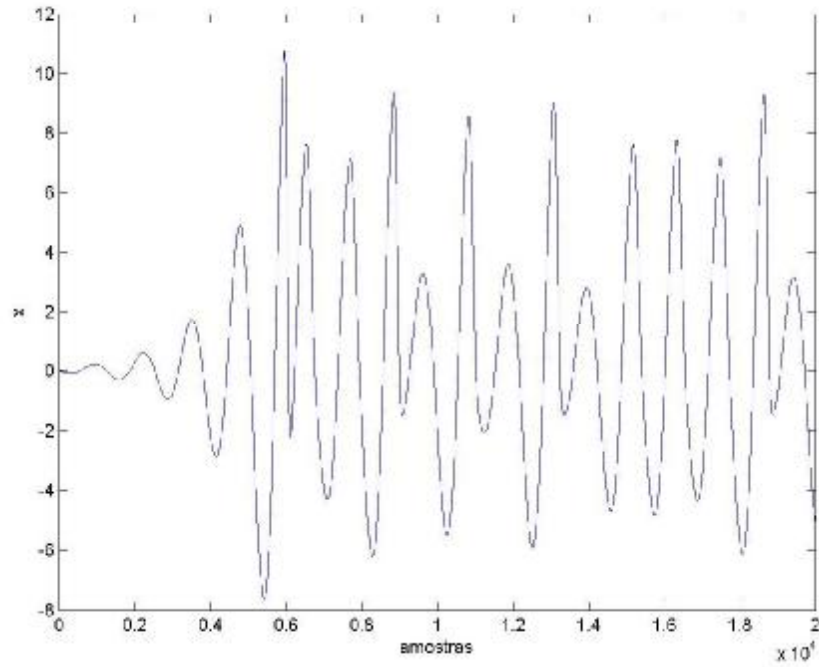


Figura 6.5. Coordenada  $x$  do sistema de Rössler.

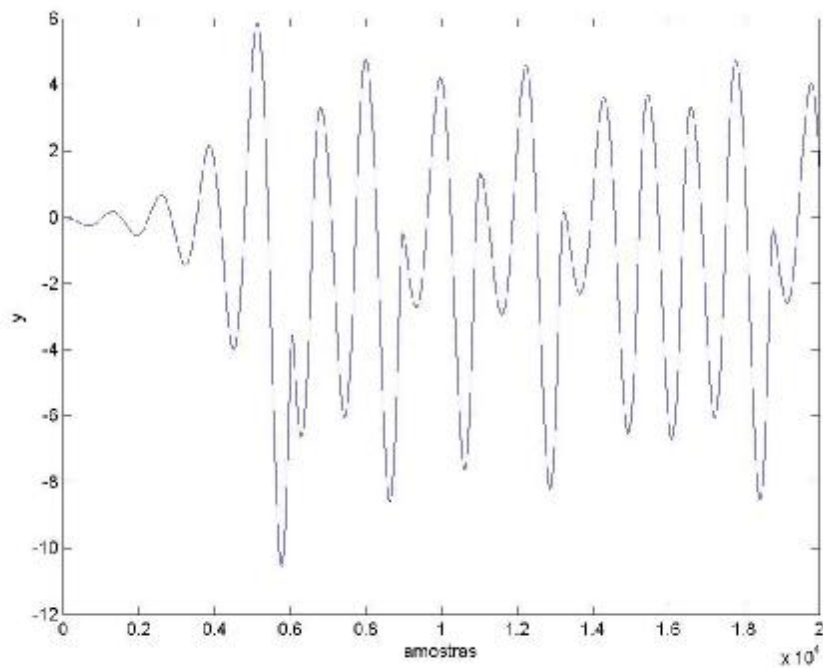


Figura 6.6. Coordenada  $y$  do sistema de Rössler.

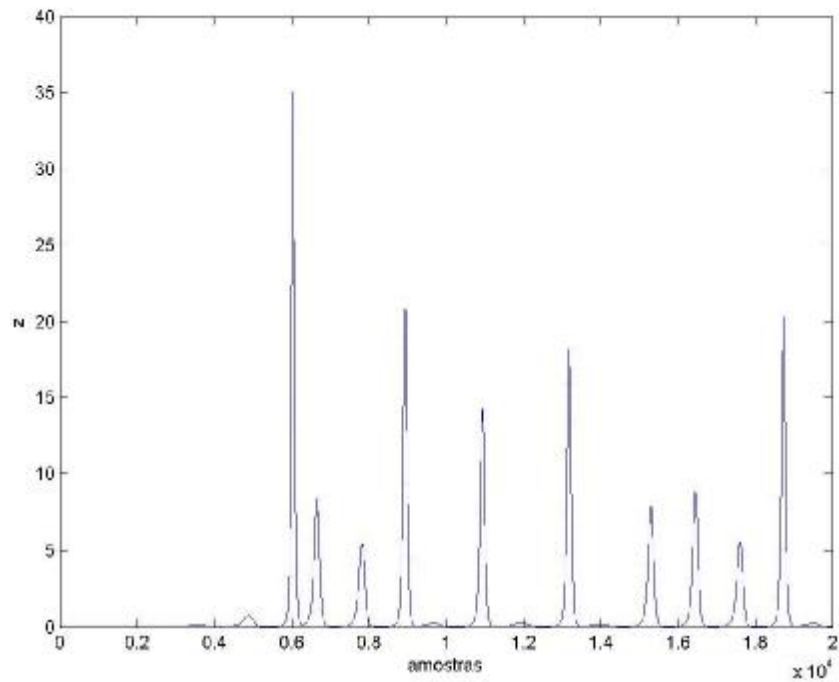


Figura 6.7. Coordenada  $z$  do sistema de Rössler.

#### 6.2.4 Fornalha de Gás de Box e Jenkins

A fornalha de gás de Box e Jenkins consiste de uma fornalha onde a entrada é a taxa de alimentação de gás metano e a saída é a concentração de dióxido de carbono ( $\%CO_2$ ) em uma mistura de gases [BOX94]. Para este sistema têm-se 296 amostras e os sinais de entrada e saída estão ilustrados nas figuras 6.8 e 6.9, respectivamente [JAN06].

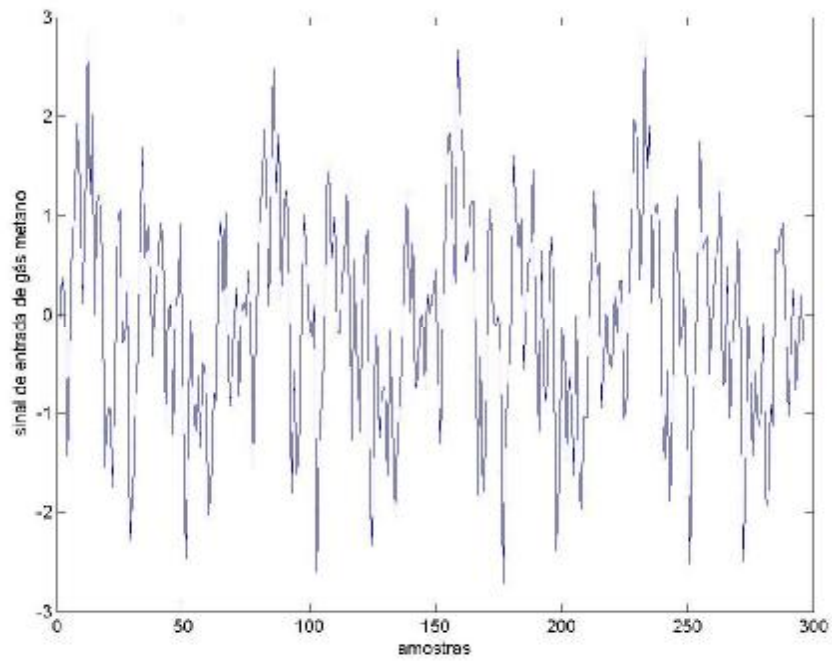


Figura 6.8. Sinal de entrada de gás metano.

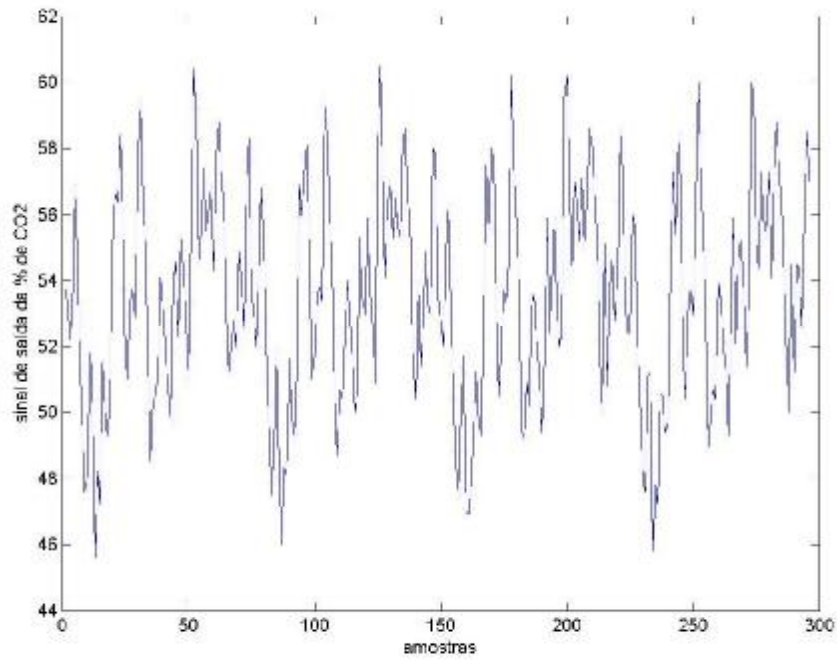


Figura 6.9. Sinal de saída da %CO<sub>2</sub>.

# Capítulo 7

## 7 Resultados de Simulação

No presente capítulo são apresentados resultados da identificação um-passo-à-frente ( $k=1$ ) dos seguintes sistemas: (i) trocador de calor, (ii) sistema de Mackey-Glass, (iii) sistema caótico de Rössler e a (iv) fornalha de gás de Box-Jenkins. Neste contexto, a RN-RBF na configuração série-paralelo, com quatro algoritmos de agrupamento,  $k$ -médias,  $c$ -médias, Gustafson-Kessel e Gath-Gheva, com atualização dos pesos pela pseudo-inversa foram empregados. A arquitetura geral da RN-RBF é ilustrada na figura 7.1, onde  $u$  representa o vetor de entradas do sistema,  $y$  o vetor de saídas reais,  $N_u$  é o número de entradas atrasadas,  $N_y$  é o número de saídas atrasadas,  $n$  é o número de grupos,  $c$  consiste do vetor dos centros das gaussianas,  $w$  o vetor dos pesos da camada oculta e  $\hat{y}(t)$  é a saída estimada da rede.

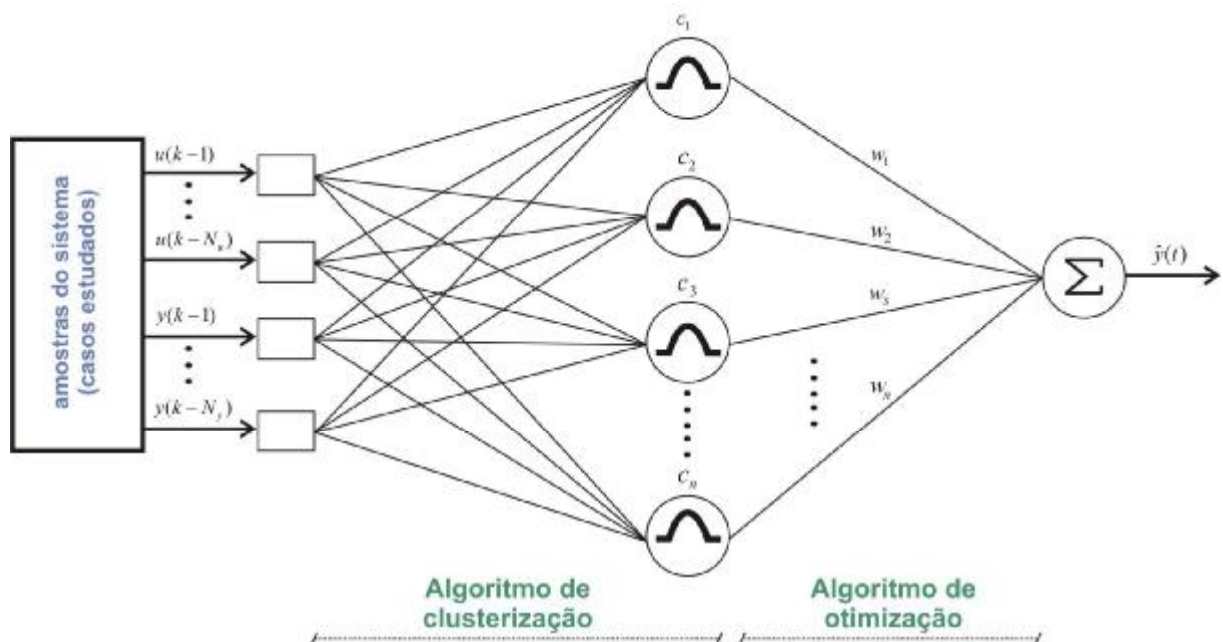


Figura 7.1. Arquitetura geral da RN-RBF para os sistemas analisados.

Para cada estudo de caso foi construída uma tabela de resultados das simulações das fases de estimação e validação e também figuras dos melhores resultados de cada algoritmo de agrupamento. Cada tabela contém todas as informações necessárias, tais como: número de entradas atrasadas ( $N_u$ ), número de saídas atrasadas ( $N_y$ ), números de grupos ( $n$ ), e os resultados encontrados, como:  $R^2$  da fase de estimação ( $R^2_{est}$ ),  $R^2$  da fase de validação ( $R^2_{val}$ ),  $MSE$  da fase de estimação ( $MSE_{est}$ ),  $MSE$  da fase de validação ( $MSE_{val}$ ) e o número de operações algébricas. Esta informação de número de operações algébricas foi normalizada para cada estudo de caso, para se ter uma idéia melhor da diferença desses números. Para normalizar esta informação foi utilizado o maior número de operações de ponto flutuante (comando *flops* do Matlab 5.3) de cada estudo de caso. Os maiores números de operações encontrados foram:

- Trocador de calor: 1.858.800.000;
- Mackey-Glass: 110.789.694;
- Coordenada  $x$  do sistema de Rössler: 54.163.000.000;
- Coordenada  $y$  do sistema de Rössler: 557.149.694;
- Coordenada  $z$  do sistema de Rössler: 426.905.935;
- Box-Jenkins: 104.860.433.

Existem alguns detalhes importantes das simulações que devem ser observados. O primeiro é que sempre é utilizada a metade das amostras disponíveis de cada estudo de caso para a fase de estimação dos parâmetros da RN-RBF, e a outra metade para a fase de validação. O segundo, é que as aberturas das gaussianas foram mantidas fixas em 10. E finalmente, o terceiro, é que a melhor simulação é definida através da maior  $R^2$ .

Os métodos GG e GK possuem dois parâmetros de projeto cada um, ou seja,  $m$  e  $e$  para o GG, e  $w$  e  $e$  para o GK. Neste caso, foram realizados vários testes para obtenção de melhor desempenho dos métodos de agrupamento mencionados. Neste caso, para os algoritmos GG e GK foram escolhidos os seguintes valores para seus parâmetros de projeto:

- GG:  $m = 2$  e  $e = 10^{-6}$ ;
- GK:  $w = 2$  e  $e = 10^{-4}$ .

## 7.1 Resultados para o Trocador de calor

Nas tabelas 7.1 e 7.2 são apresentados os resultados das simulações na fase de estimação e validação, respectivamente, para o caso do trocador de calor.

Tabela 7.1: Resultados das simulações na fase de estimação para o trocador de calor.

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{est}$	$MSE_{est}$
<i>k</i> -médias	1	2	1	10	0,9941	0,0184
	2	2	1	12	0,9944	0,0174
	3	2	2	10	0,9113	0,2764
	4	2	2	20	0,9914	0,0269
	<b>5</b>	<b>2</b>	<b>2</b>	<b>25</b>	<b>0,9968</b>	<b>0,0100</b>
<i>c</i> -médias	1	2	1	10	0,9938	0,0193
	2	2	1	15	0,9942	0,0182
	3	2	2	10	0,9155	0,2634
	<b>4</b>	<b>2</b>	<b>2</b>	<b>20</b>	<b>0,9955</b>	<b>0,0142</b>
	5	2	2	25	0,9760	0,0747
GK	1	2	1	10	0,9676	0,1011
	<b>2</b>	<b>2</b>	<b>1</b>	<b>15</b>	<b>0,9960</b>	<b>0,0124</b>
	3	2	2	15	0,8547	0,4529
	4	2	2	17	0,8698	0,4056
	5	2	2	20	0,8202	0,5602
GG	1	2	1	6	0,8897	0,3437
	<b>2</b>	<b>2</b>	<b>1</b>	<b>9</b>	<b>0,9875</b>	<b>0,0300</b>
	3	2	2	10	0,8581	0,4421
	4	2	2	15	0,8748	0,3900
	5	2	2	17	0,9629	0,1155

Tabela 7.2: Resultados das simulações na fase de validação para o trocador de calor.

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{val}$	$MSE_{val}$	nº de op. algébricas
<i>k</i> -médias	1	2	1	10	0,9873	0,0356	0,0053
	2	2	1	12	0,9871	0,0361	0,0045
	3	2	2	10	0,7857	0,5985	0,0081
	4	2	2	20	0,9814	0,0518	0,0118
	<b>5</b>	<b>2</b>	<b>2</b>	<b>25</b>	<b>0,9922</b>	<b>0,0218</b>	<b>0,0203</b>
<i>c</i> -médias	1	2	1	10	0,9869	0,0365	0,0159
	2	2	1	15	0,9837	0,0454	0,0556
	3	2	2	10	0,8116	0,5262	0,0231
	<b>4</b>	<b>2</b>	<b>2</b>	<b>20</b>	<b>0,9900</b>	<b>0,0279</b>	<b>0,0900</b>
	5	2	2	25	0,9759	0,0679	0,1130
GK	1	2	1	10	0,9417	0,1629	0,1474
	<b>2</b>	<b>2</b>	<b>1</b>	<b>15</b>	<b>0,9942</b>	<b>0,0162</b>	<b>0,4316</b>
	3	2	2	15	0,7615	0,6661	0,5223
	4	2	2	17	0,8131	0,5221	0,5242
	5	2	2	20	0,8020	0,5532	0,5228
GG	1	2	1	6	0,7604	0,6690	0,0335
	<b>2</b>	<b>2</b>	<b>1</b>	<b>9</b>	<b>0,9798</b>	<b>0,0563</b>	<b>0,0773</b>
	3	2	2	10	0,7400	0,7262	0,1378
	4	2	2	15	0,7216	0,7777	0,5399
	5	2	2	17	0,9362	0,1783	1



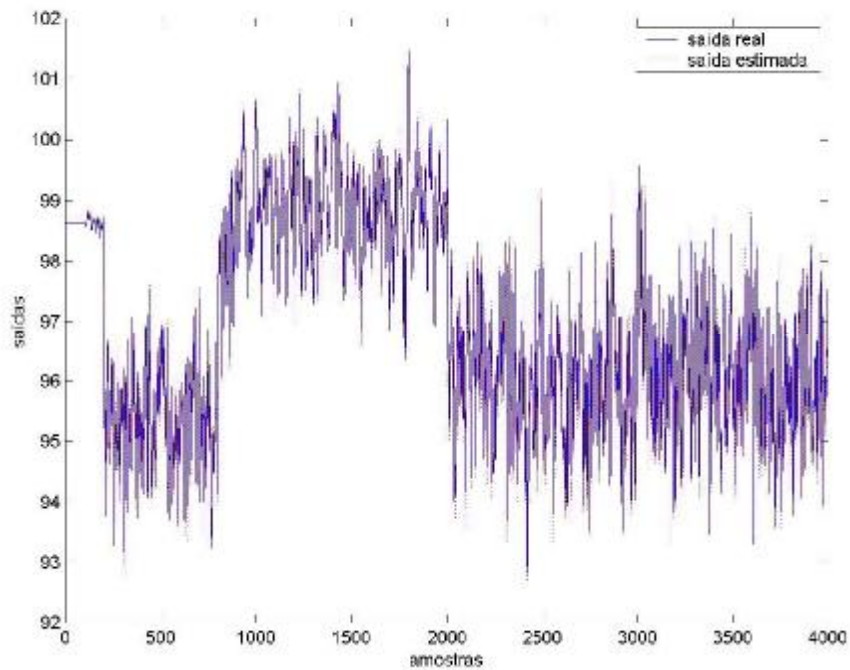


Figura 7.2. Saída do melhor resultado para identificação do trocador de calor com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 5).

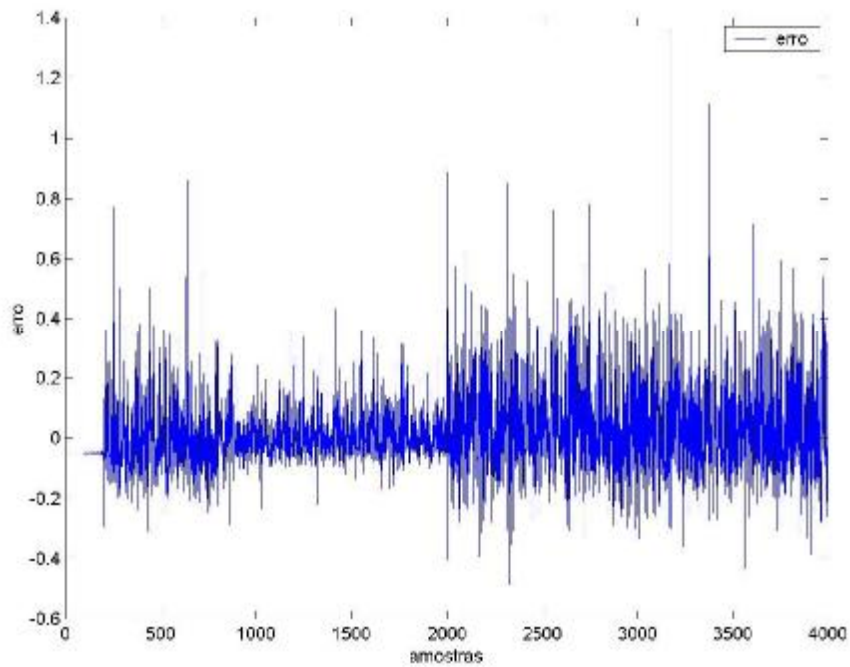


Figura 7.3. Erro do melhor resultado para identificação do trocador de calor com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 5).

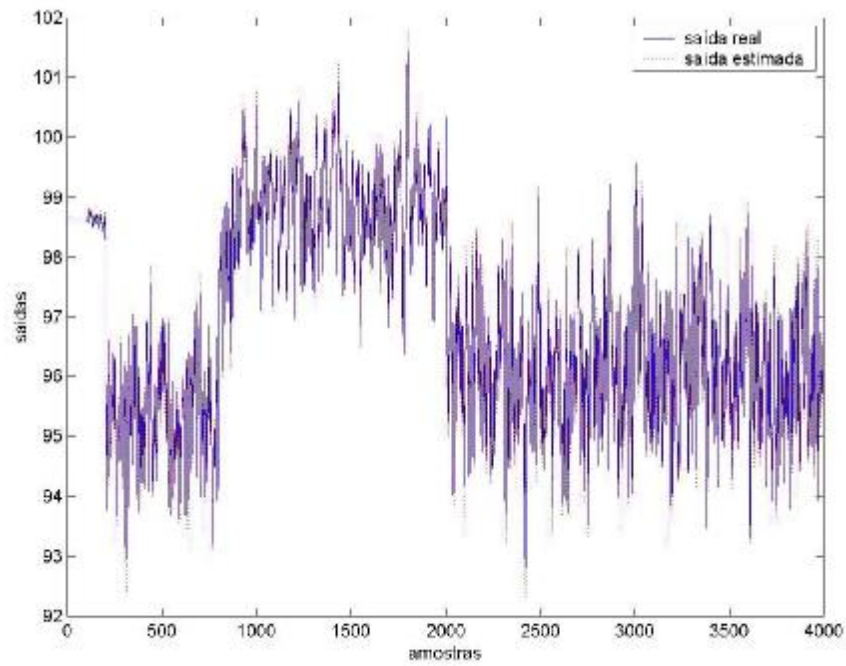


Figura 7.4. Saída do melhor resultado para identificação do trocador de calor com agrupamento *c*-médias e otimização pela pseudo-inversa (simulação 4).

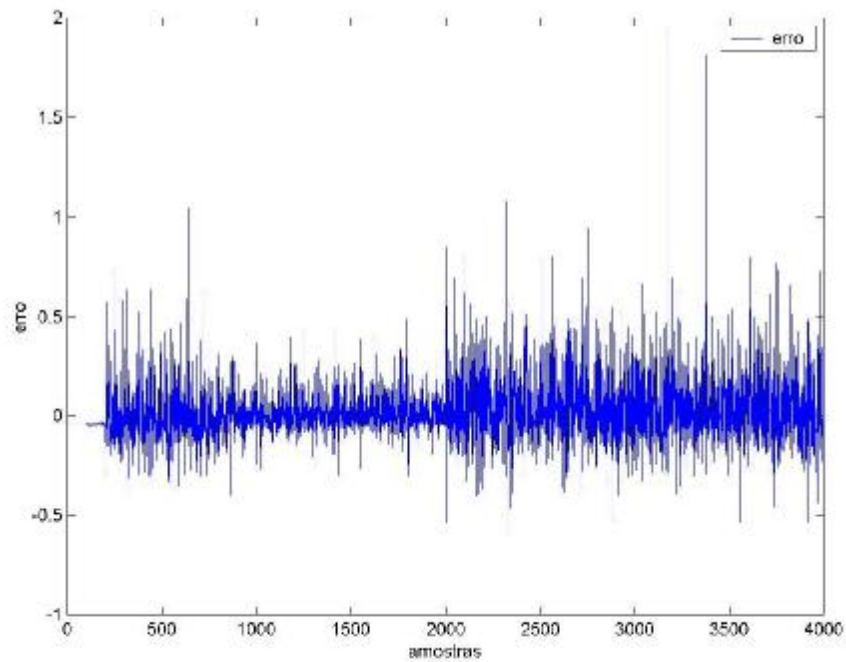


Figura 7.5. Erro do melhor resultado para identificação do trocador de calor com agrupamento *c*-médias e otimização pela pseudo-inversa (simulação 4).

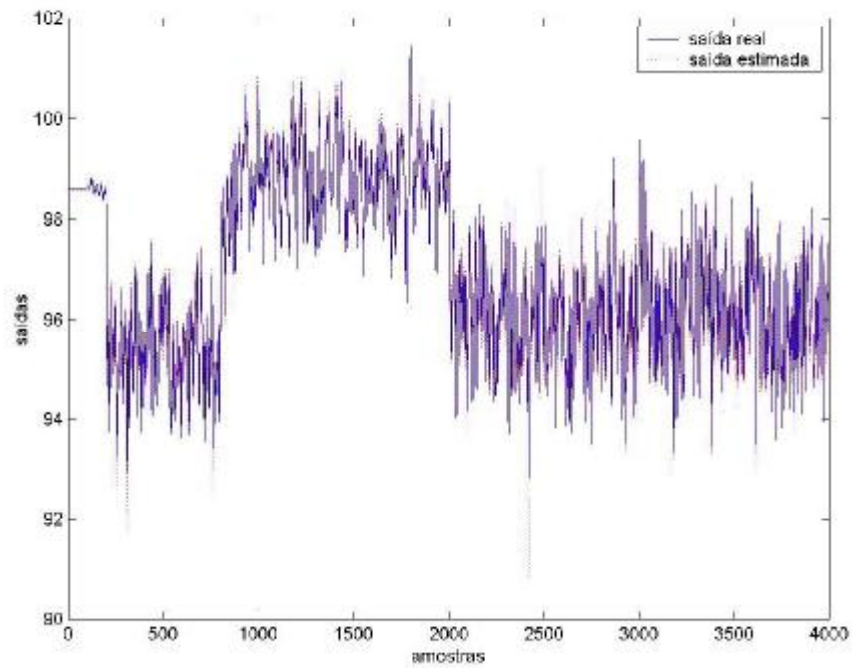


Figura 7.6. Saída do melhor resultado para identificação do trocador de calor com agrupamento GK e otimização pela pseudo-inversa (simulação 2).

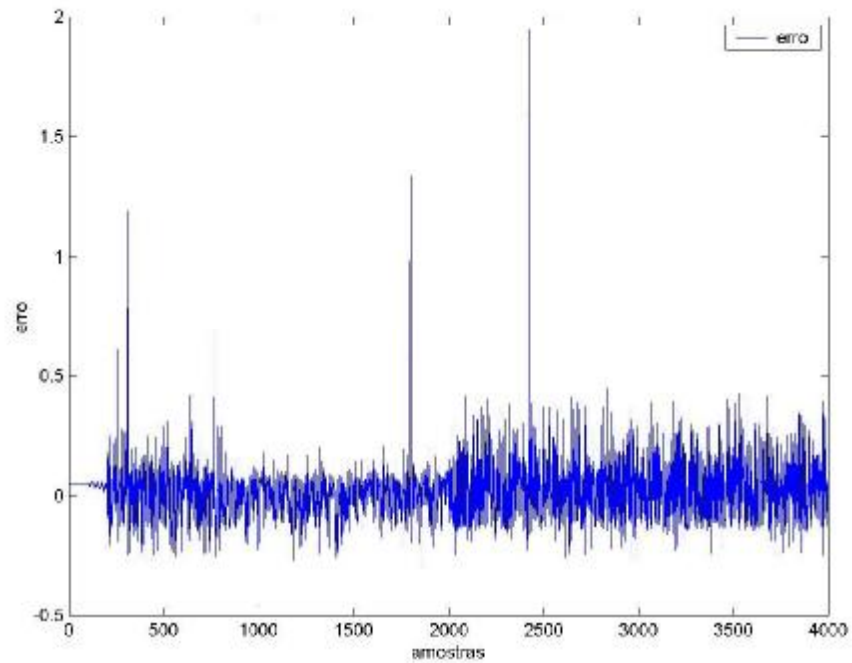


Figura 7.7. Erro do melhor resultado para identificação do trocador de calor com agrupamento GK e otimização pela pseudo-inversa (simulação 2).

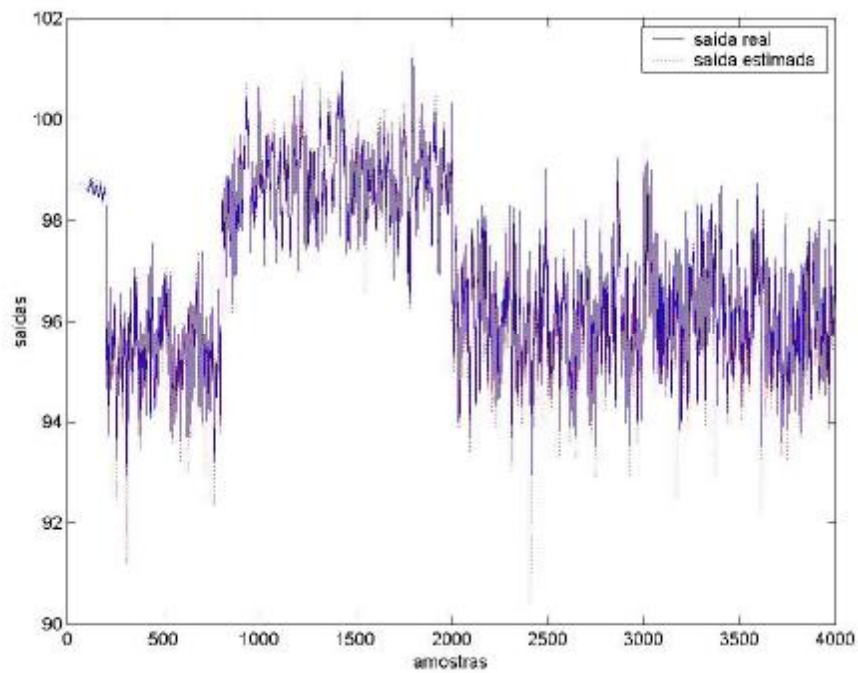


Figura 7.8. Saída do melhor resultado para identificação do trocador de calor com agrupamento GG e otimização pela pseudo-inversa (simulação 2).

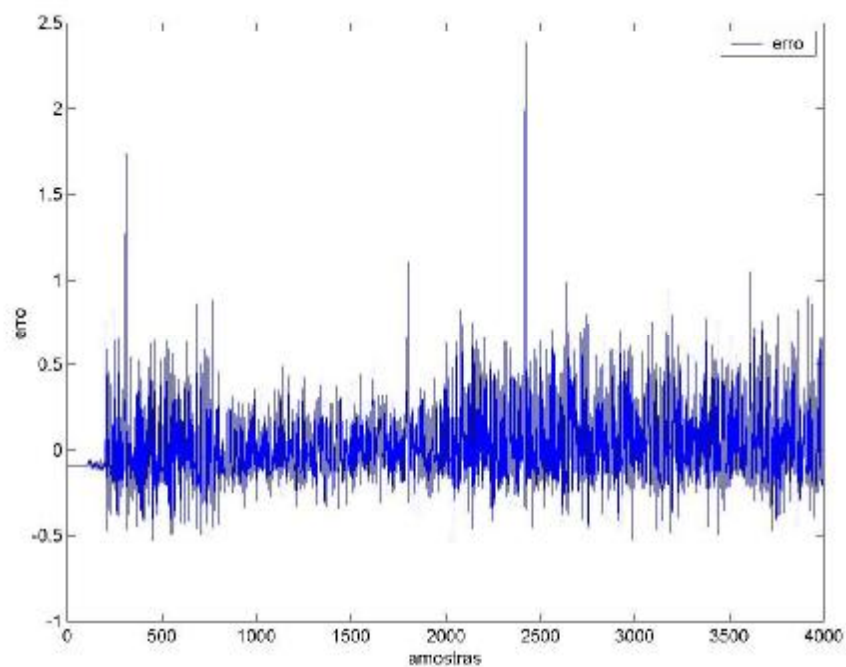


Figura 7.9. Erro do melhor resultado para identificação do trocador de calor com agrupamento GG e otimização pela pseudo-inversa (simulação 2).

### 7.1.1 Comentários dos resultados para o Trocador de Calor

Nota-se em relação aos resultados mostrados na tabela 7.1 (fase de estimação), que os métodos de agrupamentos foram bem sucedidos no projeto da rede neural de base radial, pois os valores obtidos ficaram no intervalo  $0,8202 \leq R^2_{est} \leq 0,9968$ . Nota-se também que o melhor resultado foi obtido com método de treinamento usando  $k$ -médias com  $R^2_{est} = 0,9968$  e  $MSE = 0,0100$  para um projeto usando  $N_u = N_y = 2$  e 25 grupos (*clusters*). Um detalhe que fica evidente é que o método GK nas simulações 3 a 5 ( $R^2_{est}$  com valor próximo a 0,8) tem um desempenho inferior aos da simulações 1 e 2. Isto decorre do fato dos modelos de menor ordem ( $N_u$  e  $N_y$ ) terem sido os mais promissores para o ajuste pelo GK.

Em relação à tabela 7.2 (fase de validação), observa-se que ao contrário da tabela 7.1 o melhor resultado não foi o obtido pelo  $k$ -médias, mas pelo método de GK com valor de  $R^2_{val}$  próximo à unidade. Entretanto, em geral, os métodos de GK e GG apresentam elevada complexidade computacional (número de operações algébricas) em relação aos outros três métodos abordados neste estudo de caso.

## 7.2 Resultados para o Sistema de Mackey-Glass

Nas tabelas 7.3 e 7.4 são resumidos os resultados das simulações na fase de estimação e validação, respectivamente, para o caso do sistema de Mackey-Glass.

Tabela 7.3: Resultados das simulações na fase de estimação para o sistema de Mackey-Glass.

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{est}$	$MSE_{est}$
<i>k</i> -médias	1	0	2	2	0,9535	0,0028
	2	0	2	4	0,9981	0,0001
	3	0	3	2	0,9191	0,0048
	4	0	3	4	0,9884	0,0006
	<b>5</b>	<b>0</b>	<b>3</b>	<b>6</b>	<b>0,9990</b>	<b>0,00005</b>
<i>c</i> -médias	1	0	2	2	0,9532	0,0028
	2	0	2	4	0,9538	0,0027
	3	0	3	2	0,9191	0,0048
	4	0	3	4	0,9801	0,0012
	<b>5</b>	<b>0</b>	<b>3</b>	<b>6</b>	<b>0,9991</b>	<b>0,00005</b>
GK	1	0	2	2	0,9562	0,0026
	<b>2</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9994</b>	<b>0,00003</b>
	3	0	3	2	0,9288	0,0042
	4	0	3	4	0,9983	0,00009
	5	0	3	6	0,9992	0,00004
GG	1	0	2	2	0,9544	0,0027
	<b>2</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9993</b>	<b>0,00003</b>
	3	0	3	2	0,9178	0,0049
	4	0	3	4	0,9982	0,0001
	5	0	3	6	0,9919	0,0004

Tabela 7.4: Resultados das simulações na fase de validação para o sistema de Mackey-Glass.

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{val}$	$MSE_{val}$	nº de op. algébricas
<i>k</i> -médias	1	0	2	2	0,9531	0,0026	0,0020
	2	0	2	4	0,9983	0,00009	0,0048
	3	0	3	2	0,9186	0,0045	0,0022
	4	0	3	4	0,9895	0,0005	0,0093
	<b>5</b>	<b>0</b>	<b>3</b>	<b>6</b>	<b>0,9991</b>	<b>0,00004</b>	<b>0,0137</b>
<i>c</i> -médias	1	0	2	2	0,9531	0,0026	0,0044
	2	0	2	4	0,9537	0,0026	0,0132
	3	0	3	2	0,9186	0,0045	0,0059
	4	0	3	4	0,9819	0,0010	0,0238
	<b>5</b>	<b>0</b>	<b>3</b>	<b>6</b>	<b>0,9991</b>	<b>0,00005</b>	<b>0,0497</b>
GK	1	0	2	2	0,9561	0,0024	0,0229
	<b>2</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9994</b>	<b>0,00003</b>	<b>0,0918</b>
	3	0	3	2	0,9283	0,0040	0,0514
	4	0	3	4	0,9984	0,00008	0,2524
	5	0	3	6	0,9992	0,00005	1
GG	1	0	2	2	0,9543	0,0025	0,0556
	<b>2</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9994</b>	<b>0,00003</b>	<b>0,2450</b>
	3	0	3	2	0,9173	0,0046	0,0830
	4	0	3	4	0,9983	0,00009	0,2743
	5	0	3	6	0,9914	0,0004	0,49537

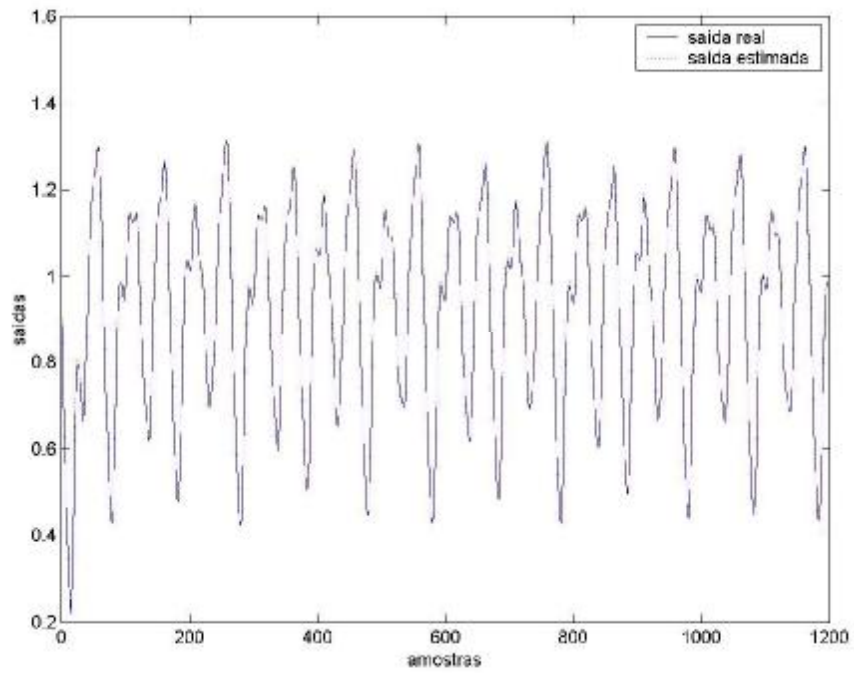


Figura 7.10. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 5).

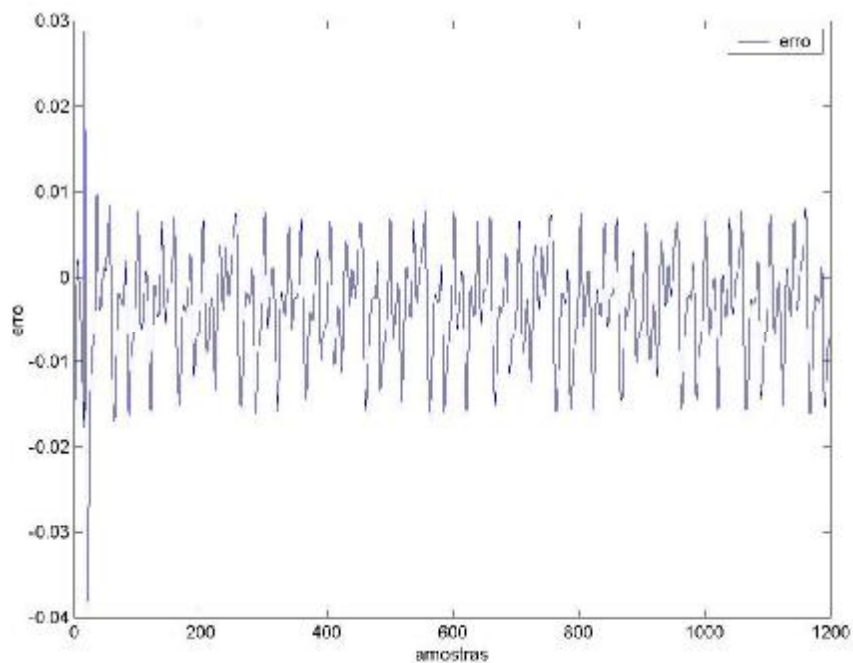


Figura 7.11. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 5).



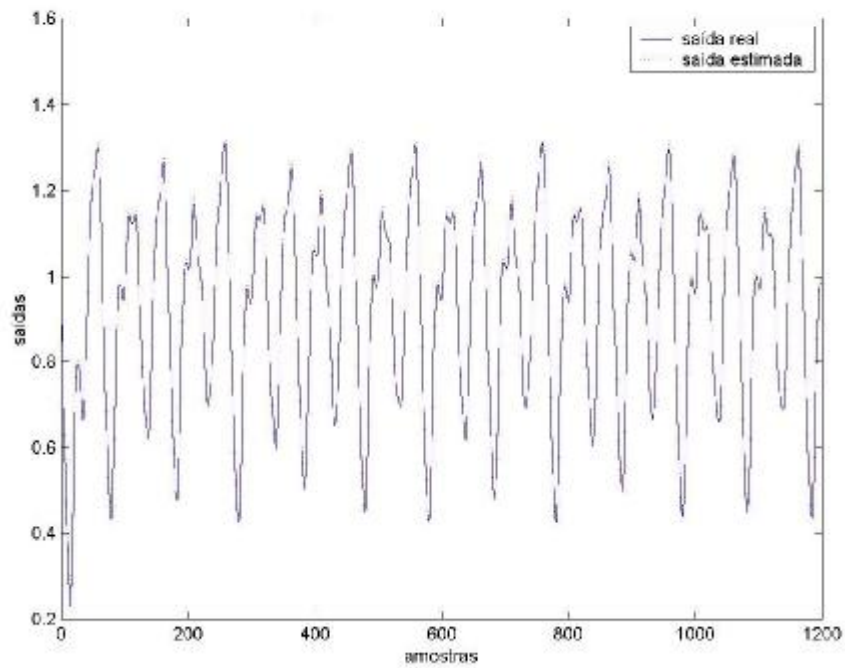


Figura 7.12. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 5).

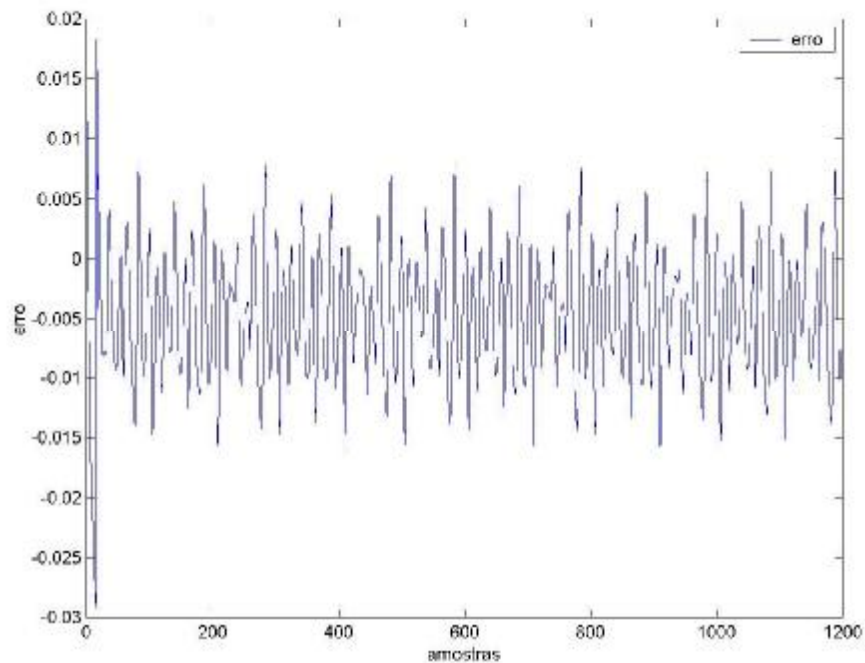


Figura 7.13. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 5).

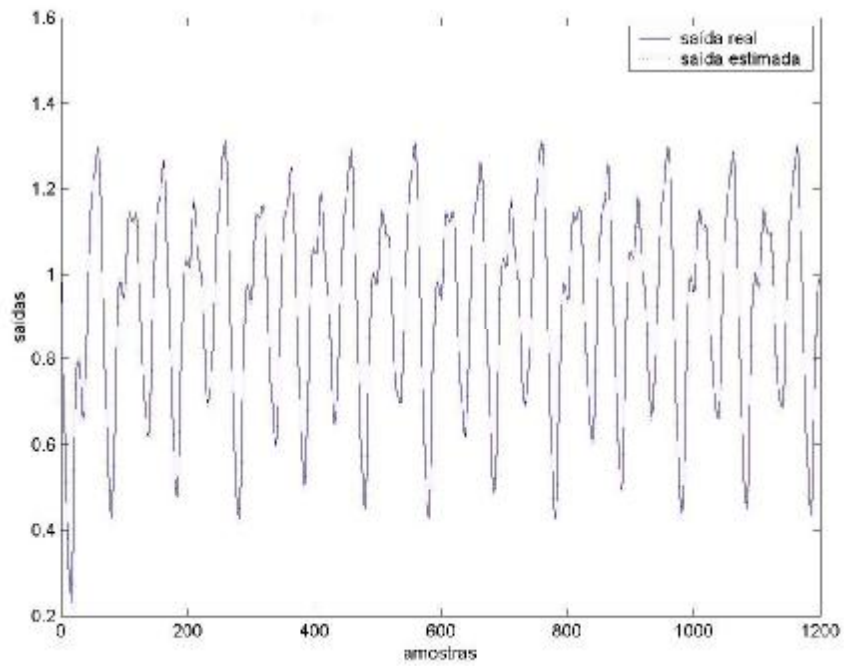


Figura 7.14. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GKe otimização pela pseudo-inversa (simulação 2).

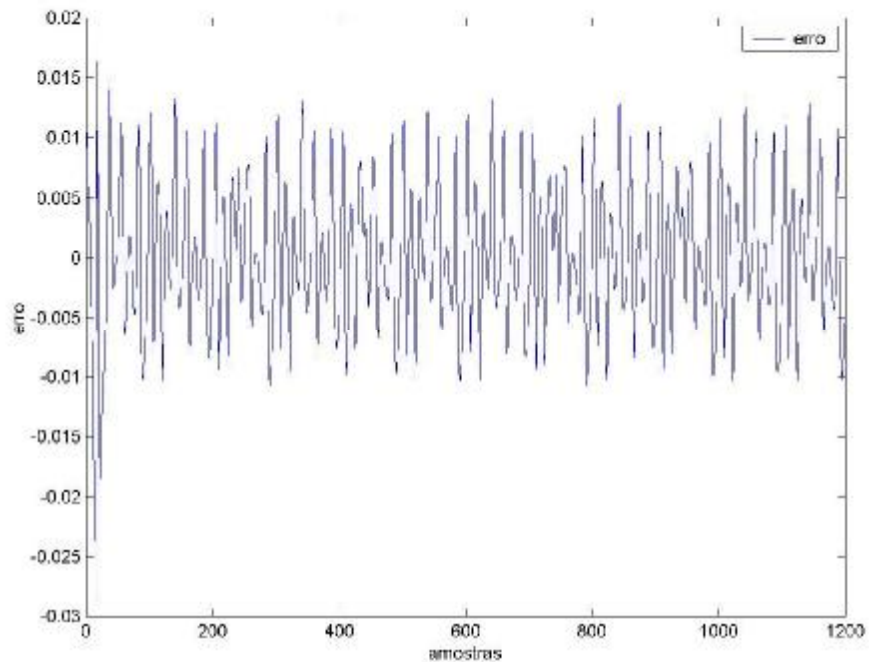


Figura 7.15. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GK e otimização pela pseudo-inversa (simulação 2).

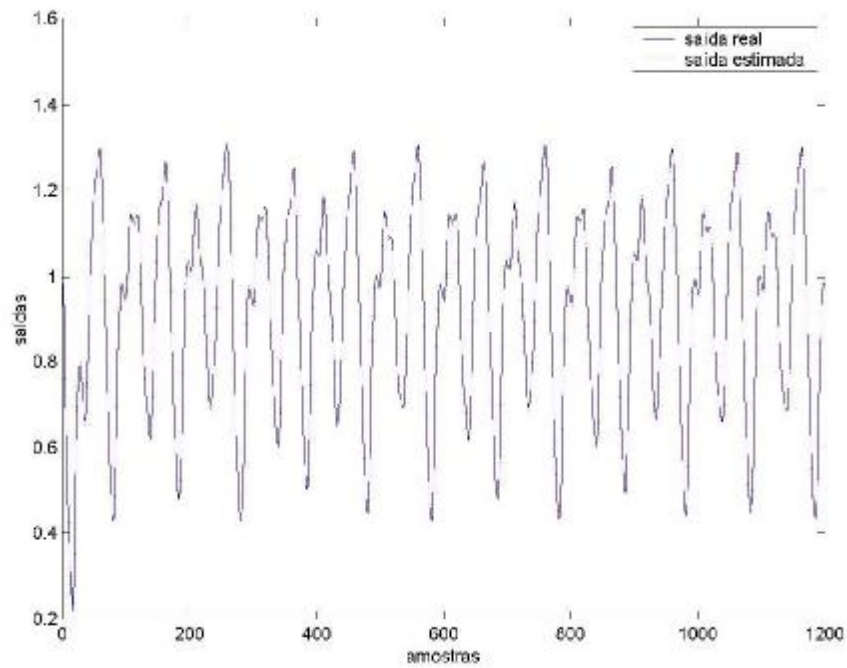


Figura 7.16. Saída do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GG e otimização pela pseudo-inversa (simulação 2).

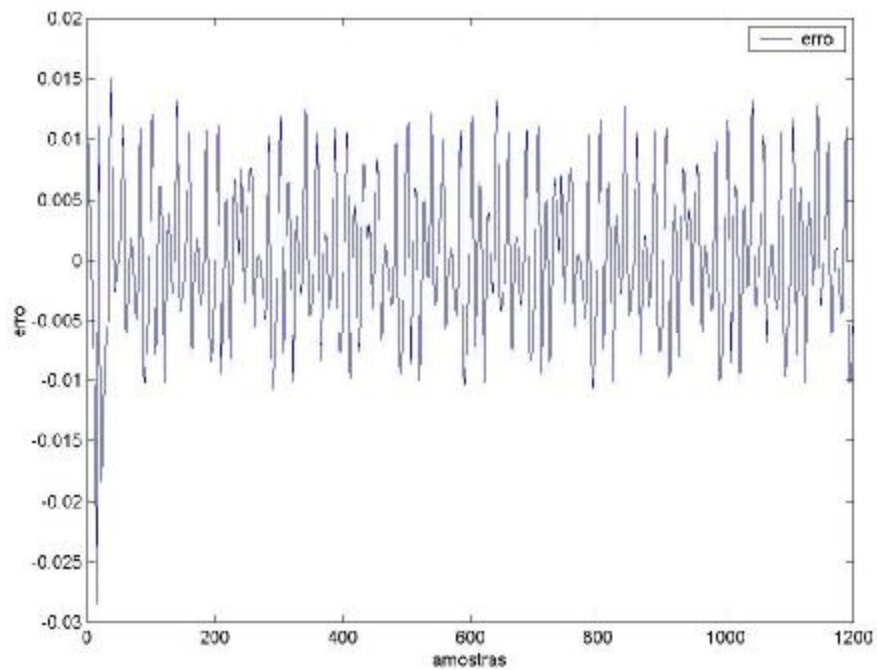


Figura 7.17. Erro do melhor resultado para identificação da equação diferencial de Mackey-Glass com agrupamento GG e otimização pela pseudo-inversa (simulação 2).

### 7.2.1 Comentários dos resultados para o Sistema de Mackey-Glass

Em relação aos resultados da fase de estimação mostrados na tabela 7.3 observa-se que o método de GK com  $N_u = 0$  e  $N_y = 2$  usando 4 grupos (*clusters*) na camada oculta da rede neural de base radial obteve o melhor resultado. Um aspecto a ser mencionado é que os métodos foram eficientes, onde o pior desempenho obteve  $R^2_{est}$  foi de 0,9178, o que mostra uma boa aproximação do comportamento dinâmico do sistema de Mackey-Glass identificado.

Em relação a tabela 7.4 (fase de validação), observa-se o excelente desempenho do GG e GK (usando  $N_u = 0$  e  $N_y = 2$  e 4 grupos), mas com maior custo computacional que os métodos  $k$ -médias e  $c$ -médias. Deve-se mencionar também que apesar do desempenho do GG e GK, tanto o  $k$ -médias quanto o  $c$ -médias projetaram a rede neural de base radial com êxito, pois o mapeamento da rede ficou com um  $R^2_{val} \geq 0,9186$ .

## 7.3 Resultados para o Sistema Caótico de Rössler

### 7.3.1 Resultados para a coordenada $x$

Nas tabelas 7.5 e 7.6 são resumidos os resultados das simulações na fase de estimação e validação, respectivamente, para a coordenada  $x$  do sistema caótico de Rössler.

Tabela 7.5: Resultados das simulações na fase de estimação para a coordenada  $x$ .

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{est}$	$MSE_{est}$
$k$ -médias	<b>1</b>	<b>0</b>	<b>2</b>	<b>2</b>	<b>0,9998</b>	0,000008
	2	0	2	3	0,9998	0,000008
	3	0	3	2	0,9996	0,000001
	4	0	3	3	0,9996	0,000001
	5	0	3	4	0,9998	0,000008
$c$ -médias	<b>1</b>	<b>0</b>	<b>2</b>	<b>2</b>	<b>0,9998</b>	<b>0,000008</b>
	2	0	2	3	0,9998	0,000008
	3	0	3	2	0,9996	0,000001
	4	0	3	3	0,9996	0,000001
	5	0	3	4	0,9998	0,000008
GK	1	0	2	2	0,9998	0,000008
	2	0	2	3	0,9998	0,000008
	3	0	3	2	0,9996	0,000001
	4	0	3	3	0,9996	0,000001
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9998</b>	<b>0,000008</b>
GG	1	0	2	2	0,9998	0,000001
	2	0	2	3	0,9998	0,000008
	<b>3</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9999</b>	<b>0,000003</b>
	4	0	2	5	0,9999	0,000002
	5	0	2	6	0,9708	0,0013

Tabela 7.6: Resultados das simulações na fase de validação para a coordenada  $x$ .

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{val}$	$MSE_{val}$	nº de op. algébricas
$k$ -médias	<b>1</b>	<b>0</b>	<b>2</b>	<b>2</b>	<b>0,9998</b>	<b>0,000008</b>	<b>0,00007</b>
	2	0	2	3	0,9998	0,000008	0,00013
	3	0	3	2	0,9997	0,00001	0,00012
	4	0	3	3	0,9997	0,00001	0,00017
	5	0	3	4	0,9998	0,000008	0,00027
$c$ -médias	1	<b>0</b>	<b>2</b>	<b>2</b>	<b>0,9998</b>	<b>0,000008</b>	<b>0,00026</b>
	2	0	2	3	0,9998	0,000008	0,00038
	3	0	3	2	0,9997	0,00001	0,00032
	4	0	3	3	0,9997	0,00001	0,00050
	5	0	3	4	0,9998	0,000008	0,00110
GK	1	0	2	2	0,9998	0,000008	0,00038
	2	0	2	3	0,9998	0,000008	0,00090
	3	0	3	2	0,9997	0,0001	0,00090
	4	0	3	3	0,9997	0,0001	0,0017
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9999</b>	<b>0,000007</b>	<b>0,0053</b>
GG	1	0	2	2	0,9998	0,000009	0,0012
	2	0	2	3	0,9998	0,000008	0,0398
	<b>3</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9999</b>	<b>0,000002</b>	<b>0,0454</b>
	4	0	2	5	0,9999	0,000002	0,2986
	5	0	2	6	0,9756	0,0013	1

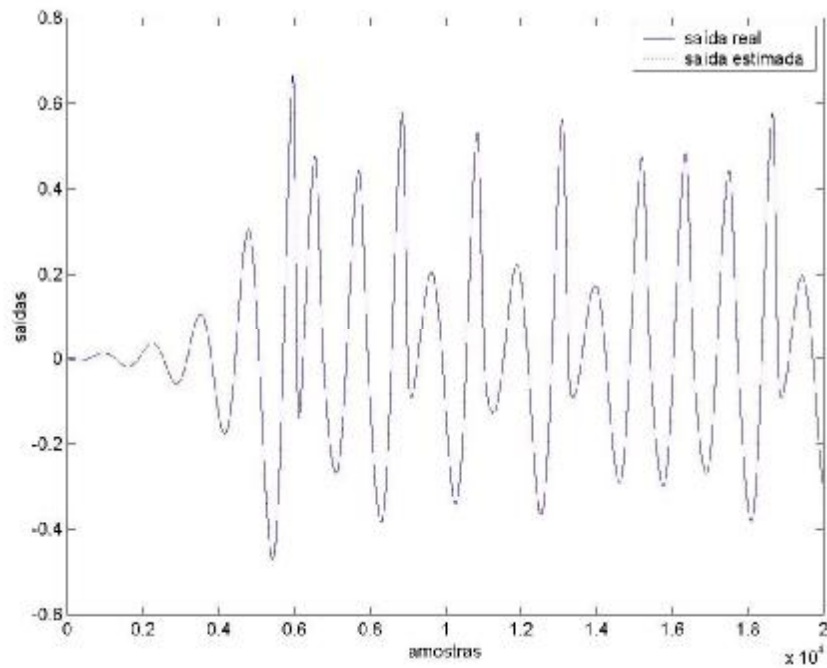


Figura 7.18. Saída do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 1).

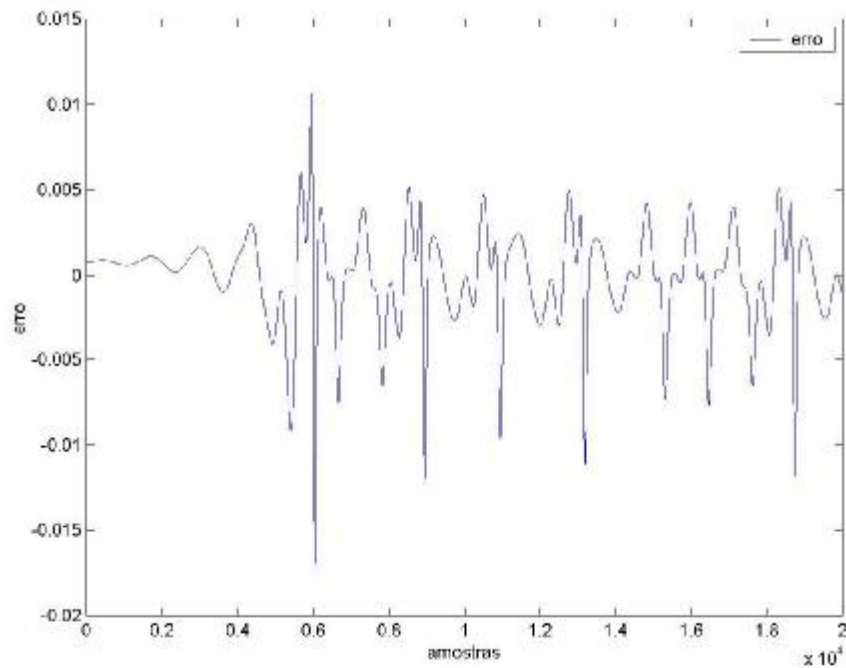


Figura 7.19. Erro do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 1).

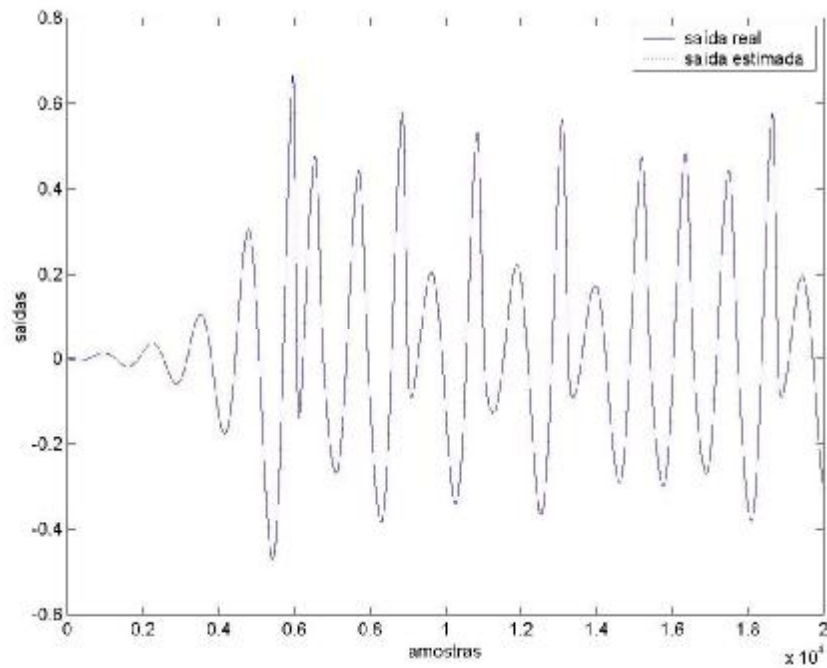


Figura 7.20. Saída do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 1).

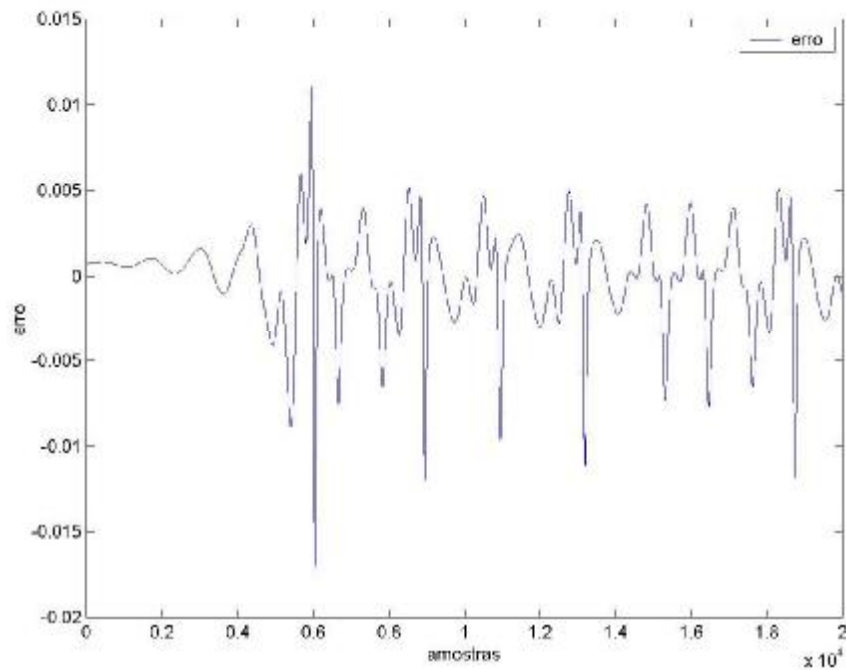


Figura 7.21. Erro do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 1).



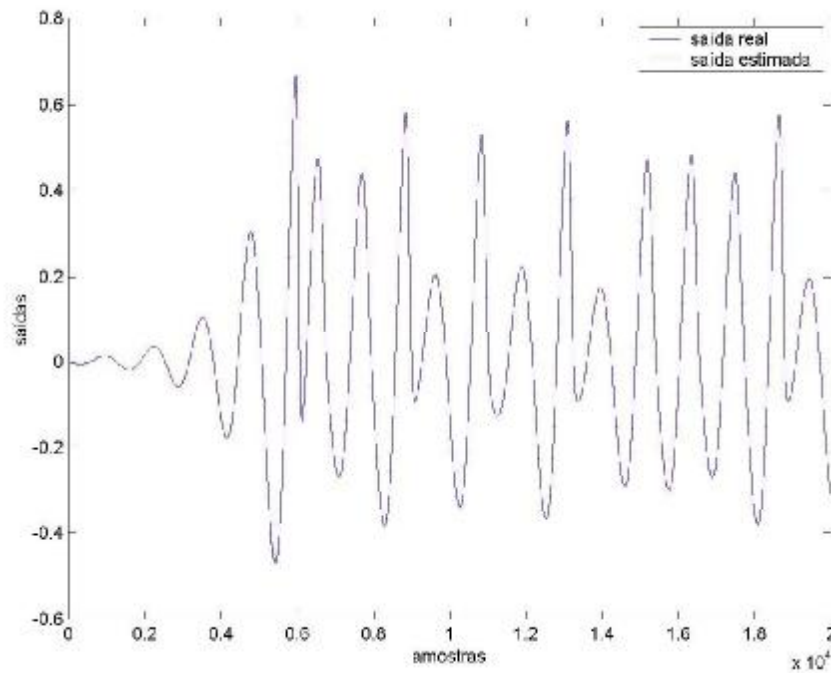


Figura 7.22. Saída do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5).

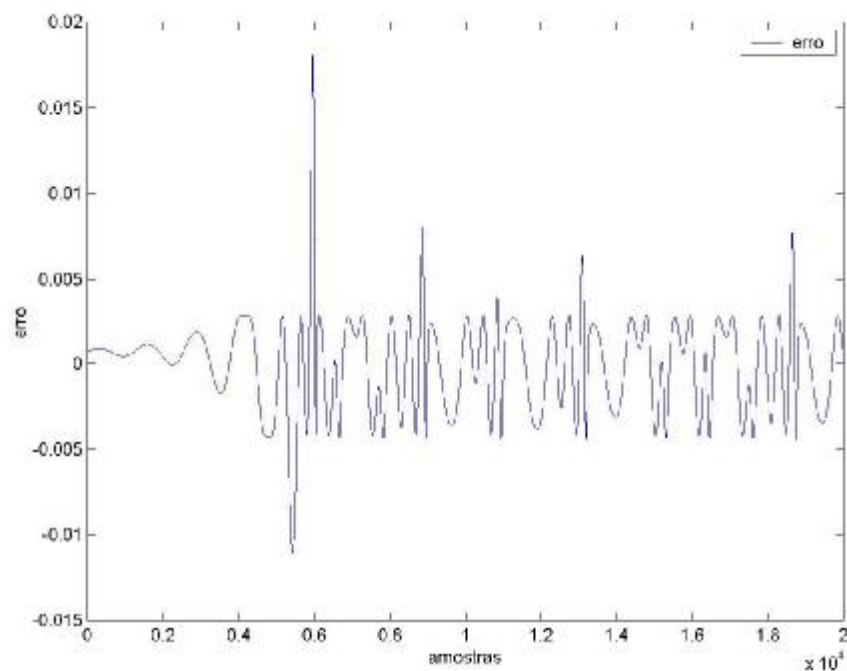


Figura 7.23. Erro do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5).

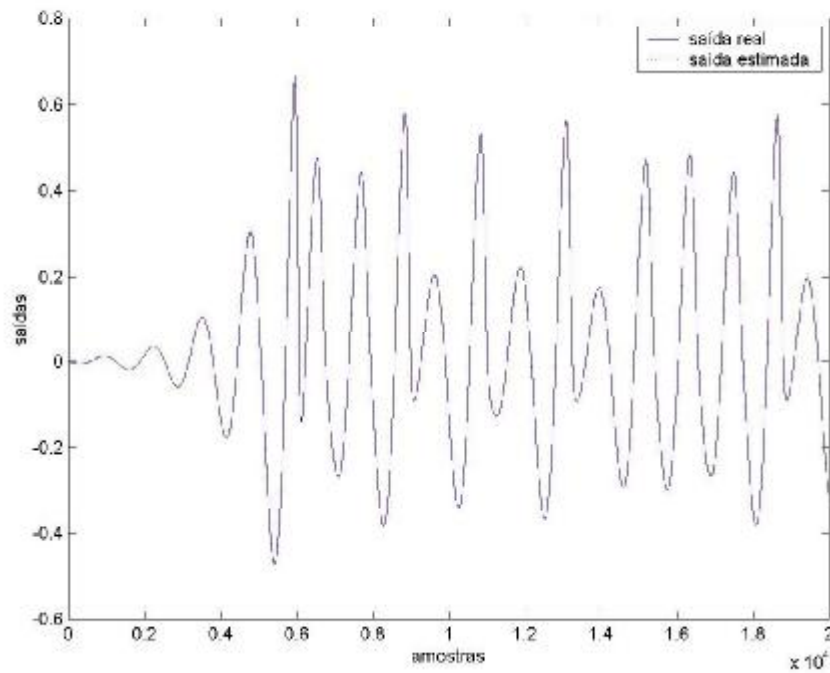


Figura 7.24. Saída do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 3).

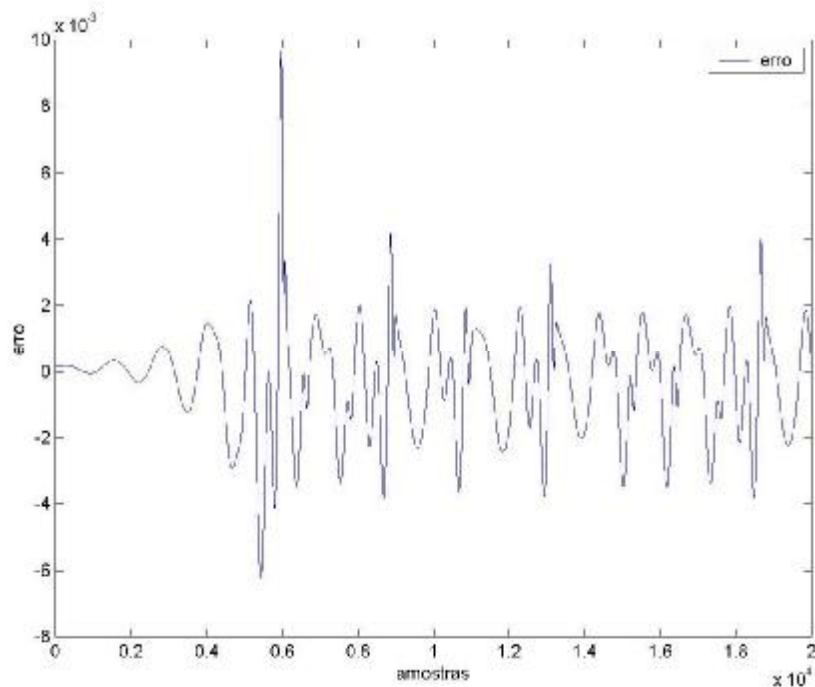


Figura 7.25. Erro do melhor resultado para identificação da coordenada  $x$  do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 3).

### **7.3.1.1 Comentários dos resultados para a coordenada $x$**

Em relação aos resultados da fase de estimação e validação, estes mostrados nas tabelas 7.5 e 7.6, respectivamente, observa-se que todos os métodos de agrupamento testados foram eficientes e obtiveram um projeto de rede neural de base radial que apresenta mapeamento preciso para identificação um-passo-à-frente da coordenada  $x$  de Rössler. Neste contexto, os resultados ficaram muito próximos tanto para o  $R^2_{est}$  quanto para o  $R^2_{val}$ . Nota-se também que o método GG apresentou maior custo computacional que os outros métodos testados.

### 7.3.2 Resultados para a coordenada $y$

Nas tabelas 7.7 e 7.8 são apresentados os resultados das simulações na fase de estimação e validação, respectivamente, para a coordenada  $y$  do sistema caótico de Rössler.

Tabela 7.7: Resultados das simulações na fase de estimação para a coordenada  $y$ .

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{est}$	$MSE_{est}$
$k$ -médias	1	0	2	2	0,9992	0,0001
	2	0	2	3	0,9993	0,0001
	3	0	3	2	0,9983	0,0002
	4	0	3	3	0,9984	0,0002
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9999</b>	<b>0,00001</b>
$c$ -médias	1	0	2	2	0,9993	0,0001
	2	0	2	3	0,9993	0,0001
	3	0	3	2	0,9984	0,0002
	4	0	3	3	0,9984	0,0002
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9999</b>	<b>0,00001</b>
GK	1	0	2	2	0,9992	0,0001
	2	0	2	3	0,9993	0,0001
	<b>3</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9999</b>	<b>0,000007</b>
	4	0	2	5	0,9999	0,000008
	5	0	3	2	0,9971	0,0004
GG	1	0	2	2	0,9993	0,0001
	2	0	2	5	0,9973	0,003
	3	0	3	2	0,9984	0,0002
	<b>4</b>	<b>0</b>	<b>3</b>	<b>5</b>	<b>0,9999</b>	<b>0,0000007</b>
	5	0	3	6	0,9997	0,00004

Tabela 7.8: Resultados das simulações na fase de validação para a coordenada  $y$ .

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	n° de grupos	$R^2_{val}$	$MSE_{val}$	n° de op. algébricas
$k$ -médias	1	0	2	2	0,9994	0,00008	0,0077
	2	0	2	3	0,9995	0,00008	0,0112
	3	0	3	2	0,9998	0,0001	0,0140
	4	0	3	3	0,9988	0,0001	0,0159
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9999</b>	<b>0,00001</b>	<b>0,0213</b>
$c$ -médias	1	0	2	2	0,9994	0,00008	0,0242
	2	0	2	3	0,9995	0,00008	0,0388
	3	0	3	2	0,9988	0,0001	0,0296
	4	0	3	3	0,9988	0,0001	0,0530
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9999</b>	<b>0,00001</b>	<b>0,1207</b>
GK	1	0	2	2	0,9994	0,00009	0,0606
	2	0	2	3	0,9994	0,00008	0,3730
	<b>3</b>	<b>0</b>	<b>2</b>	<b>4</b>	<b>0,9999</b>	<b>0,000008</b>	<b>0,6591</b>
	4	0	2	5	0,9999	0,000008	1
	5	0	3	2	0,9978	0,0003	0,0822
GG	1	0	2	2	0,9994	0,00008	0,1762
	2	0	2	5	0,9980	0,0003	0,0904
	3	0	3	2	0,9988	0,001	0,1693
	<b>4</b>	<b>0</b>	<b>3</b>	<b>5</b>	<b>0,9999</b>	<b>0,0000007</b>	<b>0,3877</b>
	5	0	3	6	0,9997	0,00004	0,4923

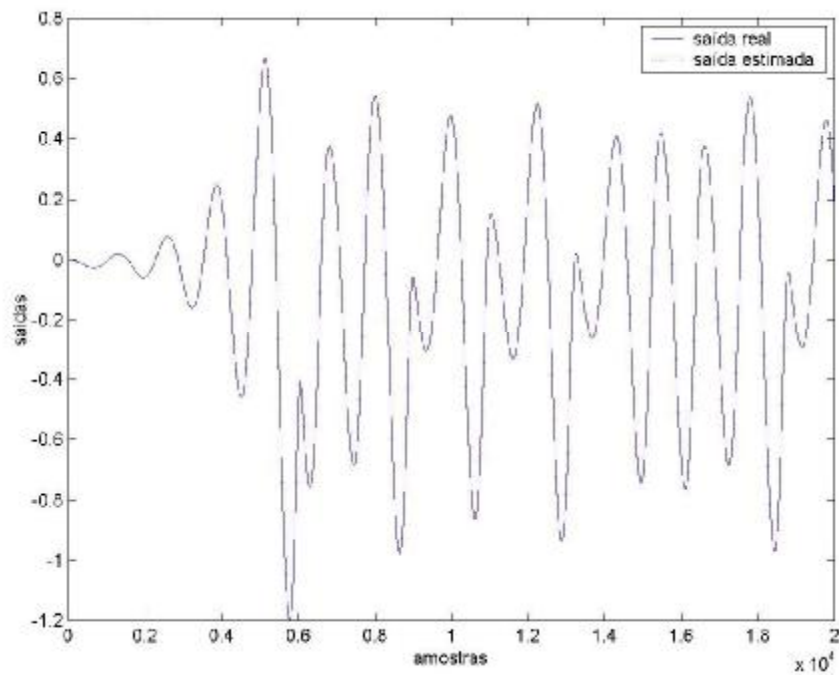


Figura 7.26. Saída do melhor resultado para identificação da coordenada y do sistema de Rössler com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 5).

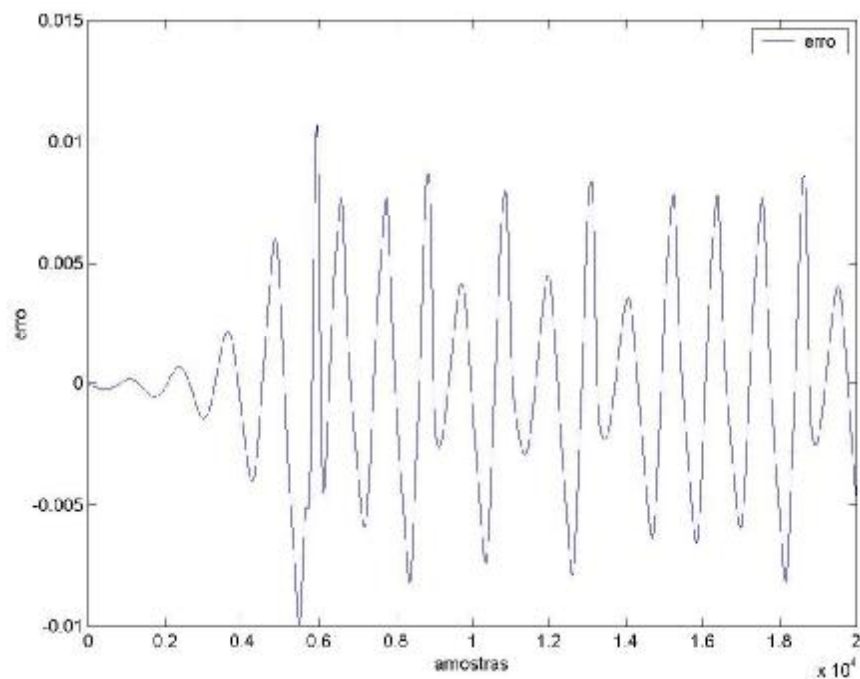


Figura 7.27. Erro do melhor resultado para identificação da coordenada y do sistema de Rössler com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 5).

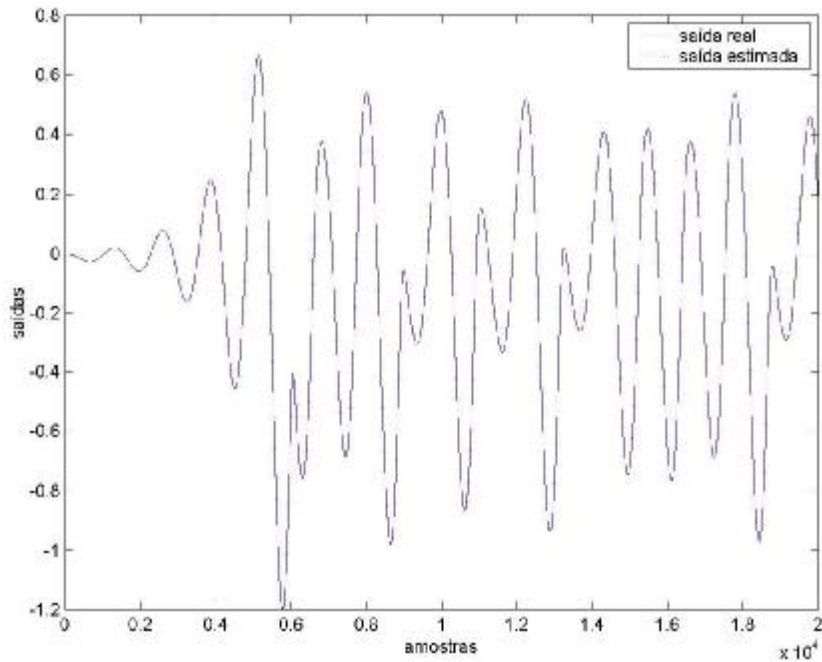


Figura 7.28. Saída do melhor resultado para identificação da coordenada  $y$  do sistema de Rössler com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 5).

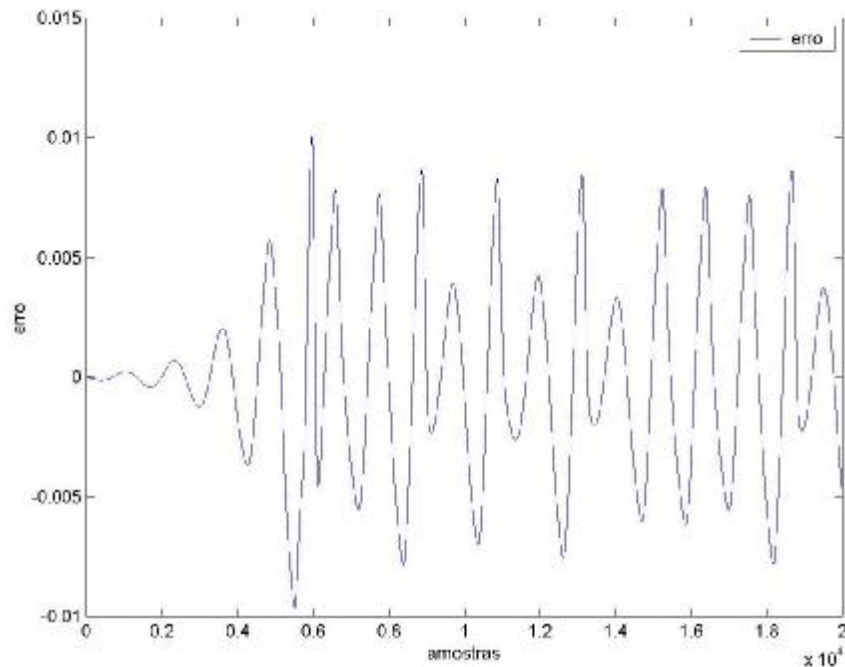


Figura 7.29. Erro do melhor resultado para identificação da coordenada  $y$  do sistema de Rössler com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 5).

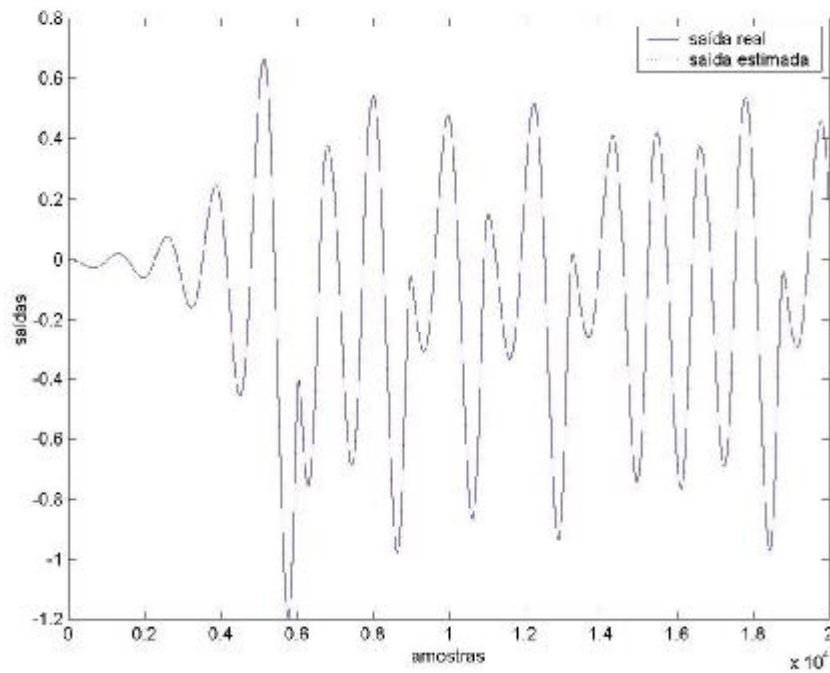


Figura 7.30. Saída do melhor resultado para identificação da coordenada y do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 3).

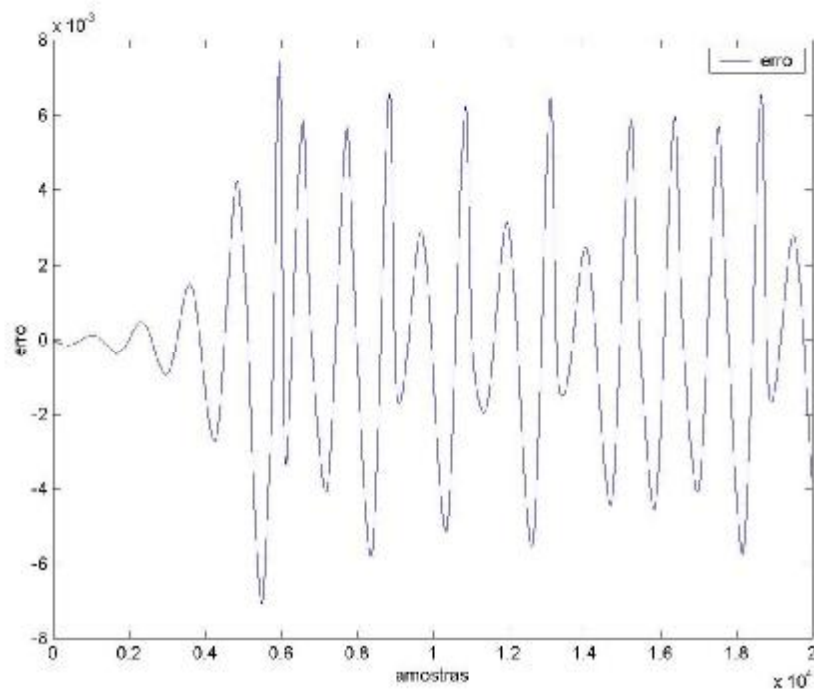


Figura 7.31. Erro do melhor resultado para identificação da coordenada y do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 3).



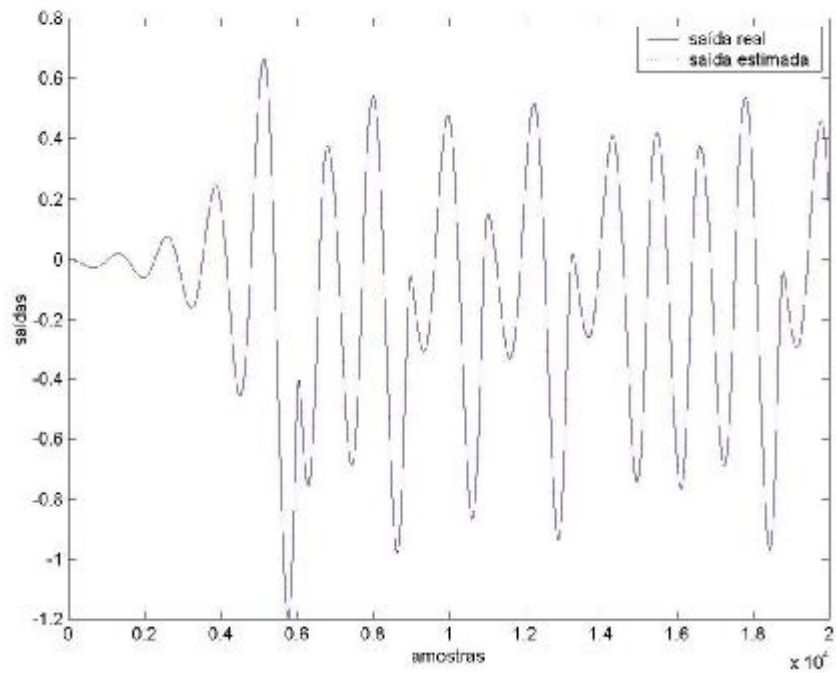


Figura 7.32. Saída do melhor resultado para identificação da coordenada y do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 4).

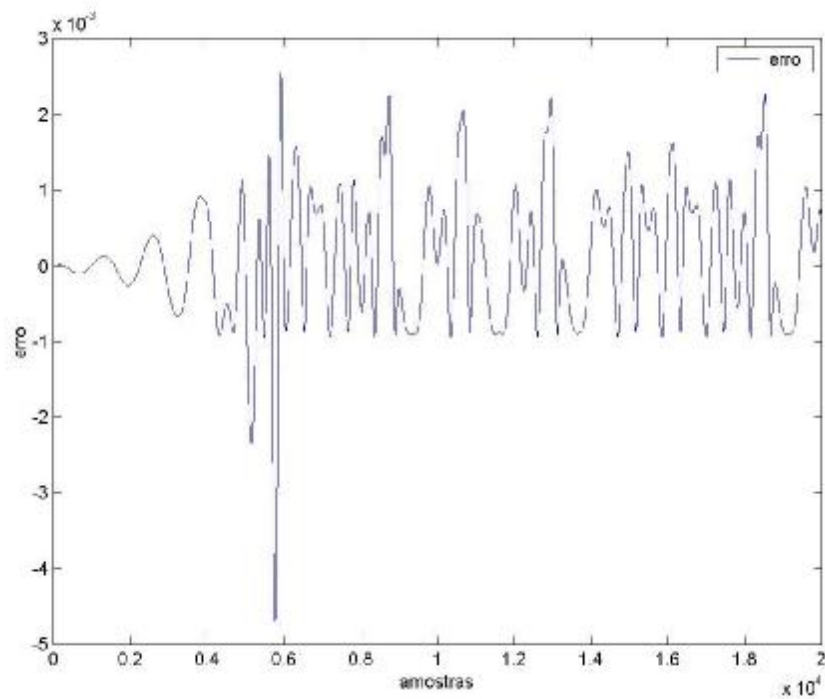


Figura 7.33. Erro do melhor resultado para identificação da coordenada y do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 4).

### **7.3.2.1 Comentários para os resultados para a coordenada $y$**

As tabelas 7.5 e 7.6 mostraram os resultados da fase de estimação e validação, respectivamente para a coordenada  $y$  do sistema caótico de Rössler. Observa-se que todos os métodos de agrupamento testados foram eficientes e obtiveram um projeto de rede neural de base radial que apresenta mapeamento preciso para identificação um-passo-à-frente com  $R^2 = 0,9999$  tanto na fase de estimação quanto na validação. Nota-se também que o método GK apresentou maior custo computacional que os outros métodos testados.

### 7.3.3 Resultados para a coordenada $z$

Nas tabelas 7.9 e 7.10 são resumidos os resultados das simulações na fase de estimação e validação, respectivamente, para a coordenada  $z$  do sistema caótico de Rössler.

Tabela 7.9: Resultados das simulações na fase de estimação para a coordenada  $z$ .

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{est}$	$MSE_{est}$
<i>k</i> -médias	1	0	2	2	0,9993	0,000002
	<b>2</b>	<b>0</b>	<b>2</b>	<b>3</b>	<b>0,9994</b>	<b>0,000002</b>
	3	0	3	2	0,9988	0,000005
	4	0	3	3	0,9989	0,000006
	5	0	3	4	0,9990	0,000004
<i>c</i> -médias	1	0	2	2	0,9993	0,000003
	<b>2</b>	<b>0</b>	<b>2</b>	<b>3</b>	<b>0,9994</b>	<b>0,000006</b>
	3	0	3	2	0,9988	0,000005
	4	0	3	3	0,9989	0,000004
	5	0	3	4	0,9989	0,000006
GK	1	0	2	2	0,9993	0,000003
	2	0	2	3	0,9994	0,000002
	3	0	3	2	0,9988	0,000005
	4	0	3	3	0,9989	0,000004
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9999</b>	<b>0,00000003</b>
GG	<b>1</b>	<b>0</b>	<b>2</b>	<b>2</b>	<b>0,9992</b>	<b>0,000003</b>
	2	0	2	3	0,9990	0,000004
	3	0	3	2	0,9985	0,000006
	4	0	3	3	0,9989	0,000004
	5	0	3	4	0,9980	0,000008

Tabela 7.10: Resultados das simulações na fase de validação para a coordenada  $z$ .

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{val}$	$MSE_{val}$	nº de op. algébricas
<i>k</i> -médias	1	0	2	2	0,9994	0,000002	0,0105
	<b>2</b>	<b>0</b>	<b>2</b>	<b>3</b>	<b>0,9995</b>	<b>0,000002</b>	<b>0,0181</b>
	3	0	3	2	0,9989	0,000004	0,0141
	4	0	3	3	0,9990	0,000003	0,0229
	5	0	3	4	0,9991	0,000003	0,0481
<i>c</i> -médias	1	0	2	2	0,9994	0,000006	0,0316
	<b>2</b>	<b>0</b>	<b>2</b>	<b>3</b>	<b>0,9995</b>	<b>0,000002</b>	<b>0,0563</b>
	3	0	3	2	0,9989	0,000004	0,0418
	4	0	3	3	0,9990	0,000003	0,0669
	5	0	3	4	0,9990	0,000003	0,1233
GK	1	0	2	2	0,9994	0,000002	0,0859
	2	0	2	3	0,9995	0,000002	0,3248
	3	0	3	2	0,9989	0,000004	0,7376
	4	0	3	3	0,9991	0,000003	1
	<b>5</b>	<b>0</b>	<b>3</b>	<b>4</b>	<b>0,9999</b>	<b>0,00000003</b>	<b>0,5174</b>
GG	<b>1</b>	<b>0</b>	<b>2</b>	<b>2</b>	<b>0,9993</b>	<b>0,000002</b>	<b>0,1140</b>
	2	0	2	3	0,9990	0,000003	0,2475
	3	0	3	2	0,9986	0,000005	0,2175
	4	0	3	3	0,9990	0,000003	0,3727
	5	0	3	4	0,9980	0,000007	0,8823

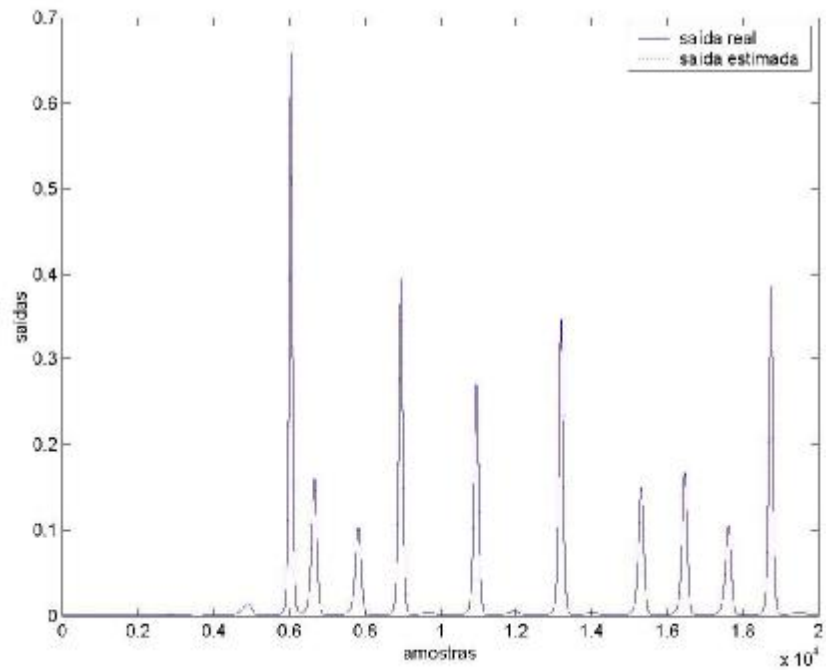


Figura 7.34. Saída do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 2).

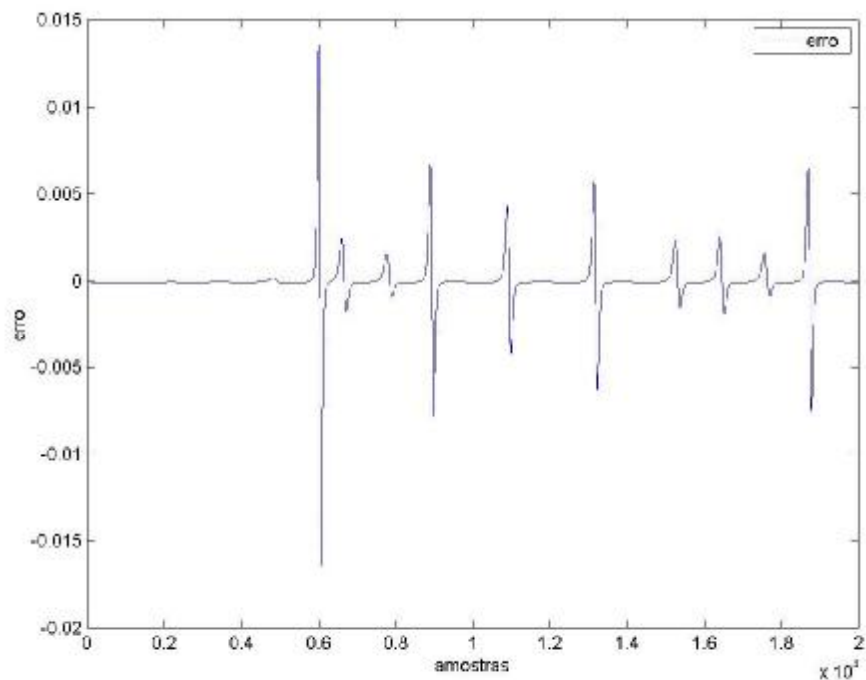


Figura 7.35. Erro do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 2).

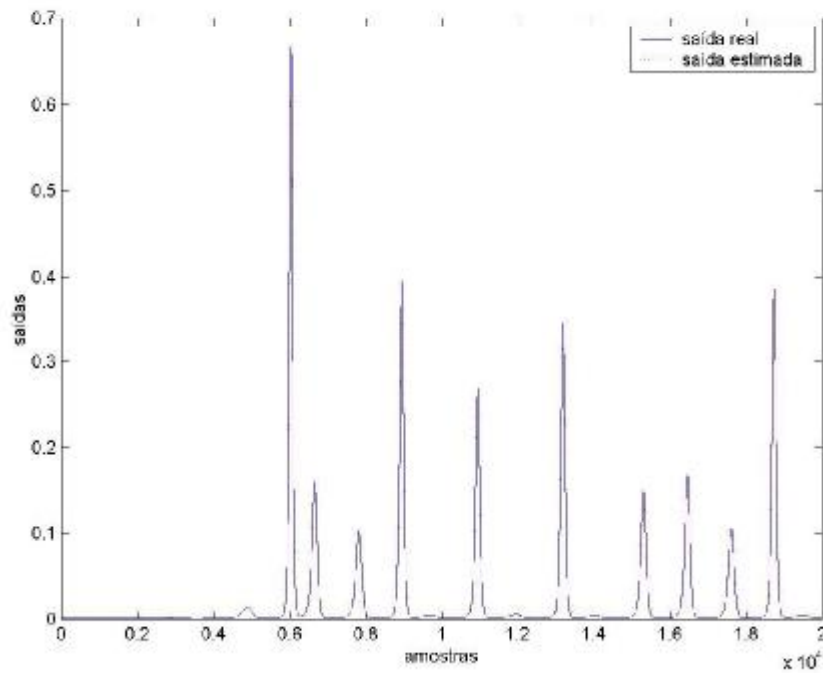


Figura 7.36. Saída do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 2).

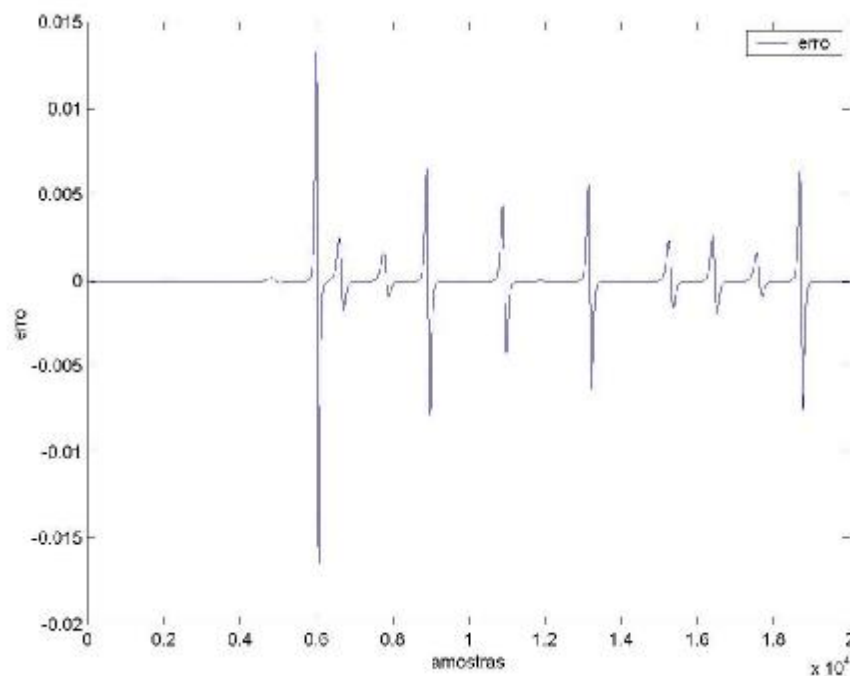


Figura 7.37. Erro do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 2).

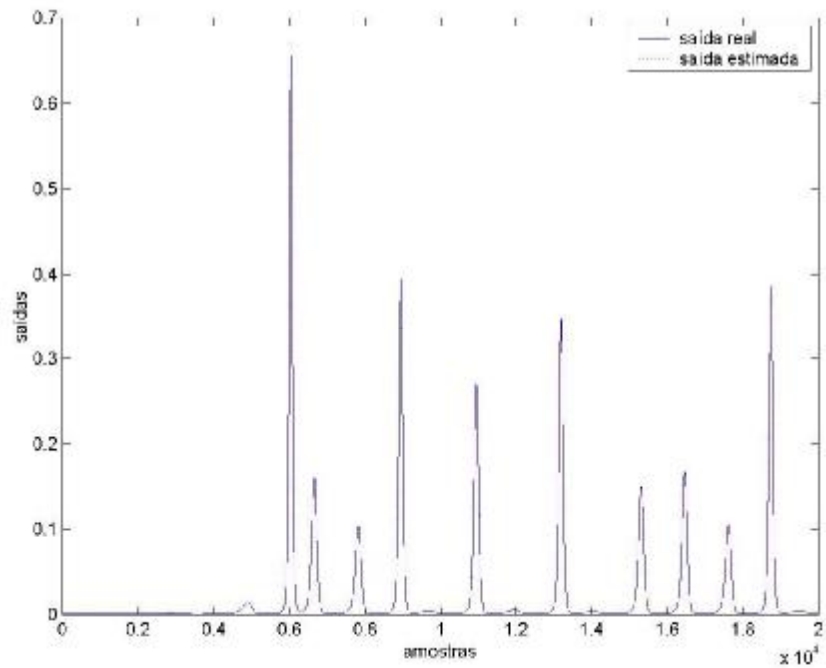


Figura 7.38. Saída do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5).

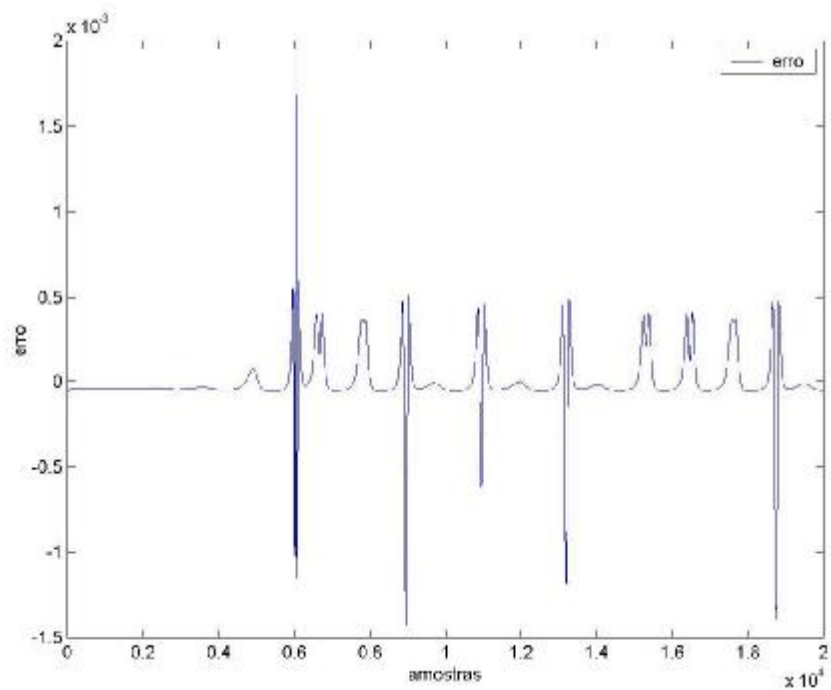


Figura 7.39. Erro do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento GK e otimização pela pseudo-inversa (simulação 5).

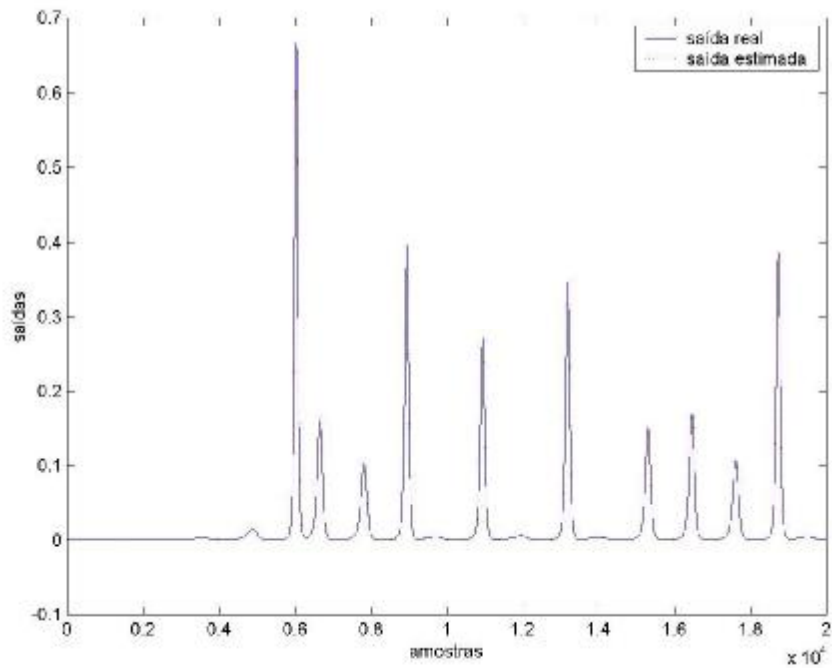


Figura 7.40. Saída do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 1).

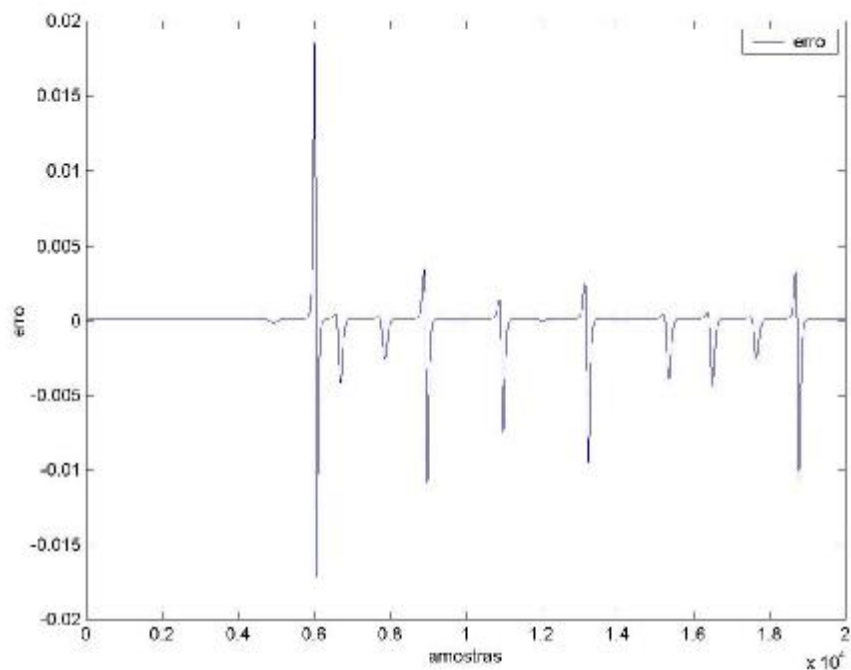


Figura 7.41. Erro do melhor resultado para identificação da coordenada  $z$  do sistema de Rössler com agrupamento GG e otimização pela pseudo-inversa (simulação 1).



### 7.3.3.1 Resultados para a coordenada z

Em relação aos resultados da fase de estimação mostrados na tabela 7.9 observa-se que o método de GK com  $N_u = 0$  e  $N_y = 3$  usando 4 grupos (*clusters*) na camada oculta da rede neural de base radial obteve o melhor resultado. E também obteve a maior complexidade computacional levando em consideração o número de operações algébricas.

Em relação a tabela 7.10 (fase de validação), observa-se o excelente desempenho de todos os algoritmos de agrupamento pois o mapeamento da rede ficou com um  $R^2_{val} \geq 0,9993$ .

## 7.4 Resultados para a Fornalha de Box e Jenkins

Nas tabelas 7.11 e 7.12 são apresentados os resultados das simulações na fase de estimação e validação, respectivamente, para o caso da fornalha de Box e Jenkins.

Tabela 7.11: Resultados das simulações na fase de estimação para a fornalha de Box-Jenkins.

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{est}$	$MSE_{est}$
<i>k</i> -médias	1	2	1	30	0,9836	0,1641
	2	2	1	35	0,9945	0,0552
	3	2	1	40	0,9986	0,0140
	<b>4</b>	<b>2</b>	<b>2</b>	<b>45</b>	<b>0,9995</b>	<b>0,0052</b>
	5	2	2	45	0,9719	0,2801
<i>c</i> -médias	1	2	1	25	0,9700	0,3007
	2	2	1	30	0,9892	0,1086
	3	2	1	35	0,9906	0,0942
	<b>4</b>	<b>2</b>	<b>2</b>	<b>40</b>	<b>0,9960</b>	<b>0,0406</b>
	5	2	2	50	0,9269	0,7286
GK	1	2	1	25	0,9131	0,8725
	2	2	1	30	0,9816	0,1849
	3	2	1	35	0,9854	0,1463
	<b>4</b>	<b>2</b>	<b>1</b>	<b>40</b>	<b>0,9957</b>	<b>0,0433</b>
	5	2	2	40	0,6181	3,8049
GG	1	2	1	25	0,9274	0,7285
	2	2	1	30	0,9787	0,2140
	3	2	1	35	0,9933	0,0673
	<b>4</b>	<b>2</b>	<b>1</b>	<b>40</b>	<b>0,9987</b>	<b>0,0135</b>
	5	2	2	40	0,8898	1,0977

Tabela 7.12: Resultados das simulações na fase de validação para a fornalha de Box-Jenkins.

Algoritmo de agrupamento	simulações	$N_u$	$N_y$	nº de grupos	$R^2_{val}$	$MSE_{val}$	nº de op. algébricas
<i>k</i> -médias	1	2	1	30	0,9802	0,2020	0,0120
	2	2	1	35	0,9906	0,0965	0,0157
	3	2	1	40	0,9965	0,0353	0,0193
	<b>4</b>	<b>2</b>	<b>2</b>	<b>45</b>	<b>0,9989</b>	<b>0,0109</b>	<b>0,0234</b>
	5	2	2	45	0,9438	0,5712	0,0246
<i>c</i> -médias	1	2	1	25	0,9536	0,4741	0,0705
	2	2	1	30	0,9871	0,1321	0,0842
	3	2	1	35	0,9852	0,1514	0,1465
	<b>4</b>	<b>2</b>	<b>2</b>	<b>40</b>	<b>0,9928</b>	<b>0,0740</b>	<b>0,1280</b>
	5	2	2	50	0,8722	1,2984	0,1157
GK	1	2	1	25	0,8961	1,0612	0,9609
	2	2	1	30	0,9724	0,2818	1
	3	2	1	35	0,9816	0,1881	0,9189
	<b>4</b>	<b>2</b>	<b>1</b>	<b>40</b>	<b>0,9917</b>	<b>0,0748</b>	<b>0,3684</b>
	5	2	2	40	0,5162	4,9156	0,3826
GG	1	2	1	25	0,9109	0,9102	0,1643
	2	2	1	30	0,9730	0,2754	0,1606
	3	2	1	35	0,9912	0,0898	0,2657
	<b>4</b>	<b>2</b>	<b>1</b>	<b>40</b>	<b>0,9968</b>	<b>0,0330</b>	<b>0,3284</b>
	5	2	2	40	0,7722	2,3147	0,4007

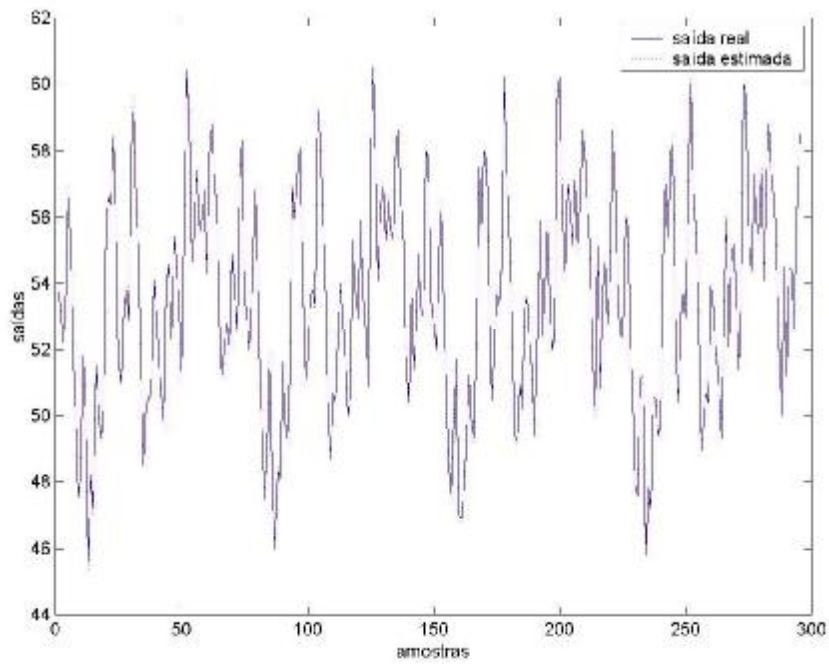


Figura 7.42. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 4).

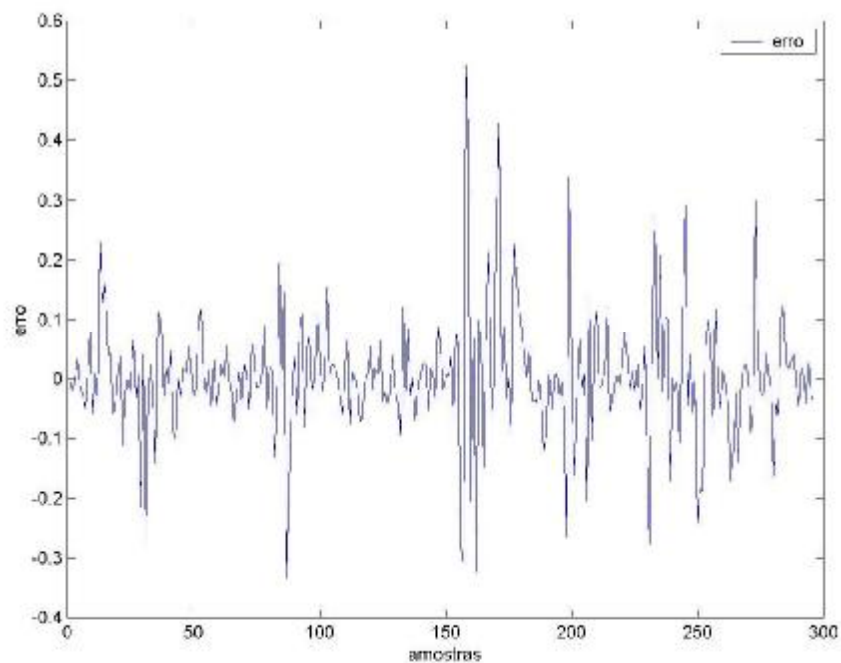


Figura 7.43. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento  $k$ -médias e otimização pela pseudo-inversa (simulação 4).

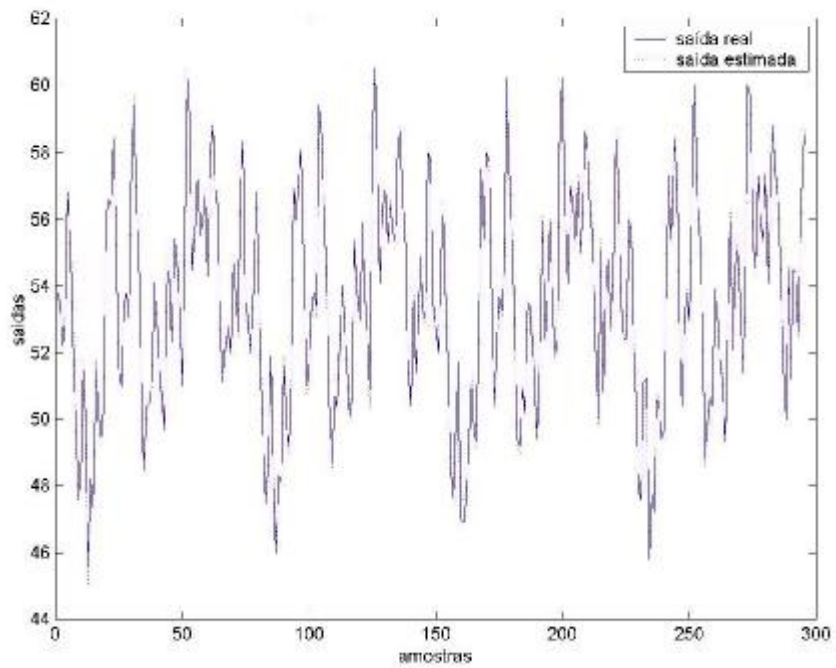


Figura 7.44. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 4).

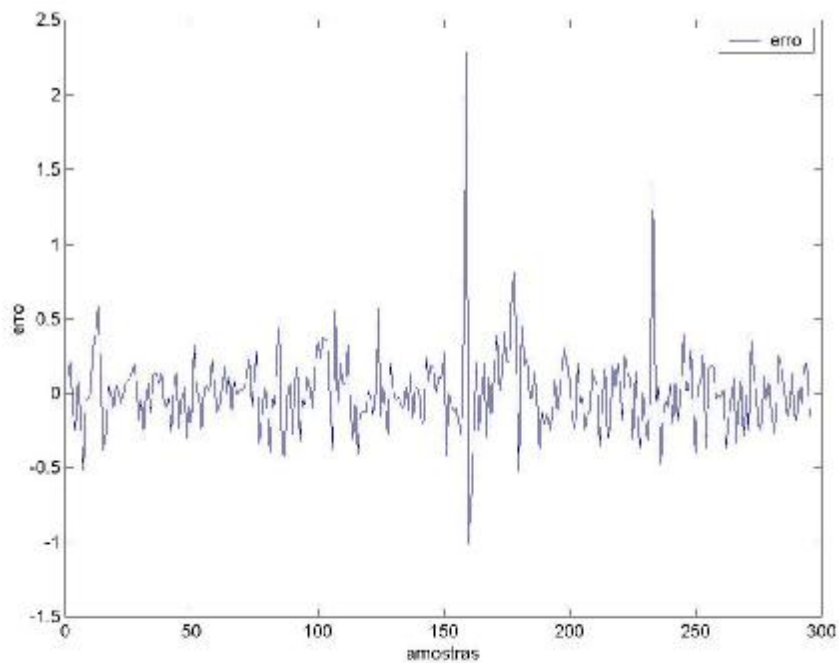


Figura 7.45. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento  $c$ -médias e otimização pela pseudo-inversa (simulação 4).

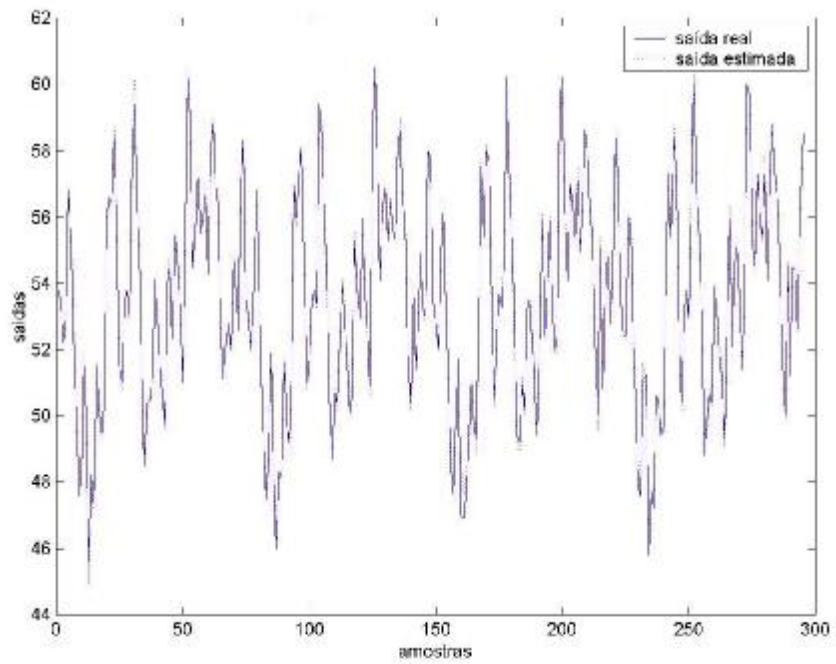


Figura 7.46. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GK e otimização pela pseudo-inversa (simulação 4).

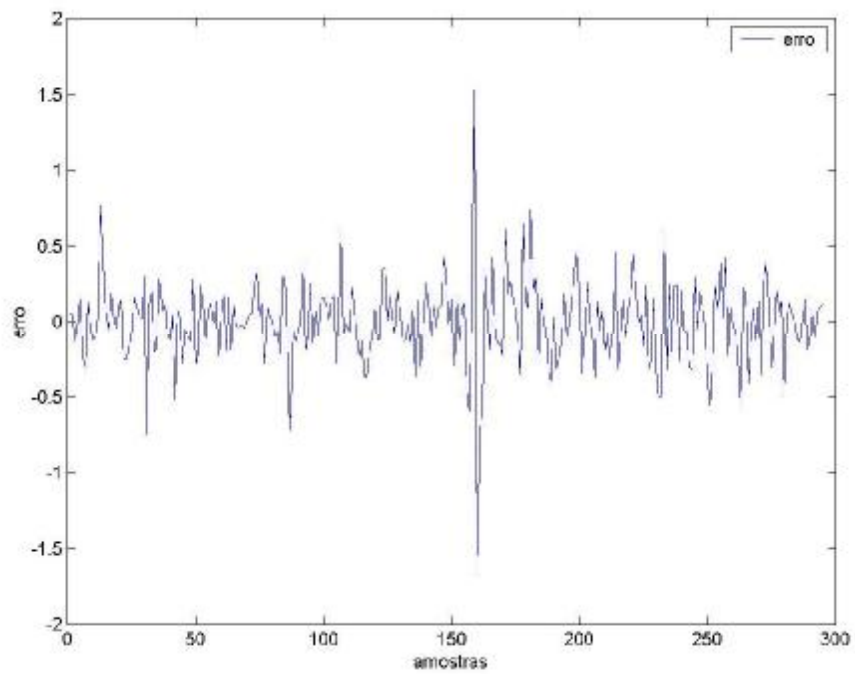


Figura 7.47. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GK e otimização pela pseudo-inversa (simulação 4).

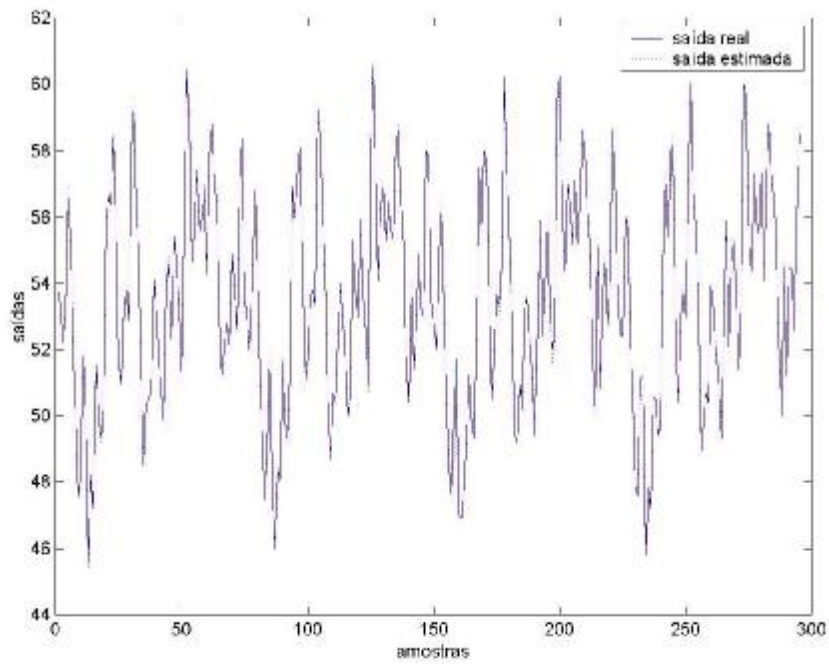


Figura 7.48. Saída do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GG e otimização pela pseudo-inversa (simulação 4).

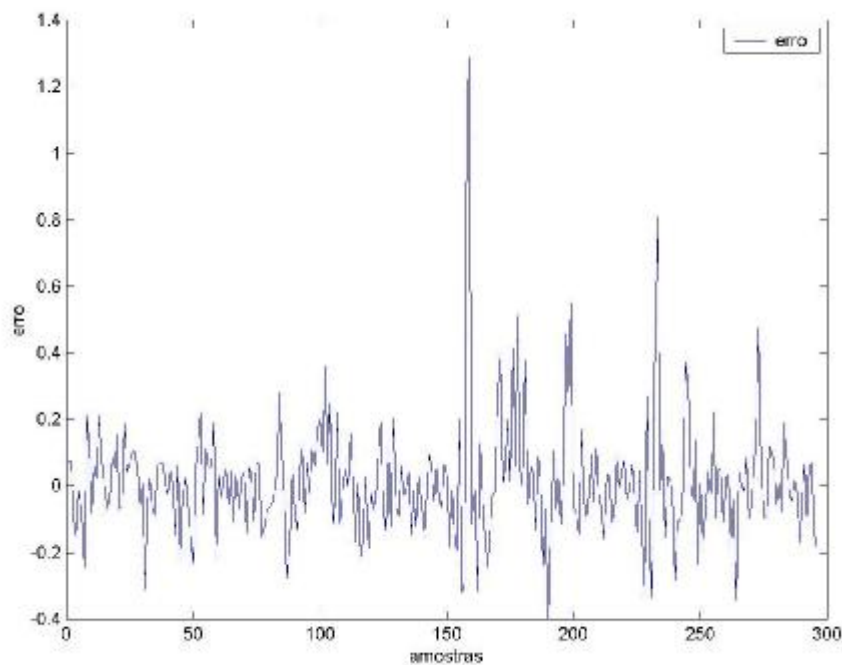


Figura 7.49. Erro do melhor resultado para identificação da fornalha de Box e Jenkins com agrupamento GG e otimização pela pseudo-inversa (simulação 4).

#### 7.4.1 Comentários dos resultados para a fornalha de Box-Jenkins

Nota-se em relação aos resultados mostrados na tabela 7.11 (fase de estimação), que os métodos de agrupamentos GK e GG não obtiveram bons resultados com  $N_u = N_y = 2$  e 40 grupos, mas no restante dos resultados, os métodos foram bem sucedidos no projeto da rede neural de base radial.

Em relação a tabela 7.12 (fase de validação), observa-se ainda que com  $N_u = N_y = 2$  e 40 grupos os métodos GK e GG não obtiveram bons resultados e também realizaram um maior número de operações algébricas se comparados ao  $k$ -médias e  $c$ -médias.

O algoritmo de agrupamento  $k$ -médias obteve os melhores resultados tanto na fase de estimação quanto na validação e obteve bons resultados com a variação do número de grupos. Diferente dos outros métodos que na fase de validação com número de grupos maiores ou iguais a 40 e  $N_u = N_y = 2$  não obtiveram bons resultados.



## 7.5 Comentários Gerais

Alguns detalhes devem ser comentados sobre cada algoritmo de agrupamento. Iniciando pelo  $k$ -médias, uma vantagem deste algoritmo é que este faz com que cada dado do conjunto pertença a um grupo. Outra vantagem é que a localização inicial dos centros pode variar o que permite estabelecer outras condições iniciais para que o algoritmo possa melhorar seu desempenho. Sobre a primeira vantagem comentada, esta pode se tornar uma desvantagem, pois mesmo o dado não estando tão próximo ao centro ele pertencerá a este, isto faz com o que o algoritmo decresça em sua eficiência. Outra desvantagem, sendo esta a principal, é que antes do algoritmo ser iniciado o número de grupos deve ser escolhido.

O algoritmo  $c$ -médias é sensível às escolhas iniciais dos grupos, precisando de uma inicialização adequada para assegurar a obtenção de resultados corretos. A utilização de algoritmos para realizar essas escolhas pode resolver o problema, inicializando com centros bem separados e próximos dos centros finais assegurando assim uma correta convergência e redução do número de iterações necessárias.

Um detalhe importante do algoritmo GK é que este insere uma solução para permitir que os grupos possam ser alongados (elipsóides), introduzindo uma métrica corrigida por uma matriz definida positiva, com as covariâncias entre as dimensões do espaço de atributos. Isto pelo fato de que a distância euclidiana empregada constrói apenas grupos hiperesféricos, e a correção permite a detecção de grupos hiperelipsoidais. Esta métrica utiliza a distância de Mahalanobis. Já para o algoritmo GG, sua formulação pode ser considerada uma extensão do GK onde a distância ao centro é relacionada a uma distribuição de probabilidades e sua função de pertinência decresce mais lentamente a partir dos respectivos centros, de modo que suas funções vizinhas se sobrepõem em demasia, e o algoritmo fica mais propenso a convergir indesejavelmente a um mínimo local.

Uma análise diretamente dos resultados também pode ser feita. O primeiro detalhe a ser comentado sobre os resultados encontrados é que, na maioria dos casos, quando é inserido um número maior de saídas atrasadas nas entradas, a identificação melhora. O motivo para esta melhora é que a rede obtém mais informações sobre a dinâmica não-linear do processo assim melhorando sua resposta. Por esta razão que o número de entradas atrasadas ( $N_u$ ) e

saídas atrasadas ( $N_y$ ) afetam os resultados. Mas um cuidado especial deve ser tomado, pois quanto maior o número de  $N_u$  e  $N_y$ , maior é a complexidade do modelo.

Para se obter a melhor resposta da identificação, foram apresentados os índices  $R^2$  e  $MSE$ , tanto na fase de estimação quanto na fase de validação, e a melhor resposta foi retirada do maior  $R^2$ . Esta maneira pode não ser considerada o melhor método de se descobrir o melhor resultado, pois pode ser levado em consideração a complexidade do modelo encontrado. Para a análise da complexidade do modelo, vários pontos podem ser abordados, que são os números  $N_u$  e  $N_y$ , que já foram comentados anteriormente, e também o número de grupos utilizados. Por exemplo, um modelo com número de grupos igual a 20 com a média do  $R^2$  igual a 0,9999, é melhor que um modelo com número de grupos igual a 15 com a média do  $R^2$  igual a 0,9500. A resposta a esta questão depende exclusivamente do modelador, como foi descrito na teoria de identificação. É ele que dirá se a complexidade no modelo é um ponto relevante para a sua aplicação. Nesta dissertação apenas foi considerado o modelo com maior média de  $R^2$  sem um foco na complexidade do modelo encontrado.

Outro aspecto interessante é a dificuldade em analisar qual dos algoritmos de agrupamento obteve um melhor resultado, pois todos obtiveram resultados ótimos e muito próximos. Pôde-se observar que os pares de valores de  $R^2_{est}$  e  $R^2_{val}$ ,  $MSE_{est}$  e  $MSE_{val}$ , sempre foram diferentes. Isto se deve pela diferença entre a fase de estimação e de validação. A primeira fase é a de estimação, onde todos os parâmetros necessários são obtidos. E na fase de validação, os dados que são passados para a rede são todos desconhecidos, podendo assim, testar a generalização da rede. Nesta diferenciação das etapas de estimação e validação, às vezes esta não se encaixa para sistemas caóticos. Sobre estes sistemas, pode-se observar que o  $R^2_{val}$  pode ser maior que o  $R^2_{est}$  devido à complexidade deste tipo de sistema.

Os resultados obtidos e a maneira de como foram obtidos dependem exclusivamente do modelador, pois não existem métodos que demonstrem qual o número de  $N_u$  e  $N_y$  deve ser utilizado, qual o número razoável de grupos e também que grau de complexidade é aceitável. Tudo depende da experiência do modelador, e de como ele deseja utilizar o modelo encontrado.

# Capítulo 8

## 8 Conclusão e Futura Pesquisa

Esta dissertação teve por objetivo apresentar um estudo comparativo de desempenho de métodos de agrupamento. Para a RN-RBF foram contextualizados e testados os seguintes métodos: (i)  $k$ -médias, (ii)  $c$ -médias nebuloso, (iii) Gustafson-Kessel e (iv) Gath-Gheva, para o treinamento de redes neurais de base radial. Estas redes neurais são então aplicadas em uma configuração série-paralelo com intuito de identificação de sistemas não-lineares. Os sistemas não-lineares validados foram os seguintes: (i) um trocador de calor, (ii) sistema de Mackey-Glass, (iii) sistema caótico de Rössler e (iv) fornalha de gás de Box-Jenkins.

Os objetivos de estudar e analisar os resultados de simulação em identificação, tanto da fase de estimação quanto validação dos modelos da rede neural de base radial, foram promissores para aplicações futuras na área de identificação, controle e previsão de séries temporais, pois foram atingidos com sucesso.

Para futura pesquisa, a utilização de outros algoritmos de otimização e também algoritmos para otimizar os centros encontrados pelos algoritmos de agrupamento é uma área promissora. Uma outra extensão interessante desta pesquisa seria, não só a identificação um-passo-à-frente mas também multi-passos-à-frente conforme publicado em [SIE05]. E ainda mais, para complementar, na parte de identificação realizar também para sistemas multivariáveis.

A contribuição desta dissertação foi a de validar em vários estudos de casos métodos de agrupamento para o treinamento de redes neurais que muitas vezes, como o caso do GG e GK são geralmente utilizados em sistemas nebulosos e não em projeto de redes neurais. Neste contexto, esta pesquisa gerou a publicação de artigos científicos em eventos nacionais e internacionais, a citar as seguintes publicações: [GUE05a], [GUE05b], [GUE04a], [GUE04b] [GUE04c], [GUE04d] e [SIE05].

# Referências Bibliográficas

- [AGU04] AGUIRRE, L. A. *Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Ed. UFMG, Belo Horizonte, 2º Ed., 2004.
- [AHM00] AHMAD, S.M. & SUSTON, R. *Linear, Nonlinear Modeling and AUV Retrieval*. Technical Report, Dept. of Mechanical and Marine Engineering, University of Plymouth, United Kingdom, 2000.
- [AST90] ASTRÖM, K. J. & WITTENMARK, B. *Computer Controlled Systems - Theory and Design*. Prentice-Hall, 2º Ed., 1990.
- [BAB02] BABUSKA, R.; VAN DER VEEN P. J. & KAYMAK, U. *Improved covariance estimation for Gustafson-Kessel clustering*. IEEE International Conference on Fuzzy Systems, 2002, p. 1081-1085.
- [BEZ75] BEZDEK, J. C. & DUNN, J. C. *Optimal fuzzy partitions: a heuristic for estimating the parameters in a mixture of normal distributions*. IEEE Transactions on Computers, 1975, p. 835-838.
- [BEZ81] BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Ed. Plenum Press, New York, USA, 1981.
- [BIL80] BILLINGS, S. A. *Identification of Nonlinear Systems - A Survey*. IEE Proceedings, Control Theory and Applications, Part D, vol. 127, no. 6, 1980, p. 272-285.
- [BIL91] BILLINGS, S. A. & TAO, Q. H. *Model Validity Tests for Nonlinear Signal Processing Applications*. International Journal Control, vol. 54, no. 1, 1991, p. 157-199.

- [BIL95] BILINGS, S. A. & ZHENG, G. L. *Radial Basis Function Network Configuration Using Genetic Algorithms*. Pergamon, Neural Networks, vol. 8, no. 6, 1995, p. 877-890.
- [BIS95] BISHOP, C. M. *Neural Networks for Pattern Recognition*. Ed. Clarendon Press-Oxford, New York, USA, 1995.
- [BOR01] BORS, A. G. *Introduction of the Radial Basis Function (RBF) Networks*. Online Symposium for Electronics Engineers, DSP Algorithms: Multimedia, <http://www.osee.net/>, 13/05/2004, vol. 1, 2001, p. 1-7.
- [BOX94] BOX, G.; JENKINS, G. & REINSEL, G. *Time Series Analysis Forecasting and Control*. Ed. Prentice Hall, 3<sup>rd</sup> ed., Englewood Cliffs, New Jersey, USA, 1994.
- [BRA00] BRAGA, A. P.; CARVALHO, A. C. P. L. F. & LUDEMIR, T. B. *Redes Neurais Artificiais, Teoria e Aplicações*. Rio de Janeiro, Ed. LTC, 2000.
- [CAS99] CASSINI, C. C. S. & AGUIRRE L. A. *Uma Introdução à Identificação de Sistemas Não-Lineares. Laboratório de Modelagem, Análise e Controle de Sistemas Não-Lineares*. Relatório Técnico, Dept. de Eng. Elétrica, UFMG, 1999.
- [CAS01] CASTRO, M. C. F. *Predição Não-Linear de Séries Temporais Usando Redes Neurais RBF por Decomposição em Componentes Principais*. Tese de Doutorado, UNICAMP, Dept. de Engenharia Elétrica, São Paulo, 2001.
- [CHE92] CHEN, H. & LIU, R.-W. *Adaptive Distributed Orthogonalization Processing for Principal Components Analysis*. International Conference on Acoustics, Speech and Signal Processing, San Francisco, CA, USA, vol. 2, 1992, p. 293-296.
- [COE00] COELHO, L. S. *Identificação e Controle de Processos Multivariáveis via Metodologias Avançadas e Inteligência Computacional*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica, UFSC, Florianópolis, 2000.

- [COE04] COELHO, A. A. R. & COELHO, L. S. *Identificação de Sistema Dinâmicos Lineares*. Ed. UFSC, Florianópolis, SC, 2004.
- [COH02] COHEN, S. & INTRATOR, N. *A Hybrid Projection-based and Radial Basis Function Architecture: Initial Values and Global Optimisation*. *Pattern Analysis & Applications*, vol. 5, 2002, p.113-120.
- [COR01] CORRÊA, M. V. *Identificação Caixa-Cinza de Sistemas Não-Lineares Utilizando Representações NARMAX Racionais e Polinomiais*. Tese de Doutorado, Dept. de Eng. Elétrica, UFMG, 2001.
- [COV65] COVER, T. M. *Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition*. *IEEE Transactions on Electronic Computers*, vol. EC-14, 1965, p. 326-334.
- [DUD73a] DUDA, R. O. & HART, P. E. *Pattern Classification and Scene Analysis*. Ed. Willey, New York, USA, 1973.
- [DUN73b] DUNN, J. C. *A Fuzzy Relative of the ISODATA Process and it's Use in Detecting Compact Well-Separated Clusters*. *Journal of Cybernetics*, vol. 3, 1973, p. 32-57.
- [DUD04] DUDUL. S. V. *Prediction of a Lorenz chaotic Attractor using Two-layer Perceptron Neural Network*. *Applied Soft Computing*, 2004.
- [EVE80] EVERITT, B. *Cluster Analysis*. New York, USA: Academic Press, 1980.
- [FOL04] FOLLAND, R.; HINES, E.; DUTTA, R., BOILOT, P. & MORGAN, D. *Comparison of neural network predictors in the classification of tracheal-bronchial breath sounds by respiratory auscultation*. *Artificial Intelligence in Medicine*, vol. 31, no. 3, July 2004, p. 211-220.

- [GAT89] GATH, I. & GEVA, A.B. *Unsupervised optimal fuzzy clustering*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 7, 1989, p. 773-781.
- [GOR00] GORP, J. V. *Nonlinear Identification with Neural Network and Fuzzy Logic*. PhD thesis, Vrije Universiteit Brussel, Faculteit Toegepaste Wetenschappen, Dienst Algemene Elektriciteit & Instrumentatie (ELEC), Brussel, Belgium, 2000.
- [GRE89] GREBLICK, W. & PAWLAK, M. *Recursive Nonparametric Identification of Hammerstein Systems*. Journal of the Franklin Institute, vol. 326, no. 4, 1989, p. 461-481.
- [GUE05a] GUERRA, F.; SIERAKOWSKI, C. A. & COELHO, L. S. *Rede neural função de base radial usando otimização por aproximação estocástica em identificação de sistemas não-linear com comportamento caótico*. VII Simpósio Brasileiro de Automação Inteligente, São Luís, MA, vol. 1, 2005, p. 1-5.
- [GUE05b] GUERRA, F. & COELHO, L. S. *Radial basis neural network learning based on particle swarm optimization to multistep prediction of chaotic Lorenz's system*. 5th International Conference on Hybrid Intelligent Systems, Rio de Janeiro, RJ, 2005. 5th International Conference on Hybrid Intelligent Systems, Los Alamitos, CA, USA, IEEE Press, 2005, vol. 1, p. 521-524.
- [GUE04a] GUERRA, F. & COELHO, L. S. *Identificação multivariável de um sistema caótico de Rössler usando sistema nebuloso de Takagi-Sugeno*. XV Congresso Brasileiro de Automática, Gramado, RS, 2004, vol. 1, p. 1-6.
- [GUE04b] GUERRA, F. & COELHO, L. S. *Neural network conception based on genetic algorithm optimization applied to multivariable system identification: implementation and practical case study*. VI INDUSCON (VI Conferência Internacional de Aplicações Industriais), Joinville, SC, 2004, vol. 1, p. 1-6.

- [GUE04c] GUERRA, F. & COELHO, L. S. *Identification of Rössler's System using a Radial Basis Function Neural Network*. VI INDUSCON (VI Conferência Internacional de Aplicações Industriais), Joinville, SC, 2004, vol. 1.
- [GUE04d] GUERRA, F. & COELHO, L. S. *An approach of neural network for the Chua's chaotic system identification*. VI INDUSCON (VI Conferência Internacional de Aplicações Industriais), Joinville, SC, 2004, vol. 1.
- [GUS79] GUSTAFSON, D. E. & KESSEL, W. C. *Fuzzy clustering with fuzzy covariance matrix*. Proceedings of the IEEE Control and Decision Conference, San Diego, 1979, p. 761-766.
- [HAG96] HAGAN, M. T.; DEMUTH H. B. & BEALE, M. *Neural Network Design*. Ed. PWS Publishing Company, Ohio, USA, 1996.
- [HAR71] HARDY, R. L. *Multiquadric equations of topography and other irregular surfaces*. Journal of Geophysics Research, vol. 76, 1971, p. 1905-1915.
- [HAY01] HAYKIN, S. *Redes Neurais: Princípios e Prática*. 2º Edição, Ed. Bookman, Porto Alegre, RS, 2001.
- [HER03] HERNANDEZ, E. D. M. *Inteligência Computacional e Redes Neurais em Engenharia Elétrica*. PSI2222 Práticas de Eletricidade e Eletrônica II, Seminário 6, 2003.
- [JAN97] JANG, J. S. R.; SUN, C. T. & MIZUTANI, E. *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence*. Ed. Prentice Hall, New Jersey, USA, 1997.
- [JAN05] JAN, T. & KIM, M. *Vector quantized radial basis function neural network with embedded multiple local linear models for financial prediction*. Neural Networks, IJCNN '05, IEEE International Joint Conference, vol. 4, Aug. 2005, p.2538- 2543.



- [JAN06] JANG, J. S. R. *Data Sets of Working Group on Data Modeling Benchmarks*. IEEE Neural Networks Council Standards Committee, Taiwan, URL: <http://neural.cs.nthu.edu.tw/jang/benchmark>, 13/01/2006.
- [JOH93] JOHANSSON, R. *System Modeling and Identification*. Ed. Prentice Hall, Upper Saddle River, NJ, USA, 1993.
- [KHA90] KHANNA, T. *Foundation of Neural Networks*. Ed. Addison-Wesley Publishing Company, Princeton, USA, 1990.
- [KLI98] KLIMASAUSKAS, C. C. *Hybrid Modeling for Robust Nonlinear Multivariable Control*. ISA Transactions, vol. 37, 1998, p. 291-297.
- [KOD02] KODOGIANNIS, V. & LOLIS, A. *Forecasting Financial Time Series using Neural Network and Fuzzy System-based Techniques*. Neural Computing & Applications, vol. 11, 2002, p. 90-102.
- [KRO96] KRÖSE. B. & SMAGHT, P. D. V. *An Introduction to Neural Networks*. Eighth edition, University of Amsterdam, Amsterdam, 1996.
- [LJU01] LJUNG, L. *Black-box Models from Input-output Measurements*. IEEE Instrumentation and Measurement, Technology Conference, Budapest, Hungary, May, 2001, p. 21-23.
- [LJU87] LJUNG, L. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, NJ, 1987.
- [LJU94] LJUNG, L. & GLAD, T. *Modeling of Dynamic Systems*. Prentice-Hall, Upper Saddle River, NJ, 1994.

- [LJU96] LJUNG, L. *Development of System Identification*. 13<sup>th</sup> IFAC World Congress, San Francisco, CA, USA, 1996, p. 141-146.
- [LOW89] LOWE, D. *Adaptive Radial Basis Functions Nonlinearities and the Problem of Generalization*. 1<sup>st</sup> IEE International Conference on Artificial Neural Network, London, UK, 1989, p. 171-175.
- [LOW91] LOWE, D. *What have Neural nNetwork to Offer Statistical Pattern Processing?* Proceedings of the SPIE Conference on Adaptive Signal Processing, San Diego, USA, 1991, p. 460-471.
- [LU92] LU, Y. - Z. *The New Generation of Advanced Process Control*. Control Engineering International, Mid-March, 1992, p. 21-23.
- [MAC77] MACKEY, M. C. & GLASS, L. *Oscillation and Chaos in Physiological Control Systems*. Science, vol. 197, 1977, p. 287-289.
- [MAS04] MASHOR, M.Y. *Performance Comparison between HMLP, MLP and RBF Networks with Application to On-line System Identification*. Proceedings IEEE, Conference on Cybernetics and Intelligent Systems, Singapore, 2004.
- [MIC86] MICCHELLI, C. A. *Interpolation of scattered data: Distance matrices and conditionally positive definite functions*. Constructive Approximation, vol. 2, 1986, p. 11-22.
- [MOO04a] Moor, B. L. R. D. *DaISy: Database for the Identification of Systems*. Department of Electrical Engineering, ESAT/SISTA, K.U.Leuven, Belgium, URL: <http://www.esat.kuleuven.ac.be/sista/daisy/>, 01/12/2004. [Used dataset: exchanger.dat, Process Industry Systems, 97-002.].

- [NAR90] NARENDRA, K. S. & PARTHASARATHY, K. *Identification and Control of Dynamical Systems using Neural Networks*. IEEE Transactions on Neural Networks, vol. 1, no. 1, march, 1990.
- [NEP02] NEPOMUCENO, E. G. *Identificação Multiobjetivo de Sistemas Não-Lineares*. Dissertação de Mestrado, Dept. de Eng. Elétrica, UFMG, 2002.
- [NOR04] NORVING, P. & RUSSEL, S. *Intelligence Artificial*, 2º Edição, Ed. Campus, 2004.
- [OGA93] OGATA, K. *Engenharia de Controle Moderno*. Prentice-Hall do Brasil, 2º Ed, 1993.
- [PAN06] PANIGRAHI, S.; BALASUBRAMANIAN, S.; GU, H.; LOGUE, C. & MARCHELLO, M. *Neural-network-integrated electronic nose system for identification of spoiled beef*. LWT - Food Science and Technology, vol. 39, no. 2, 2006, p. 135-145.
- [PEN03] PENG, H.; OZAKI, T.; OZAKI, V. H. & TOYODA, Y. *A Parameter Optimization Method for Radial Basis Function Type Models*. IEEE Transactions on Neural Networks, vol. 14, no. 2, 2003.
- [REZ03] REZENDE, S. O. *Sistemas Inteligentes: Fundamentos e Aplicações*. Ed. Manole, São Paulo, SP, 2003.
- [ROS76] Rössler, O. E. *An Equation for Continuous Chaos*. Physical Letters, vol. 35A, 1976, p. 397-398.
- [SAC04] SACCHI, R.; CARNEIRO, A. A. F. M. & ARAÚJO, A. F. R. *A RBF Network Trained by the SONARX Model and Applied to Obtain the Operation Policies of the Hydropower Systems*. 8th SBRN, São Luis, MA, 2004.
- [SCH97a] SCHALKOFF, R. J. *Artificial Neural Networks*. Ed. The McGraw-Hill Companies, Clemson, USA, 1997.

- [SCH97b] SCHAIBLE, B.; XIE, H. & LEE Y. C. *Fuzzy Logic Models for Raking Process Effects*. IEEE Transactions on Fuzzy Systems, vol. 5, no. 4, 1997, p. 545-556.
- [SHI04] SHIGUEMORI, E. H.; SILVA, J. D. S.; VELHO, H. F. C. & CARVALHO, J. C. *Radial Basis Function Neural Network in the Inversion of Temperature Vertical Profiles from Satellite Radiance Data*. 8th SBRN, São Luis, MA, 2004.
- [SIE05] SIERAKOWSKI, C. A.; GUERRA, F. & COELHO, L. S. *Particle swarm optimization approach for multi-step-ahead prediction using radial basis function neural network*. 15TH IFAC WORLD CONGRESS, Praga, 2005, p. 1-6.
- [SJO95a] SJOBERG, J. *Non-Linear System Identification with Neural Networks*. PhD thesis, Dept. of Electrical Engineering, Linkoping University, Sweden, 1995.
- [SJO95b] SJÖBERG, J.; ZHANG Q.; LJUNG, L.; BENVENISTE, A.; DEYLON, B.; GLORENNEC, P.-Y.; HJALMARSSON, H. & JUDITSKY, A. *Nonlinear Black-Box Modeling in System Identification: a Unified Overview*. Technical Report, June, 1995.
- [TAN95] TAN, S.; HAO, J. & VANDEWALLE, J. *Efficient Identification of RBF Neural Net Models for Nonlinear Discrete-time Multivariable Dynamical Systems*. Neurocomputing, vol. 9, 1995, p. 11-26.
- [VEE95] VEELANTURF, L. P. J. *Analysis and Applications of Artificial Neural Networks*. Ed. Prentice Hall, Hertfordshire, USA, 1995.
- [WIE58] WIENER, T. *Nonlinear problems in random theory*. Ed. The Technology Press. M.I.T., New York, 1958.
- [YAN00] YAN, L.; SUNDARARAJAN, N. & SARATCHANDRAN, P. *Nonlinear System Identification Using Lyapunov Based Fully Tuned Dynamic RBF Networks*. Neural Processing Letters, vol. 12, 2000, p. 291-303.

- [YAT05] YATES, J.W.T.; GARDNER, J.W.; CHAPPELL, M.J. & DOW, C.S. *Identification of bacterial pathogens using quadrupole mass spectrometer data and radial basis function neural networks*. Science, Measurement and Technology, IEE Proceedings, vol. 152, no. 3, May 2005, p.97- 102.
- [YIN97] YINGWEI, L.; SUNDARARAJAN, N. & SARATCHANDRAN, P. *Identification of Time-varying Nonlinear Systems using Minimal Radial Basis Function Neural Networks*. IEE Proc-Control Theory Appl., vol. 144, no. 2, 1997.
- [YU00] YU, D. L.; GOMM, J. B. & WILLIAMS, D. *Neural Model Input Selection for a MIMO Chemical Process*. Engineering Applications of Artificial Intelligence, vol. 13, 2000, p. 15-23.
- [YU02] YU, D. L. & GOMM, J. B. *Enhanced Neural Network Modelling for a Real Multivariable Chemical Process*. Neural Computing & Applications, vol. 10, 2002, p. 289-299.
- [ZAD65] ZADEH, L. A. *Fuzzy sets*. Information and Control, vol. 8, 1965, p. 338-353.
- [ZHA96] ZHANG, Y. & RONG X. L. *Hybrid Training of RBF Networks with Application to Nonlinear Systems Identification*. Proceedings of the 35th Conference on Decision and Control, Kobe, Japan, 1996.