

CARLOS ALBERTO CLAUMANN

**DESENVOLVIMENTO E APLICAÇÕES DE REDES NEURAIIS
WAVELETS E DA TEORIA DE REGULARIZAÇÃO NA MODELAGEM
DE PROCESSOS**

Tese submetida ao Programa de Pós-Graduação em Engenharia Química, Departamento de Engenharia Química e Engenharia de Alimentos, Universidade Federal de Santa Catarina, como requisito parcial à obtenção do título de Doutor em Ciências da Engenharia Química.

Orientador: Prof. Nestor Roqueiro, Dr.

Florianópolis, 2003

AGRADECIMENTOS

Aos meus pais, pela paciência, apoio e incentivo.

Aos meus familiares pelo apoio.

Ao professor Nestor Roqueiro pela orientação e principalmente pela amizade.

Ao professor Jáuber Cavalcante de Oliveira pelos seus esclarecimentos e ensinamentos em Análise Funcional.

Aos amigos do Laboratório de Controle de Processos (LCP) pelo auxílio e contribuições, principalmente a Adriano, Mazzuco, Luís Henrique e aos professores Ricardo Antônio Francisco Machado e Ariovaldo Bolzan.

Aos amigos que encontrei na UFSC, em especial a José Luciano e Ivan.

Ao Departamento de Engenharia Química e Alimentos pela oportunidade concedida.

Ao Edivilson pelo apoio, atenção e auxílio.

Aos professores que participaram das bancas examinadoras e que contribuíram com a realização deste trabalho.

Ao CNPQ pelo apoio financeiro.

RESUMO

CLAUMANN, Carlos Alberto. **Desenvolvimento e aplicações de redes neurais wavelets e da teoria de regularização na modelagem de processos**. Florianópolis, 2003. 153p. Tese (Doutorado em Engenharia Química) - Curso de Pós-Graduação em Engenharia Química, Universidade Federal de Santa Catarina.

Orientador: Nestor Roqueiro

Defesa: 28/02/03

Neste trabalho foram propostas modificações na *wavenet* para melhorar a capacidade de generalização e reduzir o número de neurônios. O desempenho da nova rede foi superior quando comparada com as tradicionais redes *feedforward* e de base radial na identificação de processos não-lineares: um atrator caótico, um reator biológico em que ocorre inibição do crescimento de células pela formação de produto e um tanque de *pH*. O desempenho de previsão da nova rede proposta, que é do tipo estática, foi também validado utilizando essa como um simulador de processos não-lineares. Foram efetuados testes sem ruído para um *CSTR (Continuous Stirred-Tank Reactor)* encamisado com múltiplos estados estacionários e um reator cujo comportamento apresenta resposta inversa variável. Alguns testes foram efetuados para o reator biológico e para o tanque de *pH* com a adição de ruído. Foi realizado um estudo a respeito da minimização das curvaturas da superfície de um modelo empírico e efetuada sua conexão com o problema de sobre-treinamento. A nova rede proposta foi utilizada como modelo para o controle de dois processos não-lineares: um tanque de nível com alteração significativa de área (experimental) e um tanque de *pH* (simulado).

Palavras-chave: Wavelets; Teoria da regularização; Redes neurais; Identificação de processos; Controle de processos

ABSTRACT

In this work modifications were proposed in the *wavenet* to enhance its generalization capability and to reduce the number of neurons. The modified net exhibited superior performance when compared with traditional feedforward and radial base nets in the identification of nonlinear process: a caotic atractor, a biological reactor in which inhibition of the growing cells takes place due to product formation and a pH tank. The prediction performance of the static proposed net was also validated using it as a nonlinear process simulator. Tests without noise were realized for a jacketed, multiple steady-state CSTR (Continuous Stirred-Tank Reactor) and a reactor whose behavior presents a variable inverse response. Some tests including noise were accomplished for the biological reactor and a pH tank. A study referring to the surface curvature minimization of an empirical model was elaborated an its connection with the overtraining problem was evaluated. The new proposed net was applied as a model for control purposes in two nonlinear processes: A level tank with variable diameter (experimental) and a pH tank (simulated).

SUMÁRIO

LISTA DE FIGURAS.....	x
LISTA DE TABELAS.....	xiv
1 INTRODUÇÃO	1
2 TEORIA DE <i>WAVELETS</i> E MULTIRESOLUÇÃO.....	4
2.1 Introdução	4
2.2 Famílias de <i>Wavelets</i>.....	6
2.3 Obtenção de <i>Wavelets</i> Mãe	7
2.4 Aproximação de Funções através de Multiresolução.....	11
2.5 Aproximação de Funções Unidimensionais.....	13
2.6 Aproximação de Funções Multivariáveis	17
3 ALTERAÇÕES NA <i>WAVENET</i>	19
3.1 Multiresolução Apenas com Funções de Escala	19
3.2 Determinação de uma Função de Ativação Adequada para Identificação de Processos.....	25
3.3 Determinação do Número Inicial de Funções de Escala (Primeira Resolução).....	28
4 REDES NEURAIIS PARA IDENTIFICAÇÃO DE SISTEMAS.....	31
4.1 Redes Neurais Utilizadas em Identificação de Sistemas	31
5 ESTUDO DE MÉTODOS DE TREINAMENTO DE REDES <i>FEEDFORWARD</i>.....	34
5.1 Descrição do Método de Treinamento da Retroprogação do Erro.....	34
5.2 Descrição do Método de Treinamento Levenberg-Marquardt.....	35
5.3 Comparação de Desempenho entre um Método de Primeira e Segunda Ordem	38
5.4 Metodologia de Treinamento Baseada em Algoritmos Genéticos e Mínimos Quadrados	45
5.4.1 Introdução.....	45

5.4.2	Termos Técnicos Utilizados em Algoritmos Genéticos	46
5.4.3	Diferenças Entre Algoritmos Genéticos e Técnicas Clássicas de Otimização ..	47
5.4.4	Descrição da Técnica Baseada em Mínimos Quadrados e Algoritmos.....	47
5.5	Comparação de Desempenho Entre um Método de Segunda Ordem e um Método Baseado em Algoritmos Genéticos	48
5.6	Conclusões	52
6	REGULARIZAÇÃO E REPRESENTAÇÃO DO CONHECIMENTO	53
6.1	Introdução	53
6.2	Introdução de Regularizadores no Problema de Mínimos Quadrados.....	54
6.3	Normalização de Regularizadores	57
6.4	Influência da Regularização pela Norma dos Pesos no Condicionamento da Matriz de Covariância.....	58
6.5	Influência da Regularização na Capacidade de Generalização de Modelos Empíricos.....	62
6.6	Resultados	67
6.6.1	Identificação do Tanque de <i>pH</i> Utilizando uma Rede de Base Radial.....	67
6.6.2	Identificação do Tanque de <i>pH</i> Utilizando uma Rede de Funções de Escala ..	70
6.6.3	Identificação do Tanque de <i>pH</i> Utilizando uma Rede Neural <i>Feedforward</i>	73
6.7	CONCLUSÕES.....	76
7	COMPARAÇÃO DE DESEMPENHO DE ARQUITETURAS DE REDES NEURAIIS	78
7.1	Identificação do Atrator Caótico	78
7.2	Identificação do Reator Biológico	84
7.3	Identificação do Tanque de <i>pH</i>	93
7.4	Conclusões	96
8	PREDIÇÃO PARA HORIZONTE DE LONGO ALCANCE DO COMPORTAMENTO DE SISTEMAS DINÂMICOS.....	98
8.1	<i>CSTR</i> Encamisado	98

8.2	Reação de Van de Vusse.....	105
8.3	Conclusões	109
9	IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS NA PRESENÇA DE RUÍDO UTILIZANDO REDES DE FUNÇÕES DE ESCALA.....	110
9.1	Introdução	110
9.2	Geração de Dados com Ruído	111
9.3	Identificação do Tanque de <i>pH</i> a partir de Dados Corrompidos com Ruído .	112
9.4	Identificação do Reator Biológico a partir de Dados Corrompidos com Ruído 114	
9.5	Conclusões	118
10	CONTROLE PREDITIVO DO NÍVEL DE UM TANQUE COM ALTERAÇÃO DE DIÂMETRO	119
10.1	Descrição do Tanque de Nível	119
10.2	Treinamento da Rede de Funções de Escala.....	121
10.3	Validação da Rede de Funções de Escala	123
10.3.1	Simulação	123
10.3.2	Análise de Resíduos de Treinamento	124
10.4	Princípios Básicos de Controle Preditivo	127
10.5	Implementação Computacional do Controlador	128
10.6	Projeto do Controlador	129
10.7	Resultados	130
10.8	Conclusões	133
11	CONTROLE DE UM TANQUE DE PH.....	134
11.1	Identificação do Tanque de <i>pH</i>	134
11.2	Objetivos do Controlador	136
11.3	Preditor Utilizado no Controlador.....	136
11.4	Resultados	137

11.5	Conclusões	141
12	CONCLUSÕES FINAIS	142
13	APÊNDICE	144
13.1	Apêndice 1	144
13.2	Apêndice 2 - Calibração do Sensor de Pressão	146
13.2.1	Procedimento para Determinação da Curva de Calibração	146
14	REFERÊNCIAS BIBLIOGRÁFICAS	148

LISTA DE FIGURAS

Figura 2-1 - Função de escala e <i>wavelet</i> (base de Haar)	9
Figura 2-2 - Função de escala e <i>wavelet</i> (base de Hat)	9
Figura 2-3 - Função de escala e <i>wavelet</i> (base <i>spline</i> quadrática).....	9
Figura 2-4 - Função de escala e <i>wavelet</i> (base de Daubechies).....	10
Figura 2-5 - Função de escala e <i>wavelet</i> (base <i>spline</i> bicúbica).....	10
Figura 2-6 - Base de Haar em duas dimensões: (a) função de escala e <i>wavelet</i> (uma dimensão); (b) função de escala (duas dimensões); (c) três <i>wavelets</i> mãe(duas dimensões).....	18
Figura 3-1 - Dados de entrada: X1, X2 possuem distribuição uniforme.....	21
Figura 3-2 - Dados de entrada: X1 possui distribuição uniforme, X2 distribuição gaussiana.....	21
Figura 3-3 - Superfície de previsão: rede de funções de escala (primeiro grupo de dados).....	23
Figura 3-4 - Superfície de previsão: <i>wavenet</i> (primeiro grupo de dados)	23
Figura 3-5 - Superfície de previsão: rede de funções de escala (segundo grupo de dados)	24
Figura 3-6 - Superfície de previsão: <i>wavenet</i> (segundo grupo de dados).....	24
Figura 3-7 - Aumento do número de extremos conforme cresce a dimensão do espaço de entrada	26
Figura 3-8 - (a) função de escala <i>spline</i> ; (b) derivada de primeira ordem; (c) derivada de segunda ordem.....	27
Figura 3-9 - <i>Spline</i> dividida em 3 regiões	28
Figura 3-10 - Expansão contendo 8 funções de escala.....	29
Figura 3-11 - Expansão contendo 5 funções de escala.....	29
Figura 5-1 - Perturbações aplicadas na vazão de base do tanque de <i>pH</i>	40
Figura 5-2 – Evolução do <i>pH</i> no tanque.....	41
Figura 5-3 - Dados reais \times previsão da rede <i>feedforward</i> treinada por <i>backpropagation</i>	42
Figura 5-4 - Dados reais \times previsão da rede <i>feedforward</i> treinada por Levenberg-Marquardt.....	42
Figura 5-5 - (a) Erros de previsão da rede <i>feedforward</i> treinada por <i>backpropagation</i> (b) Erros de previsão da rede <i>feedforward</i> treinada por Levenberg-Marquardt.....	43
Figura 5-6 - Evolução do erro de treinamento utilizando os métodos <i>backpropagation</i> e Levenberg-Marquardt.....	44
Figura 5-7 - Dados reais \times previsão da rede <i>feedforward</i> treinada por <i>GA</i> + mínimos quadrados.....	50

Figura 5-8 - (a) Erros de previsão da rede <i>feedforward</i> treinada por Levenberg-Marquardt (b) Erros de previsão da rede <i>feedforward</i> treinada por GA + mínimos quadrados.....	50
Figura 5-9 - Evolução do treinamento da rede <i>feedforward</i> utilizando Levenberg-Marquardt e GA + mínimos quadrados.....	51
Figura 6-1 - Comparação da aproximação do mesmo conjunto de dados para diferentes modelos.....	62
Figura 6-2 - Valor absoluto dos pesos para alguns valores do parâmetro de regularização.....	68
Figura 6-3 - Influência do parâmetro de regularização no número de condição da matriz de covariância.....	68
Figura 6-4 - Influência do parâmetro de regularização no valor de $100-R^2$	69
Figura 6-5 - Valor absoluto dos pesos para alguns valores do parâmetro de regularização.	70
Figura 6-6 - Influência do parâmetro de regularização no número de condição da matriz de covariância (1° nível de resolução).....	71
Figura 6-7 - Influência do parâmetro de regularização no número de condição da matriz de covariância (2° nível de resolução).....	71
Figura 6-8 - Influência do parâmetro de regularização no valor de $100-R^2$	72
Figura 6-9 - Valor absoluto dos pesos para alguns valores do parâmetro de regularização (μ)	74
Figura 6-10 - Norma do vetor de pesos da camada de saída _x valor do parâmetro de regularização (μ).....	74
Figura 6-11 - Número de condição da matriz de covariância para alguns valores do parâmetro de regularização (μ).....	75
Figura 6-12 - $100-R^2$ x número de épocas para alguns valores do parâmetro de regularização (μ).....	76
Figura 7-1 - (a) Comportamento oscilatório ($\alpha= 3,5$); (b) Comportamento caótico ($\alpha= 3,9$)	78
Figura 7-2 - O valor de $100-R^2$ em função do número de parâmetros ajustáveis para as redes testadas (treinamento).....	81
Figura 7-3 - Valor de <i>AIC</i> em função do número de parâmetros ajustáveis para as redes testadas.....	83
Figura 7-4 - O valor de $100-R^2$ em função do número de parâmetros ajustáveis para as redes testadas (Teste).....	84
Figura 7-5 - Perturbações aplicadas na vazão de alimentação.....	86
Figura 7-6 - Concentração de células.....	86

Figura 7-7 - Concentração de substrato	86
Figura 7-8 - Concentração de produto	87
Figura 7-9 - Perturbações aplicadas na vazão de alimentação	87
Figura 7-10 - Dados da saída (concentração de células)	88
Figura 7-11 - Dados da saída (concentração de substrato).....	88
Figura 7-12 - Dados da saída (concentração de produto).....	88
Figura 7-13 - O valor de $100-R^2_x$ número de parâmetros ajustáveis das redes (treinamento) (a) concentração de células; (b) concentração de substrato; (c) concentração de produto	90
Figura 7-14 - O valor de AIC_x número de parâmetros ajustáveis das redes (treinamento) (a) concentração de células; (b) concentração de substrato; (c) concentração de produto	91
Figura 7-15 - O valor de $100-R^2_x$ número de parâmetros ajustáveis das redes (teste) (a) concentração de células; (b) concentração de substrato; (c) concentração de produto	92
Figura 7-16 - $100-R^2_x$ número de parâmetros ajustáveis (treinamento)	94
Figura 7-17 - AIC_x número de parâmetros ajustáveis (treinamento).....	95
Figura 7-18 - $100-R^2_x$ número de parâmetros ajustáveis (teste).....	95
Figura 8-1 - Perturbações aplicadas na vazão de alimentação	101
Figura 8-2 - Dados treinamento (concentração de reagente <i>A</i>)	102
Figura 8-3 - Dados treinamento (temperatura no reator).....	102
Figura 8-4 - Dados treinamento (temperatura na camisa)	102
Figura 8-5 - Perturbações aplicadas na vazão de alimentação (teste)	103
Figura 8-6 - Dados teste x simulação rede (concentração de reagente <i>A</i>).....	103
Figura 8-7 - Dados teste x simulação rede (temperatura no reator)	104
Figura 8-8 - Dados teste x simulação rede (temperatura na camisa).....	104
Figura 8-9 - Perturbações aplicadas na taxa de diluição (dados de treinamento).....	106
Figura 8-10 - Concentração de <i>A</i> (dados de treinamento).....	106
Figura 8-11 - Concentração de <i>B</i> (dados de treinamento).....	107
Figura 8-12 - Perturbações aplicadas na taxa de diluição (dados de teste)	107
Figura 8-13 – Dados teste $_x$ rede atuando como simulador (concentração de <i>A</i>)	108
Figura 8-14 - Dados teste $_x$ rede atuando como simulador (concentração de <i>B</i>).....	108
Figura 9-1 - Grupos de dados de treinamento com ruído obtido a partir do caso sem ruído	113
Figura 9-2 - Grupos de dados de teste com ruído obtido a partir do caso sem ruído	113

Figura 9-3 - Comparação previsão da rede em aberto \times dados do teste.....	114
Figura 9-4 - Dados de treinamento com ruído obtidos a partir do caso sem ruído; (a) concentração de células; (b) concentração de substrato; (c) concentração de produto	115
Figura 9-5 - Dados de teste com ruído obtidos a partir do caso sem ruído; (a) concentração de células; (b) concentração de substrato; (c) concentração de produto	116
Figura 9-6 - Comparação rede atuando como simulador \times dados do teste (a) concentração de células; (b) concentração de substrato; (c) concentração de produto	117
Figura 10-1 - Tanques de nível LCP; em detalhe o tanque com alteração brusca de diâmetro no qual foram realizados os experimentos de controle	120
Figura 10-2 - Diagrama do sistema de controle do tanque de nível	121
Figura 10-3 - Perturbações aplicadas no sinal enviado pelo sistema de controle (treinamento)	122
Figura 10-4 - Dados de altura (treinamento); (A) seção inferior; (B) seção intermediária; (C) seção superior tanque	122
Figura 10-5 - Comparação dados reais \times previsão rede (treinamento) (A) seção inferior; (B) seção intermediária; (C) seção superior tanque	123
Figura 10-6 - Perturbações aplicadas no sinal enviado pelo sistema de controle (teste)....	124
Figura 10-7 - Comparação dos dados teste \times curva de simulação gerada pela rede (A) seção inferior; (B) seção intermediária; (C) seção superior tanque	124
Figura 10-8 - Testes de correlação baseados em resíduos (95% confiança)	126
Figura 10-9 - Transições de <i>setpoint</i> para os controladores neural e PI. (A) seção inferior; (B) seção intermediária; (C) seção superior tanque	131
Figura 10-10 - Ações de controle tomadas pelos controladores neural e PI	132
Figura 11-1 – Comparação dados reais processo \times previsão do modelo linear	135
Figura 11-2 Comparação dados reais processo \times previsão do modelo linear (correspondente às amostras entre 530 e 630 do grupo de dados)	136
Figura 11-3 - Desempenho do controlador preditivo baseado em modelo linear ($N_2= 5$, $\lambda=0,55$).....	138
Figura 11-4 - Desempenho do controlador preditivo baseado em modelo linear ($N_2= 5$, $\lambda=0,1$)	138
Figura 11-5 - Comparação entre o desempenho dos controladores preditivos baseados nos modelos linear e neural (perturbações <i>setpoint</i>) (A) <i>pH</i> (variável controlada); (B) vazão de base (variável manipulada)	139

Figura 11-6 - Comparação de desempenho dos controladores preditivos baseados em modelo linear e neural (perturbações carga) (A) pH (variável controlada); (B) vazão de base (variável manipulada).....	140
--	-----

LISTA DE TABELAS

Tabela 2.1 - Coeficientes utilizados para geração de algumas <i>wavelets</i> mais conhecidas.....	8
Tabela 2.2 - Suporte da função de escala e de suas compressões diádicas (caso $N=3$).....	11
Tabela 2.3 - Valor da interseção entre funções adjacentes utilizadas em uma expansão para alguns tipos de função de escala.....	11
Tabela 3.1 –Comparação entre desempenho da <i>wavenet</i> e da rede de funções de escala	22
Tabela 5.1 - Variáveis e parâmetros do processo de neutralização.....	39
Tabela 5.2 - Comparação do desempenho dos métodos de <i>backpropagation</i> e Levenberg-Marquardt (treinamento).....	41
Tabela 5.3 - Comparação do desempenho dos métodos Levenberg-Marquardt e <i>GA</i> +mínimos quadrados.....	49
Tabela 6.1 - Influência da regularização nos autovalores da matriz $F(X)^T F(X)$	60
Tabela 7.1 - Estrutura das redes utilizadas na identificação do atrator caótico.....	80
Tabela 7.2 - Parâmetros do reator biológico.....	85
Tabela 7.3 - Estrutura das redes utilizadas na identificação do reator biológico.....	89
Tabela 7.4 - Estrutura das redes utilizadas na identificação do tanque de <i>pH</i>	94
Tabela 8.1 - Parâmetros do <i>CSTR</i> encamisado.....	100
Tabela 8.2 - Parâmetros do reator Van de Vusse.....	105
Tabela 10.1 - Parâmetros dos controladores.....	131

1 INTRODUÇÃO

Nos últimos anos, foram realizados grandes avanços na análise e identificação de processos não-lineares com a utilização de modelos provenientes da inteligência computacional. Nesta área destacam-se os modelos obtidos a partir de regras (lógica difusa) e, principalmente, os que utilizam funções de ativação (redes neurais). Na visão da inteligência computacional, uma rede neural é composta de camadas de unidades processadoras interconectadas através de pesos e conhecidas por neurônios. Um sinal ao passar por um neurônio sofre uma transformação, em geral não-linear, aplicada pela função de ativação desse.

As redes neurais mais utilizadas em identificação e controle não-linear são as do tipo *feedforward*. Muitos trabalhos encontrados na literatura as utilizam. Grande parte desse sucesso pode ser atribuído ao algoritmo de treinamento supervisionado e iterativo conhecido por retropropagação do erro, também conhecido por *backpropagation*, RUMELHART e MCCLELLAND(1986). No entanto, a identificação de sistemas pode tornar-se uma tarefa complexa devido ao grande número de parâmetros da estrutura da rede (número de camadas intermediárias, número de neurônios por camada) e do método de treinamento (seleção inicial dos pesos, determinação do fator de aprendizagem, taxa de momento e critério de parada) HAYKIN(1999). Os parâmetros da estrutura de rede e do método de treinamento são determinados por tentativa e erro ou através de conhecimento heurístico. Devido ao grande número de parâmetros e pela ausência de fundamentação matemática das redes *feedforward*, tem-se estudado outras estruturas de modelagem, tais como modelos não-lineares, mas lineares nos parâmetros. Este último tipo de estrutura é atrativo, pois o treinamento pode ser formalizado como um problema de regressão linear e resolvido por mínimos quadrados. Dois tipos de redes não-lineares, mas lineares nos parâmetros têm sido utilizados: as redes de funções de base radial (*RBFN*) e ultimamente, as de *wavelets*. As *RBFN* possuem apenas uma camada intermediária, sendo que, os neurônios utilizam funções de ativação definidas em torno de centros HAYKIN(1999). No caso das *RBFN*, os parâmetros de estrutura de rede correspondem à determinação do número, dimensões e localização dos centros, um número bem menor de parâmetros quando se compara esse tipo de rede com as *feedforward*. As redes de *wavelets* são compostas de funções localizadas assim como as *RBFN*; no entanto, são mais bem fundamentadas matematicamente que estas últimas. As redes de *wavelets* utilizam o conceito de multiresolução MALLAT(1989) que é uma metodologia de representação de

funções em diferentes escalas ou resoluções. Uma função, no contexto da análise em multiresolução, é representada como a soma de sucessivas aproximações realizadas em espaços de funções definidos pela teoria das *wavelets* DAUBECHIES(1992), STRANG e NGUYEN(1996). Aplicações de *wavelets* para problemas de aproximação e construção de redes neurais surgem com BAKSHI e STEPHANOPOULOS(1993) através das *wavenets* e ZHANG e BENVENISTE(1992) com as redes de frames.

A aproximação de uma função $f(x)$ é efetuada, no contexto de multiresolução, obtendo sua decomposição em termos de versões deslocadas e comprimidas de uma função básica, conhecida como "*wavelet* mãe". As translações e compressões são definidas pela teoria de *wavelets*. Dessa forma, são fixados previamente a localização e o suporte (os pontos do domínio de uma função que são diferentes de zero) da função de ativação de cada neurônio. Neste caso, o treinamento da rede se restringe à determinação dos coeficientes (pesos) relativos às projeções. No entanto, para problemas de maior porte, o número de neurônios de uma *wavenet* cresce exponencialmente à medida que se aumenta o número de entradas. Conforme definido pela teoria de *wavelets*, o suporte das funções de ativação diminui de forma acentuada em relação ao domínio do problema à medida que cresce o número de entradas, ou seja, as funções de ativação podem se tornar muito localizadas, diminuindo a capacidade de generalização da rede treinada. Além desses fatores, como em qualquer problema de identificação, determinar um modelo, a partir de um conjunto finito de observações, sem qualquer conhecimento sobre o sistema que os gerou é um problema mal-posto, no sentido que um modelo único pode não existir TIKHONOV e ARSENIN(1977). Uma forma de diminuir incertezas é adicionar conhecimento prévio do sistema que se deseja identificar.

Neste trabalho foram propostas modificações na *wavenet* com o objetivo de melhorar sua capacidade de generalização e reduzir o número de neurônios. O desempenho da nova rede também foi superior, em termos de capacidade de aproximação e generalização, quando comparado com as tradicionais redes *feedforward* e de base radial, na identificação de processos não-lineares: a equação logística que é um atrator caótico, um reator biológico em que ocorre inibição do crescimento de células pela formação de produto e um tanque de *pH*.

O desempenho de previsão da nova rede proposta, que é do tipo estática, foi também validado utilizando essa como um simulador de processos não-lineares. Foram efetuados testes sem ruído para um *CSTR* (*Continuous Stirred Tank Reactor*) encamisado com múltiplos estados estacionários e um reator cujo comportamento apresenta resposta inversa variável.

Alguns testes foram efetuados, com a adição de ruído nas variáveis previstas, para o reator biológico e para o tanque de *pH*.

Com o objetivo de melhorar a capacidade de generalização de redes neurais, foi efetuado um estudo a respeito da minimização das curvaturas da superfície de um modelo empírico e efetuada sua conexão com o problema de sobre-treinamento.

A nova rede proposta foi utilizada como modelo de um controlador preditivo em dois processos não-lineares: um experimental, que abordou o controle de nível de um tanque com alteração da área e outro simulado, relativo ao *pH* de um tanque agitado.

Alguns testes foram efetuados para comparar o desempenho do controlador preditivo baseado na rede neural descrita com os clássicos PI e preditivo baseado em modelo linear. Em todos os casos, melhores resultados foram obtidos com o controlador não-linear.

2 TEORIA DE *WAVELETS* E MULTIRESOLUÇÃO

2.1 Introdução

As famílias de *wavelets* são bases para um espaço de funções e a representação de uma função arbitrária é realizada utilizando uma ponderação dos elementos da base; isto é, através de uma série. As redes de *wavelets* são basicamente redes de funções de base radial. Fixadas as variáveis do processo e a família de *wavelets* a ser utilizada, ficam definidos automaticamente os centros e os raios de todas as funções em todas as escalas. A partir de uma *wavelet*, pode ser gerada, por translações e mudanças de escala, uma família de funções que é base para um espaço de funções.

O desenvolvimento e o rápido crescimento da teoria de *wavelets* e análise em multiresolução fornece uma poderosa estrutura matemática para o desenvolvimento de redes neurais MALLAT(1989), DAUBECHIES(1992). A teoria de *wavelets* foi inicialmente utilizada por GROSSMANN e MORLET(1984) na área de processamento de sinais, sendo que MEYER(1986) introduziu a primeira *wavelet* contínua. A pesquisa na obtenção de *wavelets* tem progredido rapidamente desde a construção de *wavelets* de suporte compacto DAUBECHIES(1988) e da estrutura de multiresolução. Podem ser encontradas aplicações de *wavelets* nas mais diversas áreas:

- 1) Tratamento de sinais: previsão e detecção de falhas VENKATARAMAN e GONZ(2002), REN *et al.*(2000); Processamento de imagens FATEMI e BOLOUKI (2002), ANTONINI *et al.*(1998); filtragem YAMAGUCHI(2002), SHAO *et al.*(1999), KOSANOVICH e PIOVOSO(1997);
- 2) Estatística: classificação de dados HUNG *et al.*(2002); seleção de variáveis para modelos ALSBERG(1998);
- 3) Modelagem, controle e otimização de processos: identificação de sistemas BECERRA *et al.*(2002), GAO *et al.*(2001), CRISTEA *et al.*(2000); controle preditivo DONGBING e HUOSHENG(2000), LI e XI(2000), HUANG e JIN(1997), ELIAS-JUAREZ e KANTOR(1992); controle adaptativo XIA *et al.*(2002), XU e TAN(2001), XU e HO(1999); otimização SAFAVI e ROMAGNOLI(1995).

Basicamente, *wavelets* são utilizadas para aproximação de funções, para análise de sinais nos domínios do tempo e da frequência e como uma nova área de estudo da Matemática. Neste trabalho, o interesse está voltado para a utilização de *wavelets* como uma ferramenta para desenvolvimento de redes neurais.

O uso de *wavelets* no desenvolvimento de redes neurais segue duas principais correntes. A primeira é baseada no trabalho de ZHANG e BENVENISTE(1992), que introduz uma rede semelhante às do tipo *feedforward*, sendo utilizada como função de ativação uma *wavelet* ao invés de uma função hiperbólica. A segunda é a rede desenvolvida por BAKSHI e STEPHANOPOULOS(1993) denominada *wavenet*, que utiliza uma metodologia de aprendizado não iterativa e hierárquica. A aproximação de funções baseada na *wavenet* está mais próxima da metodologia de multiresolução utilizada em tratamento de sinais, enquanto preserva as características de uma rede neural.

As *wavenets* têm-se tornado uma ferramenta de aprendizado alternativa às tradicionais redes *feedforward* devido às suas vantagens, tais como: Precisão da aproximação (o erro de treinamento pode ser reduzido o quanto desejado), estrutura simples e rápido treinamento, entre outras. Embora uma *wavenet* possa ser utilizada para resolver problemas multidimensionais, a grande parte dos trabalhos encontrados na literatura descreve aplicações para um número pequeno de variáveis de entrada. Por exemplo, HAROLD *et. al.*(1992) propuseram vários tipos de arquiteturas de redes de *wavelets* que foram aplicadas na classificação de sinais vocais; PATI e KRISHNAPRASAD(1993) desenvolveram um método de síntese de *wavenets* no qual o número de funções da rede podia ser determinado a partir do grupo de dados de treinamento; DELYON *et al.*(1995) apresentaram um método aproximado para estimar os coeficientes das funções da *wavenet* (os valores dos coeficientes são restritos a uma faixa fornecida pelo método). O número de entradas utilizado em todos os casos foi menor ou igual a três e apenas uma saída. Aplicações de maior porte não foram realizadas, provavelmente, devido ao crescimento do número de funções da *wavenet* à medida que aumenta o número de entradas. Cabe ressaltar que muitos problemas da área de processos químicos são multidimensionais. Para tratar de tais problemas foram propostas algumas simplificações na *wavenet*: ROQUEIRO(1995) utilizou uma rede formada pelo somatório de sub-redes com apenas uma variável independente; a *Wavelet-Sigmoid Basis Function Neural Network (WSBFN)* ZHAO *et. al.*(1998) utiliza uma camada intermediária composta de funções de escala e *wavelets* e na camada de saída, funções de ativação sigmóides; o trabalho de KIM *et. al.*(2002) descreve uma rede em que os parâmetros das *wavelets* foram determinados por algoritmo genético e os pesos da camada de saída através do método

backpropagation, sendo que o número de neurônios deve ser definido previamente à etapa de estimação dos pesos da rede.

Nas próximas seções, apresenta-se uma descrição da teoria de *wavelets* e sua utilização no contexto da aproximação de funções.

2.2 Famílias de *Wavelets*

As *wavelets* constituem uma família de funções construídas a partir de dilatações e translações de uma função básica Ψ denominada "*wavelet mãe*". Essa família de *wavelets* é definida da seguinte forma:

$$\Psi_{(a,b)}(x) = |a|^{-1/2} \cdot \Psi\left(\frac{x-b}{a}\right) \quad (2.1)$$

$$a, b \in R; a \neq 0$$

Na Equação (2.1), o parâmetro b efetua uma translação no domínio da *wavelet* mãe e o coeficiente a , também conhecido por parâmetro de escala, efetua uma compressão. O coeficiente que precede a *wavelet* mãe tem como objetivo normalizar as funções da família, garantindo que todas as funções tenham a mesma energia, ou seja, $\int_{\Omega} \Psi_{(a,b)}^2 dx$ é constante (onde Ω é o domínio de integração).

A função ψ deve ter suporte não nulo (o suporte é um subconjunto do domínio D de uma função f definido como $\{x \in D | f(x) \neq 0\}$). Outra propriedade de ψ , de acordo com a teoria de *wavelets*, é ter média zero. Dessa forma, ψ deve ter algumas oscilações ou ser identicamente nula. As *wavelets* formam uma base para o $L^2(R)$ (espaço das funções integráveis ao quadrado) DAUBECHIES(1988). Os valores dos parâmetros a e b da Equação (2.1) podem ser restritos a um conjunto discreto e ainda assim gerar o $L^2(R)$. Se $a = a_0^{-m}$ e $b = nb_0 a_0^{-m}$ para $m, n \in Z$ (conjunto dos números inteiros) com $a_0 > 1$ e $b_0 > 0$ é possível definir a seguinte família de *wavelets* discretas, conforme Equação (2.2).

$$\Psi_{(m,n)}(x) = |a_0|^{m/2} \cdot \Psi(a_0^m x - nb_0) \quad m, n \in Z \quad (2.2)$$

Normalmente em aplicações são utilizados os valores $a_0 = 2$ e $b_0 = 1$ BAKSHI e STEPHANOPOULOS(1993). A representação de uma função $f(x)$ no espaço de funções gerado pela família $\psi_{(m,n)}$ pode ser descrita como:

$$f(x) = \sum_m \sum_n c_{m,n} \Psi_{m,n}(x) \quad (2.3)$$

Com a escolha adequada de a_0 , b_0 e ψ é possível gerar um conjunto de funções independentes, de tal forma que a família $\{\psi_{(m,n)}\}$ gere o espaço das funções integráveis ao quadrado. Dado que não é objetivo deste trabalho o desenvolvimento de famílias de *wavelets*, foram utilizados os valores $a_0 = 2$ e $b_0 = 1$. Também foram escolhidas algumas funções específicas, as bases de *Haar*, *Splines* e *Daubechies* (Ver seção 2.3), para apresentação de exemplos. As aplicações foram efetuadas utilizando apenas a *spline* quadrática; maiores razões da utilização dessa última como função de escala podem ser encontradas na seção 3.2.

2.3 Obtenção de *Wavelets Mãe*

A representação analítica das funções ψ é possível somente em alguns casos. Em geral, uma *wavelet* mãe é obtida numericamente e armazenada em um vetor para posterior uso. Versões comprimidas e/ou deslocadas desta podem ser facilmente calculadas. A obtenção de uma *wavelet* mãe começa com a solução da equação de dilatação DAUBECHIES(1988), STRANG e NGUYEN(1996):

$$\Phi(x) = \sum_{n=0}^N p_n \Phi(2x - n) \quad (2.4)$$

onde Φ é conhecida por função de escala e N é o índice do último coeficiente.

No trabalho de STRANG e NGUYEN(1996) foram descritos três métodos iterativos para resolver a Equação (2.4), fixado o conjunto de parâmetros p . O método mais simples admite inicialmente que a função Φ seja constante para um suporte finito. Esta solução provisória é aplicada na Equação (2.4), sendo obtida uma nova estimativa para Φ . O procedimento

descrito é repetido até a convergência. Em seguida, a *wavelet* mãe ψ é construída a partir da função de escala utilizando a Equação (2.4):

$$\Psi(x) = \sum_{n=0}^N (-1)^n p_n \Phi(2x - n) \quad (2.5)$$

A escolha dos coeficientes p_n está intimamente ligada às propriedades das famílias de funções que este método gera e à estabilidade do método DAUBECHIES(1988). Para cada conjunto de coeficientes existe uma *wavelet* mãe. Através de compressões e deslocamentos desta pode-se obter uma família de funções que aproxima o $L^2(\mathbb{R})$. Conforme a teoria de *wavelets*, uma condição imposta aos coeficientes para garantir a unicidade de Φ é que

$$\sum_{n=0}^N p_n = 2 \text{ STRANG e NGUYEN(1996).}$$

Nas Figuras 2.1 a 2.5 mostram-se algumas das *wavelets* mais conhecidas, bem como suas respectivas funções de escala. Os coeficientes utilizados para geração destas, através das Equações (2.4) e (2.5), podem ser vistos na Tabela 2.1.

Tabela 2.1 - Coeficientes utilizados para geração de algumas *wavelets* mais conhecidas

	Haar	Hat	<i>Spline</i> quadrática	Daubechies	<i>Spline</i> bicúbica
Coeficientes	$p_0 = 1$ $p_1 = 1$	$p_0 = \frac{1}{2}$ $p_1 = 1$ $p_2 = \frac{1}{2}$	$p_0 = \frac{1}{4}$ $p_1 = \frac{3}{4}$ $p_2 = \frac{3}{4}$ $p_3 = \frac{1}{4}$	$p_0 = \frac{1}{4} \cdot (1 + \sqrt{3})$ $p_1 = \frac{1}{4} \cdot (1 + \sqrt{3})$ $p_2 = \frac{1}{4} \cdot (1 - \sqrt{3})$ $p_3 = \frac{1}{4} \cdot (1 - \sqrt{3})$	$p_0 = \frac{1}{8}$ $p_1 = \frac{1}{2}$ $p_2 = \frac{3}{4}$ $p_3 = \frac{1}{2}$ $p_4 = \frac{1}{8}$

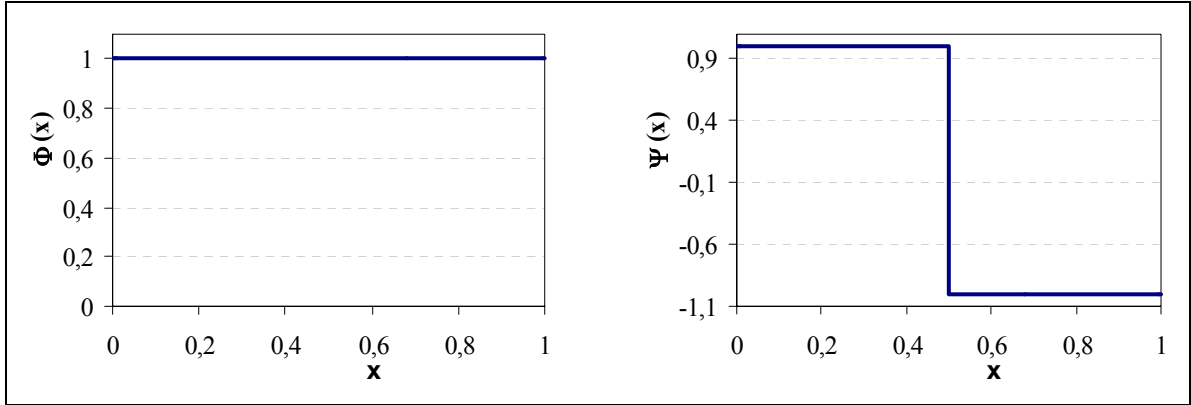


Figura 2-1 - Função de escala e *wavelet* (base de Haar)

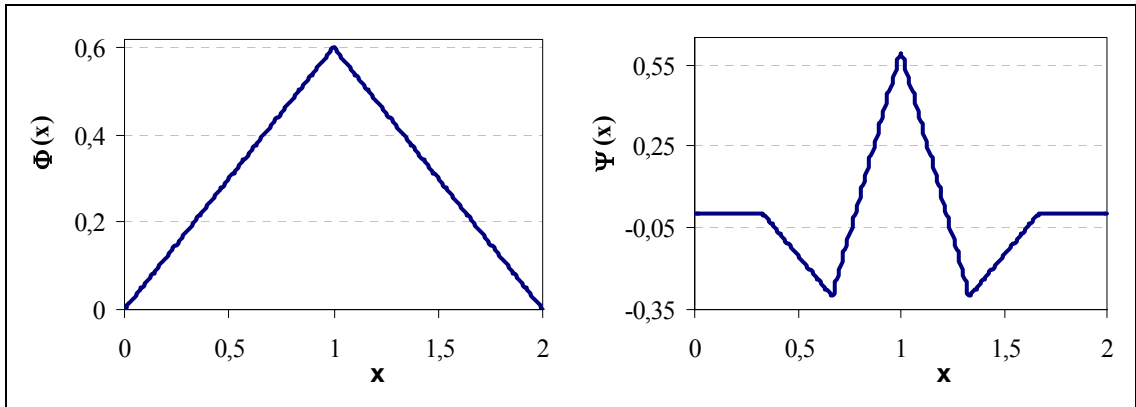


Figura 2-2 - Função de escala e *wavelet* (base de Hat)

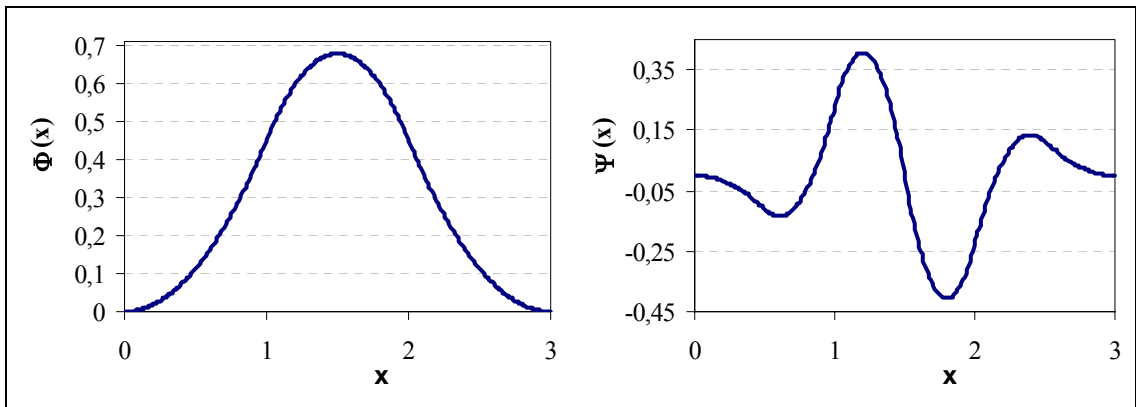


Figura 2-3 - Função de escala e *wavelet* (base *spline* quadrática)

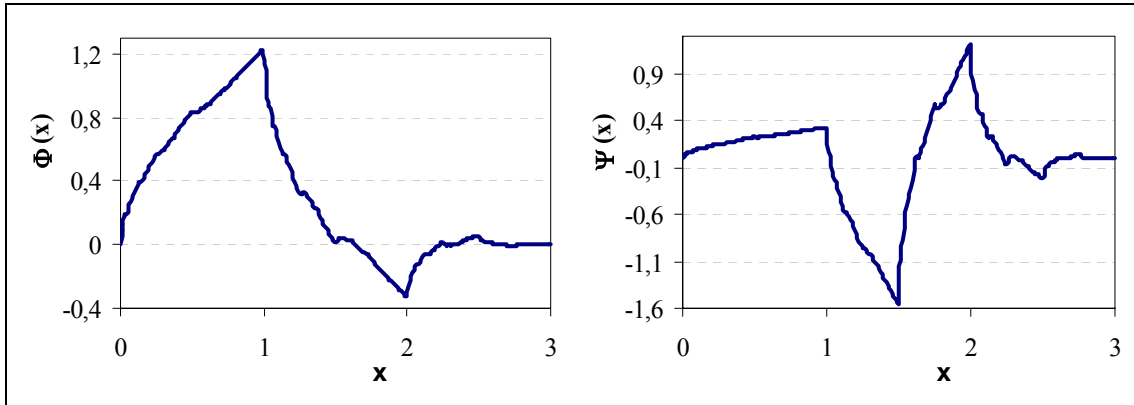


Figura 2-4 - Função de escala e *wavelet* (base de Daubechies)

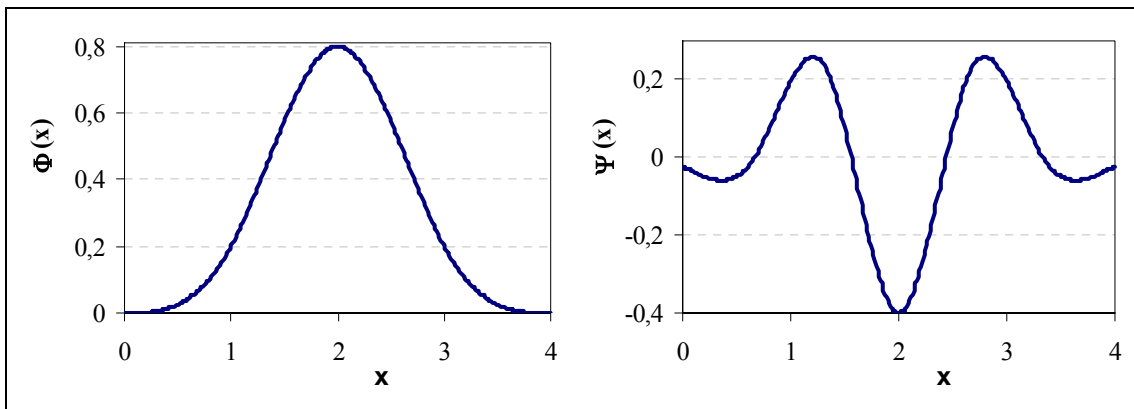


Figura 2-5 - Função de escala e *wavelet* (base *spline* bicúbica)

Uma propriedade da função de escala, obtida a partir da Equação (2.4), é que esta vale 0 fora do intervalo $[0, N]$, ou seja, seu suporte é o intervalo $[0, N]$ STRANG e NGUYEN(1996). Pode-se deduzir a partir desta propriedade que a interseção de duas funções de escala adjacentes utilizadas em uma expansão depende do número de coeficientes. Observando o lado direito da Equação (2.4), conclui-se que $N+1$ cópias de Φ , com suporte $N/2$ e igualmente espaçadas, devem ser acomodadas no intervalo $[0, N]$, pois conforme o lado esquerdo dessa equação o somatório define Φ . Exemplificando para o caso $N=3$, tem-se 4 cópias de Φ acomodadas em $[0, 3]$, cada uma delas com suporte 1,5 e espaçadas de uma distância de 0,5 (Ver Tabela 2.2)

De acordo com a Tabela 2.2, para $N=3$, a interseção entre quaisquer duas cópias comprimidas e adjacentes da função de escala, por exemplo $\Phi(2x)$ e $\Phi(2x-1)$, é de $2/3$ do suporte dessas.

Tabela 2.2 - Suporte da função de escala e de suas compressões diádicas (caso $N=3$)

Função	Suporte
$\Phi(x)$	[0; 3]
$\Phi(2x)$	[0; 1,5]
$\Phi(2x-1)$	[0,5; 2]
$\Phi(2x-2)$	[1; 2,5]
$\Phi(2x-3)$	[1,5; 3]

Mostra-se na Tabela 2.3 o valor da interseção para as funções de escala descritas na Tabela 2.1.

Tabela 2.3 - Valor da interseção entre funções adjacentes utilizadas em uma expansão para alguns tipos de função de escala

Base	Número de coeficientes	Intersecção
Haar	2	0
Hat	3	1/2
<i>Spline</i> quadrática	4	2/3
Daubechies	4	2/3
<i>Spline</i> bicúbica	5	3/4

Pode-se observar na Tabela 2.3, que a interseção entre quaisquer duas cópias, comprimidas e adjacentes, de uma função de escala utilizada em uma expansão cresce com o aumento do número de coeficientes.

Todo o raciocínio apresentado a respeito de interseção de funções de escala também é válido para as *wavelets*. A Equação (2.5), que define as *wavelets*, tem formato análogo ao da Equação (2.4) utilizada no raciocínio.

2.4 Aproximação de Funções através de Multiresolução

Uma função $f(x) \in L^2(\mathbb{R})$ é representada, no contexto da multiresolução, através de aproximações sucessivas em diferentes resoluções MALLAT(1989). A análise de multiresolução é baseada em um conjunto de subespaços $V_m \subset L^2(\mathbb{R})$ que satisfazem:

1. $V_m \subset V_{m+1} \forall m \in Z$ (Os subespaços V_m são encaixantes);
2. $\bigcap_{m \in Z} V_m = \{0\}$ e $\bigcup_{m \in Z} V_m = L^2(R)$ (A união de todos os subespaços V_m gera o $L^2(R)$);
3. $f(x) \in V_m \Leftrightarrow f(2 \cdot x) \in V_{m+1}$.

As idéias centrais da multiresolução são ($\forall m, i, j \in Z$):

1. $V_{m+1} = W_m \oplus V_m$;
2. $W_m \perp V_m$ (\perp significa complemento ortogonal);
3. $W_i \perp W_j$.

onde V_i, W_i são os subespaços gerados por translações das funções de escala $\Phi(2^i)$ e das *wavelets* $\psi(2^i)$ respectivamente. Dessa forma, as *wavelets* formam conjuntos ortogonais (conjuntos em que quaisquer duas funções diferentes são ortogonais) dentro de uma mesma resolução (quaisquer duas *wavelets* diferentes em uma resolução são ortogonais) e entre diferentes resoluções (qualquer *wavelet* de uma determinada resolução é ortogonal a uma *wavelet* contida em outra resolução diferente). No caso das funções de escala, esta propriedade não é uma exigência. Como um exemplo, tem-se a base *spline* quadrática, cujo conjunto de versões transladadas da função de escala não gera um conjunto ortogonal.

Devido às características dos subespaços utilizados na multiresolução, V_m pode ser descrito como a combinação dos subespaços com resoluções inferiores V_i e W_i ($i < m$), ou seja:

$$V_{m+1} = V_m \oplus W_m = V_{m-1} \oplus W_{m-1} \oplus W_m = V_0 \oplus W_0 \oplus W_1 \dots \oplus W_m$$

O subespaço V_0 é equivalente a $W_{-\infty} \oplus \dots \oplus W_{-2} \oplus W_{-1}$ e, portanto, as características mais grosseiras de uma função $f(x)$ são identificadas por V_0 . As informações com um nível crescente de detalhamento são identificadas pelos subespaços W_i ($i \geq 0$), à medida que i cresce. Tomando $m \rightarrow \infty$ pode-se aproximar qualquer função em $L^2(R)$ com erro arbitrário.

2.5 Aproximação de Funções Unidimensionais

A expansão em série de funções mostrada na Equação (2.3) é, em geral, dividida em duas partes. As informações mais grosseiras são identificadas por uma expansão em funções de escala e os detalhes são identificados pelas *wavelets*. A aproximação de uma função $f(x)$ é constituída da soma das duas partes. Matematicamente tem-se:

$$f(x) = \sum_{n=-\infty}^{n=\infty} d_n \Phi_{0,n}(x) + \sum_{m=0}^{m=\infty} \sum_{n=-\infty}^{n=\infty} c_{m,n} \Psi_{m,n}(x) \quad (2.6)$$

Os índices m, n contidos na Equação (2.6) podem assumir valores ilimitados, indicando que a teoria de *wavelets* é aplicável para aproximação de funções de domínio e níveis de detalhamento arbitrário pertencentes ao $L^2(R)$. Na prática, $f(x)$ é conhecida apenas em um conjunto de pontos e, portanto, pode-se aproximar $f(x)$ em um domínio limitado e até um determinado nível de precisão. O número finito de pontos impõe restrições no número de funções utilizadas na aproximação. Dessa forma, são suficientes apenas alguns níveis de resolução para atingir um pequeno erro de aproximação (o que implica em m limitado). Além disso, o grupo de dados poderá ser normalizado, o que implica em um número finito de funções da série distribuídas nesse domínio normalizado (n finito).

Se a expansão em série de funções, correspondente à Equação (2.6), for descrita no formato de uma rede neural, ou seja, como uma *wavenet*, tem-se que cada elemento da série será equivalente a um neurônio.

Neurônios podem ser adicionados até que o erro de aproximação se aproxime arbitrariamente de zero. Dessa forma, mesmo se os dados contiverem ruído, pode-se fazer as previsões da expansão em série de *wavelets* tão próximas quanto se queira dos dados. Esta situação não é desejável em identificação de sistemas, sendo que o número de resoluções deve ser escolhido de forma que sejam identificadas apenas as características relevantes de um processo e não o ruído.

O espaço de funções, considerando as limitações em n , não é mais o $L^2(R)$ e sim $L^2([\alpha, \beta])$ (o espaço das funções integráveis ao quadrado definidas entre α e β). A faixa de normalização geralmente utilizada é entre $\alpha = 0$ e $\beta = 1$.

A Equação (2.6) pode ser reescrita considerando os índices m, n finitos (sem perda de generalidade, o índice n inicia de 0):

$$f(x) = \sum_{n=0}^{n=N_\Phi} d_n \Phi_{0,n}(x) + \sum_{m=0}^{m=M_{\max}} \sum_{n=0}^{n=N_{\Psi,m}} c_{m,n} \Psi_{m,n}(x) \quad (2.7)$$

onde

N_Φ : número de funções de escala;

M_{\max} : número de resoluções utilizadas na aproximação;

$N_{\Psi,m}$: número de *wavelets* na resolução m ;

d_n ($n = 0 \dots N_\Phi$) e $c_{m,n}$ ($m = 0 \dots M_{\max}$; $n = 0 \dots N_{\Psi,m}$): Coeficientes da expansão em série de funções;

d : vetor composto dos coeficientes d_n ($n = 0 \dots N_\Phi$);

c_m : vetor composto dos coeficientes $c_{m,n}$ contidos na resolução m ($n = 0 \dots N_{\Psi,m}$).

A Equação (2.7) pode ser escrita na forma de uma expansão em subespaços de funções:

Definindo:

$$V_0 = \sum_{n=0}^{n=N_\Phi} d_n \Phi_{0,n}(x)$$

$$W_m = \sum_{n=0}^{n=N_{\Psi,m}} c_{m,n} \Psi_{m,n}(x)$$

e lembrando que $V_{m+1} = W_m \oplus V_m$, tem-se $f(x) \in V_0 \oplus W_0 \oplus W_1 \dots W_{M_{\max}}$

Uma função é representada, no contexto da multiresolução, como a soma de sucessivas aproximações. Estas aproximações são realizadas projetando a função nos subespaços V e W definidos pela teoria de *wavelets*. Considerando F_i e H_i como as projeções, ou aproximações, de $f(x)$ nos subespaços V_i e W_i respectivamente, tem-se:

$$F_0 = \sum_{n=0}^{n=N_\Phi} \bar{d}_n \Phi_{0,n}(x)$$

$$H_m = \sum_{n=0}^{n=N_{\Psi,m}} \bar{c}_{m,n} \Psi_{m,n}(x)$$

onde

\bar{d} : vetor composto dos coeficientes estimados \bar{d}_n ($n = 0 \dots N_\Phi$);

\bar{c}_m : vetor composto dos coeficientes $\bar{c}_{m,n}$ contidos na resolução m ($n = 0 \dots N_{\Psi,m}$).

o que implica que $f(x) = F_0 + H_0 + H_1 + \dots + H_{M_{\max}}$

Deve-se notar que as projeções F_i e H_i correspondem a elementos pertencentes aos subespaços V_i e W_i . Dessa forma, a notação \bar{d} e \bar{c}_m significa que estes vetores de coeficientes são casos particulares de d e c_m , respectivamente. Os coeficientes \bar{d} e \bar{c}_m podem ser determinados através do método dos mínimos quadrados. Neste caso o critério de otimização utilizado é a norma quadrática do erro de aproximação, ou seja, o somatório do erro quadrático calculado nos pontos do grupo de dados.

A metodologia de multiresolução para aproximação de uma função $f(x)$ é aplicada da seguinte forma:

Partindo-se de F_0 (a aproximação de $f(x)$ mais grosseira)

$$F_0 = \sum_{n=0}^{n=N_\Phi} \bar{d}_n \Phi_{0,n}(x)$$

os coeficientes de \bar{d} devem ser estimados, minimizando o critério:

$$J_0 = \|f(x) - F_0\|^2$$

Para obter uma aproximação mais detalhada deverá ser incorporado um conjunto de *wavelets* pertencentes à escala 0:

$$F_1 = \sum_{n=0}^{n=N_\Phi} \bar{d}_n \Phi_{0,n}(x) + \sum_{n=0}^{n=N_{\Psi,0}} \bar{c}_{0,n} \Psi_{0,n}(x) = F_0 + H_0$$

Considerando que H_0 deve aproximar o erro entre $f(x)$ e F_0

$$f(x) - F_0 = \sum_{n=0}^{n=N_{\Psi,0}} \bar{c}_{0,n} \Psi_{0,n}(x) = H_0$$

os parâmetros \bar{c}_0 podem ser calculados minimizando o critério J_1

$$J_1 = \|(f(x) - F_0) - H_0\|^2$$

A expressão $f(x) - F_0$ em J_1 foi mostrada entre parênteses para reforçar que é conhecida para todos os pontos do grupo de dados, podendo ser obtida das informações de entrada e saída do grupo de dados, além dos coeficientes \bar{d} estimados da aproximação F_0 .

Em geral, para um k qualquer diferente de 0, a aproximação F_k

$$F_k = F_{k-1} + \sum_{n=0}^{n=N_{\Psi_{(k-1)}}} \bar{c}^{(k-1),n} \Psi_{(k-1),n}(x) = F_{k-1} + H_{k-1}$$

deve aproximar o erro entre $f(x)$ e F_{k-1}

$$f(x) - F_{k-1} = \sum_{n=0}^{n=N_{\Psi_{(k-1)}}} \bar{c}^{(k-1),n} \Psi_{(k-1),n}(x) = H_{k-1}$$

sendo que os parâmetros $\bar{c}^{(k-1)}$ podem ser calculados minimizando o critério J_k :

$$J_k = \|(f(x) - F_{k-1}) - H_{k-1}\|^2$$

Normalmente, a propriedade de ortogonalidade das *wavelets* não pode ser aproveitada na simplificação dos cálculos utilizados para estimação dos pesos. A propriedade de ortogonalidade das *wavelets* é referente ao produto interno no espaço $L^2(\mathcal{R})$, ou seja a integral $\int_{\Omega} \Psi_i(x) \cdot \Psi_j(x) dx \quad \forall i \text{ e } j, i \neq j$ (onde Ω é o domínio de integração). Quando se dispõe de um conjunto de dados, este produto interno é aproximado pelo usual produto de dois vetores, correspondentes aos valores dos pontos do grupo de dados aplicados nas *wavelets*. Como os dados podem estar amostrados de forma deficiente e não uniforme, a integral pode não ser, em geral, bem aproximada. Dessa forma, os pesos das funções contidas em uma mesma resolução não podem ser calculados separadamente, sendo necessário resolver um

sistema linear, resultante da aplicação do método dos mínimos quadrados, para determinação desses MOTARD e JOSEPH(1994).

Deve-se notar que ao adicionar um nível de resolução necessita-se estimar apenas os pesos das funções pertencentes a este nível. Os coeficientes das funções correspondentes às resoluções adicionadas anteriormente, caso existam, não necessitam ser estimados novamente.

2.6 Aproximação de Funções Multivariáveis

A teoria de aproximação de funções utilizando *wavelets* descrita na seção 2.5 considera apenas o caso monovariável. No entanto, esta teoria é válida para funções multivariáveis, permitindo a identificação de processos MIMO (*Multiple-Input Multiple-Output*).

A técnica mais utilizada para desenvolvimento de *wavelets* multidimensionais é o produto tensorial entre *wavelets* e funções de escala unidimensionais DAUBECHIES (1988). Por exemplo, há uma função de escala e três *wavelets* mãe que capturam informações em diferentes orientações no caso de duas entradas:

$$\begin{aligned}\Phi(x_1, x_2) &= \Phi(x_1) \cdot \Phi(x_2) & \Psi^1(x_1, x_2) &= \Phi(x_1) \cdot \Psi(x_2) \\ \Psi^2(x_1, x_2) &= \Psi(x_1) \cdot \Phi(x_2) \\ \Psi^3(x_1, x_2) &= \Psi(x_1) \cdot \Psi(x_2)\end{aligned}$$

A família de *wavelets* de duas variáveis é portanto:

$$\Psi^{h}_{m,n_1,n_2}(x_1, x_2) = 2^m \Psi(2^m x_1 - n_1, 2^m x_2 - n_2); \quad h = \{1, 2, 3\}$$

Uma ilustração da função de escala e das três *wavelets* mãe para a base de Haar (duas variáveis de entrada) é mostrada na Figura 2-6.

O número de *wavelets* aumenta exponencialmente com a dimensão do espaço de entrada. Para N entradas há $2^N - 1$ *wavelets*. Se Φ e ψ constituem uma base ortogonal em $L^2(R)$, o conjunto de funções gerado pelo produto tensorial destas gera uma base para o $L^2(R^N)$ (o espaço N -dimensional das funções integráveis ao quadrado).

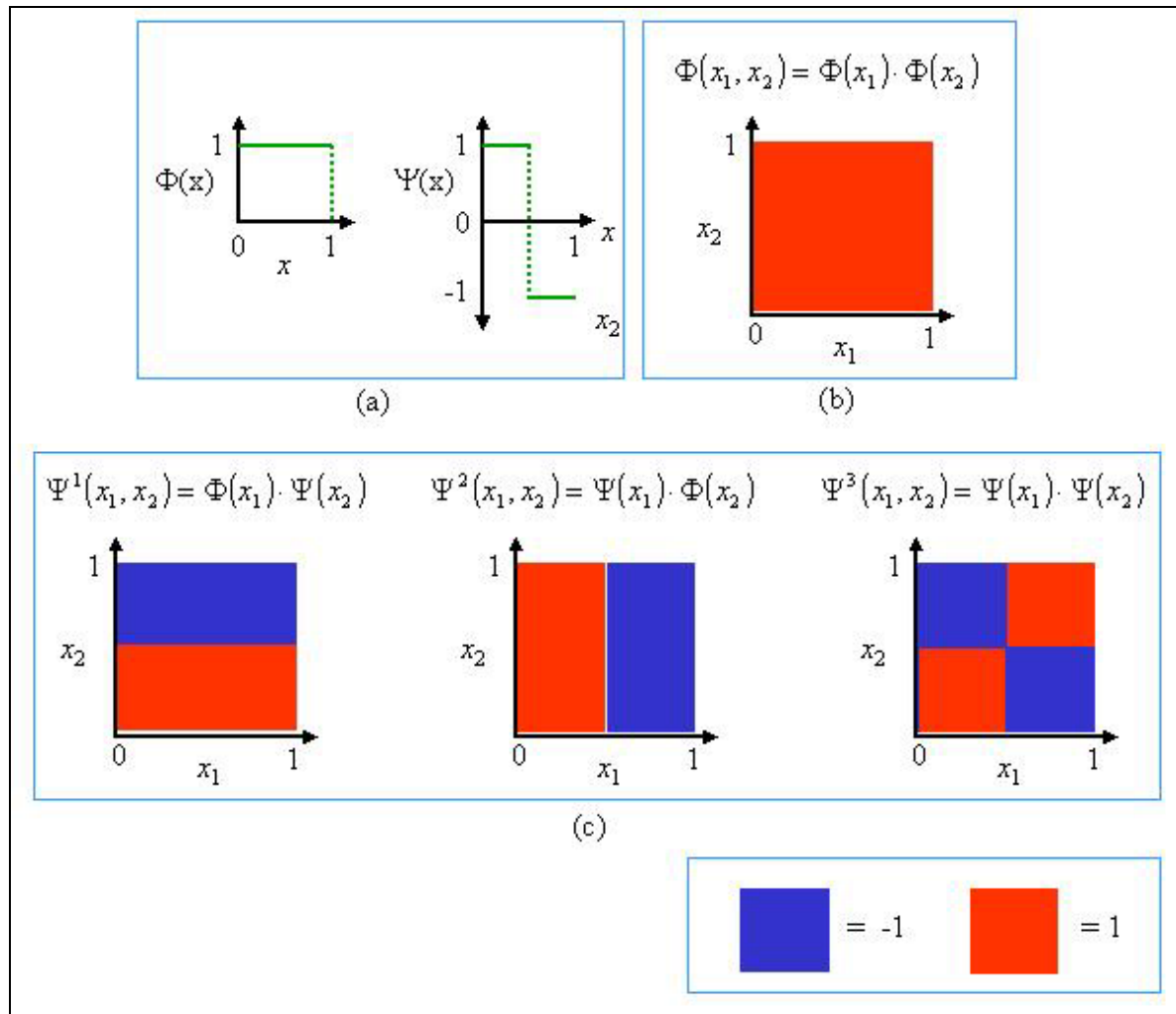


Figura 2-6 - Base de Haar em duas dimensões: (a) função de escala e *wavelet* (uma dimensão); (b) função de escala (duas dimensões); (c) três *wavelets* mãe(duas dimensões)

3 ALTERAÇÕES NA *WAVENET*

Parte das propostas deste trabalho são relativas a modificações na *wavenet*, entre as quais:

1. Multiresolução somente com funções de escala;
2. Escolha da função de ativação adequada para identificação de processos;
3. Determinação do número de funções de escala (primeira resolução).

As propostas têm como objetivo reduzir o número de funções utilizadas na expansão e melhorar a capacidade de generalização.

3.1 Multiresolução Apenas com Funções de Escala

Na *wavenet*, o primeiro nível utilizado na aproximação é constituído de funções de escala e os próximos, se necessários, de *wavelets*. Uma das propostas deste trabalho é fazer multiresolução apenas com funções de escala. Dessa forma, o primeiro nível é igual ao da *wavenet* e, nos próximos níveis, substitui-se as *wavelets* por funções de escala. As razões para esta alteração são:

- 1) No caso de problemas multivariáveis, por exemplo com d entradas, há $2^d - 1$ *wavelets* mãe para apenas uma função de escala. Para ajustar dados utilizando redes de *wavelets*, deve-se determinar um conjunto, em geral pequeno, de neurônios dentre um grupo bem maior de possíveis neurônios definidos pela teoria de *wavelets*. Isto significa que, em um determinado nível de resolução, o número de elementos do conjunto formado apenas por funções de escala é menor por um fator de $(2^d - 1)$, quando comparado ao número de elementos do conjunto contendo *wavelets*.
- 2) Nas *wavenets*, os níveis de resolução crescente de *wavelets* são subespaços ortogonais. Dessa forma, as componentes da função f que se deseja aproximar em determinado subespaço devem ser determinadas com precisão (os pesos devem ser estimados corretamente), pois erros cometidos não podem ser recuperados em outros subespaços, ou seja, pela adição de mais níveis. Este problema não acontece utilizando multiresolução somente com funções de

escala. Se algumas características de f forem perdidas para um determinado subespaço V_i , no próximo nível, correspondente ao subespaço $V_{i+1} = V_i \oplus W_i$, será possível recuperá-las. Há uma redundância entre os subespaços, V_{i+1} e V_i . V_{i+1} contém V_i (o subespaço cujas componentes de f não foram determinadas corretamente) além de W_i .

3) Uma propriedade das *wavelets* é que possuem média nula, o que implica que assumem valores positivos e negativos em seu suporte. Modelos que utilizem *wavelets* podem não apresentar boa capacidade de generalização, já que para uma determinada função de ativação (neste caso *wavelet*) a média dos pontos do grupo de dados que estão contidos em seu suporte é, em geral, não-nula. Dessa forma, pode-se obter um bom ajuste para alguns pontos e um péssimo ajuste para outros. Este problema pode ser agravado em casos onde há regiões pouco amostradas e/ou com amostragem não uniforme. Uma situação bastante comum é que os dados estão geralmente localizados na região central do grupo de dados em detrimento das bordas. Neste caso, modelos construídos a partir de *wavelets* como funções de ativação podem apresentar grandes erros de previsão nas bordas. Tal problema não ocorre com as funções de escala que possuem média não nula.

Matematicamente, uma rede somente com funções de escala pode ser descrita por:

$$f(x) = \sum_{m=0}^{m=M_{\max}} \sum_{n=0}^{n=N_{\phi,m}} d_{m,n} \Phi_{m,n}(x) \quad (3.1)$$

onde

$N_{\phi,m}$: número de funções de escala na resolução m ;

M_{\max} : número de resoluções utilizadas na aproximação;

Para mostrar as diferenças entre a aproximação obtida por uma *wavenet* e a rede que utiliza apenas funções de escala, comparou-se o desempenho dessas na aproximação e reconstrução da superfície $Y = 0,8 \cdot X1 + 0,5 \cdot X2$ utilizando a base de *Haar*. Foram gerados dois grupos de dados com aproximadamente 500 pontos escolhidos aleatoriamente: No primeiro, os valores de $X1$ e $X2$ foram distribuídos de maneira uniforme no espaço das variáveis de entrada. No segundo, os pontos correspondentes à variável $X1$ foram obtidos utilizando distribuição uniforme e os da variável $X2$ pela distribuição gaussiana (Ver Figuras 3.1 e 3.2).

Substituindo os pontos contidos nas Figuras 3.1 e 3.2 na equação do plano, obteve-se a saída Y e foram gerados os grupos de dados. Na aproximação, foram utilizadas redes com as seguintes características: um nível de funções de escala e 4 de *wavelets* para a *wavenet* e 4 resoluções de funções de escala para a rede de funções de escala.

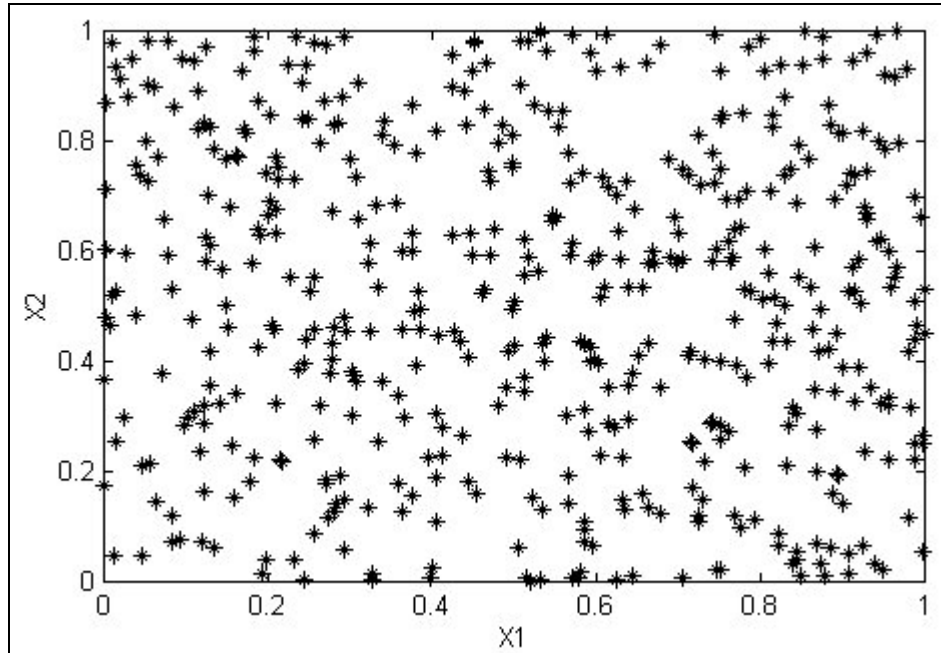


Figura 3-1 - Dados de entrada: X_1 , X_2 possuem distribuição uniforme

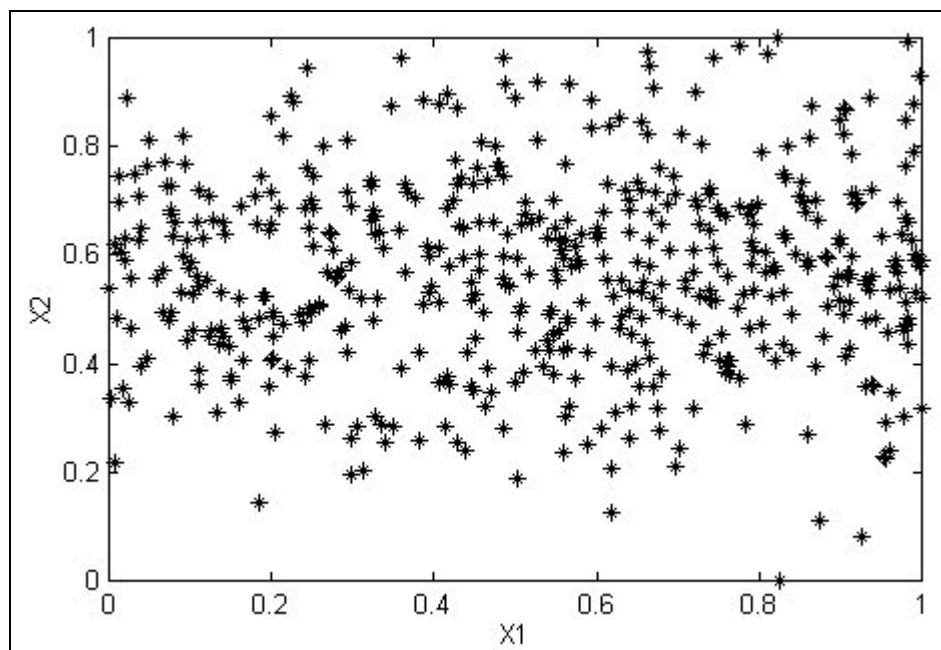


Figura 3-2 - Dados de entrada: X_1 possui distribuição uniforme, X_2 distribuição gaussiana

Na Tabela 3.1 podem ser vistas informações a respeito do treinamento das redes descritas, considerando os dois grupos de dados. Nesta tabela, mostra-se uma comparação do desempenho das redes em relação ao número de neurônios e em relação ao valor do coeficiente de correlação R^2 . O valor de R^2 é definido de acordo com a equação (3.2):

$$R^2 (\%) = \left(1 - \frac{\sum_{i=1}^{N_p} (y_i - y_i^p)^2}{\sum_{i=1}^{N_p} (y_i - \bar{y})^2} \right) \cdot 100 \quad (3.2)$$

onde

N_p : número de pontos do grupo de dados;

y_i : valores de saída para os pontos do grupo de dados ($i=1...N_p$);

y_i^p : valores previstos por um modelo para os pontos de saída do grupo de dados ($i=1...N_p$);

\bar{y} : valor médio dos pontos de saída do grupo de dados.

Na expressão para o cálculo de R^2 , o termo $\sum_{i=1}^{N_p} (y_i - \bar{y})^2$ representa a energia total contida nos dados de saída, ou seja, a energia do sinal de saída.

Tabela 3.1 – Comparação entre desempenho da *wavenet* e da rede de funções de escala

	Nº de neurônios	Grupo de dados 1		Grupo de dados 2	
		R^2	$100-R^2$	R^2	$100-R^2$
R. funções de escala	85	99,9810	0,0190	99,9269	0,0731
<i>Wavenet</i>	256	99,9501	0,0499	99,8296	0,1704

Analisando a Tabela 3.1, nota-se que o valor de $100-R^2$, que indica a porcentagem de energia do sinal de saída não identificada por um modelo, é menor no caso da rede de funções de escala, apesar dessa utilizar um número menor de neurônios que a *wavenet*.

Nas Figuras 3.3 e 3.4 mostram-se as superfícies de previsão obtidas a partir das redes neurais treinadas para o primeiro grupo de dados. Para obter as superfícies, foi efetuada uma interpolação dos valores previstos pelas redes em uma malha de passo 0,01 (1% do domínio de cada variável). Nas Figuras 3.5 e 3.6 mostram-se as superfícies de previsão obtidas das redes neurais treinadas para o segundo grupo de dados.

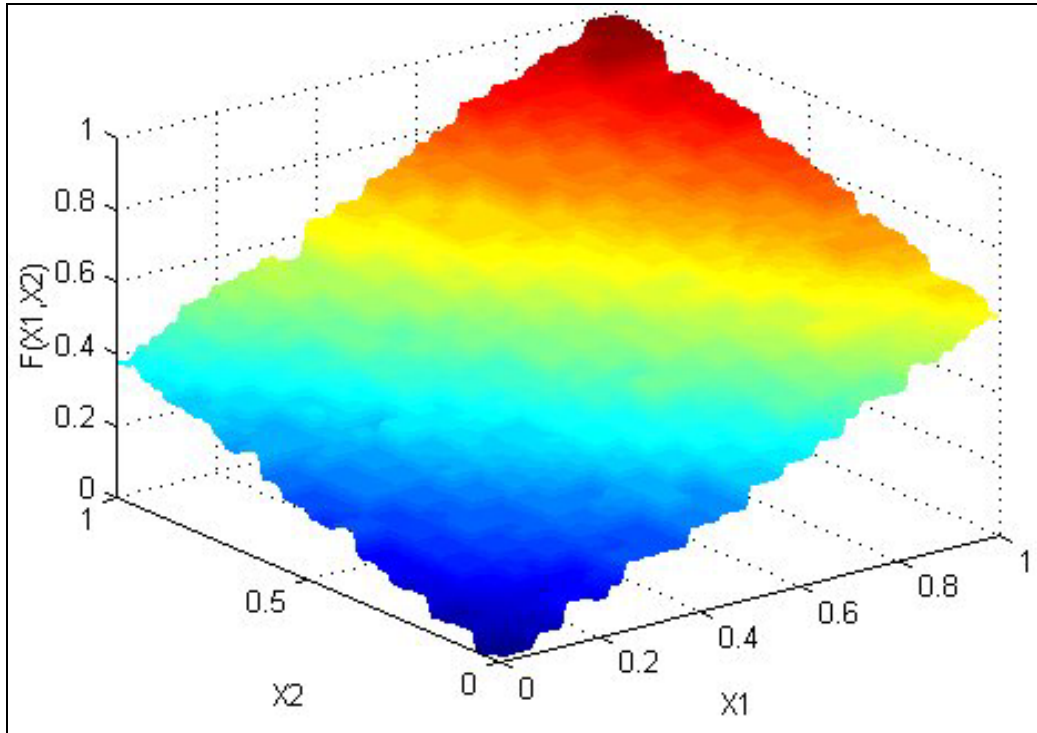


Figura 3-3 - Superfície de previsão: rede de funções de escala (primeiro grupo de dados)

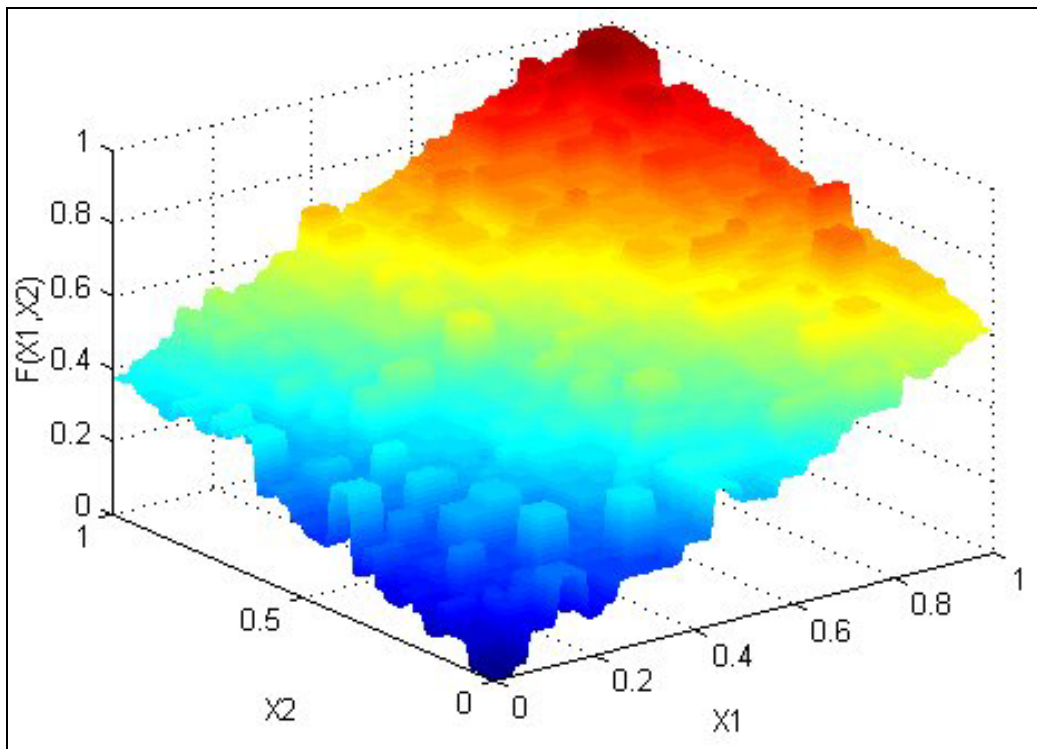


Figura 3-4 - Superfície de previsão: *wavenet* (primeiro grupo de dados)

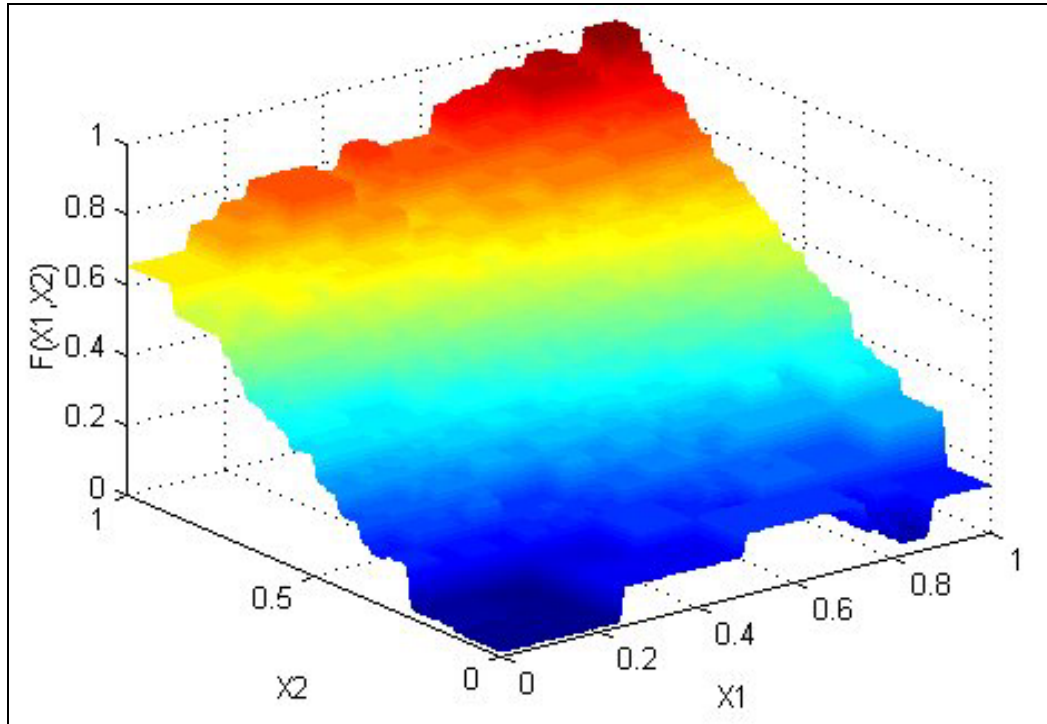


Figura 3-5 - Superfície de previsão: rede de funções de escala (segundo grupo de dados)

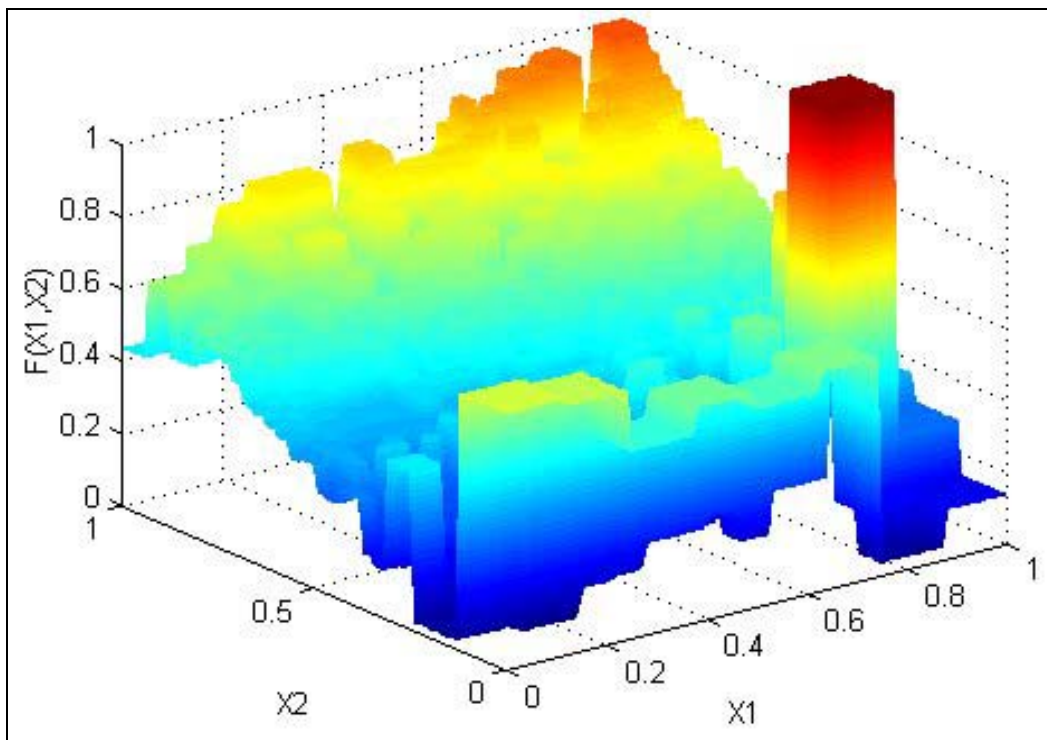


Figura 3-6 - Superfície de previsão: *wavenet* (segundo grupo de dados)

Nas Figuras 3.3 a 3.6 destaca-se a melhor capacidade de generalização da rede de funções de escala. A diferença entre a capacidade de generalização da *wavenet* e da rede de funções de escala foi maior no caso do segundo grupo de dados, que apresenta amostragem deficiente nas bordas em relação à variável X_2 .

3.2 Determinação de uma Função de Ativação Adequada para Identificação de Processos

Conforme descrito no Capítulo 2, é possível gerar funções de escala e *wavelets* dependendo dos coeficientes utilizados na Equação (2.4). Para definir uma função de escala adequada para identificação e controle foram utilizados os seguintes critérios:

Em relação ao processo que se deseja modelar, tem-se a seguinte restrição:

1. **Ausência de Descontinuidades:** Descontinuidades são características pouco comuns em processos. Dessa forma, as redes neurais utilizadas para modelar processos deveriam utilizar funções contínuas até pelo menos a primeira derivada. Isto garante a continuidade do modelo eliminando arestas da superfície de aproximação desse e outros comportamentos pouco prováveis de existirem em processos reais.

Qualquer função de escala escolhida deveria ter as seguintes características:

2. **Diferenciabilidade:** Esta propriedade está diretamente relacionada com o item (1) Ausência de Descontinuidades. Porém, há outra razão para a exigência de continuidade da função de ativação. Alguns algoritmos exigem a diferenciabilidade das funções de ativação. Não basta a rede ser linear nos parâmetros (fácil diferenciabilidade em relação aos parâmetros), o que permite o treinamento por mínimos quadrados. Em algumas aplicações é necessário diferenciar a rede com relação às suas entradas. Um exemplo é a utilização de modelos neurais em controladores preditivos. Para esta família de controladores, fixa-se uma referência (saída) e deseja-se descobrir, através de um algoritmo de otimização, as ações de controle (entradas da rede) que levam o sistema o mais próximo possível da referência. Não há solução analítica para o problema descrito e uma forma de resolvê-lo é utilizar algoritmos baseados em Cálculo. Estes algoritmos exigem que a função de ativação seja diferenciável.

3. **Pequena Intersecção:** Da teoria de *wavelets*, sabe-se que o tamanho do conjunto de neurônios utilizados na aproximação aumenta exponencialmente com o número de entradas e o nível de resolução. Na prática, só alguns neurônios, selecionados dentre todos os possíveis, são necessários no ajuste. Uma forma de diminuir o tamanho do conjunto é escolher uma *wavelet* mãe que resulte em uma família de *wavelets* com pequena intersecção entre neurônios adjacentes. Uma pequena intersecção implica que serão colocadas poucas *wavelets* em um determinado domínio, o que diminuirá a taxa de crescimento do conjunto.
4. **Número Reduzido de extremos:** Um bom modelo para identificação deveria ajustar bem os dados de treinamento (pequeno erro quadrático de aproximação) e ter boa capacidade de generalização, ou seja, deve apresentar um comportamento sem oscilações (sem excessivas curvaturas) mesmo que se tenha amostragem deficiente. De acordo com o princípio de Occam CHEN e HAYKIN(2001), POGGIO e GIROSI(1990), na ausência de qualquer conhecimento prévio, a superfície de um modelo empírico deveria ser a mais simples e suave possível. Em geral, regiões de maior curvatura estão associadas à presença de mínimos locais e, portanto, a utilização de uma função de ativação com vários extremos pode gerar modelos com comportamentos não suaves em regiões deficientes de dados. No caso multivariável, o problema é agravado pelo fato de que as funções de ativação são obtidas através do produto de funções unidimensionais e o número de extremos cresce rapidamente com o número de entradas.

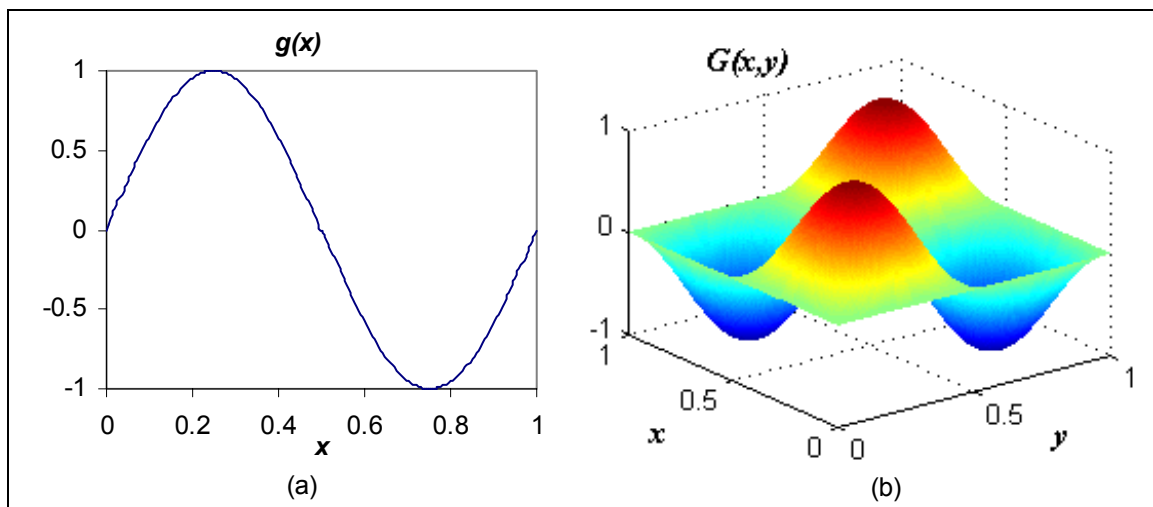


Figura 3-7 - Aumento do número de extremos conforme cresce a dimensão do espaço de entrada

Na Figura 3-7, mostra-se um exemplo do aumento do número de extremos quando cresce a dimensão do espaço de entrada. Neste exemplo $g(x)$ possui dois extremos e $G(x,y)=g(x) \cdot g(y)$ possui quatro.

Analisando vários tipos de funções de escala em relação aos critérios descritos conclui-se que a *spline* quadrática é uma função adequada para objetivos de controle e identificação. Algumas razões para esta escolha podem ser enumeradas:

1. As derivadas de primeira e segunda ordem da *spline* podem ser facilmente calculadas e originam expressões mais simples do que a *spline* original. A curva da derivada de primeira ordem é linear por partes e a de segunda constante por partes (Ver Figura 3-8).

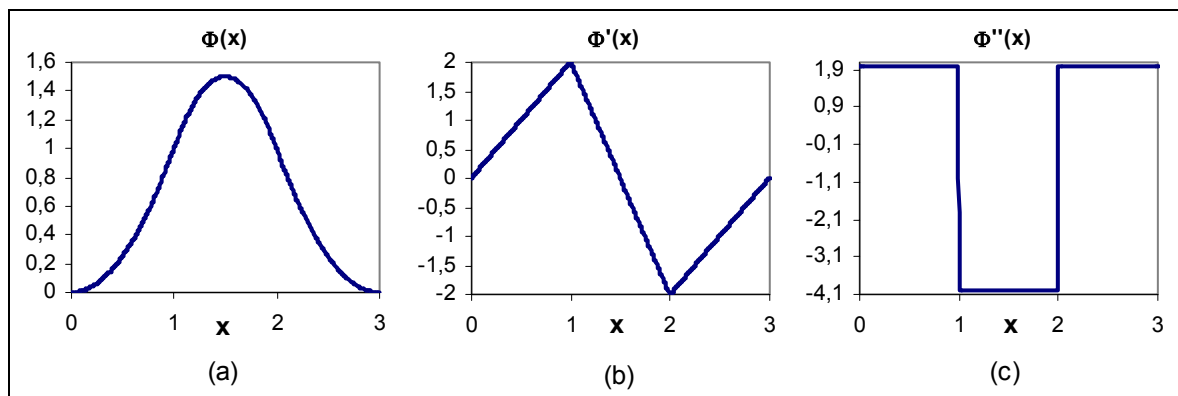


Figura 3-8 - (a) função de escala *spline*; (b) derivada de primeira ordem; (c) derivada de segunda ordem

Na Figura 3-8 pode-se observar que a derivada de segunda ordem não é definida nos pontos da curva onde há transição de uma região de valor constante para outra. Isto não representa problema já que se tratam de descontinuidades isoladas e pode-se definir o valor da segunda derivada nestes pontos de forma arbitrária.

2. Duas funções adjacentes da família de *splines* quadráticas têm interseção correspondente a $2/3$ de seu suporte (considerando o caso unidimensional), a menor possível desconsiderando as base de *Haar* (sem interseção) e a *Hat* (interseção de $1/2$ do suporte) que não satisfazem à restrição de continuidade até a primeira derivada. Para o caso multivariável, as frações de interseção modificam, mas a validade do que foi exposto permanece inalterada.

3. A *spline* quadrática possui apenas um extremo.

3.3 Determinação do Número Inicial de Funções de Escala (Primeira Resolução)

O primeiro passo para aproximar uma função em um determinado domínio é normalizar os dados de entrada e saída; por exemplo, no intervalo $[0;1]$. Em seguida, fixa-se a função de escala. A aproximação dos dados de treinamento inicia, no contexto multiresolução, a partir de uma combinação linear de funções de escala deslocadas. No caso da *spline*, o deslocamento é de $1/3$ do suporte quando se consideram as adjacentes, o que equivale a uma interseção de $2/3$ do suporte. Para a geração da família de *splines* quadráticas, a partir da função Φ correspondente, a *spline* é convenientemente dividida em três regiões conforme mostrado na Figura 3-9.

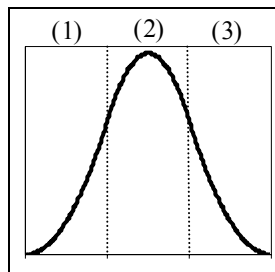


Figura 3-9 - *Spline* dividida em 3 regiões

Pode-se definir o número inicial de funções de escala de forma arbitrária. Para exemplificar, utilizando o caso unidimensional, podem ser vistas duas expansões possíveis: Uma com 8 (Figura 3-10) funções e outra com 5 (Figura 3-11) (Os números 1, 2 e 3 são relativos às regiões demarcadas na Figura 3-9). Deve-se notar que, nos dois casos, várias funções de escala (deslocadas em $1/3$ de seu suporte) estão contidas no intervalo $[0;1]$ (o mesmo que o utilizado para normalizar o grupo de dados). As funções de escala localizadas nas extremidades devem ser truncadas em razão do domínio finito da curva que se deseja aproximar.

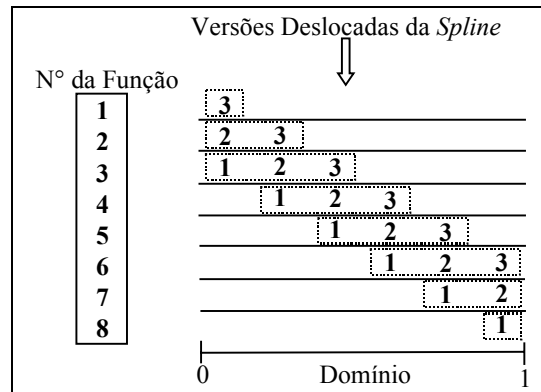


Figura 3-10 - Expansão contendo 8 funções de escala

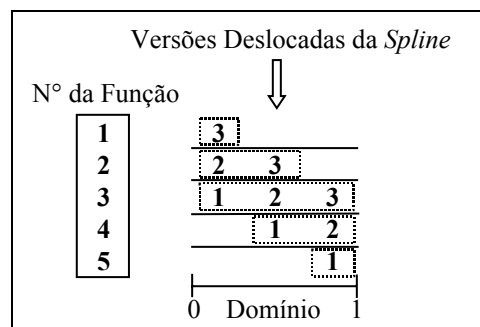


Figura 3-11 - Expansão contendo 5 funções de escala

Uma das propostas deste trabalho é definir qual o número de funções de escala que deve ser utilizado na primeira resolução. Se a expressão analítica da função que se deseja aproximar é conhecida, então não interessa o ponto de partida. No entanto, quando se dispõe apenas de um conjunto finito de pontos, a expansão que inicia com o menor número de funções de escala fornece, possivelmente, modelos com melhor capacidade de generalização. Algumas razões para isto são:

- 1) Compromisso bias-variância: Se a curva que se deseja aproximar tiver não-linearidades pouco acentuadas, então ela será certamente aproximada utilizando somente o primeiro nível de funções de escala, já que um comportamento próximo do linear está contido no espaço das funções geradas por expansões de *splines* (as *splines* são combinações de polinômios de baixa ordem e o comportamento linear é um caso particular). Dessa forma, o modelo com menor número de parâmetros é preferido em razão do compromisso Bias-Variância JOHANSEN(1997).

2) Erros na estimação dos pesos:

2.1) Característica localizada das funções de escala: Para um problema com d entradas, o suporte de cada função multidimensional é obtido pela interseção dos suportes de d funções unidimensionais localizadas. Isso implica na diminuição do suporte das funções de escala, quando comparado ao domínio de interesse do problema que é definido pelos dados de treinamento, à medida que aumenta a dimensão do espaço de entrada. Dessa forma, pode haver neurônios com muito poucos pontos em seu suporte, levando a problemas numéricos no treinamento, principalmente para casos com amostragem deficiente de dados e que apresentam ruído. Na expansão utilizando 8 funções de escala em cada entrada (com um total de 8^d , levando em conta todas as combinações), os neurônios são mais localizados que o caso com 5, sendo que maiores erros no cálculo dos pesos devem acontecer no primeiro caso.

2.2) Excesso de neurônios: Considerando um problema com d entradas, há em um caso 8^d neurônios e, portanto, o mesmo número de pesos deve ser estimado. No outro caso há 5^d . Com certeza, nesse último haverá menos problemas numéricos.

Analisando o que foi descrito, nota-se que há uma tendência para produzir melhores modelos à medida que se diminui o número de funções de escala utilizadas no primeiro nível de resolução. Dessa forma, conclui-se que a melhor expansão possui apenas 3 neurônios (o menor número possível, como se observa nas Figuras 3.10 e 3.11). A expansão com apenas 3 neurônios (cada uma das regiões da Figura 3-9) minimiza os problemas descritos.

Deve-se notar que a primeira resolução passa a ser formada por funções de escopo global, ou seja, o suporte dessas é igual ao domínio do problema. Neste caso, a rede neural, com as modificações propostas neste trabalho, pode ser caracterizada como um aproximador que utiliza funções de escopo global e localizadas. As primeiras são inicialmente incorporadas (primeira resolução) e como são muito suaves garantem boa capacidade de generalização ao modelo neural. Com as funções localizadas, incorporadas nas próximas resoluções, é possível aproximar regiões específicas do problema.

4 REDES NEURAS PARA IDENTIFICAÇÃO DE SISTEMAS

Os modelos dinâmicos para sistemas não-lineares podem ser descritos, em geral, como séries temporais não-lineares. O nome genérico dado a estes modelos é *NARMAX* (*Nonlinear Auto Regressive Moving Average with EXogenous Inputs*). Como casos particulares dos modelos *NARMAX*, podem-se citar as redes neurais, os modelos lineares *ARMAX*, muito utilizados em identificação, e *ARIMAX* para controle LJUNG(1987).

Grande parte dos trabalhos encontrados na literatura relativos a identificação e controle não-linear utilizam as redes *feedforward* e as *RBFN* DE JESUS *et. al.*(2001), LING(2000) SAXENA *et. al.*(2000), HUNT *et. al.*(1992). Neste capítulo descreve-se de forma simplificada a estrutura destes modelos empíricos, sendo que no Capítulo 7 é apresentado um estudo comparativo entre o desempenho das redes *feedforward*, *RBFN* e de funções de escala.

4.1 Redes Neurais Utilizadas em Identificação de Sistemas

A maior parte dos trabalhos relativos à identificação de sistemas por redes neurais estáticas utiliza redes *feedforward* e de base radial. Estas são descritas a seguir:

1) Redes de *feedforward*:

Redes *feedforward* correspondem a séries de séries de funções. Tal arquitetura caracteriza-se por um processamento de informações dividido em camadas. Conforme o teorema de Kolmogorov para redes neurais FAUSETT(1994), é possível aproximar qualquer função contínua com redes compostas de três camadas: entrada, apenas uma intermediária e saída. Neste caso, as redes *feedforward*, considerando por simplicidade apenas uma saída e uma camada intermediária, podem ser descritas como:

$$z = G\left(\sum_{i=1}^N w_i \cdot G\left(\sum_{j=1}^M \alpha_{i,j} \cdot x_j + b_i\right) + b\right)$$

onde

N , M : número de neurônios da camada intermediária e de variáveis de entrada, respectivamente;

z : saída prevista;

w_i e $\alpha_{i,j}$: pesos da camada de saída e entrada, respectivamente ($i=1,\dots,N$) ($j=1,\dots,M$).

b_i e b : bias da camada intermediária ($i=1,\dots,N$) e da camada de saída;

G : mapeamento não-linear. Alguns dos mais utilizados são a função sigmóide e a tangente hiperbólica;

$X = [x_1 \ x_2 \ \dots \ x_M]^T$: vetor de variáveis de entrada;

O treinamento de redes *feedforward* é iterativo, uma vez que os pesos da(s) camada(s) intermediária(s) são relacionados de forma não-linear à saída.

2) Redes de base radial

Redes de base radial podem ser descritas, considerando uma saída, por:

$$z = \sum_{i=1}^N w_i \cdot G(\|X - C_i\|^2)$$

sendo que $\|X - C_i\|^2 = \sum_{j=1}^M (x_j - c_{i,j})^2$

onde

N , M : número de centros e variáveis de entrada, respectivamente;

z : saída;

w_i : pesos da camada de saída ($i=1,\dots,N$).

G : mapeamento não-linear. Alguns dos mais utilizados são:

Função gaussiana = $\exp(-\frac{r^2}{\sigma^2})$, com $\sigma > 0$ e $r \in \mathbf{R}$;

Multiquadrática = $(r^2 + c^2)^{1/2}$, com $c > 0$ e $r \in \mathbf{R}$;;

Inversa Multiquadrática = $\frac{1}{(r^2 + c^2)^{1/2}}$, com $c > 0$ e $r \in \mathbf{R}$;;

$\|\cdot\|$: Norma quadrática;

$X = [x_1 \ x_2 \ \dots \ x_M]^T$: vetor de variáveis de entrada;

$C_i = [c_{i,1} \ c_{i,2} \ \dots \ c_{i,M}]^T$: Vetor de centros para a i -ésimo neurônio ($i = 1, \dots, N$).

Os centros devem ser determinados em uma etapa anterior ao treinamento. Estes podem ser obtidos aleatoriamente entre os pontos do grupo de dados ou através de técnicas de *clusterização*, tal como o *K-means* HARRIS(1994). Uma vez determinados os centros, os pesos da camada de saída podem ser facilmente calculados pela técnica dos mínimos quadrados.

As redes estáticas podem ser utilizadas na identificação de sistemas dinâmicos considerando as variáveis de entrada como atrasos das saídas, além dos valores das perturbações externas. Como um exemplo, para identificar o sistema dinâmico $y_{(k+1)} = f(y_{(k)}, u_{(k)})$ basta definir $z = y_{(k+1)}$ e $X = [y_{(k)}, u_{(k)}]$.

5 ESTUDO DE MÉTODOS DE TREINAMENTO DE REDES *FEEDFORWARD*

No capítulo 7, é apresentada uma comparação de desempenho das redes de tipo *feedforward*, base radial e de funções de escala. As duas últimas podem ser treinadas por mínimos quadrados e, portanto, os pesos são estimados de forma ótima. No entanto, a primeira necessita de um algoritmo iterativo e os pesos ótimos nem sempre são obtidos. Para fornecer uma comparação justa das redes descritas, em se tratando de desempenho de modelagem, este capítulo tem por objetivo fazer um estudo de algumas estratégias de treinamento supervisionado para redes tipo *feedforward*. O desempenho de um método que utiliza apenas as derivadas de primeira ordem da função objetivo é comparado a um método de segunda ordem, na identificação de um tanque de *pH*. Também estes foram comparados a uma simples metodologia de treinamento baseada em mínimos quadrados e algoritmos genéticos.

Neste capítulo, o desempenho entre algoritmos de treinamento foi comparado utilizando apenas os dados de treinamento. O método para o qual foram obtidos os melhores resultados, em termos do valor da função objetivo, foi utilizado como estratégia de treinamento de redes *feedforward* nas simulações descritas no capítulo 7.

5.1 Descrição do Método de Treinamento da Retropropagação do Erro

Redes *feedforward* têm sido aplicadas com sucesso para resolver uma ampla gama de problemas, utilizando o algoritmo da retropropagação do erro como método de treinamento supervisionado. Este é denominado popularmente de *backpropagation*.

Basicamente, o *backpropagation* consiste de dois passos aplicados seqüencialmente até a convergência: um no sentido das entradas para as saídas e outro em sentido contrário. Na primeira etapa, um ou mais padrões são aplicados na(s) entrada(s) da rede e seu efeito é propagado camada por camada até gerar as saídas da rede. Durante esta fase os pesos permanecem fixos. Na segunda etapa, os pesos devem ser ajustados de acordo com uma correção baseada no erro de aproximação. O erro é propagado através da rede no sentido das saídas para as entradas, sendo que os pesos são ajustados para que a previsão da rede seja a mais próxima possível dos alvos, em termos de erro quadrático.

O *backpropagation* é um método de primeira ordem, ou seja, utiliza somente a informação das derivadas de primeira ordem do erro quadrático de aproximação para correção dos pesos. Os pesos podem ser corrigidos por padrão ou por época (apresentação de todos os padrões do grupo de dados de treinamento). No primeiro caso, a função objetivo de treinamento é o erro quadrático de um padrão e no segundo, o erro quadrático de todo o grupo de dados. A atualização dos pesos por época fornece um método mais robusto, sendo mais apropriada em aplicações *off-line*, enquanto a correção por padrão é adequada em aplicações *on-line*.

A equação de correção dos pesos pode ser descrita genericamente por:

$$\Delta \mathbf{W}(n) = -\eta \frac{\partial E^2}{\partial \mathbf{W}}(n) \quad (5.1)$$

onde

\mathbf{W} : Vetor contendo todos os pesos da rede;

E^2 : Erro quadrático de aproximação;

n : Iteração. Pode ser padrão ou por época, depende de como os pesos são atualizados;

η : Fator de aprendizagem.

A derivação de expressões para o cálculo de $\frac{\partial E^2}{\partial \mathbf{W}}$ em função das entradas, ativações e saídas da rede podem ser encontrada em ZUPAN e GASTEIGER(1993).

Comumente, se adiciona na Equação (5.1) um termo de momento para melhorar as propriedades de convergência do método. Neste caso, a equação para correção dos pesos torna-se:

$$\Delta \mathbf{W}(n) = -\eta \cdot \frac{\partial E^2}{\partial \mathbf{W}}(n) + \alpha \cdot \Delta \mathbf{W}(n-1)$$

onde α é a taxa de momento.

5.2 Descrição do Método de Treinamento Levenberg-Marquardt

No caso do *backpropagation*, a correção dos pesos é proporcional ao gradiente do erro quadrático de aproximação. A grande vantagem desta abordagem é a simplicidade de implementação, no entanto, o *backpropagation* apresenta baixa taxa de convergência.

Para melhorar a convergência de treinamento em redes *feedforward*, deve-se incorporar informações de derivadas de segunda ordem. Isto pode ser obtido utilizando uma aproximação quadrática para a superfície da função objetivo em torno de cada ponto de linearização $\mathbf{W}(n)$. A correção ótima para os pesos considerando a aproximação de segunda ordem (Ver HAYKIN(1999)) é dada por:

$$\Delta \mathbf{W}(n) = -\mathbf{H}^{-1}(n) \cdot \frac{\partial E^2}{\partial \mathbf{W}}(n) \quad (5.2)$$

onde $\mathbf{H}(n) = \frac{\partial^2 E^2}{\partial \mathbf{W}^2}(n)$ é a matriz quadrada de derivadas parciais de segunda ordem do erro quadrático de aproximação em relação aos pesos. Esta matriz é usualmente denominada de Hessiana.

A Equação (5.2) corresponde ao método de Newton utilizado em otimização sem restrições. Na prática, esta estratégia de treinamento não é utilizada diretamente em função da grande complexidade, do elevado tempo computacional, ou mesmo impossibilidade da estimação das derivadas de segunda ordem. Existem algumas variações do método de Newton que utilizam a aproximação quadrática da função objetivo sem calcular as derivadas de segunda ordem. Citam-se os métodos de Quasi-Newton, Gradiente Conjugado, Gauss-Newton e Levenberg-Marquardt entre outros descritos em BAZARAA *et al.*(1993), sendo que aplicações da técnica de Levenberg-Marquardt são mais comumente encontradas na literatura. Este método aproxima a matriz H utilizando apenas o conhecimento de derivadas de primeira ordem, como descrito a seguir. Seja $\frac{\partial E_i(n)}{\partial \mathbf{W}}$ o vetor de derivadas parciais do erro em relação aos pesos para o i -ésimo padrão e n -ésima iteração (considerando que a atualização dos pesos é realizada em cada época). Então se pode definir a matriz $\mathbf{J}(n)$ de dimensão N_p (Número de padrões) \times N_w (Número de pesos da rede).

$$\mathbf{J}(n) = \begin{bmatrix} \frac{\partial E_1}{\partial \mathbf{W}}(n) \Lambda & \frac{\partial E_i}{\partial \mathbf{W}}(n) \Lambda & \frac{\partial E_{N_p}}{\partial \mathbf{W}}(n) \end{bmatrix}^T$$

Pode-se observar que $\mathbf{J}(n)$ é a matriz Jacobiana do vetor de erros;

A aproximação para a matriz \mathbf{H} é calculada como: $\mathbf{H}(n) = \mathbf{J}(n)^T \mathbf{J}(n)$.

A matriz é sempre semidefinida-positiva. Para garantir que esta aproximação seja não singular (neste caso definida positiva), a matriz $\mathbf{J}(n)$ deve ter posto completo, ou seja, seu posto deve ser igual ao número de pesos (considerando que o número de padrões é maior que o número de pesos). Infelizmente, não há como garantir que esta condição seja satisfeita. Para impedir que $\mathbf{J}(n)^T \mathbf{J}(n)$ torne-se singular, é comum adicionar a essa a matriz identidade multiplicada por uma pequena constante positiva δ . Dessa forma, a Equação (5.2), para correção dos pesos, torna-se:

$$\Delta \mathbf{W}(n) = (\mathbf{J}(n)^T \mathbf{J}(n) + \delta \cdot \mathbf{I})^{-1} \cdot \frac{\partial E^2}{\partial \mathbf{W}}(n) \quad (5.3)$$

A aplicação do método de Levenberg-Marquardt, de forma semelhante ao *backpropagation*, pode ser dividida em dois passos que devem ser utilizados sequencialmente até a convergência. No primeiro passo, calculam-se as ativações e saídas de todos os neurônios para o grupo de dados de treinamento (uma época). Durante esta fase os pesos permanecem fixos. Com estas informações calcula-se as derivadas parciais em relações aos pesos para todos os padrões. No segundo passo, os pesos devem ser ajustados de acordo com a Equação (5.3). Deve-se notar que um sistema linear tem que ser resolvido a cada época para o cálculo das correções. O método de Levenberg-Marquardt também utiliza uma estratégia para corrigir o valor δ durante o treinamento, garantindo que o erro quadrático global de aproximação sempre diminua (Ver BAZARAA *et. al.*(1993)). No capítulo 6 descrevem-se outras razões para a utilização do parâmetro δ , além de melhorar o condicionamento da matriz $\mathbf{J}(n)^T \mathbf{J}(n)$.

Alguns trabalhos na literatura mostram que é possível separar e estimar os pesos da rede em grupos sem diferenças significativas no valor do erro global de aproximação, quando comparados ao caso em que não se efetua nenhuma divisão. A vantagem é que sistemas lineares de menor porte devem ser resolvidos em cada época. No trabalho de SJOBERG e VIBERG(1997) foi mostrado que no caso de uma rede *feedforward*, treinada com o método de Levenberg-Marquardt, melhores resultados podiam ser obtidos, em termos do erro de aproximação, quando os pesos da camada intermediária e de saída eram estimados separadamente. A rede utilizada era composta de apenas uma camada intermediária e somente neurônios lineares na saída. As etapas de cálculo das saídas e estimação dos pesos foram efetuadas sequencialmente e separadamente para cada camada. Para a rede descrita, estimar

os pesos da camada de saída aplicando o método de Levenberg-Marquardt equivale a resolver o problema de mínimos quadrados, uma vez que essa camada é linear em relação aos pesos.

5.3 Comparação de Desempenho entre um Método de Primeira e Segunda Ordem

Uma comparação entre o desempenho de um método de primeira ordem (*backpropagation*) e um método de segunda ordem (Levenberg-Marquardt) foi efetuada na identificação de um tanque de *pH*. Para o *backpropagation*, foi efetuado um estudo prévio para descobrir o valor do fator de aprendizagem (η) e da taxa de momento (α) que resultavam em um melhor desempenho de treinamento. Após alguns testes obteve-se $\eta = 0,01$ e $\alpha = 0,9$. No caso do método de Levenberg-Marquardt, o parâmetro δ é alterado de forma adaptativa pelo próprio método, não necessitando ajustes por parte do usuário. Para este último, os pesos da camada intermediária e de saída foram treinados separadamente conforme descrito na seção 5.2.

O modelo empírico utilizado na identificação do sistema dinâmico foi uma rede neural *feedforward* com apenas uma camada intermediária (composta de neurônios tendo a tangente hiperbólica como função de ativação) e a camada de saída linear. O sistema escolhido para comparação foi um processo SISO, que envolve o comportamento dinâmico do *pH* ocorrendo em um reator de neutralização. O modelo matemático do processo foi extraído do trabalho de CHENG *et al.*(1995), sendo que considerou-se que o nível do tanque constante e que não há atraso na medida do *pH*. O modelo simplificado pode ser colocado no formato de variáveis de estado, sendo descrito por dois estados (invariantes reacionais) e uma saída (*pH*). O processo possui três correntes de entrada, ácido (HNO_3), tampão (NaHCO_3) e base (NaOH), que são misturadas no tanque e o *pH* na saída desse é medido. Destas correntes apenas a vazão de base é alterada. As equações diferenciais relacionadas aos estados são lineares e o *pH* é uma função não-linear desses.

$$(Ah) \frac{d(W_{a4})}{dt} = q_1 \cdot (W_{a1} - W_{a4}) + q_2 \cdot (W_{a2} - W_{a4}) + q_3 \cdot (W_{a3} - W_{a4}) \quad (5.4)$$

$$(Ah) \frac{d(W_{b4})}{dt} = q_1 \cdot (W_{b1} - W_{b4}) + q_2 \cdot (W_{b2} - W_{b4}) + q_3 \cdot (W_{b3} - W_{b4}) \quad (5.5)$$

$$W_{a4} + 10^{pH4-14} + W_{b4} \cdot \frac{1 + 2 \cdot 10^{pH4-pK2}}{1 + 10^{pK1-pH4} + 10^{pK1-pK2}} - 10^{-pH4} = 0 \quad (5.6)$$

onde

q_1, q_2 e q_3 : Vazões de ácido, tampão e base, respectivamente;

A e h : Área e altura do tanque, respectivamente;

W_{a1} e W_{b1} : Invariantes reacionais da corrente de ácido;

W_{a2} e W_{b2} : Invariantes reacionais da corrente de tampão;

W_{a3} e W_{b3} : Invariantes reacionais da corrente de base;

W_{a4} e W_{b4} : Invariantes reacionais na saída do tanque;

pK_1, pK_2 : constantes de dissociação.

As quantidades W_a e W_b são chamadas de invariantes reacionais porque suas concentrações não se alteram durante a reação e, portanto, as respectivas equações diferenciais não contêm o termo de reação. As condições nominais do processo podem ser vistas na Tabela 5.1.

Tabela 5.1 - Variáveis e parâmetros do processo de neutralização

Variável	Valores nominais
A	207 cm ²
h	14,0 cm
pK_1	6,35
pK_2	10,25
q_1 (HNO ₃)	16,6 ml/s
q_2 (NaHCO ₃)	0,55 ml/s
$q_3^{(*)}$ (NaOH)	15,6 ml/s
$pH^{(*)}$	7,0
W_{a1}	3e-3 M
W_{a2}	-3e-2 M
W_{a3}	3,05e-3 M
W_{b1}	0
W_{b2}	3e-2 M
W_{b3}	5e-5 M

(*): Valores de estado estacionário

As Equações (5.4) e (5.5) permitem calcular as concentrações dos invariantes reacionais em função do tempo, assumindo que as concentrações iniciais e as vazões sejam conhecidas. O pH da solução pode então ser determinado da Equação (5.6) a partir dos valores das concentrações dos invariantes reacionais. A Equação (5.6) é transcendental, ou seja, o valor do pH não pode ser obtido diretamente, no entanto, pode ser determinado iterativamente através de uma técnica para determinação de raízes. Neste trabalho, utilizou-se o método de Newton para raízes.

O grupo de dados utilizado para treinamento da rede (em torno de 1200 pontos) foi obtido amostrando o modelo do tanque de pH a cada 5 segundos (valor utilizado no trabalho de CHENG *et al.*(1995)). As perturbações aplicadas na vazão de base foram pulsos aleatórios com amplitude entre 7 e 21 ml/s e largura de 10 e 50 intervalos de amostragem em torno do valor de estado estacionário $q = 15,6$ ml/s. Foram adicionados aos pulsos ruído branco com amplitude de 5% da utilizada para geração desses.

Para treinamento da rede, os valores de $pH_{(k)}$ e $U_{(k)}$ foram utilizados como variáveis de entrada e como saída, o valor de $pH_{(k+1)}$. Os dados de entrada e saída utilizados no treinamento foram normalizados entre $-0,8$ e $0,8$. A camada intermediária da rede *feedforward* utilizada foi composta por 30 neurônios. Este número de neurônios foi escolhido, pois permitiu uma comparação de desempenho entre os métodos de treinamento e se obteve boa aproximação dos dados. Na Figura 5-1 mostram-se as perturbações aplicadas na vazão de base. Na Figura 5-2 podem ser vistos os valores de pH para as perturbações aplicadas.

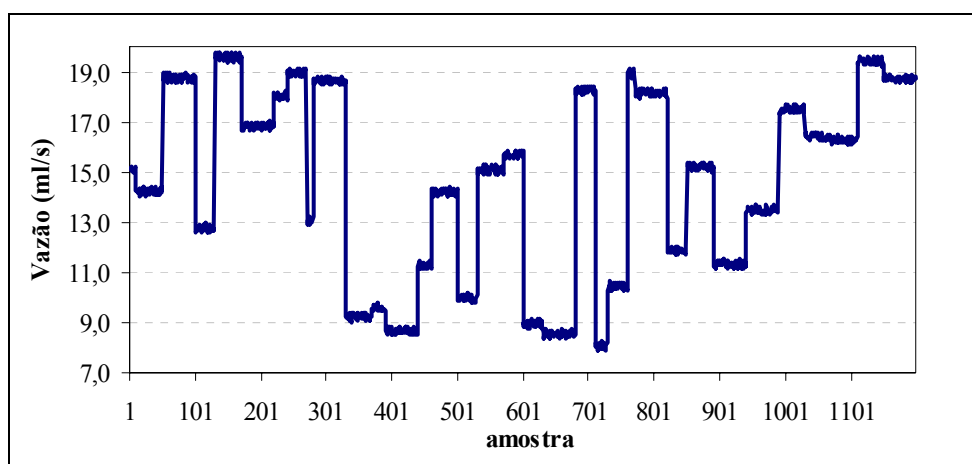


Figura 5-1 - Perturbações aplicadas na vazão de base do tanque de pH

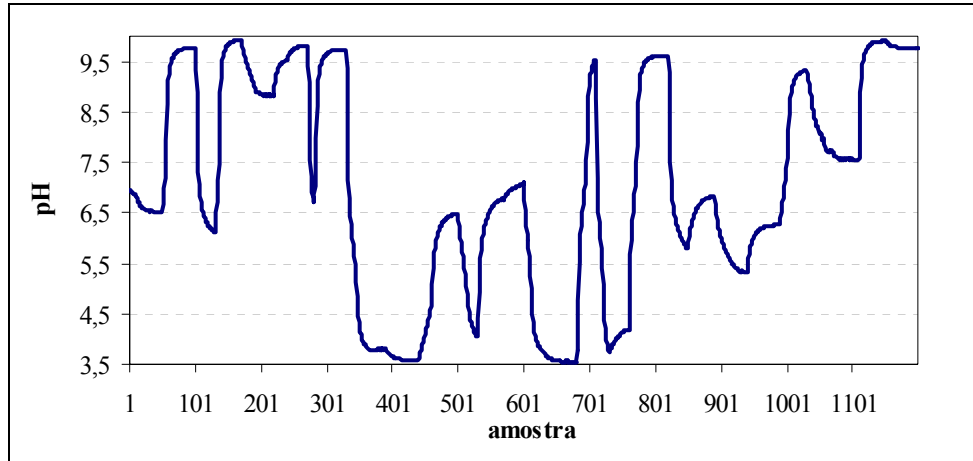


Figura 5-2 – Evolução do *pH* no tanque

A rede *feedforward* descrita foi treinada 10 vezes, pelos métodos *backpropagation* e de Levenberg-Marquardt. Em cada corrida, os dois métodos foram iniciados com os mesmos pesos, sendo esses definidos aleatoriamente entre $[-1;1]$, no entanto, os pesos inicialmente utilizados foram alterados entre corridas diferentes. Em todos os casos, foram utilizadas 1500 épocas treinamento. Na Tabela 5.2, mostra-se o resultado das corridas quando atingidas 1500 épocas, onde R^2 significa coeficiente de correlação (%) calculado a partir da equação (3.2).

Tabela 5.2 - Comparação do desempenho dos métodos de *backpropagation* e Levenberg-Marquardt (treinamento)

Corrida	<i>Backpropagation</i>		Levenberg-Marquardt	
	$R^2(\%)$	$100-R^2(\%)$	$R^2(\%)$	$100-R^2(\%)$
1	99,8028	0,1972	99,9879	0,0121
2	99,6445	0,3555	99,9871	0,0129
3	99,5841	0,4159	99,9924	0,0076
4	99,5074	0,4926	99,9936	0,0064
5	98,3656	1,6344	99,9884	0,0116
6	99,5036	0,4964	99,9851	0,0149
7	98,1779	1,8221	99,9855	0,0145
8	99,5955	0,4045	99,9907	0,0093
9	99,3106	0,6894	99,9914	0,0086
10	99,6832	0,3168	99,9900	0,0100
Média	99,3175	0,6825	99,9892	0,0108

De acordo com a Tabela 5.2, pode-se observar que o desempenho de treinamento, quando se compara o valor de R^2 , obtido com o método de Levenberg-Marquardt foi bastante superior ao *backpropagation*. O valor de $100-R^2$ que indica a porcentagem do sinal de saída, em termos da energia, que não foi identificada pelo modelo é, em média, 60 vezes menor no caso do método de Levenberg-Marquardt.

Como um exemplo, pode-se observar nas Figuras 5.3 e 5.4 as previsões das redes treinadas pelos métodos *backpropagation* e Levenberg-Marquardt, para o caso da corrida número 10. Estas redes correspondem às obtidas na última iteração de treinamento. Na Figura 5-5 podem ser observados os respectivos erros de previsão.

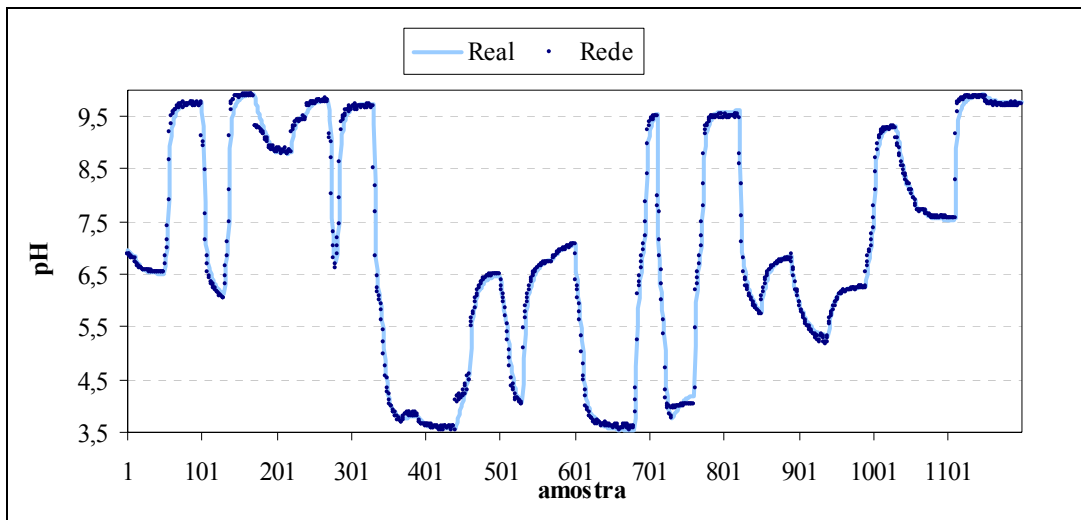


Figura 5-3 - Dados reais x previsão da rede *feedforward* treinada por *backpropagation*

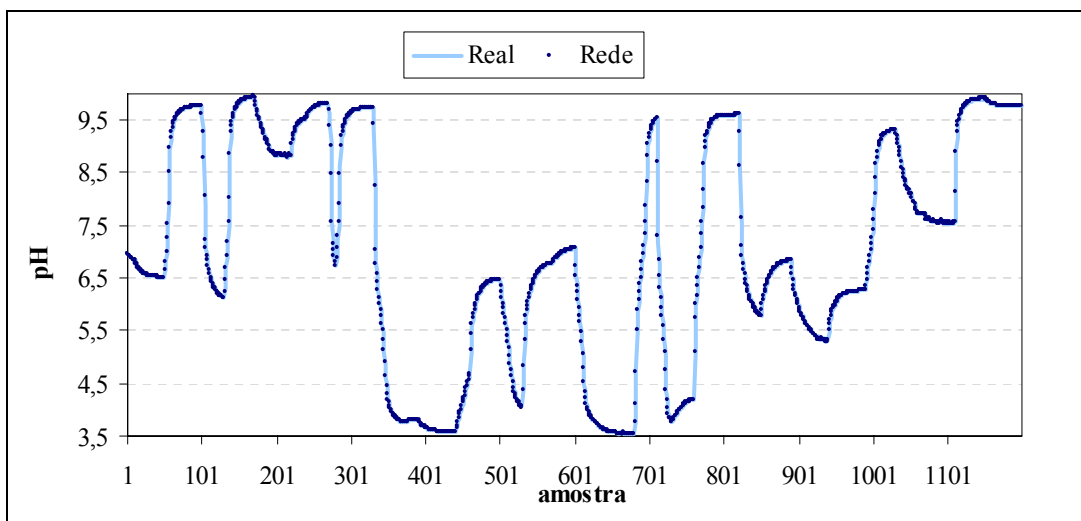


Figura 5-4 - Dados reais x previsão da rede *feedforward* treinada por Levenberg-Marquardt

Analisando as Figuras 5.3 a 5.5, nota-se que as amplitudes dos erros de treinamento foram, em geral, menores quando o treinamento foi efetuado com o método de Levenberg-Marquardt.

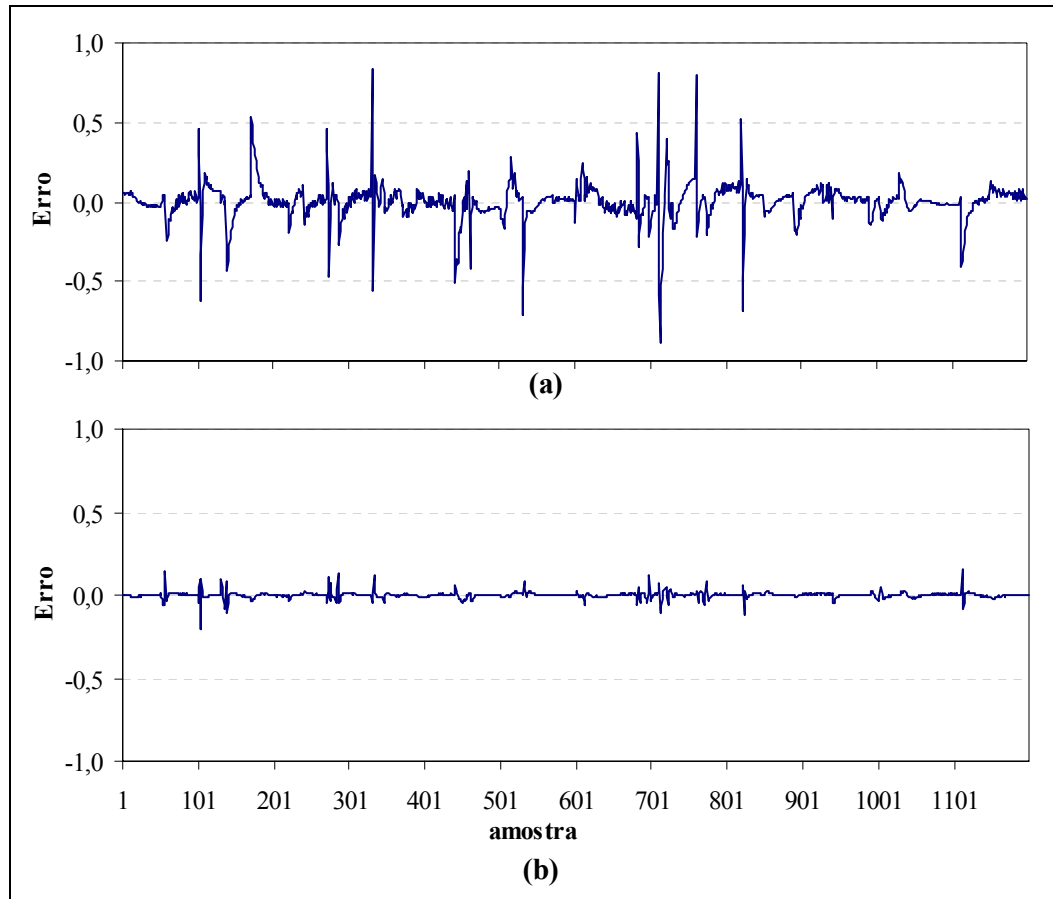


Figura 5-5 - (a) Erros de previsão da rede *feedforward* treinada por *backpropagation* (b) Erros de previsão da rede *feedforward* treinada por Levenberg-Marquardt

Também para a corrida de número 10, mostra-se na Figura 5-6 a evolução do valor do $100-R^2$ em função do número de épocas (ambos em escala log) para o treinamento efetuado por *backpropagation* e por Levenberg-Marquardt. Pode-se observar que para o *backpropagation*, seriam necessárias mais épocas, além das 1500 utilizadas, para a convergência do método. Ao contrário, no caso da técnica de Levenberg-Marquardt, a partir de 100 épocas o erro quadrático de aproximação não se alterou de forma significativa.

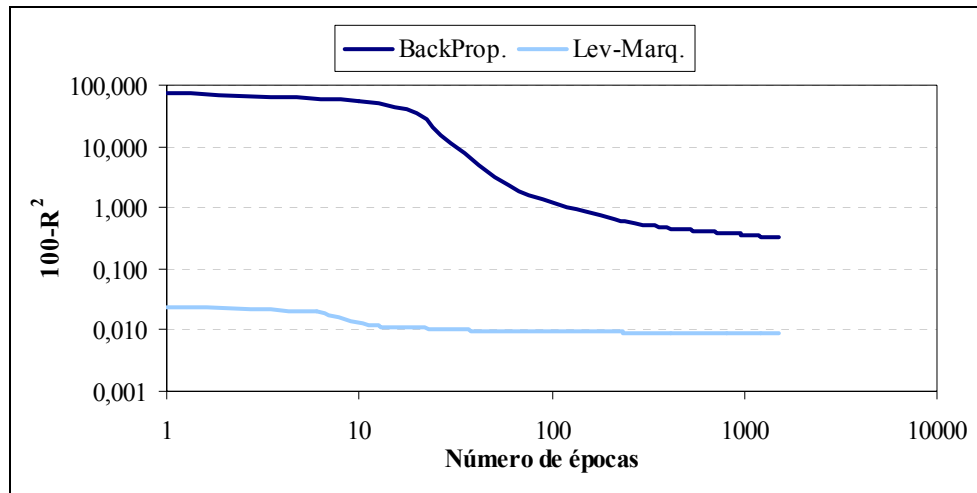


Figura 5-6 - Evolução do erro de treinamento utilizando os métodos *backpropagation* e Levenberg-Marquardt

Na Figura 5-6 pode-se observar que, no caso do método de Levenberg-Marquardt, o treinamento poderia ter sido finalizado em menos de 100 épocas, uma vez que o erro de aproximação não variou de forma significativa a partir desse ponto. Existe a possibilidade de estar ocorrendo o problema de sobre-treinamento da rede *feedforward* nesta situação. No entanto, neste capítulo deseja-se descobrir qual o melhor método, em termos da minimização do erro de aproximação, uma vez que no capítulo 7 mostra-se que o problema de sobre-treinamento pode ser minimizado alterando apenas a função objetivo de aprendizado da rede.

A diferença de desempenho entre os métodos de treinamento testados pode ser atribuída a utilização de informações relativas à interação entre as variáveis de otimização, neste caso os pesos da rede, na minimização do erro quadrático de aproximação. As informações de interação entre os pesos estão contidas nas derivadas de alta ordem do erro de aproximação. No caso da estratégia de Levenberg-Marquardt estas derivadas são utilizadas, ao contrário método de *backpropagation*. Na verdade, o *backpropagation* é uma versão bastante simplificada do método de Newton para otimização. Isto pode ser observado eliminando a dependência entre diferentes pesos da matriz H . A simplificação torna a matriz Hessiana diagonal, ou seja, todas as derivadas cruzadas são anuladas. Dessa forma, a inversa de H torna-se uma matriz diagonal em que os elementos da diagonal principal correspondem ao inverso dos respectivos elementos em H . Se comparado ao *backpropagation*, este “novo” método utiliza ganhos, fatores de aprendizagem, diferenciados para cada peso. A informação de segunda ordem está restrita a estes ganhos. Como última simplificação, considera-se que os

ganhos aplicados em todos os pesos são iguais, ou seja, o valor do fator de aprendizagem é único e serve como um parâmetro de ajuste do método. A técnica de treinamento resultante é equivalente ao clássico método *backpropagation* (sem o termo de momento).

5.4 Metodologia de Treinamento Baseada em Algoritmos Genéticos e Mínimos Quadrados

5.4.1 Introdução

Nesta seção foi aplicada uma técnica simples, em termos de implementação computacional, para treinamento de uma rede *feedforward* baseada em algoritmos genéticos e mínimos quadrados. Esta técnica é descrita em GHOSH e VERMA(2001) que utilizou um algoritmo genético para determinar os parâmetros pertencentes à camada intermediária e mínimos quadrados para estimar os pesos da camada de saída de uma rede neural *feedforward*.

No trabalho de CLAUMANN(1999), foi mostrado que o algoritmo genético descrito pelo autor obteve resultados superiores, em termos do resultado final e velocidade de convergência, quando comparado ao proposto por GOLDBERG(1989) para uma série de *benchmarks* de otimização. Por isso, foi utilizado o Algoritmo genético descrito pelo primeiro autor conjuntamente o método dos mínimos quadrados para o treinamento de uma rede *feedforward*.

Os Algoritmos Genéticos (*GAs*) são algoritmos de otimização baseados nos mecanismos de seleção natural e adaptação. Estes algoritmos operam com um conjunto de possíveis soluções, denominado “população”, para determinação do ótimo. Os elementos da população, também conhecidos por indivíduos, são combinados através de uma troca de informações de modo a encontrar melhores soluções que as da população original. A troca de informações para determinação da nova população depende do valor da função objetivo, de tal forma que melhores soluções possuem maior possibilidade de serem combinadas. Desse processo obtém-se, em geral, soluções melhores que as originais e toda a população desloca-se em direção ao ótimo. Quando um conjunto de indivíduos de tamanho igual ao da população é

obtido considera-se completada uma geração. O primeiro Algoritmo Genético foi descrito por HOLLAND(1975) e foi inspirado em um mecanismo simplificado da adaptação natural dos seres vivos ao ambiente. A adaptação pode ser considerada um processo de modificação progressiva de uma população promovendo um melhor desempenho no ambiente. O ambiente, neste caso, é a função objetivo a ser otimizada.

5.4.2 Termos Técnicos Utilizados em Algoritmos Genéticos

Em sistemas naturais um ou mais cromossomos combinam-se para formar a informação genética necessária à formação de um indivíduo. As *strings*, nos sistemas genéticos artificiais, são análogas aos cromossomos nos sistemas biológicos.

Em sistemas naturais a interação entre um organismo e seu ambiente é conhecida como fenótipo. O equivalente ao fenótipo, nos algoritmos genéticos, são as variáveis decodificadas, também chamadas de conjunto de parâmetros, solução alternativa ou ponto no espaço de solução.

Em Genética, os cromossomos dividem-se em genes que são caracterizados por duas propriedades: O alelo (função do gene) e o locus (posição do gene no cromossomo). Na pesquisa genética artificial, as *strings* são formadas por um conjunto de *detetores* ou *características* que são diferenciados entre si pela função que desempenham e pela posição que ocupam na *string*. O conjunto de todos os valores (também podem ser símbolos caso seja um número finito) que um detetor pode assumir é conhecido por alfabeto. Por exemplo, em codificação binária o alfabeto é $\{0,1\}$. Em codificação real, cada gene pode assumir qualquer valor dentro de um intervalo pré-definido.

Em populações naturais, o *fitness* é a capacidade das criaturas sobreviverem aos predadores, pestes e outros obstáculos à reprodução. No ambiente artificial, o *fitness* é uma medida relativa de desempenho entre os elementos da população. O *fitness* pode ser determinado diretamente a partir do valor da função objetivo ou de algum critério baseado nela.

5.4.3 Diferenças Entre Algoritmos Genéticos e Técnicas Clássicas de Otimização

Os algoritmos genéticos diferem dos métodos de otimização baseados em Cálculo nos seguintes aspectos:

- As variáveis de um problema a ser otimizado podem ser codificadas em algum alfabeto, tal como o binário. Neste caso, as operações efetuadas resumem-se a cópias e trocas de *bits*;
- *GAs* operam com uma população de possíveis soluções da otimização (indivíduos ou elementos) não com apenas uma solução. A robustez dos *GAs* à estagnação em mínimos locais deve-se a utilização de uma população cujos indivíduos são espalhados inicialmente por todo o espaço de pesquisa. Eventualmente alguns indivíduos da população poderão convergir para ótimos locais, porém isso dificilmente acontecerá com todos ao mesmo tempo;
- *GAs* utilizam apenas informação das avaliações da função objetivo, não empregando qualquer outro tipo de conhecimento tal como o de derivadas;
- A geração de novos indivíduos nos *GAs* não é realizada de forma determinística, mas através de modificações aleatórias nos indivíduos da população atual.

Mais detalhes a respeito de Algoritmos Genéticos ver GOLDBERG(1989).

5.4.4 Descrição da Técnica Baseada em Mínimos Quadrados e Algoritmos

Para treinamento da rede *feedforward* foi utilizado um algoritmo genético com codificação real. Neste caso, cada detetor equivale a um peso da rede a ser otimizado e, portanto, o número de detetores da *string* de cada indivíduo é igual ao número de pesos da camada intermediária. Em termos de implementação computacional, cada *string* corresponde a um vetor de números reais de comprimento igual ao número de variáveis sendo otimizadas.

Em cada geração, o algoritmo genético produz uma população de indivíduos, sendo que a *string* de cada indivíduo corresponde à camada intermediária de uma rede neural *feedforward* diferente. O valor da função objetivo (o erro quadrático de aproximação) de cada indivíduo utilizado no cálculo do *fitness* foi obtido da seguinte forma: Uma vez que os pesos da camada

intermediária foram definidos pelo algoritmo genético, calculam-se as saídas dos neurônios da camada intermediária para todos os padrões do grupo de dados de treinamento. Como os alvos são conhecidos e a camada de saída é linear, a determinação dos pesos da camada de saída implica na resolução de um problema de mínimos quadrados. Após estimar estes últimos pesos, obtém-se a saída prevista pela rede para todos os pontos do grupo de dados e, finalmente, calcula-se o erro quadrático de aproximação de treinamento.

Deve-se notar que as saídas dos neurônios da camada intermediária e de saída, considerando todos os padrões do grupo de dados de treinamento, são avaliadas apenas uma vez para o cálculo da função objetivo de um indivíduo.

Do ponto de vista da rede neural, o Algoritmo Genético é desconhecido e as únicas informações que a rede necessita desse são os pesos da camada intermediária. Do ponto de vista da aplicação do Algoritmo Genético, a arquitetura da rede é desconhecida e a única informação que o *GA* necessita é o valor da função objetivo.

As técnicas redes neurais, algoritmos genéticos e mínimos quadrados foram combinadas em um único algoritmo, no entanto, as partes atuam de forma contínua e independente.

Na identificação do tanque de *pH* foi utilizada uma população 30 indivíduos. Um número maior de indivíduos só elevou o tempo computacional e um número inferior resultou em pior desempenho de treinamento. Todos os outros parâmetros do método, tais como, tipos de operadores de crossover, mutação, seleção e taxas de crossover, mutação etc foram definidos de acordo com CLAUMANN(1999).

5.5 Comparação de Desempenho Entre um Método de Segunda Ordem e um Método Baseado em Algoritmos Genéticos

As estratégias de treinamento *GA* + mínimos quadrados e Levenberg-Marquardt foram comparadas no treinamento de uma rede *feedforward* com 30 neurônios na camada intermediária (a mesma estrutura utilizada na seção 5.3), para a identificação do tanque de *pH* descrito.

Na Tabela 5.3, mostra-se o resultado das corridas após 1500 épocas de treinamento comparando o método de Levenberg-Marquardt (resultados repetidos da Tabela 5.2) e a metodologia *GA* + mínimos quadrados. Para uma comparação adequada entre os dois métodos, em termos de convergência, foi considerado que uma geração, no caso do *GA*,

equivale a 30 épocas, ou seja, uma época associada a cada indivíduo de uma população composta por 30 desses. Dessa forma, o algoritmo genético foi finalizado em 50 gerações.

De acordo com a Tabela 5.3, pode-se observar que o desempenho de treinamento, quando se compara o valor de R^2 , obtido com a estratégia *GA* + mínimos quadrados foi superior em todas as simulações ao método de Levenberg-Marquardt. A porcentagem de energia do sinal de saída não identificada pela rede neural ($100-R^2$) foi, em média, 3 vezes menor no caso da técnica de *GA* + mínimos quadrados.

Como um exemplo, mostra-se na Figura 5-7, a previsão da rede treinada por *GA* + mínimos quadrados no caso da corrida número 10. Esta rede corresponde à obtida na última iteração de treinamento. Na Figura 5-8 pode ser vista uma comparação dos erros de previsão das técnicas de Levenberg-Marquardt e *GA* + mínimos quadrados.

Tabela 5.3 - Comparação do desempenho dos métodos Levenberg-Marquardt e *GA*+mínimos quadrados

Corrida	GA+ mínimos quadrados		Levenberg-Marquardt	
	$R^2(\%)$	$100-R^2(\%)$	$R^2(\%)$	$100-R^2(\%)$
1	99,9970	0,0030	99,9879	0,0121
2	99,9944	0,0056	99,9871	0,0129
3	99,9975	0,0025	99,9924	0,0076
4	99,9963	0,0037	99,9936	0,0064
5	99,9951	0,0049	99,9884	0,0116
6	99,9970	0,0030	99,9851	0,0149
7	99,9957	0,0043	99,9855	0,0145
8	99,9968	0,0032	99,9907	0,0093
9	99,9964	0,0036	99,9914	0,0086
10	99,9968	0,0032	99,9900	0,0100
Média	99,9963	0,0036	99,9892	0,0108

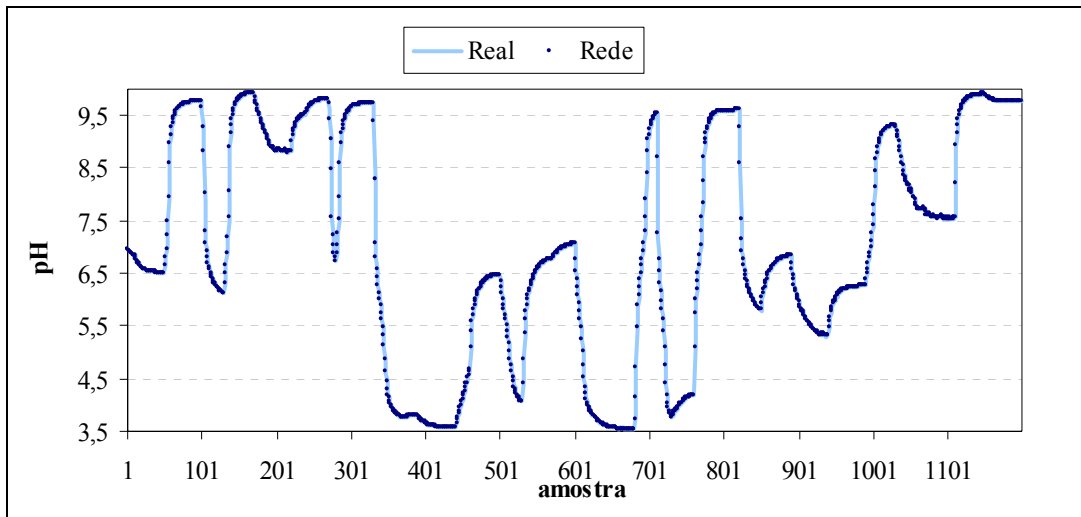
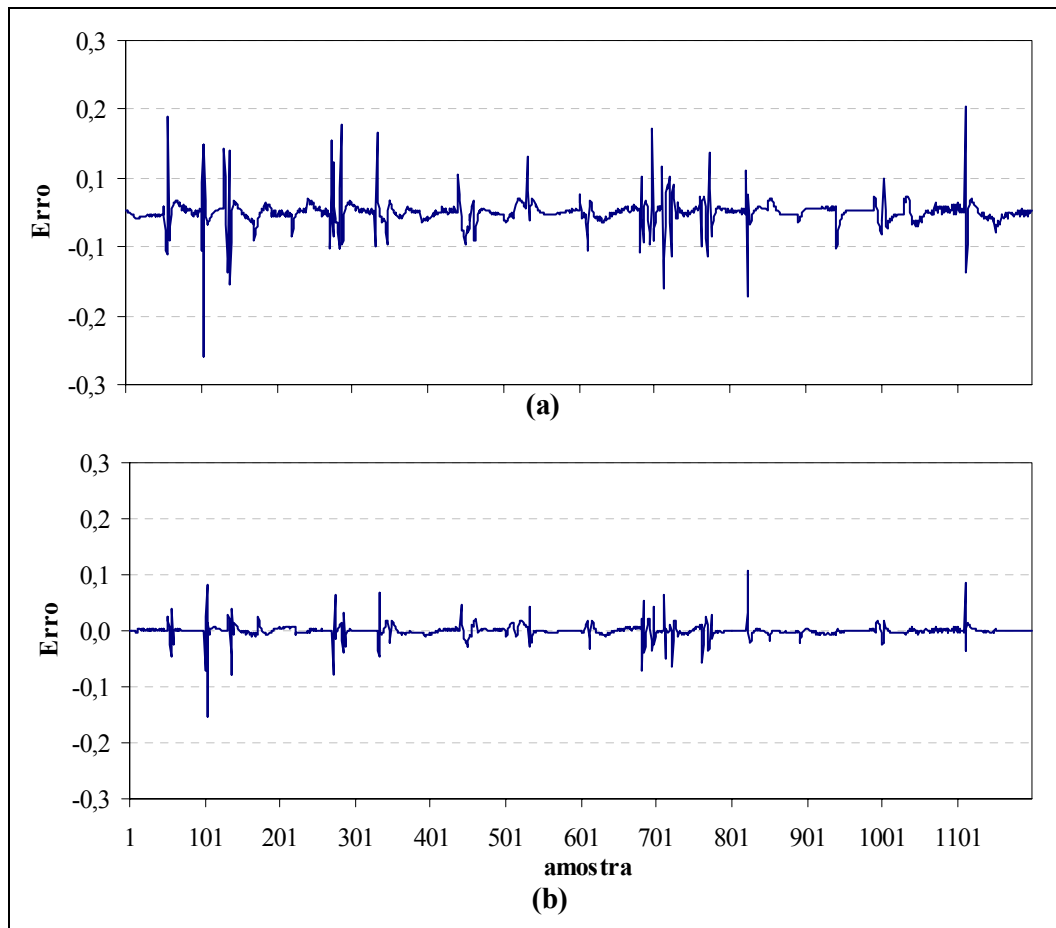


Figura 5-7 - Dados reais x previsão da rede *feedforward* treinada por *GA* + mínimos quadrados



**Figura 5-8 - (a) Erros de previsão da rede *feedforward* treinada por Levenberg-Marquardt
(b) Erros de previsão da rede *feedforward* treinada por *GA* + mínimos quadrados**

Pode ser observado na Figura 5-7, que o comportamento do sistema dinâmico pode ser dividido, de forma simplificada, em regiões onde o processo está próximo de atingir um

estado estacionário e em outras onde há rápidas transições de ponto de operação. Este comportamento ocorre em função das perturbações aplicadas na vazão de base. Comparando as Figuras 5.7 e 5.8, é possível observar que as regiões de pequeno erro estão associadas a comportamentos próximos ao estacionário e as regiões de picos de erros estão associadas à ocorrência de mudanças bruscas no processo. Para os dois tipos de comportamentos descritos, os erros de previsão da rede neural treinada pela estratégia *GA* + mínimos quadrados foram inferiores aos obtidos com a rede treinada por Levenberg-Marquardt.

Também para a corrida de número 10, mostra-se na Figura 5-9 a evolução do valor do $100-R^2$ em função do número de épocas para o treinamento efetuado por Levenberg-Marquardt e por *GA* + mínimos quadrados (uma geração = 30 épocas).

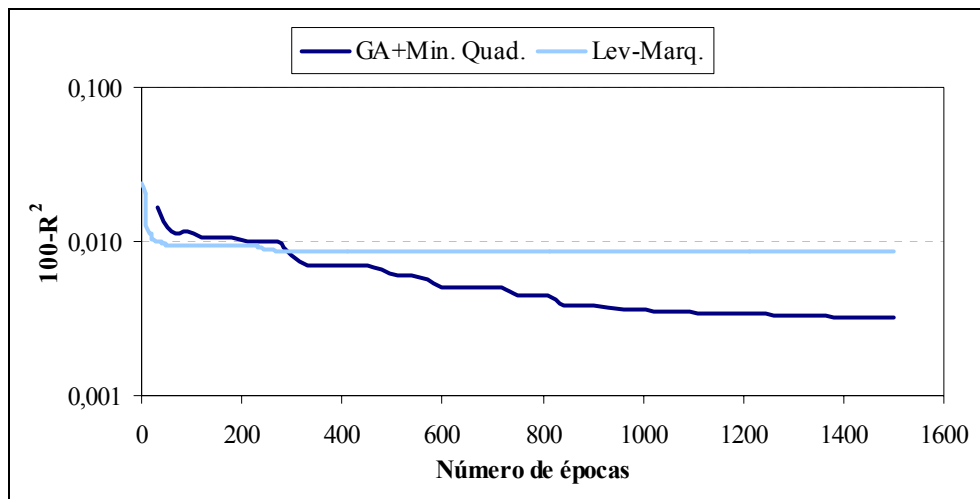


Figura 5-9 - Evolução do treinamento da rede *feedforward* utilizando Levenberg-Marquardt e *GA* + mínimos quadrados

Na Figura 5-9 pode-se observar que a estratégia de treinamento baseada em Levenberg-Marquardt converge rapidamente para um mínimo local. No caso da técnica de *GA* + mínimos quadrados, obtém-se inicialmente um desempenho inferior em termos de convergência, no entanto, o valor final de $100-R^2$ é inferior indicando a robustez dessa última estratégia à presença de mínimos locais.

Em termos de tempo computacional, observou-se que o desempenho do método *GA* + mínimos quadrados foi um pouco superior ao de Levenberg-Marquardt, se considerado o mesmo número de épocas. Este resultado pode ser explicado em termos das etapas de maior custo computacional utilizadas nos métodos e pelo número de vezes que estas etapas devem ser efetuadas por época.

Foi observado que as tarefas de maior custo computacional foram o cálculo do erro global de treinamento, que necessita da avaliação da saída da rede para todos os pontos do grupo de dados, e a estimação dos pesos a partir da técnica de mínimos quadrados. Em uma época, o erro global de treinamento deve ser avaliado apenas uma vez para as duas estratégias. No entanto, no caso da estratégia baseada em algoritmos genéticos apenas os pesos da camada de saída devem ser estimados por mínimos quadrados. Para o método Levenberg-Marquardt, além destes os pesos da camada intermediária também devem ser estimados por mínimos quadrados em cada época.

5.6 Conclusões

Embora tenha se tratado da identificação de apenas um sistema dinâmico, pode-se concluir, em função das diferenças significativas de desempenho de treinamento, que a estratégia *GA* + mínimos quadrados foi superior à técnica de Levenberg-Marquardt e essas duas foram muito superiores ao método *backpropagation*.

A estratégia *GA* + mínimos quadrados é a mais simples, em termos de implementação, pelo fato de ser baseada em ferramentas de otimização e identificação genéricas e independentes do problema. Este não é o caso dos métodos de Levenberg-Marquardt e *backpropagation* que necessitam do conhecimento de derivadas e, dessa forma, são dependentes da arquitetura da rede utilizada.

Uma rede neural *feedforward* com apenas uma camada intermediária composta de neurônios não-lineares e a camada de saída linear pode aproximar sistemas dinâmicos complexos, como é o caso do tanque de *pH*.

Embora, neste trabalho, não sejam utilizadas funções de ativação não diferenciáveis, é possível aplicar a estratégia *GA*+mínimos quadrados em problemas que utilizem tal classe de funções.

6 REGULARIZAÇÃO E REPRESENTAÇÃO DO CONHECIMENTO

6.1 Introdução

Determinar um modelo de um sistema, a partir de um conjunto finito de observações sem qualquer conhecimento prévio desse é, em geral, um problema mal-posto, no sentido de que um modelo único pode não existir TIKHONOV e ARSENIN(1977).

Na identificação de sistemas, utilizando modelos lineares nos parâmetros, problemas mal-postos produzem matrizes mal condicionadas, o que pode levar a grandes erros na determinação dos coeficientes dos modelos. Uma maneira de diminuir incertezas é introduzir conhecimento prévio do processo que se deseja identificar, além da simples aproximação de dados, de forma a obter-se uma solução única para o problema de aproximação. Um desses procedimentos é conhecido por regularização. Para transformar a identificação de sistemas em um problema bem-posto, TIKHONOV e ARSENIN(1977) criaram a teoria da regularização. Uma forma de conhecimento prévio utilizada na teoria de regularização é que descontinuidades são comportamentos pouco prováveis de ocorrer em processos. Isto evitará que o comportamento de um modelo varie abruptamente nas vizinhanças de uma observação, sendo que alguma interpolação e extrapolação dos dados pode ser justificada. Tal hipótese é válida para uma grande classe de sistemas do mundo real, mas não para todos os sistemas. A hipótese de suavidade pode ser incorporada em um modelo através do critério de identificação, ou seja, na etapa de estimação dos coeficientes ajustáveis desse O'SULLIVAN(1986), WAHBA (1990).

Regularizadores de naturezas diversas podem ser propostos, por exemplo, para penalizar a curvatura das funções utilizadas na aproximação dos dados, para impor restrições no valor assumido pelas variáveis, para forçar a estabilidade em malha aberta do modelo proposto para um sistema dinâmico, entre outros.

Alguns trabalhos que utilizam a regularização no contexto da identificação encontrados na literatura são: o de TULLEKEN(1993) que sugere o uso de restrições derivadas do conhecimento prévio do sistema para melhorar o modelo, no contexto da identificação de sistemas lineares; o de THOMPSON e KRAMER(1994) que discute como diferentes tipos de conhecimento podem ser utilizados para estruturar um modelo; o de GIROSI *et. al.*(1995) que analisa como diferentes penalizações da não suavidade levam a diferentes bases de funções

utilizadas na aproximação; o de JOHANSEN(1997) que apresenta uma estrutura unificada que permite adicionar várias formas de conhecimento durante a etapa de determinação dos coeficientes do modelo.

Nas próximas seções descreve-se como a regularização pode ser introduzida em um problema de mínimos quadrados.

6.2 Introdução de Regularizadores no Problema de Mínimos Quadrados

Matematicamente, a maior parte dos regularizadores pode ser introduzida no problema de identificação através da função objetivo. Termos adicionais, além do erro quadrático de aproximação, constituirão a nova função objetivo regularizada. Dessa forma, múltiplos critérios podem ser introduzidos. A importância de cada critério pode ser ponderada utilizando multiplicadores (constantes positivas).

Na Equação (6.1) mostra-se o formato da função objetivo regularizada J :

$$J = \|\mathbf{Y} - \mathbf{F}(\mathbf{X}) \cdot \boldsymbol{\theta}\|^2 + \sum_{i=1}^N \gamma_i \cdot J_i \quad (6.1)$$

onde,

N : Número de regularizadores;

\mathbf{Y} : Vetor de dados que se deseja aproximar;

\mathbf{X} : Matriz de dados de entrada;

$\mathbf{F}(\mathbf{X})$: Mapeamento não-linear aplicado em \mathbf{X} (Matriz);

$\boldsymbol{\theta}$: Vetor de parâmetros;

J_i - i -ésimo regularizador;

γ_i - i -ésimo multiplicador ($\gamma_i > 0$);

No caso de múltiplas saídas ($\boldsymbol{\theta}$ matriz), o problema de mínimos quadrados pode ser resolvido de forma independente para cada uma delas. Por isso, sem perda de generalidade, foi considerado o tratamento de uma saída apenas, o que implica que $\boldsymbol{\theta}$ é um vetor.

Algumas deduções efetuadas nas próximas seções dependem das variáveis definidas e, para maior compreensão, as matrizes e vetores presentes na Equação (6.1) são descritos em formato expandido:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \mathbf{M} \\ y_{N_p} \end{bmatrix}; \left\{ \begin{array}{l} y_i \in R; (i = 1, \dots, N_p) \\ \mathbf{Y}(N_p \times 1) \end{array} \right\}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{M} \\ \mathbf{X}_{N_p} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \Lambda & x_{1,N_v} \\ x_{2,1} & x_{2,2} & \Lambda & x_{2,N_v} \\ \mathbf{M} & & & \\ x_{N_p,1} & x_{N_p,2} & \Lambda & x_{N_p,N_v} \end{bmatrix}; \left\{ \begin{array}{l} x_{i,j} \in R; (i = 1, \dots, N_p; j = 1, \dots, N_v) \\ \mathbf{X}_i \in R^{N_v}; (i = 1, \dots, N_p) \\ \mathbf{X}(N_p \times N_v) \end{array} \right\}$$

$$\Phi(\boldsymbol{\varepsilon}) = \begin{bmatrix} \phi_1(\boldsymbol{\varepsilon}) \\ \phi_2(\boldsymbol{\varepsilon}) \\ \mathbf{M} \\ \phi_{N_f}(\boldsymbol{\varepsilon}) \end{bmatrix}; \left\{ \begin{array}{l} \boldsymbol{\varepsilon} \in R^{N_v}; (\boldsymbol{\varepsilon} = \mathbf{X}_i; i = 1, \dots, N_p) \\ \phi_i(\boldsymbol{\varepsilon}): R^{N_v} \rightarrow R; (i = 1, \dots, N_f) \\ \Phi(\boldsymbol{\varepsilon}): R^{N_v} \rightarrow R^{N_f} \end{array} \right\}$$

$$\mathbf{F}(\mathbf{X}) = \begin{bmatrix} \Phi^T(\mathbf{X}_1) \\ \Phi^T(\mathbf{X}_2) \\ \mathbf{M} \\ \Phi^T(\mathbf{X}_{N_p}) \end{bmatrix} = \begin{bmatrix} \phi_1(\mathbf{X}_1) & \phi_2(\mathbf{X}_1) & \Lambda & \phi_{N_f}(\mathbf{X}_1) \\ \phi_1(\mathbf{X}_2) & \phi_2(\mathbf{X}_2) & \Lambda & \phi_{N_f}(\mathbf{X}_2) \\ \mathbf{M} & & & \\ \phi_1(\mathbf{X}_{N_p}) & \phi_2(\mathbf{X}_{N_p}) & \Lambda & \phi_{N_f}(\mathbf{X}_{N_p}) \end{bmatrix}; \left\{ \begin{array}{l} \mathbf{F}(\mathbf{X}) \\ (N_p \times N_f) \end{array} \right\}$$

onde

N_p : Número de pontos do grupo de dados;

N_v : Número de variáveis;

N_f : Número de funções (regressores);

$\Phi(\cdot)$: Mapeamento não-linear aplicado aos pontos do grupo de dados.

Para exemplificar a influência dos regularizadores na solução do problema de mínimos quadrados, pode-se considerar um problema com apenas um regularizador descrito por:

$$J_1 = 2 \cdot \mathbf{B}^T \boldsymbol{\theta} + \boldsymbol{\theta}^T \mathbf{A} \boldsymbol{\theta} \quad (\mathbf{B} \text{ Vetor; } \mathbf{A} \text{ Matriz Simétrica}) \quad (6.2)$$

A minimização de J pode ser efetuada da seguinte forma:

$$\begin{aligned} J &= \|\mathbf{Y} - \mathbf{F}(\mathbf{X}) \cdot \boldsymbol{\theta}\|^2 + \gamma_1 \cdot J_1 \\ &\Downarrow \\ J &= \left(\mathbf{Y}^T \mathbf{Y} - 2 \mathbf{Y}^T \mathbf{F}(\mathbf{X}) \boldsymbol{\theta} + \boldsymbol{\theta}^T \mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X}) \boldsymbol{\theta} \right) + \gamma_1 \cdot \left(2 \mathbf{B}^T \boldsymbol{\theta} + \boldsymbol{\theta}^T \mathbf{A} \boldsymbol{\theta} \right) \\ &\quad \text{erro aproximação} \qquad \qquad \qquad J_1 \\ &\Downarrow \\ J &= \underbrace{\mathbf{Y}^T \mathbf{Y}}_{\text{constante}} + 2 \left(\gamma_1 \cdot \mathbf{B}^T - \mathbf{Y}^T \mathbf{F}(\mathbf{X}) \right) \boldsymbol{\theta} + \boldsymbol{\theta}^T \left(\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X}) + \gamma_1 \cdot \mathbf{A} \right) \boldsymbol{\theta} \\ &\quad \text{linear} \qquad \qquad \qquad \text{quadrático} \\ &\Downarrow \left(\frac{dJ}{d\boldsymbol{\theta}} = 0 \right) \text{ (condição de otimização)} \\ &\left(\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X}) + \gamma_1 \cdot \mathbf{A} \right) \boldsymbol{\theta} = \left(\mathbf{F}(\mathbf{X})^T \mathbf{Y} - \gamma_1 \cdot \mathbf{B} \right) \end{aligned} \quad (6.3)$$

No caso nominal (sem regularizadores, ou seja, $\gamma_1=0$) tem-se que a Equação (6.3) reduz-se para:

$$\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X}) \boldsymbol{\theta} = \mathbf{F}(\mathbf{X})^T \mathbf{Y} \quad (6.4)$$

onde $\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})$ é denominada de matriz de covariância.

A Equação (6.4) é conhecida por "equação normal".

Comparando as Equações (6.3) e (6.4) conclui-se que para introduzir regularizadores no problema nominal basta adicionar a matriz \mathbf{A} (multiplicada por γ) da Equação (6.2) no lado esquerdo da Equação (6.4) e subtrair vetor \mathbf{B} (multiplicado por γ) no lado direito. É importante salientar que este procedimento pode ser efetuado para qualquer regularizador que possa ser descrito por termos lineares ou quadráticos nos parâmetros. Regularizadores devem ser definidos ou semi-definidos positivos e, portanto, quando descritos no formato matricial, o termo quadrático deverá sempre dominar o termo linear, se este último existir. Dessa forma, um regularizador nunca é formado apenas pelo termo linear, pois resultaria em um

comportamento indefinido. O regularizador definido na Equação (6.2) poderia conter um termo constante, além do linear e do quadrático, porém, sem perda de generalidade, o termo constante não foi incluído, já que será eliminado quando da aplicação da condição de otimização $\frac{dJ}{d\theta} = 0$. A implementação de regularizadores que não podem ser obtidos a partir dos termos descritos levaria a um problema de otimização não-linear e, obviamente, não seria possível utilizar o método dos mínimos quadrados para determinação dos coeficientes ótimos.

Generalizando o resultado da Equação (6.4) para vários regularizadores tem-se:

$$J_i = 2\mathbf{B}_i^T \boldsymbol{\theta} + \boldsymbol{\theta}^T \mathbf{A}_i \boldsymbol{\theta} ; \quad i = 1..N \quad (\text{onde } N \text{ é o número de regularizadores})$$

⇓

$$\left(\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X}) + \sum_{i=1}^N \gamma_i \cdot \mathbf{A}_i \right) \boldsymbol{\theta} = \left(\mathbf{F}(\mathbf{X})^T \mathbf{Y} - \sum_{i=1}^N \gamma_i \cdot \mathbf{B}_i \right) \quad (6.5)$$

6.3 Normalização de Regularizadores

A Equação (6.1) pode ser escrita, por conveniência, em um formato equivalente, mas ligeiramente modificado:

$$J = \|\mathbf{Y} - \mathbf{F}(\mathbf{X}) \cdot \boldsymbol{\theta}\|^2 + \sum_{i=1}^N \gamma_i K_i J_i \quad (6.6)$$

Na Equação (6.6) tem-se:

K_i : Constante de normalização, definida como $\frac{\|\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})\|}{\|\mathbf{A}_i\|}$, utilizando termos contidos na

Equação (6.5);

γ_i : *i*-ésimo multiplicador normalizado ($\gamma_i > 0$);

$\|\cdot\|$: Norma 2 para matrizes que é igual ao autovalor máximo da matriz.

Os outros termos foram definidos na Equação (6.1)

A utilização de multiplicadores normalizados tem por objetivo:

- 1) Limitar valores de γ_i a uma faixa de pesquisa: Se $\gamma_i=0$ então o regularizador associado não possuirá importância alguma em relação ao termo de erro quadrático de aproximação, portanto $\gamma_i>0$. No outro extremo, se $\gamma_i=1$, o regularizador terá a mesma importância que o erro quadrático de aproximação dos dados. Dessa forma, os multiplicadores γ_i dos regularizadores devem assumir, na maior parte dos casos, valores entre 0 e 1;
- 2) Uma vez obtida a solução que agrada ao usuário, os valores γ_i são úteis para indicar a importância relativa de cada termo da função objetivo.

A norma 2 foi escolhida porque é a norma minimizada no problema de mínimos quadrados.

Das propriedades de norma implica que a matriz de covariância $F(X)^T F(X)$ bem como as matrizes A_i , devidamente multiplicadas por seus K_i , possuirão a mesma norma 2, ou seja, terão a mesma importância na solução do problema. A relevância de cada termo poderá ser alterada através de manipulação dos multiplicadores γ_i .

6.4 Influência da Regularização pela Norma dos Pesos no Condicionamento da Matriz de Covariância

Para um modelo não-linear, mas linear nos parâmetros, ou seja, no caso de uma série não-linear, o problema de estimação destes, considerando a função objetivo como o erro quadrático de aproximação, é equivalente a resolver um problema de mínimos quadrados.

O regularizador mais conhecido, também o mais simples, é a norma do vetor de pesos ao quadrado $\|\theta\|^2$, ou seja o comprimento do vetor de pesos ao quadrado JOHANSEN(1997). Adicionando-se este ao erro quadrático de aproximação, tem-se uma nova função objetivo:

$$G = J + \gamma \cdot \|\theta\|^2 = \|\mathbf{Y} - \mathbf{F}(\mathbf{X}) \cdot \theta\|^2 + \gamma \cdot \|\theta\|^2$$

onde $\gamma \geq 0$ é o parâmetro de regularização.

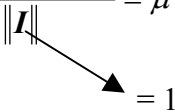
O mínimo de G pode ser obtido analiticamente, tal como para a função J , e resulta da solução de um sistema linear em relação ao vetor de parâmetros θ (Ver dedução na seção 6.2):

$$(\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X}) + \gamma \cdot \mathbf{I})\boldsymbol{\theta} = \mathbf{F}(\mathbf{X})^T \mathbf{Y}$$

Sendo que \mathbf{I} é a matriz de regularização que corresponde a uma matriz identidade com as mesmas dimensões de $\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})$.

O parâmetro γ pode ser definido, de forma conveniente, como:

$$\gamma = \mu \cdot \frac{\|\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})\|}{\|\mathbf{I}\|} = \mu \cdot \lambda^{Max}$$



onde

$\mu \geq 0$ é uma nova constante, a ser definida;

$\|\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})\|$ é norma da matriz $\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})$ cujo valor é igual ao autovalor máximo λ^{Max} dessa. A função objetivo G pode ser reescrita como:

$$G = \|\mathbf{Y} - \mathbf{F}(\mathbf{X}) \cdot \boldsymbol{\theta}\|^2 + \mu \cdot \lambda^{Max} \cdot \|\boldsymbol{\theta}\|^2$$

Esta definição tem como vantagem poder se trabalhar com uma constante normalizada, ou seja, se $\mu = 1$ a norma da matriz de covariância $\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})$, associada ao termo de aproximação dos dados $\|\mathbf{Y} - \mathbf{F}(\mathbf{X}) \cdot \boldsymbol{\theta}\|^2$, é igualada à norma da matriz de regularização \mathbf{I} multiplicada pela constante λ^{Max} . O objetivo é determinar um valor otimizado para μ que seja independente do problema.

Considerando σ o espectro de A (conjunto de todos os autovalores de matriz A) e α uma constante arbitrária tem-se, da Álgebra matricial, a seguinte relação:

$$\sigma(A + \alpha \cdot \mathbf{I}) = \sigma(A) + \alpha \tag{6.7}$$

Utilizando $A = \mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})$, $\alpha = \gamma = \mu \lambda^{Max}$ na Equação (6.7) podem-se determinar os autovalores da matriz $(\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X}) + \gamma \mathbf{I})$ a partir dos autovalores da matriz $\mathbf{F}(\mathbf{X})^T \mathbf{F}(\mathbf{X})$. Dessa forma, os seguintes resultados podem ser obtidos (Ver Tabela 6.1).

Tabela 6.1 - Influência da regularização nos autovalores da matriz $F(X)^T F(X)$

Matriz	Menor autovalor	Maior autovalor	Número de condição
$F(X)^T F(X)$	λ^{Min}	λ^{Max}	$\lambda^{Max} / \lambda^{Min}$
$F(X)^T F(X) + \gamma I$	$\lambda^{Min} + \mu \lambda^{Max}$	$(1 + \mu) \lambda^{Max}$	$(1 + \mu) \cdot \lambda^{Max} / (\lambda^{Min} + \mu \cdot \lambda^{Max})$

Na Tabela 6.1, mostra-se o valor do número de condição, para os casos com e sem regularização. Este parâmetro é definido como a relação entre o maior e o menor autovalores de uma matriz, quando se considera a norma 2 para matrizes, e indica a dificuldade de inverter uma matriz. Se o número de condição for próximo a 1, a matriz pode ser invertida facilmente, no sentido que a inversa existe e pode ser calculada acuradamente. Se o número de condição for muito alto, da ordem de muitas ordens de grandeza, a inversa pode não existir.

Para matrizes simétricas do tipo $F(X)^T F(X)$ e $(F(X)^T F(X) + \gamma I)$ os autovalores são todos não-negativos. Os maiores autovalores estão associados às funções mais importantes da série, em termos da energia do sinal de saída, para a aproximação dos dados. No outro extremo, os menores autovalores estão associados aos termos da série de pequena importância na aproximação dos dados. Como um exemplo, quando se utiliza um número de funções superior ao necessário para identificar um processo, a matriz de covariância $F(X)^T F(X)$ possuirá autovalores muito próximos a zero, já que existem regressores de pequena importância. Neste caso, o número de condição cresce arbitrariamente e os pesos podem ser estimados incorretamente em função dos problemas numéricos envolvidos. O problema é que não se conhece, a priori, quais e quantos regressores são relevantes à identificação e que, portanto deveriam ser utilizados.

A não utilização da regularização causa problemas numéricos na estimação dos pesos conforme descrito. Ao contrário, um valor excessivo do parâmetro de regularização aumenta significativamente o erro quadrático de aproximação. Dessa forma, existe uma faixa de valores que o parâmetro de regularização μ pode assumir.

Uma vez que os maiores autovalores estão associados aos termos da série de maior importância na aproximação dos dados, os primeiros não deveriam sofrer variação significativa quando se adiciona o regularizador pela norma do vetor de pesos. Dessa forma, o maior autovalor de $(F(X)^T F(X) + \gamma I)$ deve ser aproximadamente igual ao maior autovalor de $F(X)^T F(X)$, ou seja:

$$\lambda^{Max} \approx (1 + \mu) \lambda^{Max}$$

o que implica em:

$$\mu \ll 1$$

Esta restrição estabelece um limite superior para μ . Lembrando que ($\mu \geq 0$), por definição, tem-se um limite inferior para este parâmetro.

Se μ for permitido variar, por exemplo, na faixa [0;1E-4] pode-se garantir que as duas restrições sejam satisfeitas. O valor atribuído ao limite superior foi obtido através de simulações descritas na seção 6.6, onde se mostra que, para valores de μ igual ou maiores que 1E-4, o erro de aproximação poderia ter, em alguns casos, um aumento significativo quando comparado ao erro para $\mu=0$, ou seja, sem regularização. Dessa forma, fica estabelecida uma faixa de atuação para o parâmetro de regularização.

Para $\mu=0$ tem-se a solução trivial que não utiliza a regularização. Esta situação não é de interesse em função dos problemas numéricos que podem ocorrer. É possível definir um melhor limite inferior para μ relacionando o valor desse com o máximo do número de condição que não resulte em problemas numéricos. O número de condição da matriz $F(\mathbf{X})^T F(\mathbf{X}) + \gamma \mathbf{I}$ é, conforme a Tabela 6.1, igual a $\frac{(1 + \mu) \cdot \lambda^{Max}}{(\lambda^{Min} + \mu \cdot \lambda^{Max})}$. Utilizando $\mu \ll 1$ nesta expressão, o número de condição da matriz regularizada fica limitado superiormente pelo valor de μ

$$\text{Número de condição} = \frac{(1 + \mu) \cdot \lambda^{Max}}{(\lambda^{Min} + \mu \cdot \lambda^{Max})} \approx \frac{\lambda^{Max}}{(\lambda^{Min} + \mu \cdot \lambda^{Max})} \leq \frac{\lambda^{Max}}{\mu \cdot \lambda^{Max}} = \frac{1}{\mu} \quad (6.8)$$

Segundo PEDERSEN(1997), para cálculos efetuados em dupla precisão, o máximo valor de número de condição para o qual é garantida estabilidade numérica, na solução de sistemas lineares, está em torno de 1E+8. Utilizando a Equação (6.8), obtém-se que para $\mu > 1E-6$ o critério de limitação do valor do número de condição é satisfeito com segurança. Neste caso, tem-se que o número de condição $\leq 1E+6$. Dessa maneira, o intervalo $1E-6 \leq \mu \leq 1E-4$ garante estabilidade numérica para qualquer número de regressores utilizado, sem prejudicar a capacidade de aproximação da série de funções.

6.5 Influência da Regularização na Capacidade de Generalização de Modelos Empíricos

Nesta seção, prova-se a conexão entre o regularizador $\|\theta\|^2$ e a suavidade da superfície de um modelo empírico, caracterizado por uma série. Os resultados obtidos são aplicáveis às redes neurais multicamadas reconhecendo que estas podem ser representadas através de séries de funções cujos argumentos também são séries.

A capacidade de generalização de um modelo está diretamente relacionada com as curvaturas da superfície desse. Um bom modelo empírico é aquele que aproxima bem os dados e possui boa capacidade de generalização, ou seja, apresenta pequenas curvaturas (ausência de oscilações e ótimos locais) em torno dos dados utilizados na identificação.

Na Figura 6-1 mostra-se uma ilustração de duas possíveis formas de aproximação para o mesmo conjunto de dados. Tem-se, nos dois casos, um pequeno erro de modelagem. No entanto, a amostragem dos dados é deficiente gerando incertezas na aproximação. O modelo 1 generaliza melhor porque não gera comportamentos pouco prováveis de existirem na realidade. Segundo o princípio de Occam CHEN e HAYKIN(2001), na ausência de informação, o comportamento com menor probabilidade de erro de previsão é o suave. Por este motivo, um modelo empírico linear é a melhor escolha se com esse for obtido um bom ajuste dos dados.

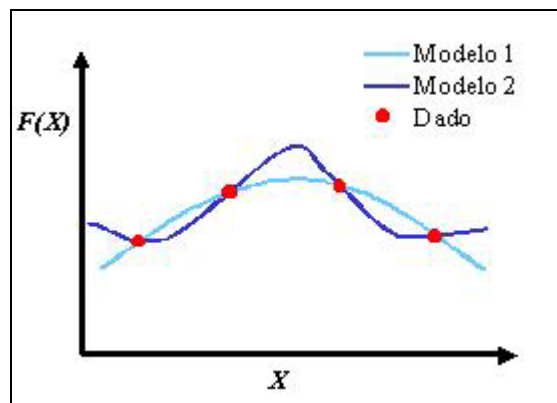


Figura 6-1 - Comparação da aproximação do mesmo conjunto de dados para diferentes modelos

Foram descritas apenas noções intuitivas a respeito da capacidade de generalização e a sua associação com a suavidade. No entanto, a propriedade de generalização deve ser formalizada matematicamente para impor um comportamento suave a um modelo não-linear.

As curvaturas presentes na superfície de uma função ficam definidas pelas derivadas de segunda ordem, caso existam. Para um modelo empírico linear todas as derivadas de segunda ordem se anulam, ou seja, todos os elementos da matriz *Hessiana* valerão 0. Dessa forma, limitar a curvatura implica em penalizar o desvio do comportamento linear. No trabalho de JOHANSEN(1997) foi definido um índice de desempenho para penalização das curvaturas presentes na superfície de um modelo empírico. O índice foi definido como a integral, com relação a todas as variáveis de entrada, do somatório das derivadas de segunda ordem ao quadrado, ou seja:

$$\int_{\Omega} \left\| \nabla^2 (\Phi(\varepsilon)^T \theta) \right\|^2 d\varepsilon \quad (6.9)$$

onde,

∇^2 : Derivadas de segunda ordem;

Ω : Domínio de integração;

ε : Vetor de variáveis de entradas, ou seja, variáveis independentes.

A Equação (6.9) representa um índice de desempenho que é positivo semi-definido, ou seja, vale 0 no caso de uma série linear e maior que 0 em qualquer outro caso. As derivadas de segunda ordem ao quadrado resultam em valores sempre maiores ou iguais a zero para o integrando. A integral garante que a penalização das curvaturas seja aplicada para todo o espaço onde o modelo é válido.

Uma vez definido o conjunto de funções, é possível demonstrar que o índice de desempenho descrito pode ser colocado no formato $\theta^T M \theta$, onde M é uma matriz a coeficientes constantes. A demonstração é dada a seguir:

Para obter uma expressão para o regularizador de curvatura em função do vetor de parâmetros θ , define-se um vetor V que corresponde à concatenação de todas as linhas da matriz de derivadas de segunda ordem da saída do modelo em relação às variáveis independentes (entradas do modelo):

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \Lambda & h_{1,N_v-1} & h_{1,N_v} \\ h_{2,1} & h_{2,2} & \Lambda & h_{2,N_v-1} & h_{2,N_v} \\ & & \mathbf{M} & & \\ h_{N_v-1,1} & h_{N_v-1,2} & \Lambda & h_{N_v-1,N_v-1} & h_{N_v-1,N_v} \\ h_{N_v,1} & h_{N_v,2} & \Lambda & h_{N_v,N_v-1} & h_{N_v,N_v} \end{bmatrix} \left(\text{com } h_{i,j} = \frac{\partial^2 (\boldsymbol{\Phi}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta})}{\partial \varepsilon_i \partial \varepsilon_j} \right)$$

↓

$$\mathbf{V} = \begin{bmatrix} h_{1,1} \Lambda h_{1,N_v} & h_{2,1} \Lambda h_{2,N_v} & \Lambda & h_{N_v-1,1} \Lambda h_{N_v,N_v} \\ \text{linha 1 de H} & \text{linha 2 de H} & & \text{linha Nv de H} \end{bmatrix}^T$$

Como o modelo empírico tem validade em um domínio limitado, pode-se associar a cada variável ε_i ($i=1, \dots, N_v$) uma faixa de trabalho, ou seja, ε_i é definida no intervalo $[a_i, b_i]$ ($i=1, \dots, N_v$). Substituindo as definições na Equação (6.9) tem-se:

$$\int_{\Omega} \left\| \nabla^2 (\boldsymbol{\Phi}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta}) \right\|^2 d\boldsymbol{\varepsilon} = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \mathbf{V}(\boldsymbol{\varepsilon})^T \cdot \mathbf{V}(\boldsymbol{\varepsilon}) d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v}$$

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \left(h_{1,1}^2 + \Lambda h_{N_v,1}^2 + h_{2,1}^2 + \Lambda h_{2,N_v}^2 \Lambda h_{N_v,1}^2 + \Lambda h_{N_v,N_v}^2 \right) (\boldsymbol{\varepsilon}) d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v}$$

Os termos na integral aparecem como um somatório. Dessa forma, pode-se deduzir a expressão para a integral de um $h_{i,j}$ genérico. O regularizador completo pode ser obtido pela soma destes termos. Considerando i e j arbitrários, mas fixos.

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \left(h_{i,j}^2 \right) (\boldsymbol{\varepsilon}) d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v} = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \left(\frac{\partial^2 (\boldsymbol{\Phi}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta})}{\partial \varepsilon_i \partial \varepsilon_j} \right)^2 d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v}$$

↓ ($h_{i,j}$ é um escalar)

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \left(\frac{\partial^2 (\boldsymbol{\Phi}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta})}{\partial \varepsilon_i \partial \varepsilon_j} \right) \cdot \left(\frac{\partial^2 (\boldsymbol{\Phi}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta})}{\partial \varepsilon_i \partial \varepsilon_j} \right) d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v}$$

↓ (A derivada parcial é um operador linear, portanto pode-se considerar :

$$h_{i,j} = \frac{\partial^2 (\boldsymbol{\Phi}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta})}{\partial \varepsilon_i \partial \varepsilon_j} = \mathbf{A}_{i,j}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta}$$

$$\begin{aligned}
& \int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} (\mathbf{A}_{i,j}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta})^T \cdot (\mathbf{A}_{i,j}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta}) d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v} \\
& \quad \Downarrow \\
& \int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \boldsymbol{\theta}^T (\mathbf{A}_{i,j}(\boldsymbol{\varepsilon}) \mathbf{A}_{i,j}(\boldsymbol{\varepsilon})^T) \cdot \boldsymbol{\theta} d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v} \\
& \quad \Downarrow \\
& \boldsymbol{\theta}^T \left(\int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \mathbf{A}_{i,j}(\boldsymbol{\varepsilon}) \mathbf{A}_{i,j}(\boldsymbol{\varepsilon})^T \cdot d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v} \right) \boldsymbol{\theta} \tag{6.10}
\end{aligned}$$

A Equação (6.10) é quadrática nos parâmetros. O termo $\mathbf{M}_{i,j} = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \Lambda \int_{a_{N_v}}^{b_{N_v}} \mathbf{A}_{i,j}(\boldsymbol{\varepsilon}) \mathbf{A}_{i,j}(\boldsymbol{\varepsilon})^T \cdot d\varepsilon_1 d\varepsilon_2 \Lambda d\varepsilon_{N_v}$ desta equação é uma matriz quadrada a coeficientes constantes com dimensões compatíveis com o número de parâmetros. Como a contribuição de todos os $h_{i,j}$, para o índice de desempenho dado pela Equação (6.9) é aditiva, pode-se definir,

$$\mathbf{M} = \sum_{i=1}^{N_v} \sum_{j=1}^{N_v} \mathbf{M}_{i,j}$$

onde \mathbf{M} também é uma matriz quadrada a coeficientes constantes com dimensões compatíveis com o número de parâmetros. Por definição, a matriz fixa \mathbf{M} é semidefinida positiva, ou seja, $\boldsymbol{\theta}^T \mathbf{M} \boldsymbol{\theta} \geq 0 \forall \boldsymbol{\theta}$.

Dessa forma tem-se que $\boldsymbol{\theta}^T \mathbf{M} \boldsymbol{\theta}$ representa o índice de desempenho dado pela Equação (6.9), ou seja:

$$\int_{\Omega} \left\| \nabla^2 (\boldsymbol{\Phi}(\boldsymbol{\varepsilon})^T \boldsymbol{\theta}) \right\|^2 d\varepsilon = \boldsymbol{\theta}^T \mathbf{M} \boldsymbol{\theta} \tag{6.11}$$

se todas as derivadas de segunda ordem existirem.

A igualdade apresentada na Equação (6.11) tem uma implicação muito importante uma vez que é possível encontrar um majorante para $\boldsymbol{\theta}^T \mathbf{M} \boldsymbol{\theta}$ em função de $\|\boldsymbol{\theta}\|^2$.

$$\begin{array}{cccccc}
\text{(a)} & \text{(b)} & \text{(c)} & \text{(d)} & \text{(e)} & \text{(f)} \\
\boldsymbol{\theta}^T \mathbf{M} \boldsymbol{\theta} = |\boldsymbol{\theta}^T \mathbf{M} \boldsymbol{\theta}| = |\boldsymbol{\theta}^T \cdot \mathbf{M} \boldsymbol{\theta}| = \|\boldsymbol{\theta}\| \cdot \|\mathbf{M} \boldsymbol{\theta}\| \cdot |\cos \alpha| \leq \|\boldsymbol{\theta}\| \cdot \|\mathbf{M} \boldsymbol{\theta}\| \leq \|\boldsymbol{\theta}\| \cdot \|\mathbf{M}\| \cdot \|\boldsymbol{\theta}\| \leq \lambda_M^{Max} \|\boldsymbol{\theta}\|^2
\end{array}$$

Considerações:

- (a) $\theta^T M \theta \geq 0 \quad \forall \theta$;
- (b) Definindo dois vetores $u = \theta$ e $v = M\theta$,
- (c) $|u^T \cdot v| = \|u\| \cdot \|v\| \cdot |\cos \alpha|$ (α é o ângulo entre u e v);
- (d) $0 \leq |\cos \alpha| \leq 1$
- (e) A norma de uma matriz é definida como $\|T\| = \sup_{x \neq 0} \frac{\|Tx\|}{\|x\|}$. Dessa forma, $\|T\|$ é o maior fator de amplificação que um elemento x (vetor) contido no domínio de T pode sofrer. De acordo com LEWIS(1991), isto implica $\|Tx\| \leq \|T\| \cdot \|x\| \quad \forall x \in$ ao domínio de T .
- (f) $\|T\| = \lambda_M^{Max}$ (o maior autovalor da matriz M).

Pode-se concluir que para minimizar $\int_{\Omega} \|\nabla^2(\Phi(\epsilon)^T \theta)\|^2 d\epsilon = \theta^T M \theta$ basta minimizar $\|\theta\|^2$,

ou seja:

$$\text{Min. } \|\theta\|^2 \Rightarrow \text{Min. } \theta^T M \theta = \text{Min. } \int_{\Omega} \|\nabla^2(\Phi(\epsilon)^T \theta)\|^2 d\epsilon \quad (\text{eq. 12-6})$$

Dessa forma, o índice de desempenho para penalização das curvaturas da superfície de um modelo com alta complexidade de cálculo $\int_{\Omega} \|\nabla^2(\Phi(\epsilon)^T \theta)\|^2 d\epsilon$ pode ser substituído, na função objetivo de treinamento, por outro muito mais simples $\|\theta\|^2 = \theta^T \theta$. O último ainda tem a vantagem de ser aplicável a séries de funções não diferenciáveis em relação às variáveis independentes.

O mapeamento “complicado” $\theta^T M \theta$, relativo à elipse cujos eixos principais (autovetores) não estão alinhados aos eixos coordenados, pode ser majorado por um mapeamento mais simples, a circunferência. Se o diâmetro da circunferência for diminuído, o mesmo deve acontecer com as dimensões da elipse para que a restrição $\theta^T M \theta \leq \lambda_M^{Max} \|\theta\|^2$ permaneça verdadeira. Assim sendo, se o simples regularizador $\|\theta\|^2$ for minimizado, então $\int_{\Omega} \|\nabla^2(\Phi(\epsilon)^T \theta)\|^2 d\epsilon = \theta^T M \theta$ também será.

6.6 Resultados

Para demonstrar os resultados obtidos nas seções 6.4 e 6.5, foram efetuados alguns testes relativos à identificação do tanque de pH descrito na seção 5.3. Para este sistema, o valor do pH , no instante $(k+1)$, pode ser identificado a partir do valor do pH e da vazão de base no tempo discreto k . Os testes foram efetuados utilizando redes de base radial, de funções de escala e *feedforward*. O parâmetro de regularização foi variado entre 0, caso sem regularização e $1E-4$, o limite superior assumido.

6.6.1 Identificação do Tanque de pH Utilizando uma Rede de Base Radial

Uma rede de base radial com 30 centros, selecionados aleatoriamente entre os pontos do grupo de dados, foi utilizada na identificação do tanque de pH . Em um estudo prévio foi determinado que para um número diferente de centros, por exemplo 20 ou 40, os resultados não seriam alterados qualitativamente, ou seja, as mesmas conclusões poderiam ser obtidas em relação à influência da regularização no treinamento de uma rede de base radial. A função de ativação utilizada foi $\exp(-\frac{r^2}{\sigma^2})$ com $\sigma=1$. A saída foi normalizada entre 0 e 1. A função objetivo que incorpora o termo de penalização da norma do vetor de pesos, além do erro quadrático de aproximação, foi utilizada para treinamento da rede de base radial.

Na Figura 6-2 mostram-se os pesos estimados da camada de saída, em valor absoluto, para alguns valores do parâmetro de regularização. A abscissa corresponde ao índice dos pesos, sendo estes ordenados em ordem decrescente de valor absoluto.

Nota-se, na Figura 6-2, que o valor absoluto dos pesos da camada de saída diminui à medida que o parâmetro de regularização μ é aumentado, o que implica em um decréscimo da norma do vetor de pesos da camada de saída. Este efeito é esperado já que a norma descrita faz parte da função objetivo minimizada. No caso sem regularização, observam-se pesos da ordem de $1E+6$, indicando uma pobre capacidade de generalização, uma vez que os dados de saída da rede foram normalizados entre 0 e 1. As soluções mais coerentes correspondem as situações $\mu= 1E-6$ a $1E-4$, onde a grande maioria da norma dos pesos é menor ou próxima a 1.

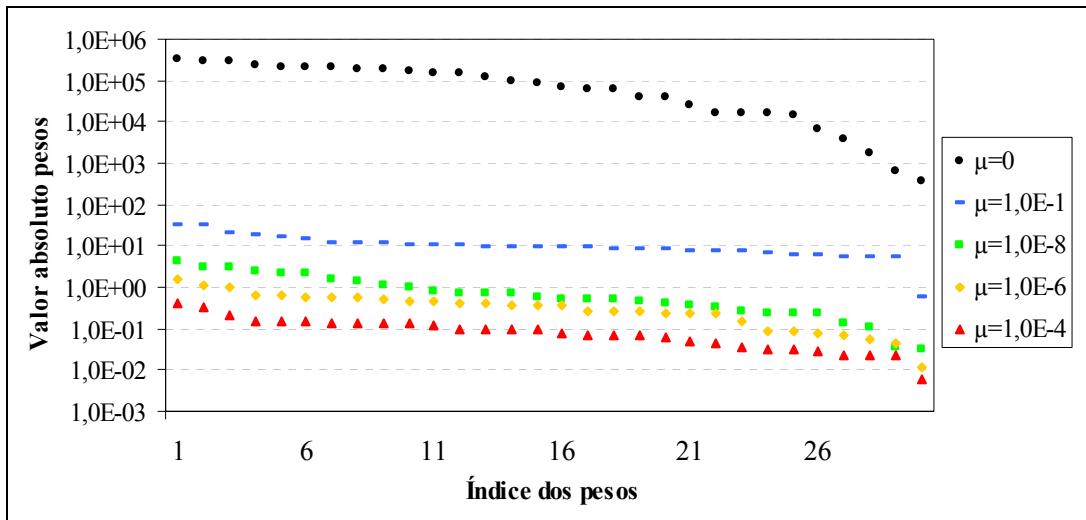


Figura 6-2 - Valor absoluto dos pesos para alguns valores do parâmetro de regularização

Na Figura 6-3 pode-se observar a influência que o parâmetro de regularização possui no número de condição da matriz de covariância.

Nota-se, na Figura 6-3, que o aumento do parâmetro de regularização diminui o número de condição da matriz de covariância. Isto ocorre pelo aumento do menor autovalor da matriz, já que o maior é pouco afetado pela regularização. Nesta figura também se pode observar a curva do número de condição máximo, considerando que o menor autovalor da matriz de covariância sem regularização é nulo. Pela proximidade das duas curvas, pode-se concluir que este é muito pequeno.

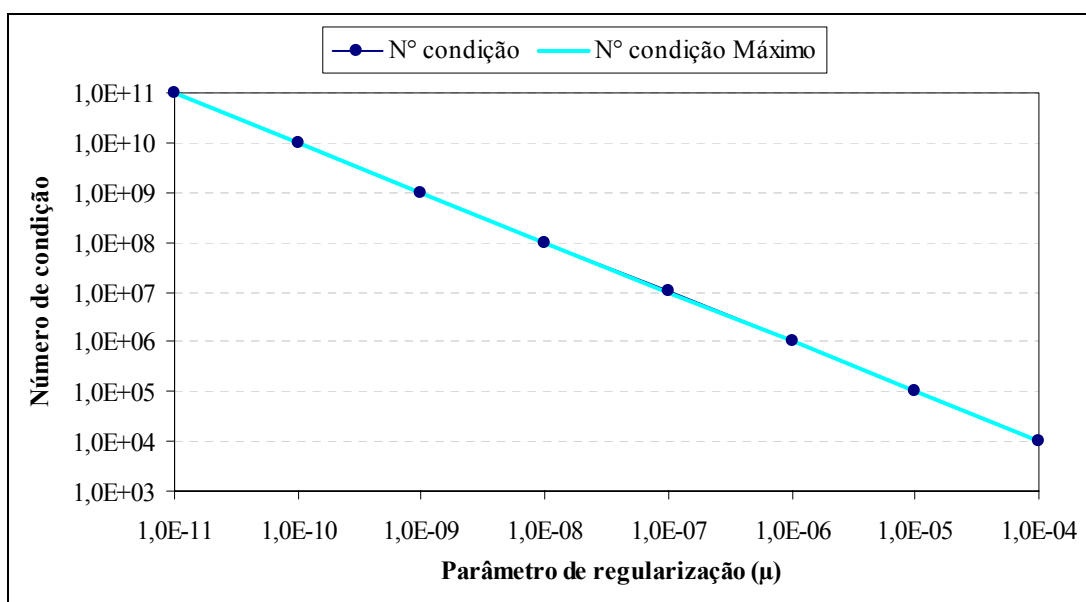


Figura 6-3 - Influência do parâmetro de regularização no número de condição da matriz de covariância

Embora não possa ser mostrado na Figura 6-3, em função da escala logarítmica para a abscissa, o valor do número de condição para a matriz de covariância sem regularização é da ordem de $1E+17$. Este valor extremamente alto explica os valores elevados para os pesos quando $\mu=0$.

Na Figura 6-4 pode-se observar o valor do erro quadrático normalizado, $100-R^2$, em função do valor do parâmetro de regularização. À medida que a importância do critério da norma do vetor de pesos é aumentada com o crescimento de μ , o valor de $100-R^2$ cresce de forma monotônica. Isto é esperado uma vez que os pesos devem ser ajustados para minimizar dois critérios: o erro quadrático de aproximação e a norma dos pesos ao invés de apenas o primeiro.

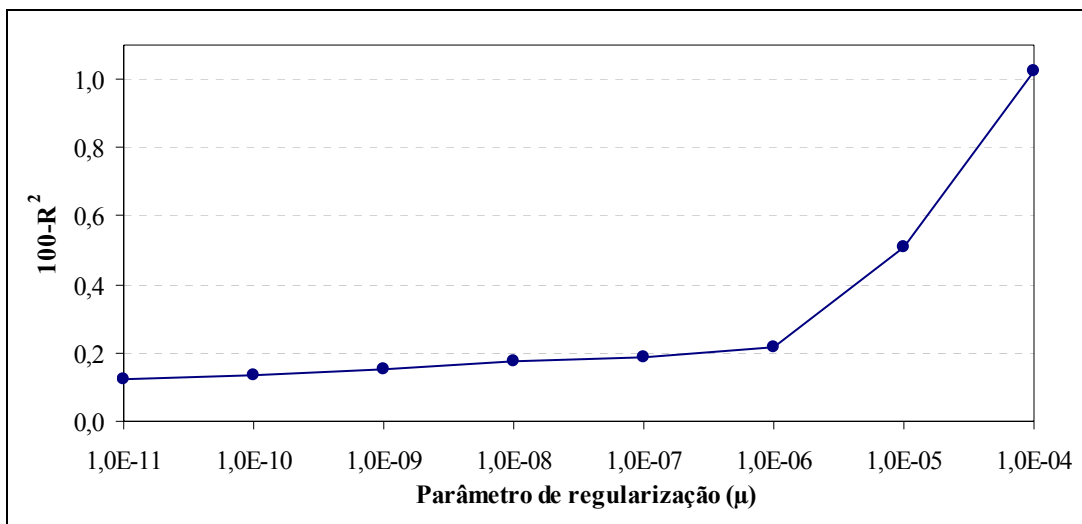


Figura 6-4 - Influência do parâmetro de regularização no valor de $100-R^2$

Na Figura 6-4 pode-se observar que $100-R^2$ sofre um aumento pouco significativo até $\mu=1E-6$. A partir deste μ há um grande acréscimo em $100-R^2$. Como o objetivo é determinar um valor do parâmetro de regularização que garanta estabilidade numérica na estimação dos pesos e baixo do erro quadrático de aproximação, $\mu=1E-6$ é o valor ótimo neste caso.

Embora não possa ser mostrado na Figura 6-4, o valor de $100-R^2$ para o caso sem regularização está em torno de 0,1, ou seja, não há uma grande variação em $100-R^2$ na faixa de μ de 0 a $1E-6$. No entanto, os pesos da camada de saída são alterados em várias ordens de grandeza. Isto implica que o erro quadrático de aproximação apresenta baixa sensibilidade a modificações na superfície do modelo empírico. Dessa forma, pode-se concluir que a utilização somente do erro quadrático de aproximação não fornece, em geral, uma função

objetivo adequada para identificação de sistemas dinâmicos. O problema deste critério é que utiliza somente a informação puntual dos dados de treinamento, o que ocorre na vizinhança destes pontos não é levado em consideração porque a informação de vizinhanças está contida nas derivadas.

A utilização do erro quadrático de aproximação conjuntamente com a norma dos pesos gera uma função objetivo de melhor desempenho do que apenas o primeiro critério. Neste caso, o conhecimento das derivadas dos pontos do grupo dados é utilizado indiretamente, uma vez que minimizando a norma do vetor de pesos estimados o mesmo ocorre com as derivadas de segunda ordem da(s) saída(s) da rede em relação à(s) entrada(s), conforme já demonstrado na seção 6.5.

6.6.2 Identificação do Tanque de pH Utilizando uma Rede de Funções de Escala

Como no caso da rede de base radial, os pesos da rede de funções de escala foram avaliados pela técnica de mínimos quadrados com regularização pela norma dos pesos. Foram utilizados dois níveis de funções de escala, totalizando $3^2 + 4^2 = 25$ funções de escala.

Na Figura 6-5 mostram-se os pesos da camada de saída, em valor absoluto, para alguns valores do parâmetro de regularização. Os vetores de pesos foram ordenados em ordem decrescente de valor absoluto.

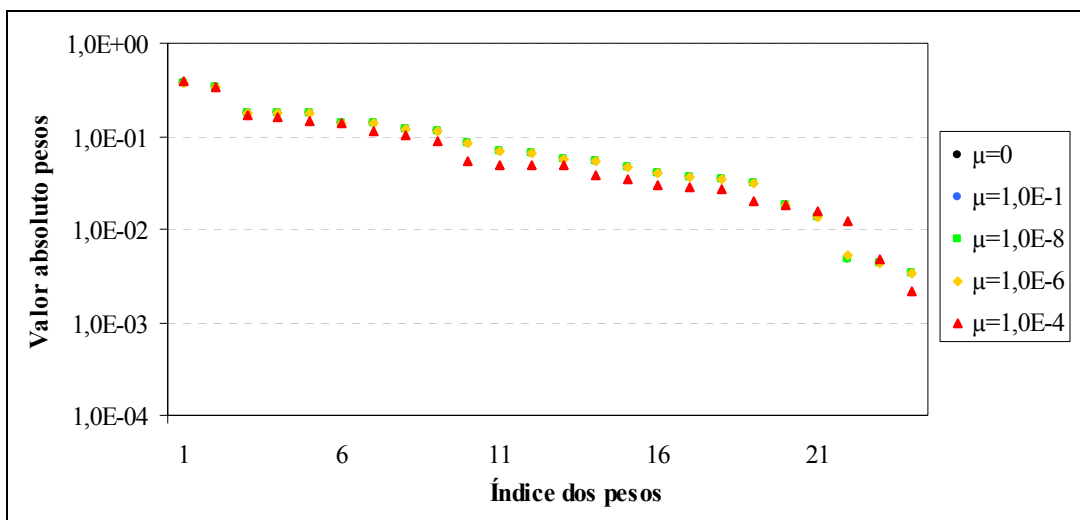


Figura 6-5 - Valor absoluto dos pesos para alguns valores do parâmetro de regularização

Na Figura 6-5 pode-se observar que, no caso da rede de funções de escala, os valores dos pesos variam muito pouco em função do valor do parâmetro de regularização. A magnitude do maior peso é inferior a 1 em qualquer caso.

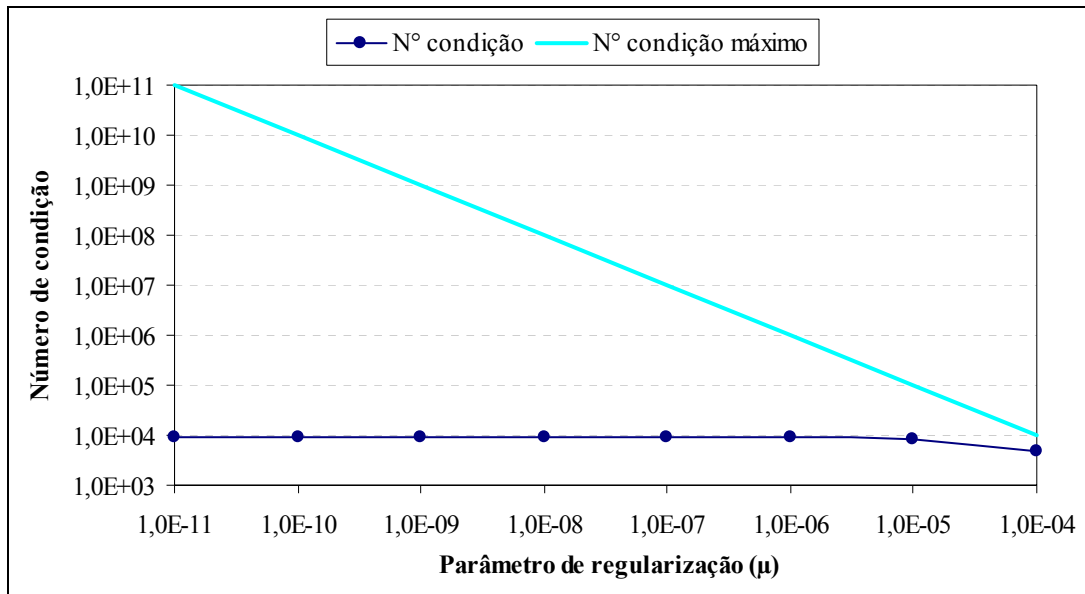


Figura 6-6 - Influência do parâmetro de regularização no número de condição da matriz de covariância (1º nível de resolução)

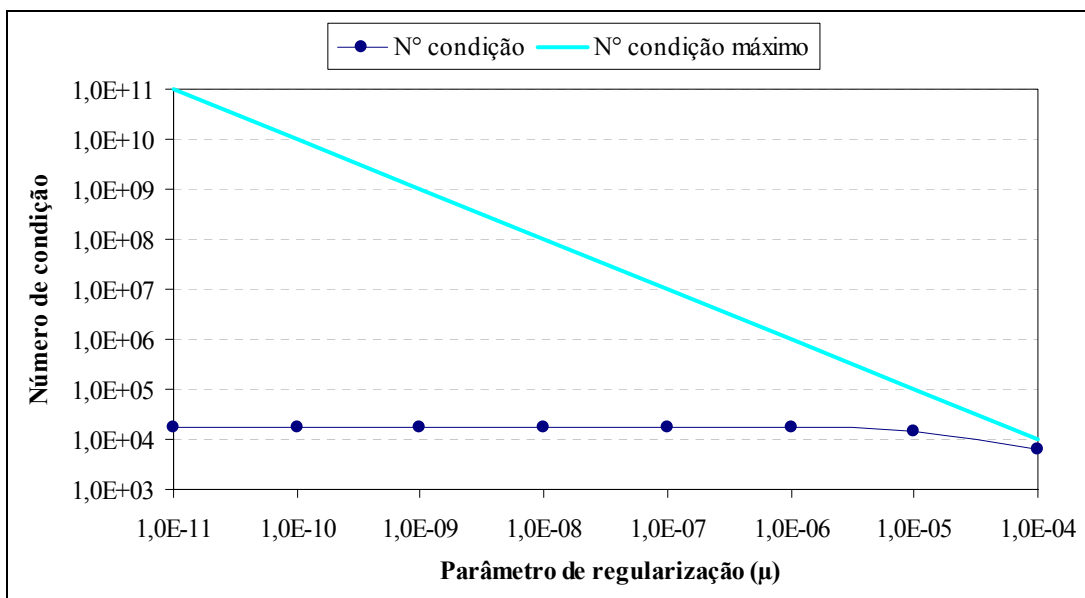


Figura 6-7 - Influência do parâmetro de regularização no número de condição da matriz de covariância (2º nível de resolução)

Nas Figuras 6.6 e 6.7 pode ser observada a evolução do número de condição em função do parâmetro de regularização, para os dois níveis utilizados na identificação do tanque de *pH*.

Nestas figuras, o número de condição da matriz de covariância, para os dois níveis de funções de escala, variou pouco indicando que o maior e menor autovalores não sofreram mudanças significativas. Dado que a contribuição do termo de regularização é definida de tal forma que o maior autovalor da matriz de covariância não sofra influência e como o número de condição não variou de forma significativa, então o menor autovalor também não sofreu modificações significativas em todos os casos. Dessa forma, o maior e o menor autovalores estão mais próximos quando se compara a rede de funções de escala com a de base radial, o que explica o menor número de condição encontrados no primeiro caso.

Na Figura 6-8 mostra-se a evolução do valor de $100-R^2$ em função do valor do parâmetro de regularização, para a rede de funções de escala. Como os pesos não sofreram mudanças significativas com a utilização da regularização, o mesmo é esperado para o valor $100-R^2$ em função do valor de μ .

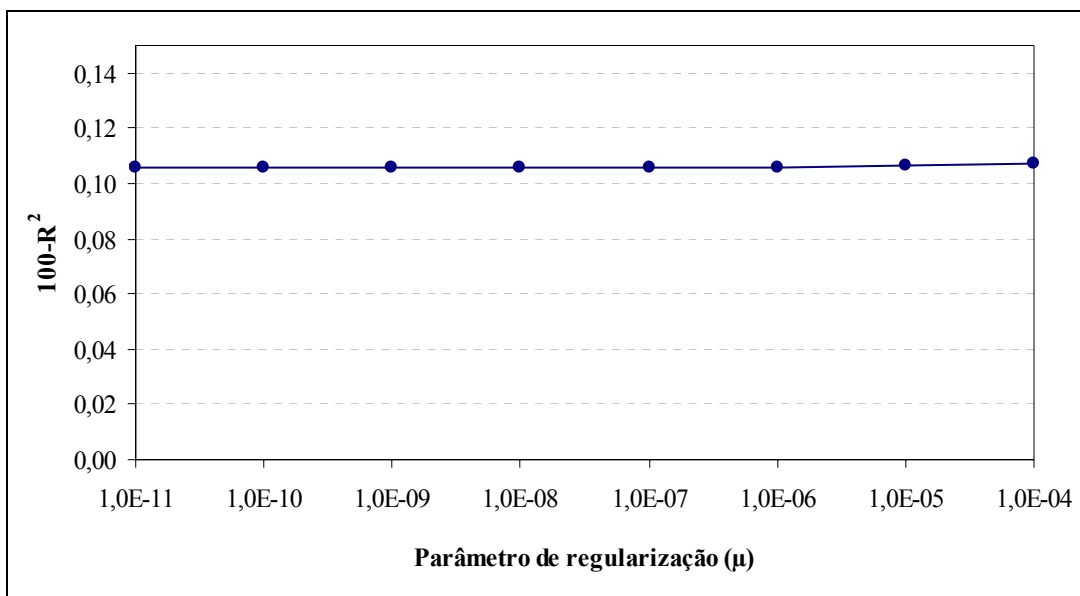


Figura 6-8 - Influência do parâmetro de regularização no valor de $100-R^2$

Para os mesmos dados de entrada e saída, o número de condição da matriz de covariância da camada de saída foi da ordem $1E+17$ para a rede de base radial e de $1E+4$ para a rede de funções de escala. A diferença de ordens de grandeza no número de condição pode ser atribuída a que as redes de base radial são *frames* e a rede de funções de escala é uma base para um espaço de funções. Um *frame* é um aproximador composto de regressores que podem ser linearmente correlacionados. Em uma base, os regressores devem ser, por definição, linearmente independentes.

No caso da rede de base radial, a aproximação não pode ser melhorada significativamente uma vez que os regressores são fortemente correlacionados. A utilização de mais neurônios só aumentará o número de condição da matriz de covariância. Embora, a regularização pela norma dos pesos garanta a estabilidade numérica na estimação dos pesos, a utilização desta técnica não diminuirá o erro quadrático de treinamento. Na verdade, há regressores em excesso, ou seja, alguns neurônios poderiam ser eliminados. Para a rede de funções de escala, ainda seria possível adicionar mais níveis para melhorar a aproximação.

6.6.3 Identificação do Tanque de *pH* Utilizando uma Rede Neural *Feedforward*

As redes utilizadas nas seções 6.6.1 e 6.6.2, para a identificação do tanque de *pH*, podem ser descritas por séries de regressores não-lineares. Este não é o caso de uma rede *feedforward*. No entanto, foram obtidos resultados semelhantes, em termos da influência do parâmetro de regularização, para a magnitude dos pesos e condicionamento da matriz de covariância da camada de saída.

Variando o valor do parâmetro de regularização entre 0 e $1E-4$, foram calculados os pesos da rede *feedforward* pela técnica de *GA* + mínimos quadrados (Ver seção 5.4). A magnitude dos coeficientes da camada intermediária, determinados pelo *GA*, foi limitada ao intervalo de $[-5;5]$, sendo que uma faixa maior de pesquisa resultava em um maior tempo computacional para convergência do método e uma faixa menor implicava em uma limitação da capacidade de aproximação no treinamento.

A matriz de covariância, associada à estimação dos pesos da camada de saída, resultante da aplicação do método de mínimos quadrados pode apresentar problemas de condicionamento. Assim sendo, monitorou-se a magnitude dos pesos, condicionamento da matriz de covariância da camada de saída e erro de aproximação em função do número de épocas.

Variando o parâmetro de regularização, foram treinadas redes *feedforward* com 30 neurônios na camada intermediária até a convergência. Em qualquer caso, utilizou-se inicialmente a mesma população. No entanto, após algumas gerações, estas populações devem ser diferentes porque as estimativas para os pesos da camada de saída são alteradas. Estes últimos também influenciam a função objetivo e, portanto, alteram o comportamento da convergência do método.

A função de ativação utilizada na camada intermediária foi a tangente hiperbólica e considerou-se a camada de saída linear.

Na Figura 6-9 mostram-se os pesos da camada de saída, em valor absoluto, para a rede *feedforward* correspondente ao melhor indivíduo da última geração, variando o parâmetro de regularização. Os pesos foram ordenados em ordem decrescente de valor absoluto.

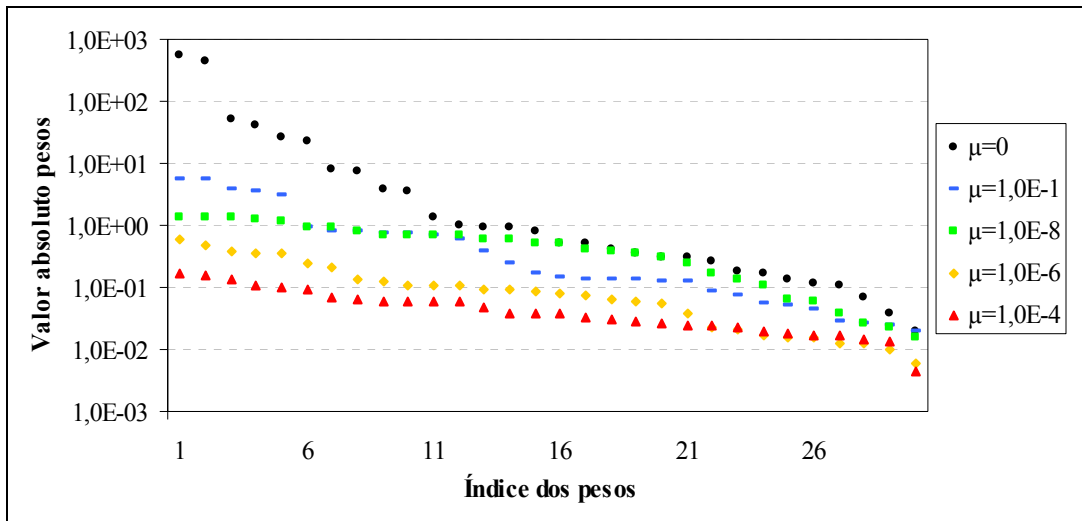


Figura 6-9 - Valor absoluto dos pesos para alguns valores do parâmetro de regularização (μ)

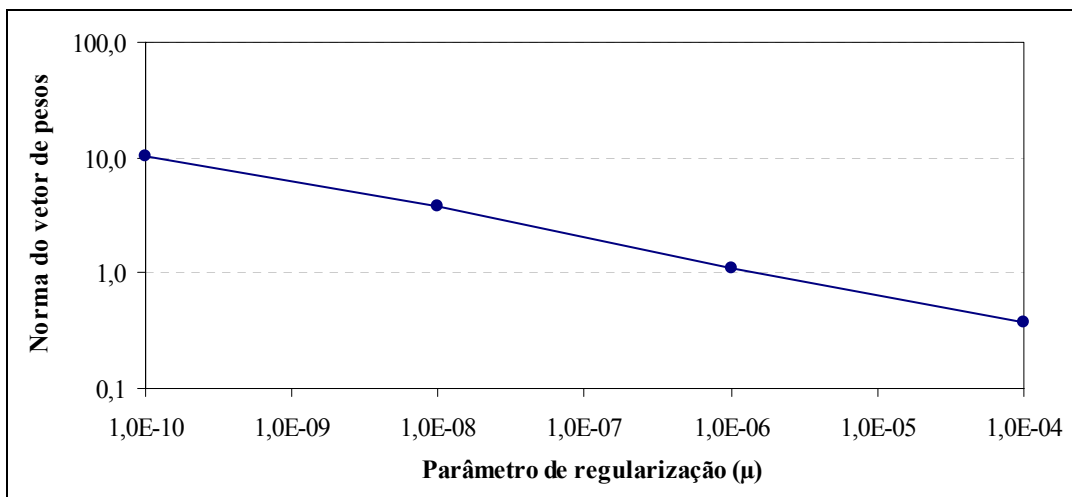


Figura 6-10 - Norma do vetor de pesos da camada de saída, valor do parâmetro de regularização (μ)

Na Figura 6-9 pode-se observar que o aumento do valor do parâmetro de regularização tende a produzir uma queda no valor absoluto dos pesos. Na verdade, a norma quadrática do vetor de pesos é que sempre decresce porque é minimizada na função objetivo de treinamento

da rede (Ver Figura 6-10). Embora não possa ser mostrado nesta última, em função da escala logarítmica para a abscissa, o valor da norma dos coeficientes para o caso $\mu=0$ fica em torno de 700.

Na Figura 6-11 mostra-se a evolução do número de condição para alguns valores do parâmetro de regularização, no caso da rede *feedforward* correspondente ao melhor indivíduo da cada geração. O valor da abscissa é mostrado em épocas sendo igual ao número de avaliações da função objetivo. A cada geração do *GA* foram efetuadas 30 avaliações de função objetivo (a população considerada foi de 30 indivíduos).

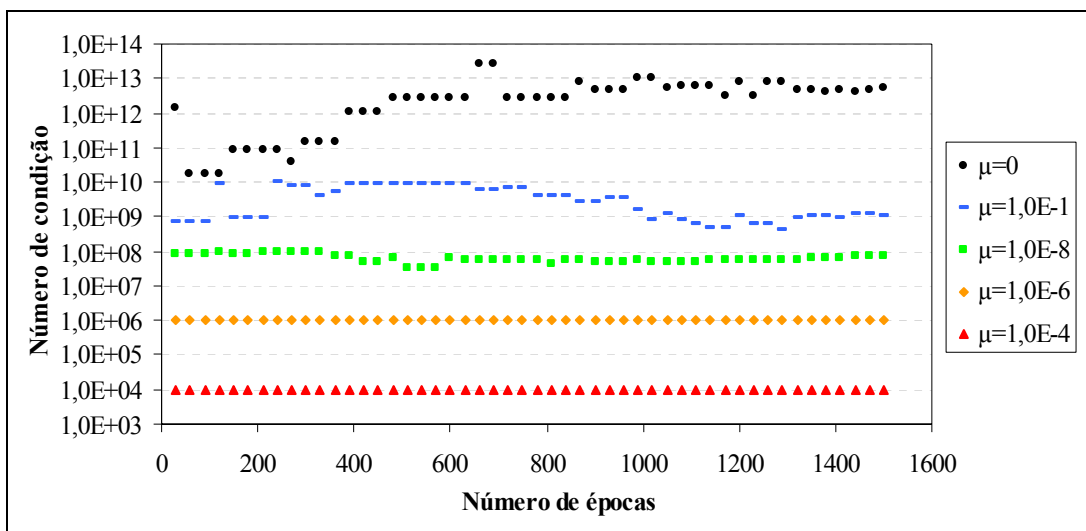


Figura 6-11 - Número de condição da matriz de covariância para alguns valores do parâmetro de regularização (μ)

Nota-se na Figura 6-11 que o aumento do parâmetro de regularização, para uma dada época, diminui o número de condição da matriz de covariância, conforme esperado. Para o caso $\mu=0$, o número de condição aumenta em função do número de épocas, ou seja, as saídas dos neurônios da camada intermediária ficam cada vez mais correlacionadas à medida que o treinamento avança. Como resultado, os coeficientes da camada de saída podem atingir algumas ordens de grandeza.

Redes do tipo *feedforward* são *frames*, tais como as redes de base radial, o que explica o valor extremamente alto do número de condição, no caso sem regularização.

Na Figura 6-12 pode-se observar a evolução do valor de $100-R^2$ em função do número de épocas. Claramente, o resultado obtido para $\mu=1E-4$ é insatisfatório causando um aumento excessivo do erro de aproximação. Este comportamento também ocorreu com a rede de base radial descrita na seção 6.6.1, cuja matriz de covariância sem regularização também era

extremamente mal condicionada. Para os outros valores de parâmetro de regularização, não é verdade que o aumento desse sempre resulte em um acréscimo do erro de aproximação. Esta foi a única diferença quando se compara os resultados obtidos com a rede de base radial e a *feedforward* para identificação do tanque de *pH*.

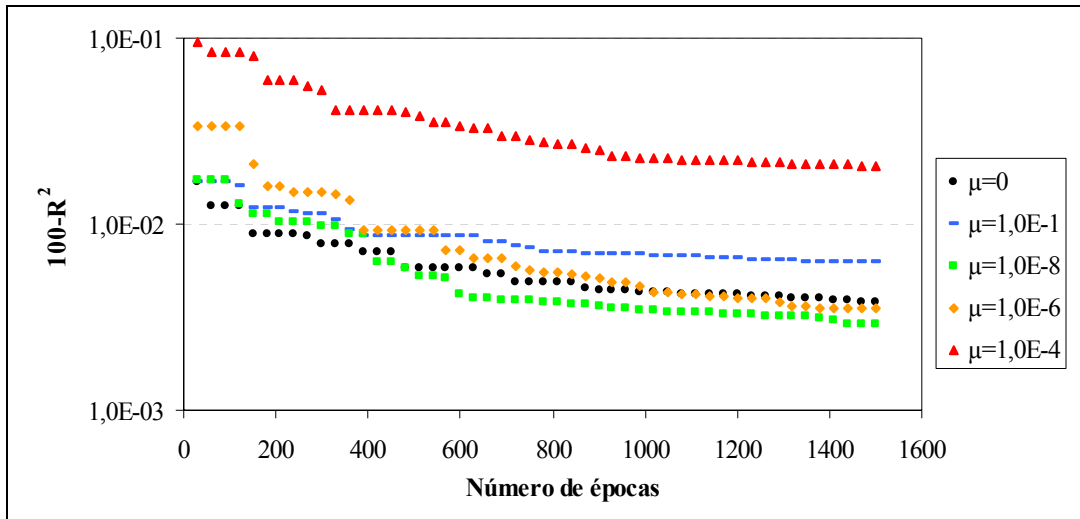


Figura 6-12 - $100-R^2$ x número de épocas para alguns valores do parâmetro de regularização (μ)

6.7 CONCLUSÕES

Problemas numéricos na estimação dos coeficientes de um modelo empírico podem ser minimizados se a função objetivo contiver, além do erro quadrático de aproximação, um termo que penalize a norma quadrática do vetor de coeficientes.

Um regularizador que penalize as curvaturas da superfície de um modelo pode ser obtido a partir do conhecimento das derivadas de segunda ordem desse, se estas estiverem definidas. Na prática, o cálculo do regularizador descrito, mesmo que possível, é bastante complexo e custoso computacionalmente.

A minimização da norma dos coeficientes de um modelo empírico também minimiza as derivadas de segunda ordem da(s) saída(s) desse em relação às variáveis de entrada, o que implica em limitar as curvaturas da superfície de previsão do modelo.

Para um modelo empírico, o compromisso entre aproximar os dados, que corresponde à minimização do erro quadrático de aproximação na função objetivo para estimação dos coeficientes, e eliminar excessivas curvaturas da superfície do modelo, que corresponde ao

termo que minimiza a norma do vetor de coeficientes, impõe restrições no valor do parâmetro de regularização, ou seja, o fator que multiplica esse último termo.

O aumento do fator de penalização da norma dos coeficientes de um modelo empírico, na função objetivo de treinamento, implica na minimização das curvaturas da superfície de previsão do modelo, no entanto, ocorre um acréscimo do valor do erro quadrático de aproximação.

Redes de base radial e do tipo *feedforward* são *frames* e a rede de funções de escala é uma base para um espaço de funções. A estimação dos coeficientes no último caso é menos suscetível a problemas numéricos, uma vez que as funções de uma base são, por definição, linearmente independentes. Esta característica não é, em geral, verdadeira no caso de um *frame*.

Para modelos empíricos que não são lineares nos parâmetros, como é o caso da rede *feedforward*, mostrou-se que a utilização da regularização pela norma dos pesos produz resultados semelhantes aos observados no caso de modelos lineares nos parâmetros, tais como as redes de funções de escala e de base radial.

7 COMPARAÇÃO DE DESEMPENHO DE ARQUITETURAS DE REDES NEURAIAS

O desempenho de redes *feedforward*, de base radial e a de funções de escala proposta foi avaliado em relação à previsão do comportamento de três sistemas dinâmicos: Um reator biológico, um atrator caótico e o tanque de *pH* descrito na seção 5.5.

Após as redes terem sido treinadas, ordenaram-se os pesos da camada de saída de cada uma delas, em termos de importância na redução do erro de aproximação, utilizando a técnica de mínimos quadrados ortogonais KORENBERG *et. al.*(1992). Com estas informações foram levantadas curvas do erro de aproximação de treinamento em função do número de parâmetros ajustáveis da camada de saída. O desempenho de previsão das redes também foi avaliado para um grupo de dados de teste.

As funções de ativação utilizadas nas redes de base radial e *feedforward* foram as descritas no capítulo 6.

7.1 Identificação do Atrator Caótico

O atrator caótico é descrito pelo mapeamento quadrático (equação logística):

$$y(k+1) = \alpha y(k)[1 - y(k)] \quad (7.1)$$

que é oscilatório para $\alpha \leq 3,7$ e apresenta comportamento caótico para $\alpha > 3,7$ BAKSHI e STEPHANOPOULOS(1993) (Ver Figura 7-1).

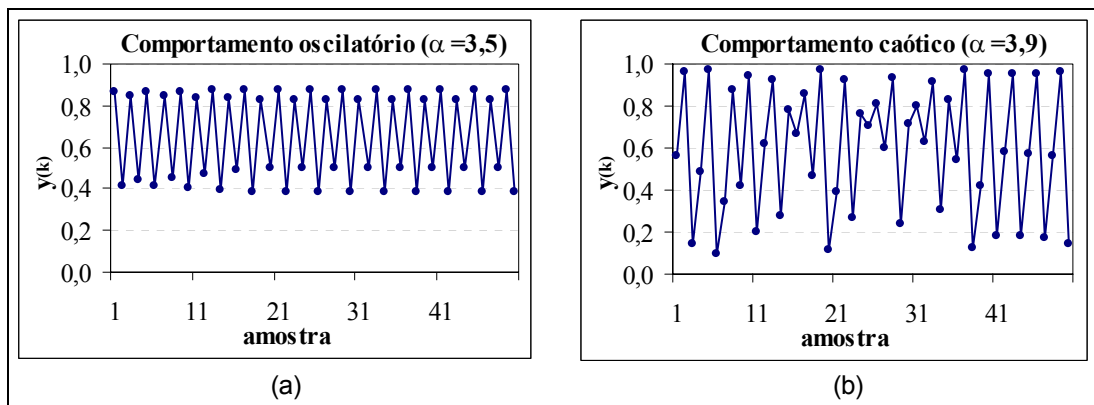


Figura 7-1 - (a) Comportamento oscilatório ($\alpha= 3,5$); (b) Comportamento caótico ($\alpha= 3,9$)

A equação logística representa a dinâmica de um sistema biológico simples. A população no tempo futuro $y_{(k+1)}$ é proporcional à população no instante k , $y_{(k)}$, e aos recursos disponíveis no mesmo instante, descrito de forma simplificada por $[I-y_{(k)}]$. Dessa forma, a equação logística é relevante no estudo de sistemas bioquímicos.

Para incorporar uma entrada externa sem alterar as características principais do sistema, o parâmetro α foi alterado e, portanto, tem-se $\alpha = \alpha_{(k)}$.

O objetivo estabelecido foi prever a população futura $y_{(k+1)}$, conhecendo-se o valor atual dessa $y_{(k)}$ e da variável de perturbação $\alpha_{(k)}$. Neste caso, a rede tem duas entradas e uma saída.

Para obtenção dos dados de treinamento calculou-se recursivamente a saída da equação logística para 400 pontos. Os 200 primeiros foram utilizados para treinamento e o restante foi utilizado para o teste. O parâmetro α foi alterado a cada iteração, tomando um valor aleatório entre 3,5 e 4, o que faz o sistema entrar e sair do caos.

As redes de base radial, do tipo *feedforward* e de funções de escala foram comparadas com relação ao erro de aproximação de treinamento, ao critério de Akaike e ao erro de aproximação para um grupo de dados de teste, na identificação do atrator caótico.

O critério para determinação da estrutura das redes utilizado foi baseado no desempenho da rede de funções de escala. Uma vez definido o número de parâmetros ajustáveis por treinamento para esta rede, foram utilizadas redes de base radial e do tipo *feedforward* com o mesmo número de parâmetros ajustáveis.

O número de parâmetros ajustáveis da rede de funções de escala, que é igual ao número de neurônios da rede e pesos da camada de saída, foi obtido considerando a resolução em que o erro de treinamento era pequeno e não mais variasse de forma significativa. Dessa forma, a quantidade de parâmetros ajustáveis é igual ao somatório do número de neurônios contidos em cada resolução utilizada.

Conforme o critério estabelecido, o número de centros para uma rede de base radial é igual ao número de neurônios da rede de funções de escala, uma vez que cada centro está conectado a um parâmetro ajustável, ou seja, um peso da camada de saída.

Para a rede *feedforward*, o número de neurônios da camada intermediária foi definido de forma que o número total de parâmetros ajustáveis entre uma saída e suas entradas seja igual, ou o mais próximo possível, do número de parâmetros ajustáveis utilizado no caso da rede de funções de escala. Dessa forma, considera-se tanto os pesos da camada de saída quanto os da camada intermediária.

Na Tabela 7.1 mostra-se a estrutura de cada rede para 25 parâmetros ajustáveis. Este número de parâmetros corresponde a uma rede de funções de escala com dois níveis de resolução.

Tabela 7.1 - Estrutura das redes utilizadas na identificação do atrator caótico

Rede de funções de escala
Nº de neurônios = 25
Nº de níveis = 2
Nº de parâmetros ajustáveis = 25
Rede de base radial
Nº de centros = 25
Nº de parâmetros ajustáveis = 25
Rede <i>feedforward</i>
Nº de neurônios da camada intermediária = $N_{Intm} = 6$
Nº de parâmetros ajustáveis (camada intermediária) = $N_{Intm} \times (N^\circ \text{ entradas} + 1^{(*)}) = 18$
Nº de parâmetros ajustáveis (camada saída) = $N_{Intm} + 1 = 7^{(*)}$
Nº total de parâmetros ajustáveis = 25
(*) Inclui bias

Na Figura 7-2 mostra-se a variação do valor de $100-R^2$ para as redes testadas em função do número de parâmetros ajustáveis. Para as redes de base radial e de funções de escala, cada parâmetro ajustável corresponde ao peso de um neurônio. Considerando estas redes, a Figura 7-2 pode ser interpretada da seguinte forma: para um determinado número N ($N=1,.. 25$, o número total de parâmetros ajustáveis) calculou-se o valor de $100-R^2$ da rede formada com apenas os N neurônios mais importantes, em termos da minimização do erro de aproximação. Os N neurônios mais importantes foram determinados através da técnica de mínimos quadrados ortogonais. Variando N de 1 a 25, gerou-se as curvas das redes de base radial e de funções de escala mostradas na Figura 7-2.

No caso da rede *feedforward*, a interpretação da Figura 7-2 pode ser feita de forma semelhante: para um determinado número N ($N=1,.. 7$, o número de neurônios da camada intermediária mais o bias de saída) calculou-se o valor de $100-R^2$ da rede formada apenas dos pesos conectados aos N aproximadores mais importantes (que podem ser neurônios da camada intermediária ou o bias da saída) na minimização do erro quadrático de treinamento. Os N aproximadores mais importantes foram determinados através da técnica de mínimos

quadrados ortogonais, uma vez que a camada de saída é linear nos parâmetros. Variando N de 1 a 7 gerou-se a curva da rede *feedforward* mostrada na Figura 7-2.

Deve-se notar que a incorporação de um neurônio na camada intermediária da rede *feedforward* equivale ao crescimento da rede em um número de parâmetros ajustáveis igual ao número total de pesos conectados a esse.

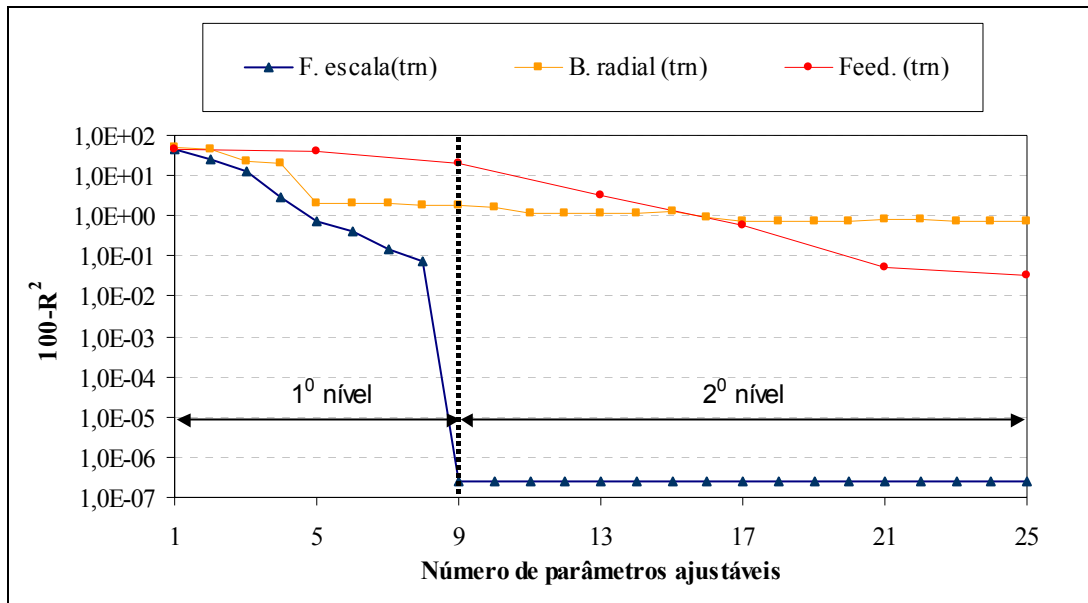


Figura 7-2 - O valor de $100-R^2$ em função do número de parâmetros ajustáveis para as redes testadas (treinamento)

Na Figura 7-2 pode-se observar que foi obtido o pior resultado, em termos de erro de aproximação, com a rede de base radial. Para este modelo, o valor de $100-R^2$, diminui significativamente até 5 neurônios. No caso da rede de *feedforward*, obtém-se um melhor desempenho se considerada a parte final da curva de $100-R^2$, sendo que pela tendência dessa curva é possível concluir que adicionando mais neurônios na camada intermediária, o erro de aproximação ainda pode ser diminuído. Para a rede de funções de escala duas características podem ser destacadas: O valor muito pequeno de $100-R^2$, quando se compara esse aos outros tipos de modelos, e a inflexão da curva de $100-R^2$ para 9 neurônios, correspondente ao primeiro nível de resolução.

A rede de funções de escala é uma base para um espaço de funções e as características descritas indicam que o atrator caótico pertence, ou seja, é um elemento do subespaço gerado pelo primeiro nível de resolução da rede proposta neste trabalho.

Para validação de modelos vários critérios podem ser utilizados, tais como: Critérios estatísticos, curvas de correlação, previsão utilizando outro grupo de dados diferente do de treinamento (validação cruzada). Neste capítulo foram adotados como métodos de comparação e validação de modelos, o critério de Akaike (*AIC*) e a validação cruzada.

O critério de Akaike é dado pela seguinte equação:

$$AIC = N \ln \left[\frac{\sum_{k=1}^N E^2(k)}{N} \right] + 4N_{\theta}$$

onde

N : número de pontos do grupo de dados;

$E(k)$: erro associado ao k -ésimo padrão;

N_{θ} : número de parâmetros;

O critério de Akaike é constituído de dois termos. Um associado ao erro médio quadrático de aproximação, sendo que diminuindo o valor desse último diminuirá o valor do critério. O outro é relacionado ao número de parâmetros do modelo e cujo valor diminui com o decréscimo do número de parâmetros. Segundo o critério de Akaike, quando dois modelos são comparados, o melhor é aquele que tiver o menor valor nesse índice desempenho.

O critério de Akaike pode ser utilizado para comparar modelos com a mesma estrutura, mas com um número diferente de parâmetros e, portanto, pode ser aplicado para determinar o número ótimo de parâmetros.

O critério de Akaike também pode ser utilizado para comparar modelos com estruturas diferentes como o feito neste trabalho, em que o desempenho das redes de base radial, de funções de escala e *feedforward* foi avaliado. A comparação foi efetuada considerando o melhor modelo de cada estrutura, ou seja, aquele que obteve o menor valor de *AIC*. Na Figura 7-3 mostra-se a variação do valor do *AIC* em função do número de parâmetros ajustáveis das redes testadas.

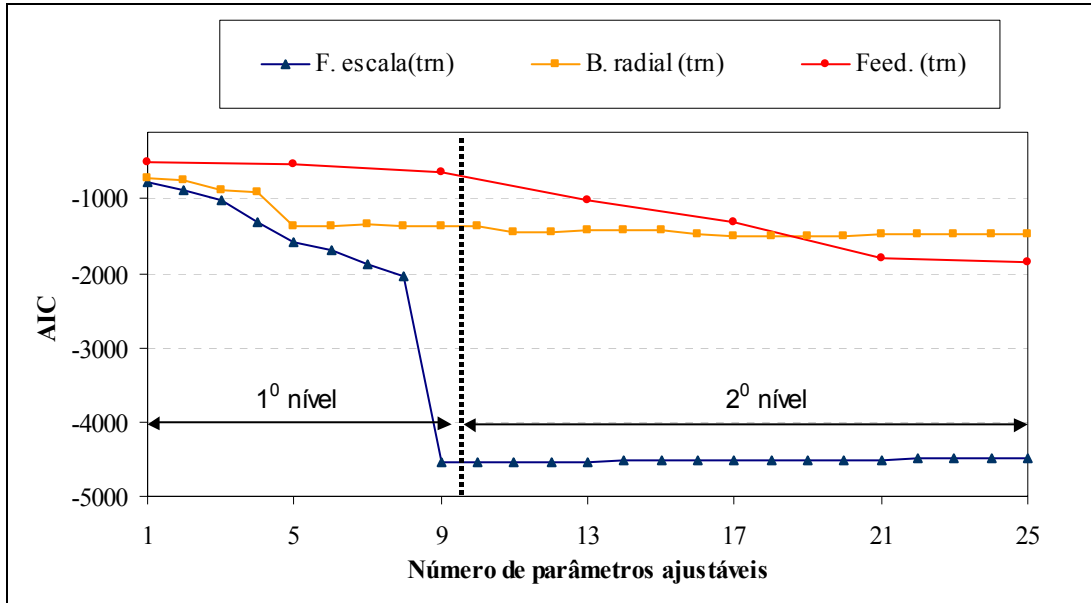


Figura 7-3 - Valor de AIC em função do número de parâmetros ajustáveis para as redes testadas

Analisando a Figura 7-3, pode-se concluir que a rede *feedforward* é um melhor modelo que a de base radial, quando se compara o menor valor das curvas de AIC , e a rede de funções de escala é muito melhor que ambas.

Na Figura 7-4 mostra-se o valor de $100-R^2$ para o grupo de dados de teste em função do número de parâmetros ajustáveis. Este teste tem por objetivo verificar a capacidade de generalização das redes descritas, uma vez que os dados são diferentes dos utilizados no treinamento e, conseqüentemente, não influenciaram no cálculo dos pesos. Pode-se observar que a rede de funções de escala é muito melhor que as redes *feedforward* e de base radial, o que está de acordo com o critério de Akaike. Outra característica importante é que, para todos os modelos, aumentando o número de parâmetros, o valor de $100-R^2$, em geral, diminui. Dessa forma, pode-se concluir que foi evitado o problema de sobre-ajuste pelo excesso do número de funções. No caso da rede *feedforward*, utilizou-se a metodologia de *GA* + mínimos quadrados para treinamento, sendo que essa foi finalizada quando $100-R^2$ não mais se alterou. Assim sendo, foi evitado o sobre-treinamento por excesso de épocas. Dessa forma, pode-se concluir que a penalização da norma do vetor de pesos, adicionada ao erro quadrático de aproximação, resultou na minimização de excessivas curvaturas das superfícies dos modelos utilizados na identificação do atrator caótico.

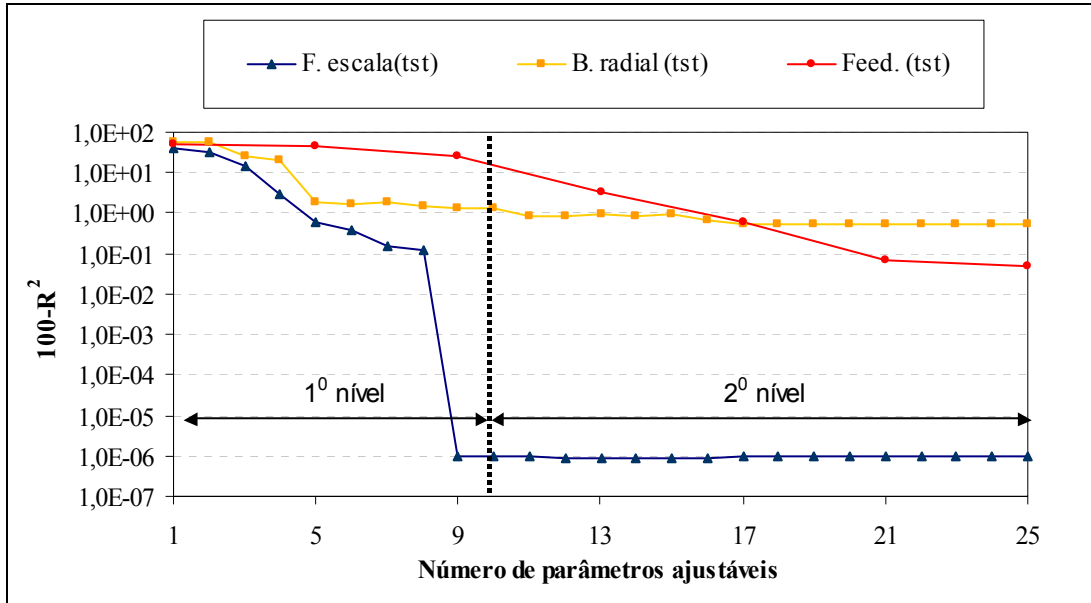


Figura 7-4 - O valor de $100-R^2$ em função do número de parâmetros ajustáveis para as redes testadas (Teste)

7.2 Identificação do Reator Biológico

O processo a ser identificado é um reator biológico em que as células sofrem inibição pelo aumento da concentração de produto. O modelo do reator biológico apresenta não-linearidades de ganho e pode ser representado através de um sistema de três equações diferenciais:

$$\dot{x} = -dx + \mu x$$

$$\dot{s} = -d(s_f - s) - \frac{1}{y_{x/s}} \mu x$$

$$\dot{p} = -dp + (\alpha\mu + \beta)x$$

$$\mu = \frac{\mu_m \left(1 - \frac{P}{p_m}\right) s}{k_m + s + \frac{s^2}{k_i}}$$

onde

d - vazão de alimentação do reator biológico (variável de perturbação);

s_f - concentração de substrato na alimentação;

p, x, s - Concentração de produto, células e substrato, respectivamente;
 μ, μ_m - taxa específica de crescimento instantânea e máxima, respectivamente;
 $y_{x/s}$ - rendimento em células/substrato consumido;
 α e β - parâmetros cinéticos;
 p_m, k_m - constante de saturação do produto e substrato, respectivamente;
 k_i - constante de inibição do substrato.

Estas equações, assim como os valores dos parâmetros (Ver tabela Tabela 7.2), foram extraídos do trabalho de HENSON e SEBORG,(1991).

Tabela 7.2 - Parâmetros do reator biológico

Parâmetros:
$y_{x/s} = 0,4 \text{ g/g};$
$\alpha = 2,2 \text{ g/g};$
$\beta = 0,2 \text{ h}^{-1};$
$\mu_m = 0,48 \text{ h}^{-1};$
$p_m = 50 \text{ g/l};$
$k_m = 1,2 \text{ g/l};$
$k_i = 22 \text{ g/l};$
$s_f = 20 \text{ g/l};$
d - (variável de perturbação).
Estado estacionário: ($d = 0,202 \text{ g/l}$)
$x = 6,0 \text{ g/l} \quad s = 5,0 \text{ g/l} \quad p = 19,14 \text{ g/l}.$

As variáveis de estado correspondem à concentração de substrato, de células e de produto. A variável de perturbação utilizada foi a vazão de alimentação do reator (d). As perturbações de entrada foram pulsos aleatórios com amplitude máxima de 20 % em torno do estado estacionário $d = 0,202 \text{ l/h}$ e largura de 6 intervalos de amostragem. (Ver Figura 7-5). Em um estudo prévio foi verificado que pulsos de largura constante eram suficientes para identificação do sistema.

O objetivo estabelecido foi prever o valor futuro (um passo à frente) das variáveis de estado, o que implica em três saídas. Em um estudo prévio determinou-se que apenas os valores atuais dos estados e da variável de perturbação eram suficientes para identificar o reator biológico. Dessa forma, as redes comparadas foram compostas por quatro entradas. Nas

Figuras 7.6 a 7.8 mostra-se o comportamento do processo para as perturbações aplicadas a esse.

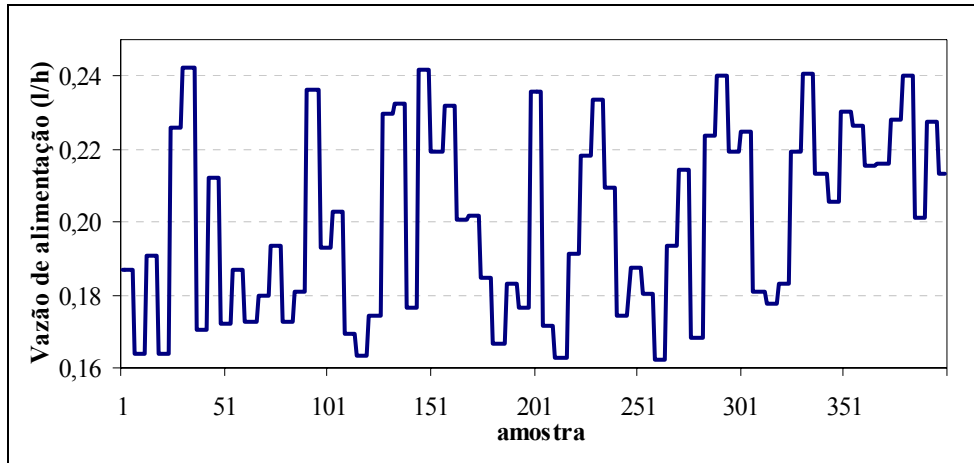


Figura 7-5 - Perturbações aplicadas na vazão de alimentação

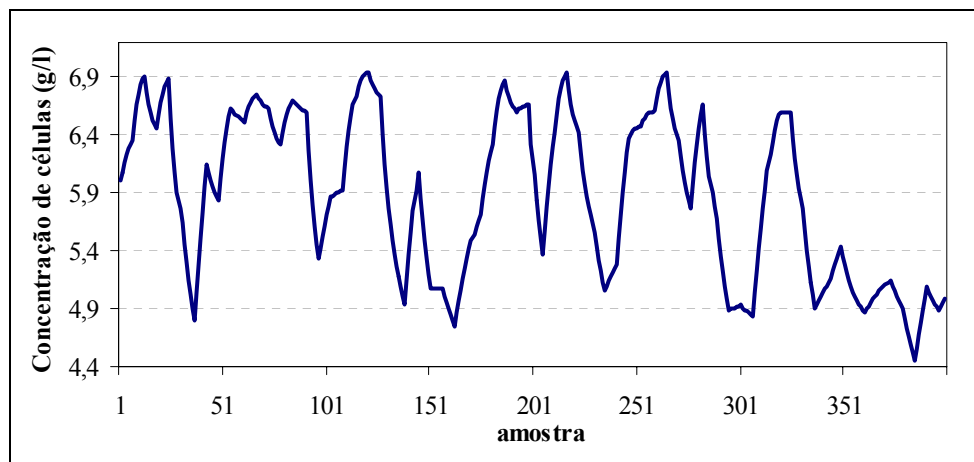


Figura 7-6 - Concentração de células

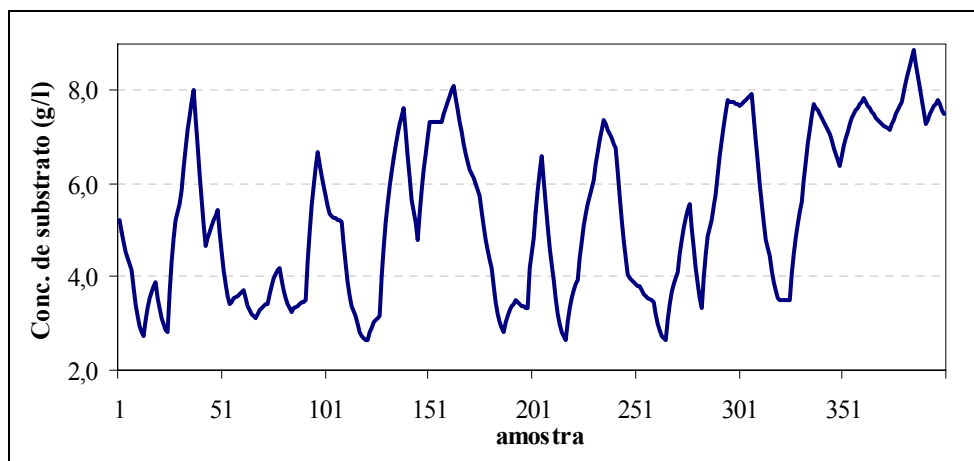


Figura 7-7 - Concentração de substrato

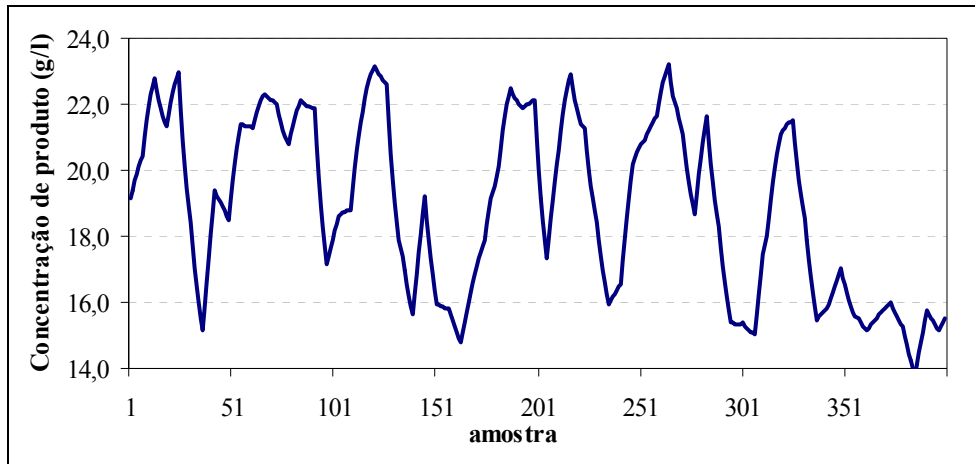


Figura 7-8 - Concentração de produto

Além de o reator biológico apresentar não-linearidade de ganho, há outra dificuldade para sua identificação: As concentrações de células e de produto são fortemente correlacionadas. Para avaliar o desempenho do modelo obtido, gerou-se um grupo dados de teste e efetuou-se a validação. As perturbações aplicadas na vazão de alimentação d foram pulsos aleatórios com amplitude máxima de 10% em torno do estado estacionário ($d = 0,202$ l/h) e largura de 40 intervalos de amostragem. Na Figura 7-9 mostram-se as perturbações aplicadas na vazão de alimentação.

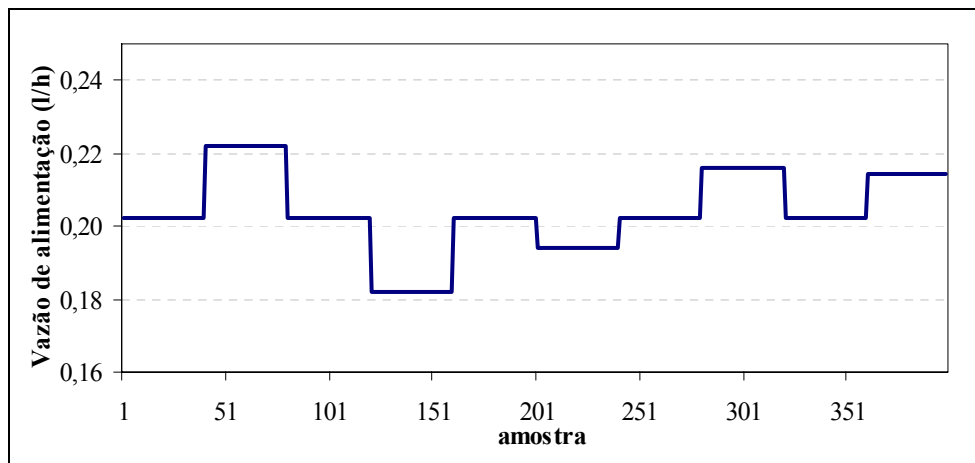


Figura 7-9 - Perturbações aplicadas na vazão de alimentação

Nas Figuras 7.10 a 7.12 mostram-se as saídas do grupo de dados de teste.

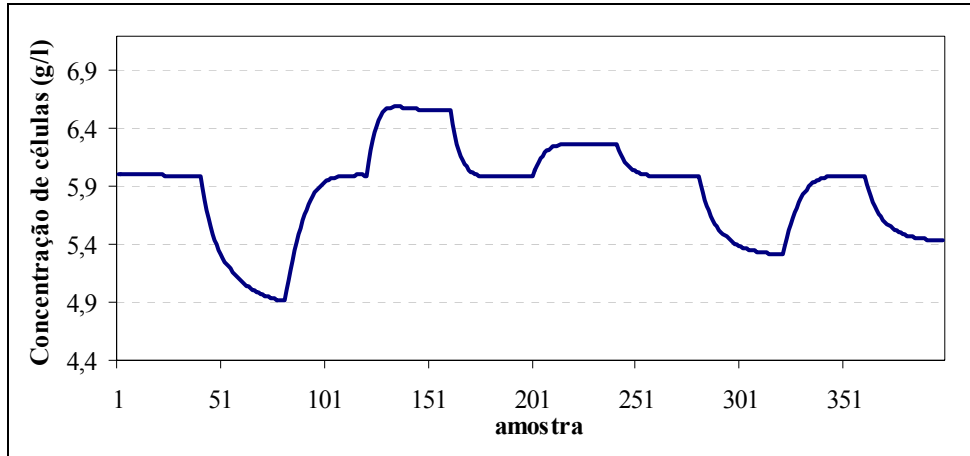


Figura 7-10 - Dados da saída (concentração de células)

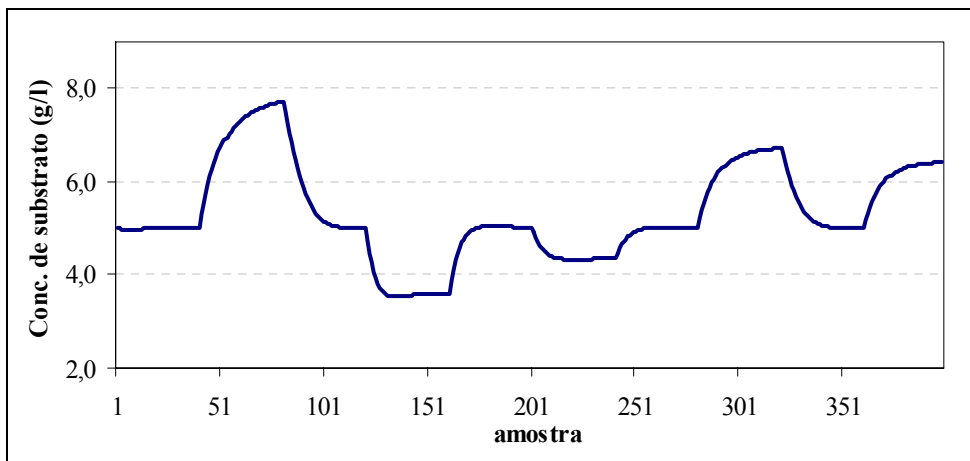


Figura 7-11 - Dados da saída (concentração de substrato)

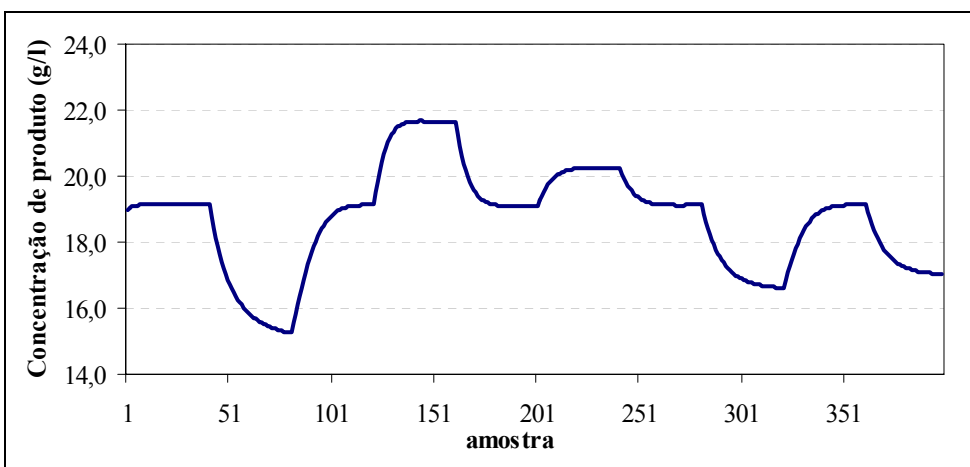


Figura 7-12 - Dados da saída (concentração de produto)

Na Tabela 7.3 pode-se observar a estrutura das redes de funções de escala, *feedforward* e de base radial utilizadas na identificação do reator biológico. Foi considerado, na determinação do número de parâmetros ajustáveis, o critério de identificação baseado na rede de funções de escala.

Tabela 7.3 - Estrutura das redes utilizadas na identificação do reator biológico

Rede de funções de escala
Nº de neurônios = 81
Nº de níveis = 1
Nº de parâmetros ajustáveis = 81
Rede de base radial
Nº de centros = 81
Nº de parâmetros ajustáveis = 81
Rede <i>feedforward</i>
Nº de neurônios da camada intermediária = $N_{esc} = 13$
Nº de parâmetros ajustáveis (camada intermediária) = $N_{esc} \times (N^\circ \text{ entradas} + 1) = 65$ (*)
Nº de parâmetros ajustáveis (camada saída) = $N_{esc} + 1 = 14$ (*)
Nº total de parâmetros ajustáveis = 79 (**)
(*) Inclui bias ;
(**) Rede <i>feedforward</i> cujo número de parâmetros ajustáveis mais se aproxima de 81.

Nas Figuras 7.13 e 7.15 mostra-se a variação do valor de $100-R^2$ em função do número de parâmetros ajustáveis das redes, considerando o treinamento e o teste respectivamente. Na Figura 7-14 mostra-se a variação do valor de *AIC* em função do número de parâmetros ajustáveis das redes testadas.

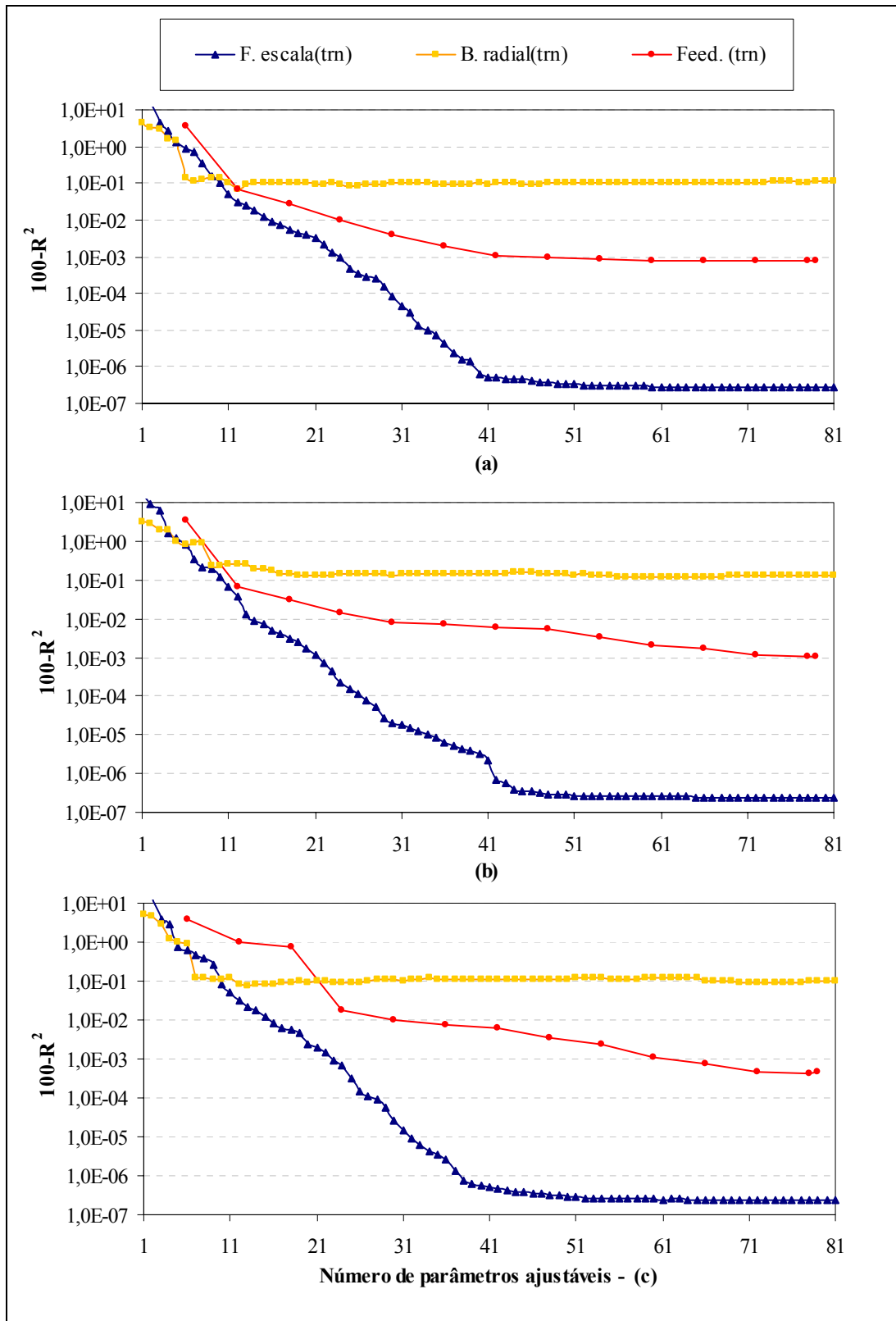


Figura 7-13 - O valor de $100-R^2$ x número de parâmetros ajustáveis das redes (treinamento)

(a) concentração de células; (b) concentração de substrato; (c) concentração de produto

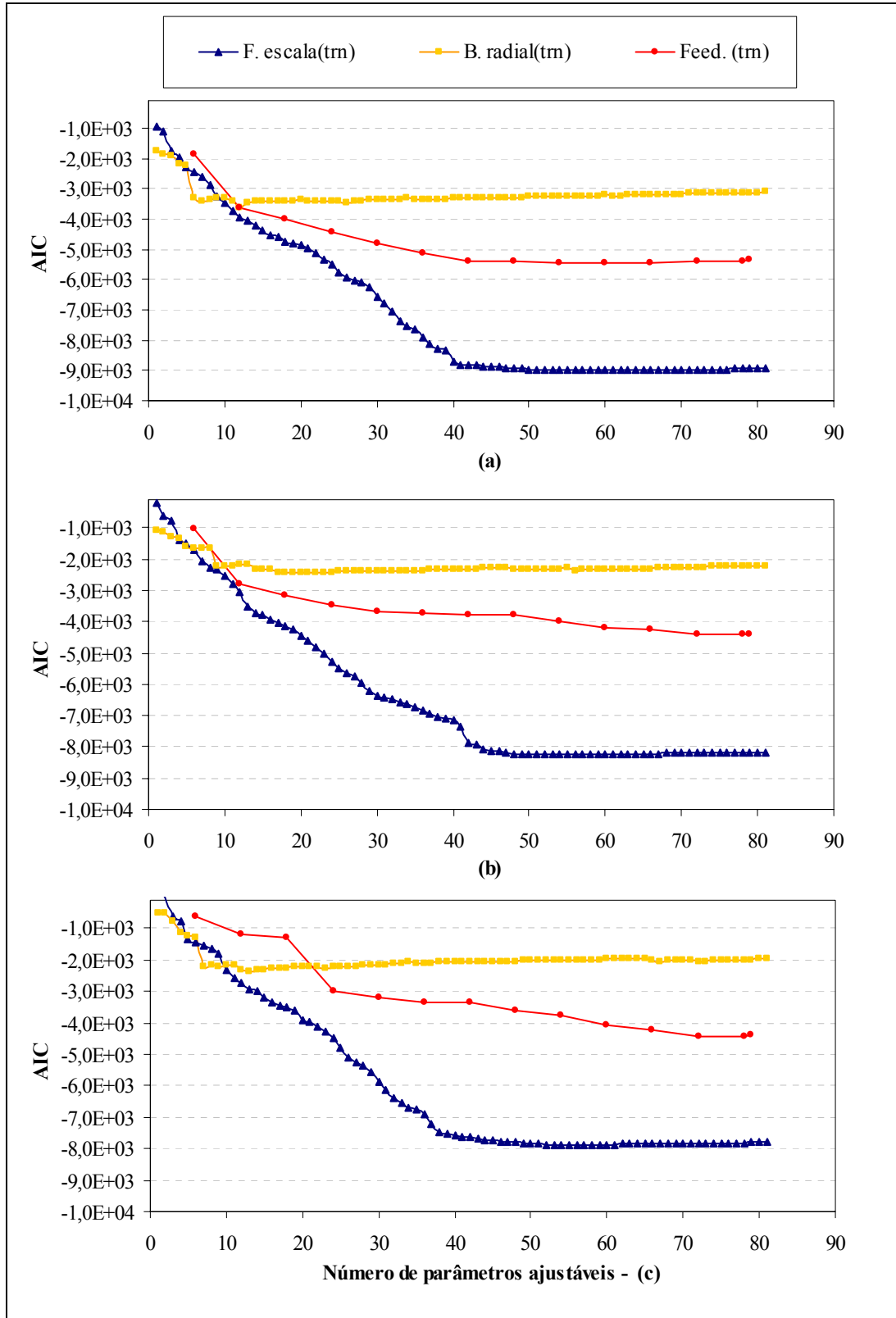


Figura 7-14 - O valor de AIC_x número de parâmetros ajustáveis das redes (treinamento) (a) concentração de células; (b) concentração de substrato; (c) concentração de produto

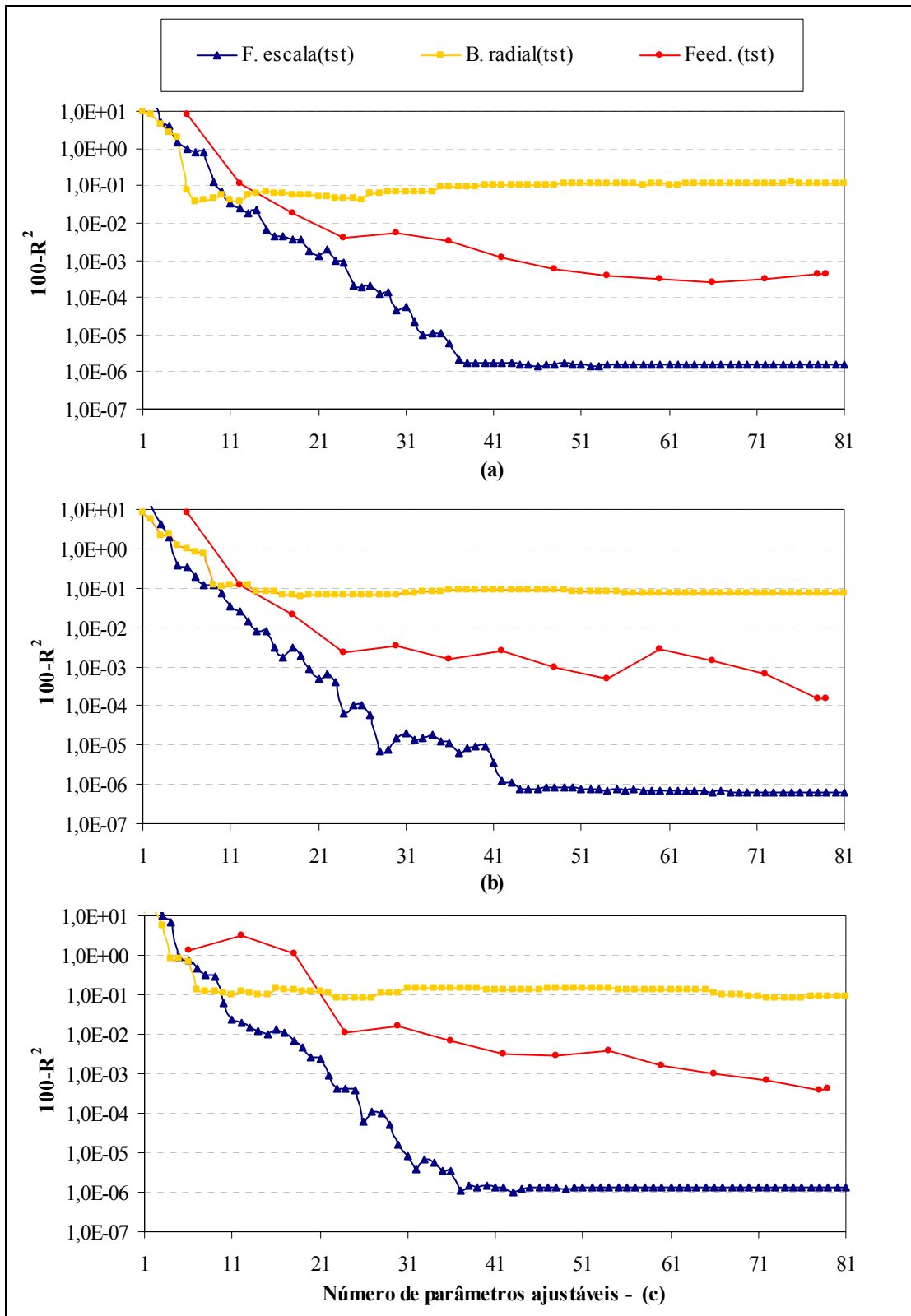


Figura 7-15 - O valor de $100-R^2$ x número de parâmetros ajustáveis das redes (teste)
 (a) concentração de células; (b) concentração de substrato; (c) concentração de produto

Analisando as Figuras 7.13 a 7.15 podem-se obter conclusões bastante semelhantes às do atrator caótico. Foram obtidos os piores resultados para todas as saídas, em termos de erro de aproximação de treinamento, teste e do critério de Akaike, com a rede de base radial. No caso da rede de *feedforward*, obtém-se um melhor desempenho, quando se compara com a de base radial. Pela tendência das curvas, é possível concluir que o desempenho da primeira pode ser melhorado adicionando mais neurônios na camada intermediária. Para a rede de funções de escala podem-se destacar os baixos valores do erro de aproximação de treinamento e de teste, quando compara esse aos outros tipos de modelos. Em termos do critério de Akaike, a rede de funções de escala também foi bastante superior. Pode-se observar que no caso deste modelo houve uma inflexão, para todos os critérios, em torno de 40 neurônios.

A rede de funções de escala é uma base para um espaço de funções e as características descritas indicam que o modelo do reator biológico, na faixa dos dados utilizados, pertence ao espaço gerado pelo primeiro nível de resolução, tal como o atrator caótico. Assim sendo, a proposta deste trabalho de começar a multiresolução com um pequeno número de funções de escala foi válida em função dos bons resultados obtidos, em termos de identificação, para os sistemas testados.

Na Figura 7-15 pode-se observar que, para todas as redes neurais, aumentando o número de parâmetros (pesos), determinados a partir do grupo de dados de treinamento, há uma tendência para que o valor de $100-R^2$, correspondente ao grupo de dados de teste, em geral diminua. Dessa forma, a regularização pela norma dos pesos garantiu a capacidade de generalização dos modelos obtidos mesmo havendo forte correlação entre as variáveis de entrada.

7.3 Identificação do Tanque de *pH*

O processo identificado foi o tanque de *pH* descrito na seção 5.3. Como grupo de dados de treinamento foram utilizados os primeiros 800 pontos correspondentes às Figuras 5.1 e 5.2 e para o teste os dados restantes.

Na

Tabela 7.4 mostra-se a estrutura de cada rede, considerando o critério de identificação baseado na rede de funções de escala para determinar o número de parâmetros ajustáveis.

Foram utilizados 3 níveis de resolução na identificação do tanque de pH no caso da rede de funções de escala.

Tabela 7.4 - Estrutura das redes utilizadas na identificação do tanque de pH

Rede de funções de escala
Nº de neurônios = 61
Nº de níveis = 3
Nº de parâmetros ajustáveis = 61
Rede de base radial
Nº de centros = 61
Nº de parâmetros ajustáveis = 61
Rede <i>feedforward</i>
Nº de neurônios da camada intermediária= $N_{esc} = 15$
Nº de parâmetros ajustáveis (camada intermediária) = $N_{esc} \times (N^\circ \text{ entradas} + 1) = 45$ (*)
Nº de parâmetros ajustáveis (camada saída)= $N_{esc} + 1 = 16$ (*)
Nº total de parâmetros ajustáveis = 61

(*) Inclui bias

Nas Figuras 7.16 e 7.18 mostra-se a variação do valor de $100-R^2$ em função do número de parâmetros ajustáveis das redes comparadas, considerando o treinamento e o teste respectivamente. Na Figura 7-17 mostra-se a variação do valor de AIC em função do número de parâmetros ajustáveis das redes.

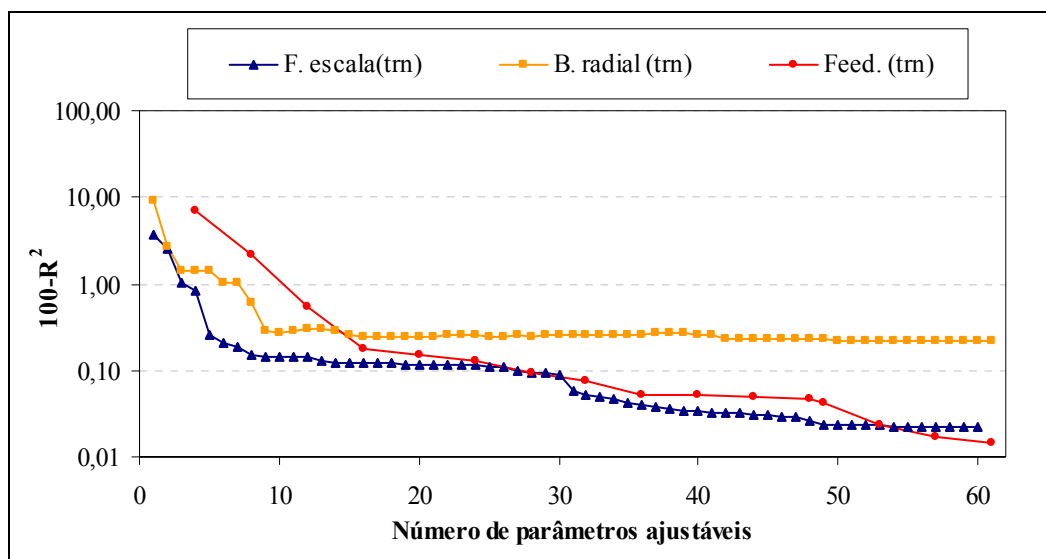


Figura 7-16 - $100-R^2$ x número de parâmetros ajustáveis (treinamento)

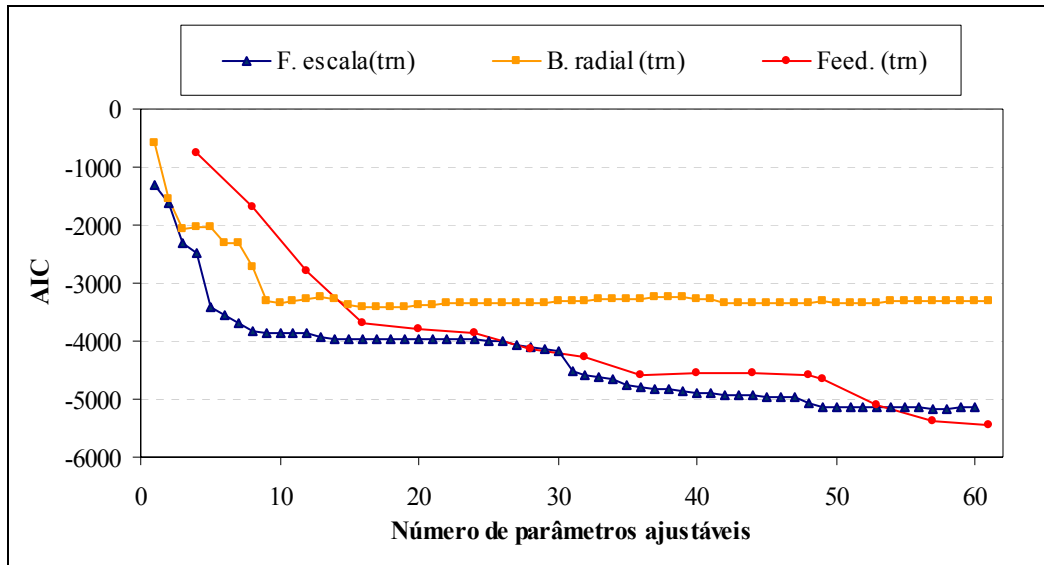


Figura 7-17 - AIC \times número de parâmetros ajustáveis (treinamento)

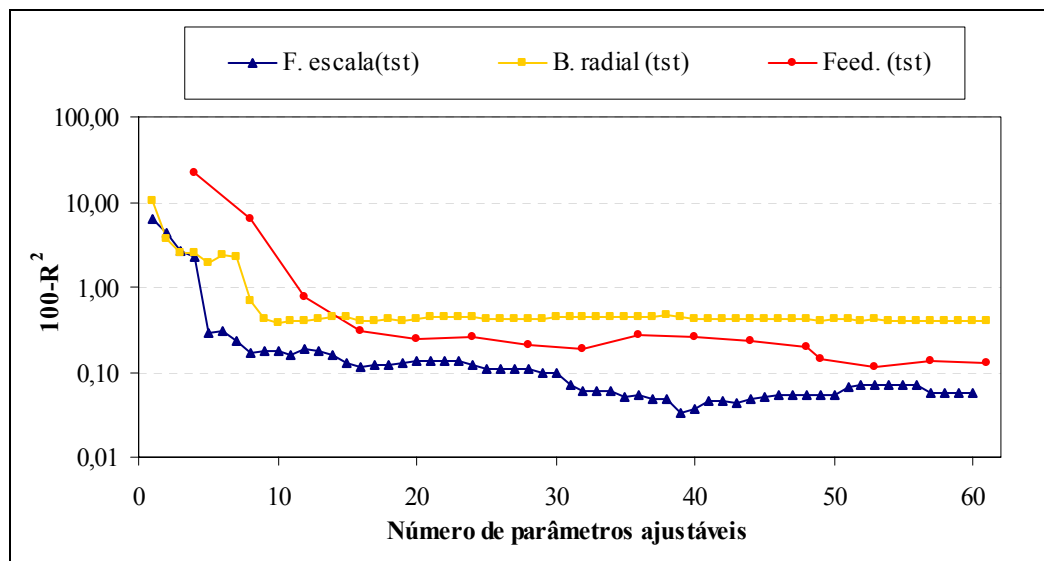


Figura 7-18 - $100-R^2$ \times número de parâmetros ajustáveis (teste)

Analisando as Figuras 7.16 a 7.18, pode ser observado que resultados tão bons, como os conseguidos com os outros sistemas dinâmicos testados, nem sempre são obtidos com a rede de funções de escala. Na identificação do tanque de pH não foi atingido um valor de $100-R^2$ tão pequeno quanto no caso do atrator caótico e do reator biológico. Também não foram observadas inflexões tão marcantes nas curvas de erro de aproximação de treinamento, teste e do critério de Akaike como apresentadas para os outros processos.

Para identificar o comportamento do tanque de pH , um modelo deve identificar tanto o comportamento dinâmico quanto o estacionário, sendo que esse último é semelhante a uma tangente hiperbólica (curva de titulação). Pode-se concluir que para este tipo particular de não-linearidade, a rede de funções de escala necessita de um número maior de níveis de aproximação. Mesmo assim, a rede *feedforward*, que é composta de neurônios cujas funções de ativação são tangentes hiperbólicas, não obteve desempenho superior ao da rede de funções de escala em termos dos critérios de validação. (Ver Figuras 7.16 a 7.18).

A rede de funções de escala é caracterizada pela soma e produtório de funções quadráticas unidimensionais. Para muitos processos, o(s) estado(s) e/ou saída(s) podem ser descritos por tais aproximadores, o que explica o bom desempenho obtido para o atrator caótico e o reator biológico. Isto não ocorre no caso do tanque de pH , uma vez que o pH é obtido da solução de uma equação transcendental e não-linear em relação aos estados.

Em termos de capacidade de generalização das redes neurais, as conclusões obtidas para o atrator caótico e o reator biológico também são válidas para o tanque de pH .

7.4 Conclusões

Na identificação dos sistemas dinâmicos atrator caótico e reator biológico pode-se concluir que os piores resultados, em termos de erro de aproximação de treinamento, teste e do critério de Akaike, foram obtidos com a rede de base radial. No caso da rede *feedforward*, obteve-se um melhor desempenho quando se compara essa com a de base radial. Para a rede de funções de escala, pode-se destacar, em comparação com os outros modelos, os baixos valores do erro de aproximação de treinamento e de teste. Em termos do critério de Akaike, a rede de funções de escala também foi bastante superior.

Para alguns tipos particulares de não-linearidade, tal como a existente no comportamento de um tanque de pH , um maior número de níveis de resolução pode ser necessário para correta identificação do sistema.

No caso da rede de funções de escala, que é uma base para um espaço de funções, a presença de inflexões marcantes nas curvas do erro de aproximação de treinamento, teste e do critério de Akaike de sistemas dinâmicos indica que esses estão contidos nos subespaços gerados pelos diferentes níveis de resolução.

O atrator caótico e reator biológico pertencem ao subespaço gerado pelo primeiro nível de resolução da rede de funções de escala proposta.

O aumento do número de neurônios faz decrescer o erro de aproximação de treinamento, como esperado, sendo que, em geral, o mesmo ocorreu no caso do grupo de dados de teste. Dessa forma, a regularização pela norma dos pesos garantiu a capacidade de generalização das redes de base radial, *feedforward* e funções de escala.

8 PREDIÇÃO PARA HORIZONTE DE LONGO ALCANCE DO COMPORTAMENTO DE SISTEMAS DINÂMICOS

Na avaliação da capacidade de previsão de um passo à frente de modelos empíricos, são utilizados os valores atuais e passados das saídas do processo para prever as saídas da planta no próximo intervalo de amostragem, ou seja, um passo à frente. Um método de validação de modelos mais exigente do que o descrito é a simulação do comportamento do processo. Neste último, utiliza-se o(s) primeiro(s) ponto(s) do grupo de dados de validação como condição inicial para a rede. Para os demais pontos, utiliza-se apenas os dados das variáveis de perturbação como informação externa e faz-se a realimentação das saídas da rede.

Neste trabalho, o desempenho da rede de funções de escala foi avaliado utilizando essa como simulador de dois sistemas dinâmicos reconhecidos como *benchmarks* de identificação e controle não-linear: um *CSTR* encamisado que apresenta multiplicidade de estados estacionários e um reator que apresenta resposta inversa variável.

8.1 *CSTR* Encamisado

O processo a ser identificado é uma reação química exotérmica do tipo $A \rightarrow B$ de primeira ordem, ocorrendo em um *CSTR* com camisa de resfriamento. Este processo apresenta dois estados estacionários estáveis e um instável. O modelo matemático rigoroso do *CSTR* encamisado foi apresentado na literatura de diversas formas, sendo adotado neste trabalho o modelo utilizado por EMBIRUÇU(1993). O modelo do *CSTR* pode ser representado através de um sistema de três equações diferenciais que descrevem a evolução dos estados (concentração de reagente no reator, temperatura no reator e na camisa) e uma equação algébrica não-linear relacionando a constante da reação de primeira ordem com a temperatura no reator:

$$\frac{dx_1}{d\tau} = -\phi \cdot k \cdot x_1 + q \cdot (x_{1f} - x_1)$$

$$\frac{dx_2}{d\tau} = \beta \cdot \phi \cdot k \cdot x_1 - (q + \delta) \cdot x_2 + \delta \cdot x_3 + q \cdot x_{2f}$$

$$\frac{dx_3}{d\tau} = \frac{q_c \cdot (x_{3f} - x_3)}{\delta_1} + \frac{\delta \cdot (x_2 - x_3)}{\delta_1 \cdot \delta_2}$$

$$k = \exp\left[\frac{x_2}{1 + x_2/\gamma}\right]$$

O processo é descrito nas variáveis adimensionais:

$$\beta = \frac{-\Delta H \cdot C_f \cdot \gamma}{\rho \cdot C_p \cdot T_{f0}} \quad \phi = \frac{V \cdot k_0 \cdot \exp(-\gamma)}{Q_0}$$

$$x_3 = \frac{(T_c - T_{f0}) \cdot \gamma}{T_{f0}} \quad \delta = \frac{U \cdot A}{\rho \cdot C_p \cdot Q_0}$$

$$x_2 = \frac{(T - T_{f0}) \cdot \gamma}{T_{f0}} \quad \gamma = \frac{E}{R \cdot T_{f0}}$$

$$\tau = \frac{Q_0 \cdot t}{V} \quad x_{2f} = \frac{(T_f - T_{f0}) \cdot \gamma}{T_{f0}}$$

$$\delta_1 = \frac{V_c}{V} \quad x_1 = \frac{C}{C_{f0}}$$

$$\delta_2 = \frac{\rho_c \cdot C_{pc}}{\rho \cdot C_p} \quad x_{1f} = \frac{C_f}{C_{f0}}$$

$$x_{3f} = \frac{(T_{cf} - T_{f0}) \cdot \gamma}{T_{f0}} \quad q = \frac{Q}{Q_0}$$

A carga térmica é definida como: $m = q \cdot x_{2f} \Rightarrow x_{2f} = m/q$, sendo que:

ΔH - Calor de reação;

C - Composição no reator;

C_f - Composição de alimentação;

C_{f0} - Composição de referência de alimentação;

ρ - Densidade do meio reacional;

ρ_c - Densidade do fluido refrigerante;

C_p - Capacidade calorífica do meio reacional;

C_{pc} - Capacidade calorífica do fluido refrigerante;

T - Temperatura do reator;

T_f - Temperatura de alimentação do reator;

T_c - Temperatura do fluido refrigerante;

T_{c0} -Temperatura de alimentação do fluido refrigerante

E -Energia de ativação;

R - Constante universal dos gases;

U - Coeficiente global de troca térmica;

A - Área de troca térmica;

V - Volume do reator;

V_c -Volume da camisa;

k_0 -Fator de Frequência;

Q -Fluxo de Alimentação;

Q_0 -Fluxo de Alimentação de referência.

Os valores numéricos utilizados podem ser vistos na tabela Tabela 8.1:

Tabela 8.1 - Parâmetros do CSTR encamisado

Parâmetros
$\beta = 8,0;$
$x_{1f} = 1,0;$
$\phi = 0,072;$
$m = 0,2;$
$\delta = 0,3;$
$q = 1,0;$
$\gamma = 20,0;$
$x_3 = 0;$
$\delta_1 = 0,1;$
$\delta_2 = 0,5;$
$x_{3f} = -1,0;$
$q_c = 1,65102;$
d - (variável de perturbação).
Estados estacionários estáveis ($d = 1.0$)
1) ($x_1 = 0,8933, x_2 = 0,5193, x_3 = -0,5950$)
2) ($x_1 = 0,1890, x_2 = 5,1373, x_3 = -0,6359$)

As variáveis de estado correspondem à concentração do reagente A , à temperatura do reator e à da camisa. A variável de perturbação utilizada foi a vazão de alimentação do reator.

O grupo de dados utilizado para treinamento da rede (em torno de 800 pontos) foi obtido amostrando o modelo do *CSTR* a cada 0,25 segundos, como recomendado no trabalho de TAHUATA(1992).

As perturbações aplicadas na vazão de alimentação adimensionalizada (q) foram pulsos aleatórios com amplitude máxima de 0,75 e largura de 5 e 15 intervalos de amostragem em torno do valor de estado estacionário $q = 1,0$. As perturbações na vazão de alimentação foram suficientes para provocar transições entre os estados estacionários.

Na Figura 8-1 mostram-se as perturbações aplicadas na vazão de alimentação e utilizadas para treinamento da rede de funções de escala. Nas Figuras 8.2 a 8.4 podem ser vistas as saídas correspondentes às perturbações aplicadas na vazão de alimentação.

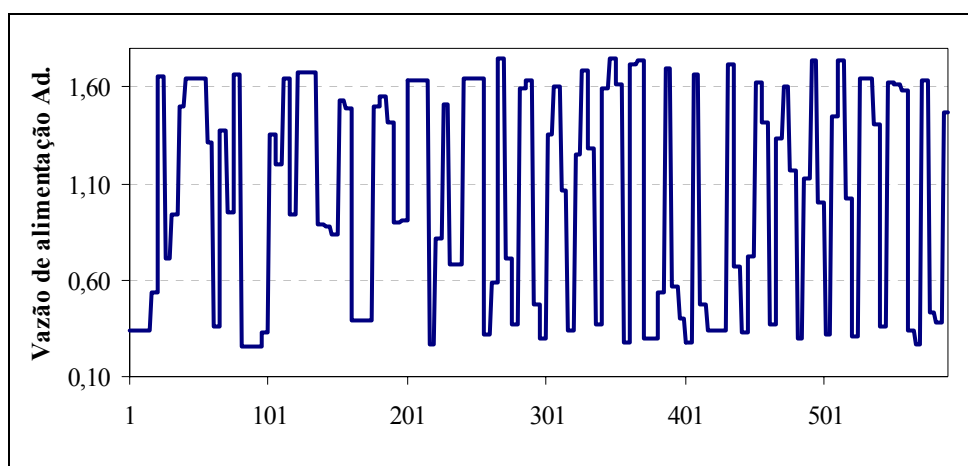


Figura 8-1 - Perturbações aplicadas na vazão de alimentação

O objetivo estabelecido foi prever o valor futuro das variáveis de estado a partir do valor atual e anterior dessas e da variável de perturbação. Assim sendo, a rede foi composta de três saídas correspondentes aos estados. Foi constatado que apenas os valores atuais dos estados e da variável de perturbação eram suficientes para identificar o processo uma vez que o coeficiente de correlação de aproximação dos dados em cada saída foi superior a 99,99%. Dessa forma, não houve necessidade de utilizar mais valores atrasados das saídas e, portanto, a rede de funções de escala foi composta de 4 entradas.

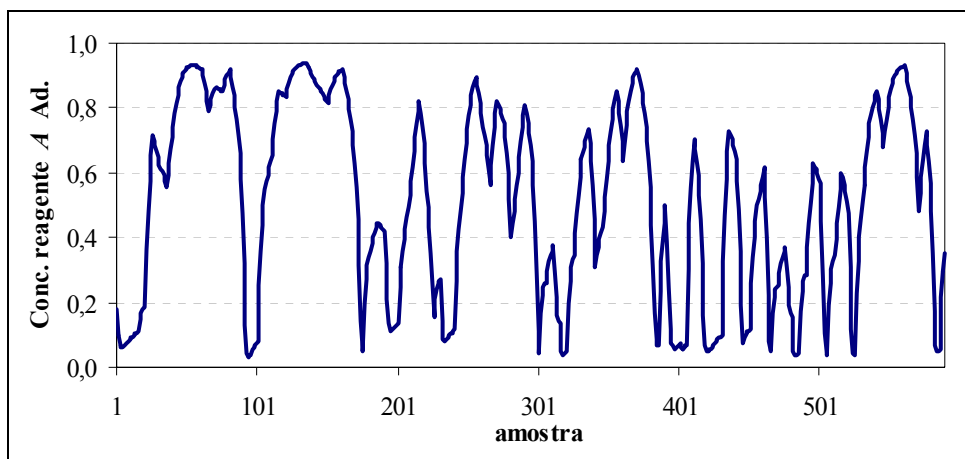


Figura 8-2 - Dados treinamento (concentração de reagente A)

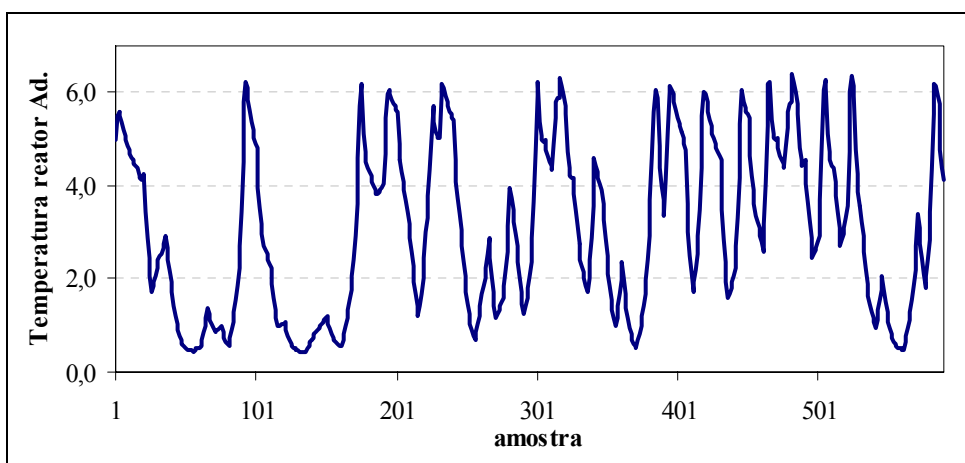


Figura 8-3 - Dados treinamento (temperatura no reator)

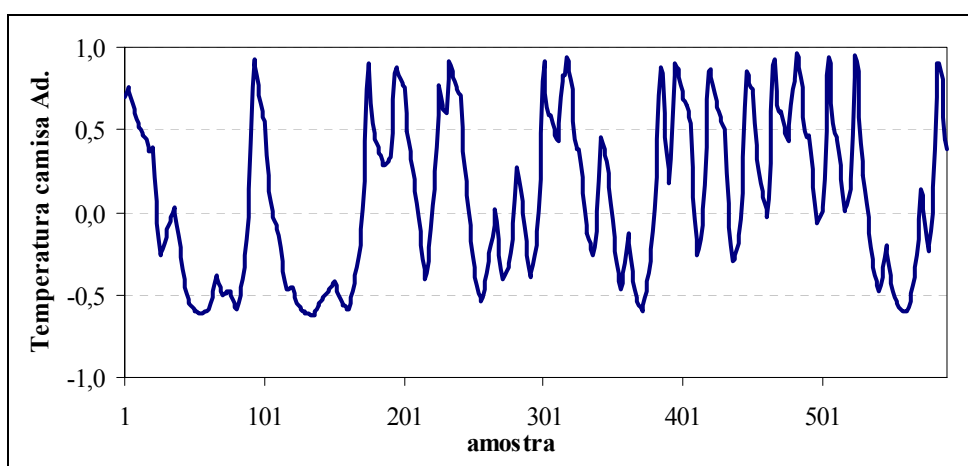


Figura 8-4 - Dados treinamento (temperatura na camisa)

Para avaliar o desempenho do modelo obtido, gerou-se um grupo de dados de teste, sendo que as perturbações aplicadas na vazão de alimentação foram pulsos aleatórios com amplitude

máxima de 0.7 e largura de 10 intervalos de amostragem em torno do valor do estado estacionário $q = 1,0$. Na Figura 8-5 mostram-se as perturbações aplicadas na vazão de alimentação.

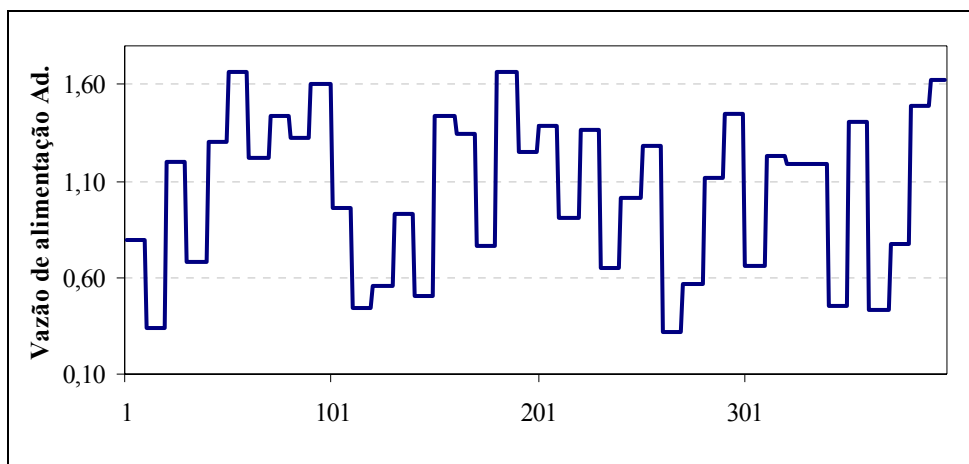


Figura 8-5 - Perturbações aplicadas na vazão de alimentação (teste)

Nas Figuras 8.6 a 8.8 mostra-se uma comparação das saídas previstas pela rede e os dados reais. A rede foi utilizada como um simulador, ou seja, os valores das saídas foram realimentados para cálculo das previsões. O coeficiente de correlação de aproximação para cada saída foi superior a 99,99%.

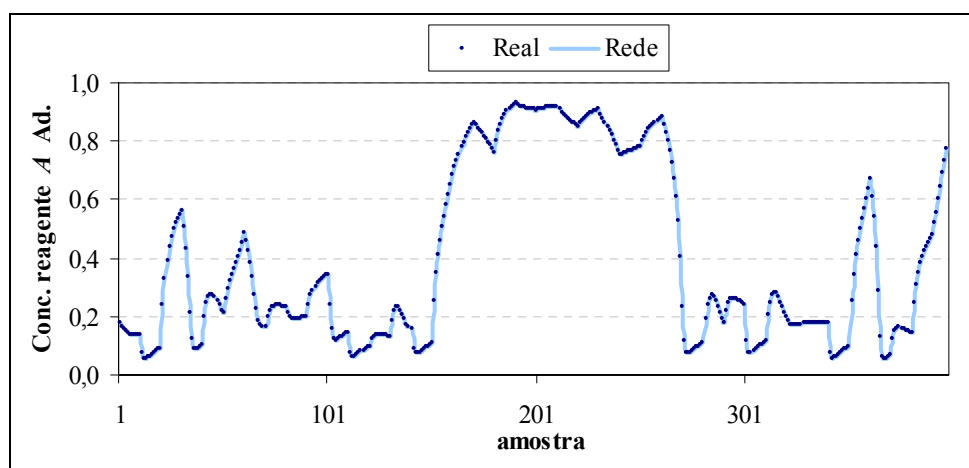


Figura 8-6 - Dados teste x simulação rede (concentração de reagente A)

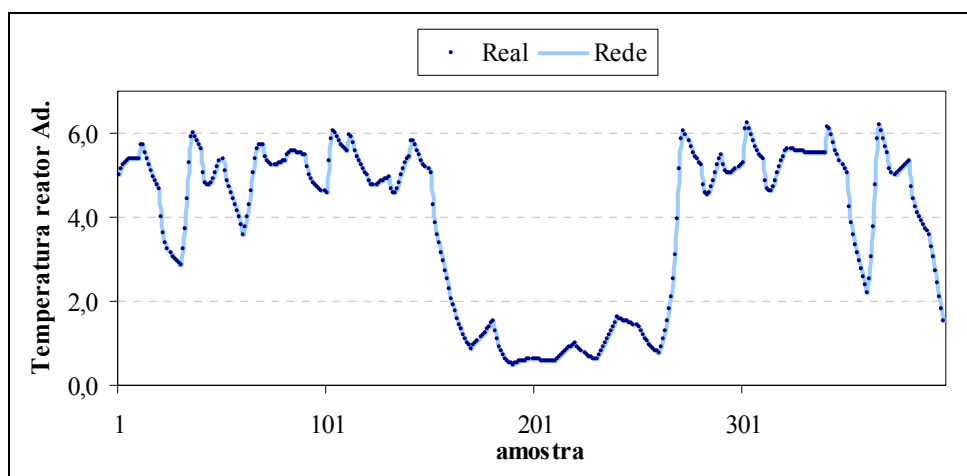


Figura 8-7 - Dados teste x simulação rede (temperatura no reator)

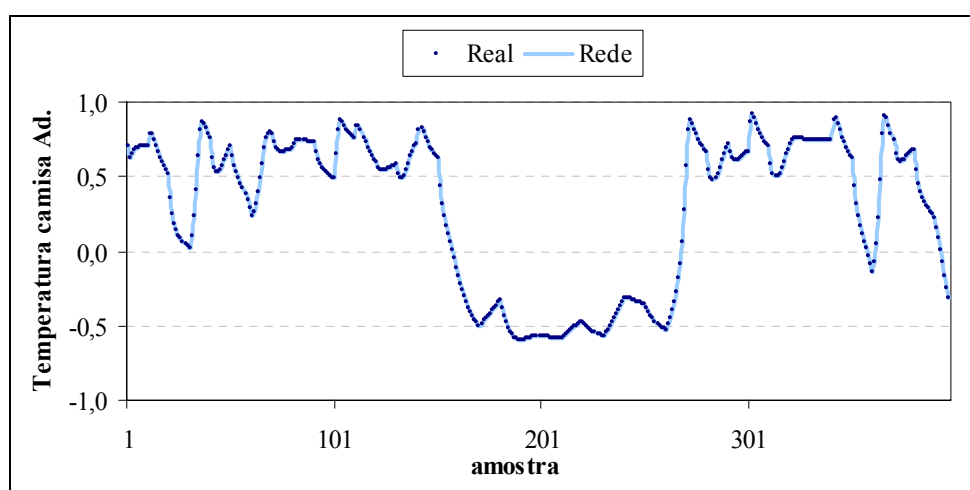


Figura 8-8 - Dados teste x simulação rede (temperatura na camisa)

Pode ser observado nas Figuras 8.6 a 8.8 que a rede de funções de escala conseguiu identificar perfeitamente a multiplicidade de estados estacionários e os comportamentos dinâmicos nas vizinhanças desses. Dessa forma, foi comprovada a capacidade identificação de redes baseadas em *wavelets*, o que valida as alterações propostas.

Pode-se concluir que a rede de funções de escala generalizou muito bem, ou seja, essa foi treinada para prever um passo a frente, no entanto, foi capaz de prever um número arbitrário de passos. O desempenho obtido, em termos de previsão em aberto, foi semelhante ao das redes neurais recorrentes CLAUMANN(1999).

8.2 Reação de Van de Vusse

A síntese do ciclopentenol a partir do ciclopentadieno segue a mesma seqüência de reações como as ocorridas no reator de Van de Vusse que apresenta resposta inversa variável NIEMICE e KRAVARIS(1998). O esquema de reação é o seguinte,



sendo que, neste caso, A é o ciclopentadieno, B é o ciclopentenol, C é o ciclopentandiol e D é o Diciclopentadieno. Considerando o volume de reator constante e que as reações são isotérmicas, o processo pode ser descrito pelas seguintes equações:

$$\begin{aligned} \frac{dC_a}{dt} &= -k_1 \cdot C_a - k_3 \cdot C_a^2 + u \cdot (C_{a0} - C_a) \\ \frac{dC_b}{dt} &= k_1 \cdot C_a - k_2 \cdot C_a - u \cdot C_b \\ y &= C_b \end{aligned} \tag{8.1}$$

onde

C_a e C_b : Concentrações de A e B no reator, respectivamente;

y : Produto de interesse que é a concentração de ciclopentenol (B);

C_{a0} : Concentração do reagente A na corrente de alimentação ;

u : Taxa de diluição;

k_1, k_2, k_3 : Parâmetros cinéticos.

Nas simulações efetuadas, os valores dos parâmetros de reação utilizados foram os descritos em PEREZ *et. al.*(2002) e podem ser vistos na Tabela 8.2.

Tabela 8.2 - Parâmetros do reator Van de Vusse

Parâmetro de reação	Valor
k_1	50 h ⁻¹
k_2	100 h ⁻¹
k_3	50 l/(mol.h)
C_{a0}	10 mol/l

Deve-se notar que embora a variável de interesse seja o ciclopentenol (B), a concentração do reagente A também deve ser conhecida para ser possível simular o comportamento da primeira, conforme a Equação (8.1). Dessa forma, na síntese do ciclopentenol, as variáveis de estado de interesse correspondem às concentrações das substâncias A e B . A variável de perturbação escolhida foi a taxa de diluição da corrente de alimentação.

O grupo de dados utilizado para treinamento da rede (em torno de 1100 pontos) foi obtido amostrando o modelo descrito a cada $1\text{ s} \approx 2,7778 \text{ E-}004 \text{ h}$. As perturbações aplicadas na taxa de diluição (u) foram pulsos aleatórios com valores na faixa de $[20; 190] \text{ h}^{-1}$ e largura entre 10 a 50 intervalos de amostragem. Na Figura 8-9 mostram-se as perturbações aplicadas na taxa de diluição e utilizadas para treinamento da rede de funções de escala. Nas Figuras 8.10 e 8.11 pode ser vista a evolução das concentrações de A e B para as perturbações aplicadas.

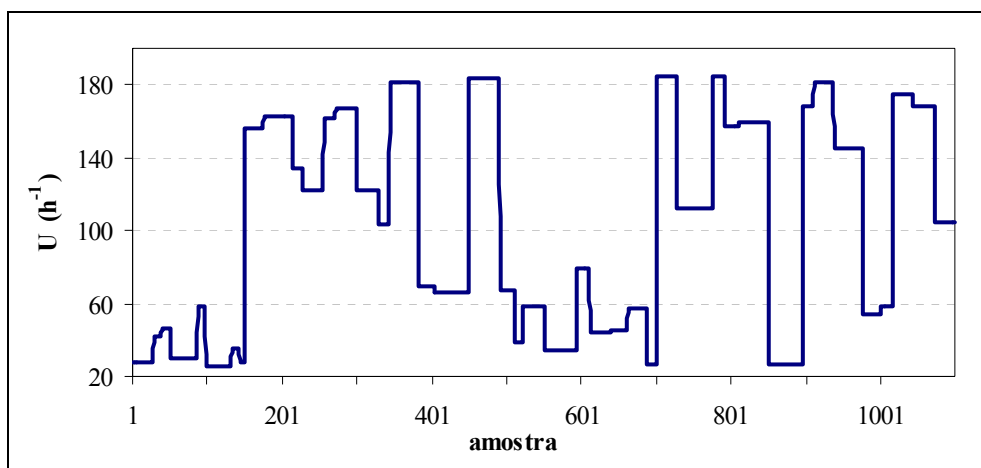


Figura 8-9 - Perturbações aplicadas na taxa de diluição (dados de treinamento)

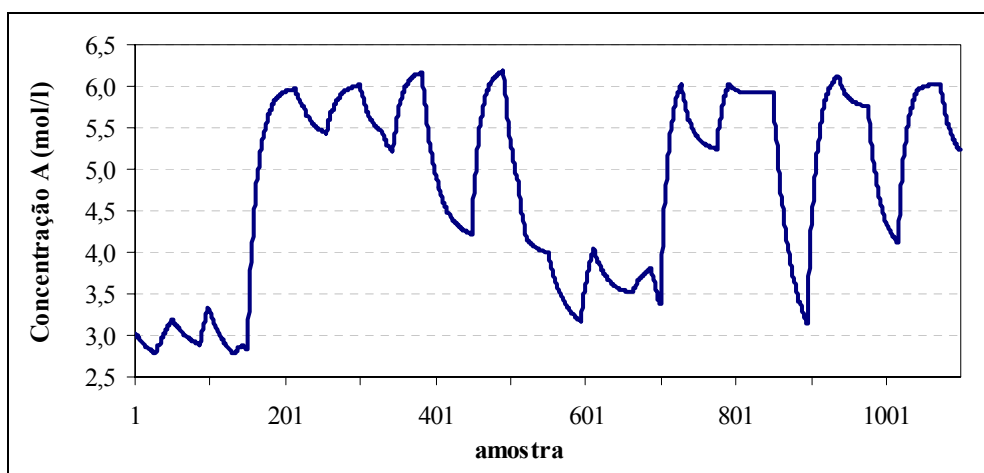


Figura 8-10 - Concentração de A (dados de treinamento)

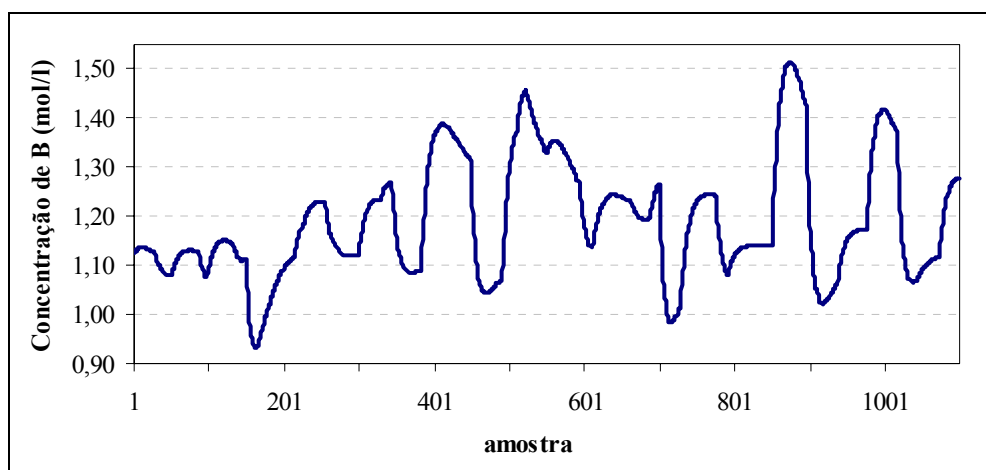


Figura 8-11 - Concentração de B (dados de treinamento)

O objetivo estabelecido foi prever o valor futuro das variáveis de estado a partir do valor atual e anterior dessas e da variável de perturbação. Assim sendo, a rede foi composta de duas saídas correspondentes aos estados de interesse. Em um estudo prévio, constatou-se que apenas os valores atuais dos dois estados e da variável de perturbação foram suficientes para identificar o processo, sendo que o coeficiente de correlação de aproximação dos dados em cada saída foi superior a 99,99%. Dessa forma, a rede de funções de escala foi composta de 3 entradas.

Para avaliar o desempenho em aberto do modelo obtido, gerou-se um grupo dados de teste, sendo que as perturbações aplicadas na taxa de diluição foram pulsos crescentes entre 30 e 180 h^{-1} com largura de 200 intervalos de amostragem. Na Figura 8-12 mostram-se as perturbações aplicadas na taxa de diluição.

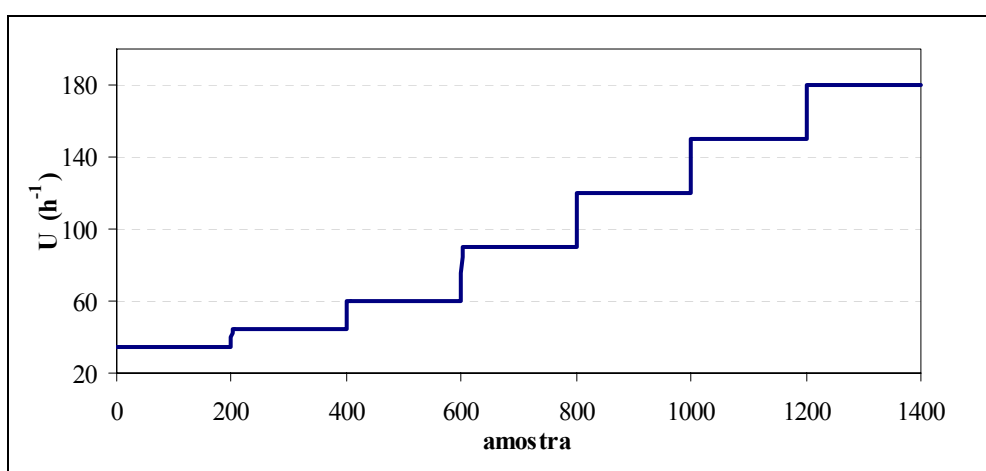


Figura 8-12 - Perturbações aplicadas na taxa de diluição (dados de teste)

Nas Figuras 8.13 e 8.14 pode-se observar uma comparação entre as saídas previstas pela rede e os dados de teste, sendo a rede utilizada como um simulador do processo. O coeficiente de correlação de aproximação do teste para cada saída foi superior a 99,99%.

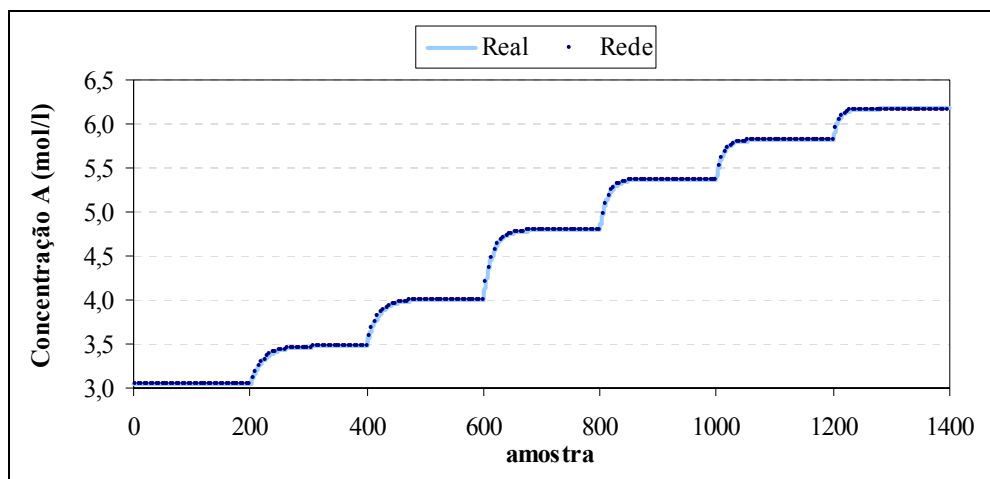


Figura 8-13 – Dados teste \times rede atuando como simulador (concentração de A)

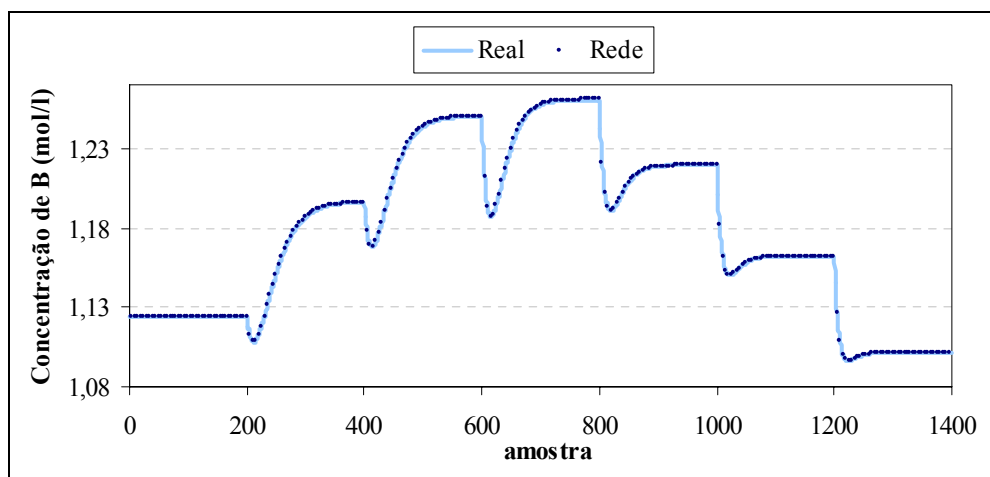


Figura 8-14 - Dados teste \times rede atuando como simulador (concentração de B)

Pode ser observado nas Figuras 8.13 e 8.14 que a rede de funções de escala conseguiu identificar perfeitamente a comportamento não-linear de resposta inversa variável. Como no caso do *CSTR* com múltiplos estados estacionários, pode-se concluir que a rede de funções de escala generalizou muito bem uma vez que foi treinada para prever um passo a frente; no entanto, foi capaz de prever um número arbitrário de passos.

Os *benchmarks* de identificação e controle não-lineares descritos neste capítulo foram identificados utilizando apenas um nível de resolução, o que confirma a validade da proposta de iniciar a multiresolução com o menor número possível de funções de escala.

8.3 Conclusões

O desempenho da rede de funções de escala foi validado utilizando essa como simulador de dois sistemas dinâmicos reconhecidos como *benchmarks* de identificação e controle não-linear: um *CSTR* encamisado que apresenta multiplicidade de estados estacionários e um reator que apresenta resposta inversa variável (reator de Van de Vusse). No caso do *CSTR*, a rede de funções de escala conseguiu identificar perfeitamente a multiplicidade de estados estacionários e os comportamentos dinâmicos nas vizinhanças desses. Para o reator de Van de Vusse, o modelo neural proposto conseguiu identificar com precisão o comportamento não-linear de resposta inversa variável.

A rede de funções de escala generalizou muito bem, pois foi treinada para prever o comportamento dos processos para apenas um passo a frente, no entanto, foi capaz de prever um número arbitrário de passos para um grupo de dados de teste, diferente do utilizado no treinamento.

Pelos bons resultados obtidos na identificação do *CSTR* encamisado e do reator de Van de Vusse, pode-se concluir que estes processos pertencem ao subespaço gerado pelo primeiro nível de resolução da rede de funções de escala proposta.

9 IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS NA PRESENÇA DE RUÍDO UTILIZANDO REDES DE FUNÇÕES DE ESCALA

9.1 Introdução

A identificação de processos que não apresentam ruído foi abordada nos capítulos anteriores. Este capítulo trata da identificação de sistemas onde ocorre ruído.

Segundo NERRAND *et. al.*(1994), dependendo de como o ruído, quando existe, influencia um sistema dinâmico, a estrutura de modelo mais adequada para fazer previsões poderia ser estática ou recorrente. Para o caso sem ruído, qualquer uma das estruturas poderia ser utilizada. No trabalho referido foram descritos dois tipos principais de modelos dependendo do comportamento do ruído. Considerando, por simplificação, o caso de uma entrada e uma saída tem-se:

1) Modelo do erro na saída: Neste caso, é assumido que a saída do processo $y_{p(k)}$ obedece as seguintes equações:

$$\begin{aligned}x_{p(k)} &= \Phi[X_{p(k-1)}, U_{(k-1)}] \\y_{p(k)} &= x_{p(k)} + w_{(k)}\end{aligned}$$

onde

$X_{p(k-1)}$: Estados processo $\{x_{p(k-1)}, x_{p(k-2)}, \dots, x_{p(k-N)}\}$;

$U_{(k-1)}$: Vetor constituído dos atrasos na entrada externa $\{u_{p(k-1)}, u_{p(k-2)}, \dots, u_{p(k-M)}\}$;

$\{w_{(k)}\}$: Seqüência de ruído branco.

Dessa forma, a saída $y_{(k)}$ do modelo utilizado para fazer previsões, ou seja, do preditor é dada por (considerando $w_{(k)} = y_{p(k)} - y_{(k)}$),

$$y_{(k)} = \Phi[y_{(k-1)}, y_{(k-2)}, \dots, y_{(k-N)}, U_{(k-1)}]$$

O preditor associado ao processo que apresenta erro na variável de saída é recorrente e depende de N valores atrasados da variável prevista. Se existir uma rede neural que pode

aproximar o mapeamento Φ , essa rede deve ser recorrente para que o preditor possa ser implementado.

2) Modelo de erro na equação: Neste caso, é assumido que o processo evolui de acordo com as seguintes equações:

$$\begin{aligned}x_{p(k)} &= \Phi[X_{p(k-1)}, U_{(k-1)}] + w_{(k)} \\y_{p(k)} &= x_{p(k)}\end{aligned}$$

A saída $y(k)$ do preditor associado é dada por:

$$y_{(k)} = \Phi[Y_{p(k-1)}, U_{(k-1)}]$$

onde

$Y_{p(k-1)}$: Vetor dos valores atrasados da saída $\{y_{p(k-1)}, y_{p(k-2)}, \dots, y_{p(k-N)}\}$;

Dessa forma, o preditor do processo que apresenta erro na equação deve ser não recorrente, sendo que utiliza a saída medida do processo. Se existe uma rede neural que pode aproximar o mapeamento e implementar o preditor, esta rede deve ser estática uma vez que o preditor não é recorrente.

9.2 Geração de Dados com Ruído

Como é mais fácil treinar redes estáticas do que recorrentes, tem-se motivação para tentar utilizar redes estáticas para processos que apresentam ruído na variável de saída, ou seja, quando uma rede recorrente seria o preditor mais adequado. Também, para muitos casos a influência do ruído pode ser desconhecida e seria desejável que uma rede estática pudesse ser utilizada na identificação.

Para testar a rede estática de funções de escala em problemas com ruído, adicionou-se ruído branco aos grupos de dados utilizados para teste e treinamento dos processos, já descritos, de tanque de pH e do reator biológico. O ruído foi adicionado nas variáveis de saída, o que caracteriza erro na equação de saída. Dessa forma, o objetivo do teste foi verificar

a capacidade de previsão da rede para uma situação desfavorável, em que o preditor adequado seria uma rede recorrente.

Para gerar os grupos de dados com ruído, foram adicionadas perturbações aleatórias, com distribuição uniforme, nas variáveis de saída dos grupos de dados sem ruído já utilizados para identificação do tanque de pH e do reator biológico. A amplitude das perturbações aplicadas correspondeu, em todos os casos, a 5% da faixa dos dados originais.

Deve-se notar que atrasos das variáveis de saída também aparecem como entradas da rede. Dessa forma, as redes de funções de escala foram treinadas com dados de entrada e saída corrompidos por ruído.

A presença de ruído branco nos dados de saída não representa grave problema, uma vez que o algoritmo de mínimos quadrados pode lidar com tal característica. O problema maior ocorre quando o ruído aparece nas variáveis de entrada porque tal comportamento não é manipulado corretamente pelo método de mínimos quadrados. Este fato é agravado pela transformação não-linear dos dados de entrada efetuada pelos neurônios da rede de funções de escala.

9.3 Identificação do Tanque de pH a partir de Dados Corrompidos com Ruído

Para geração de dados com ruído, aplicou-se a metodologia descrita aos dados sem ruído mostrados na Figura 5-2, correspondentes as perturbações aplicadas na vazão de base (Ver Figura 5-1). Os primeiros 800 pontos foram utilizados para treinamento e os 400 restantes para o teste. A variável de perturbação, que corresponde à vazão de ácido, não foi alterada. Nas Figuras 9.1 e 9.2 mostram-se os grupos de dados de treinamento e teste obtidos com ruído a partir da situação sem ruído.

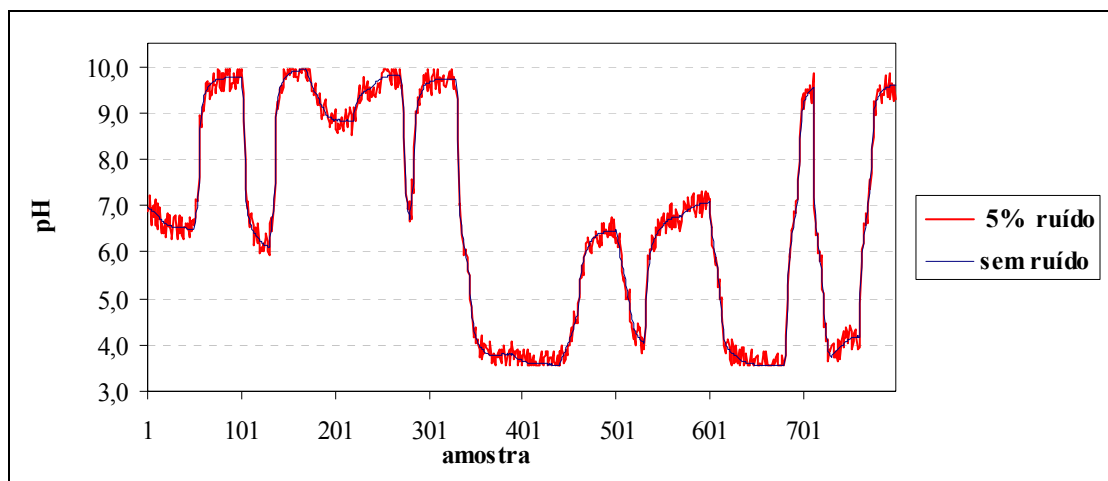


Figura 9-1 - Grupos de dados de treinamento com ruído obtido a partir do caso sem ruído

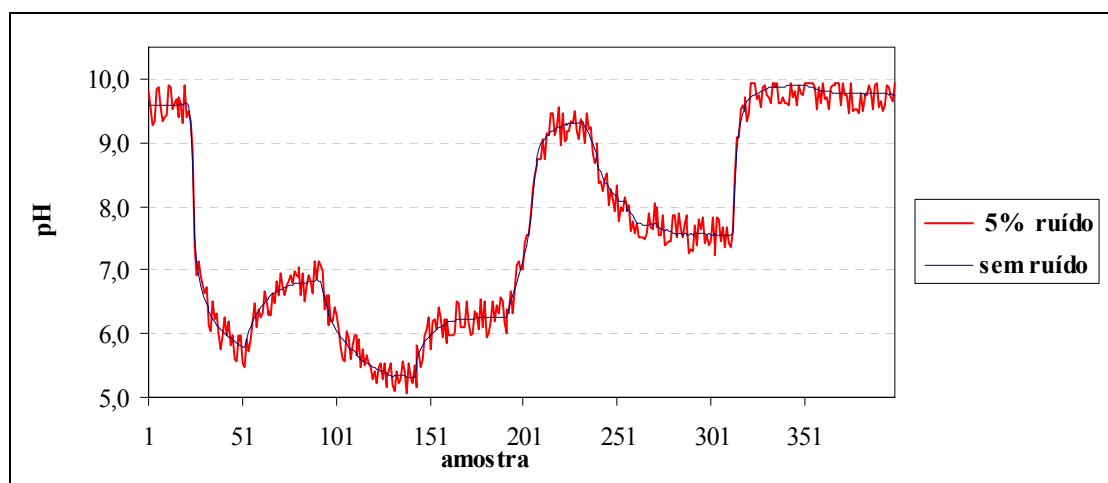


Figura 9-2 - Grupos de dados de teste com ruído obtido a partir do caso sem ruído

Nas Figuras 9.1 e 9.2 pode-se observar a grande quantidade de ruído adicionado aos dados originais. Uma rede de funções de escala, composta de três níveis de resolução, foi treinada com o grupo de dados mostrado na Figura 9-1, para identificar o valor do pH no instante $k+1$ em função do valor do pH e vazão de base no momento k . Utilizando este modelo, uma previsão do comportamento do processo, correspondente ao grupo de dados de teste, foi obtida por simulação. Na Figura 9-3 mostra-se uma comparação entre as previsões do valor de pH utilizando a rede de funções de escala e o grupo de dados de teste.

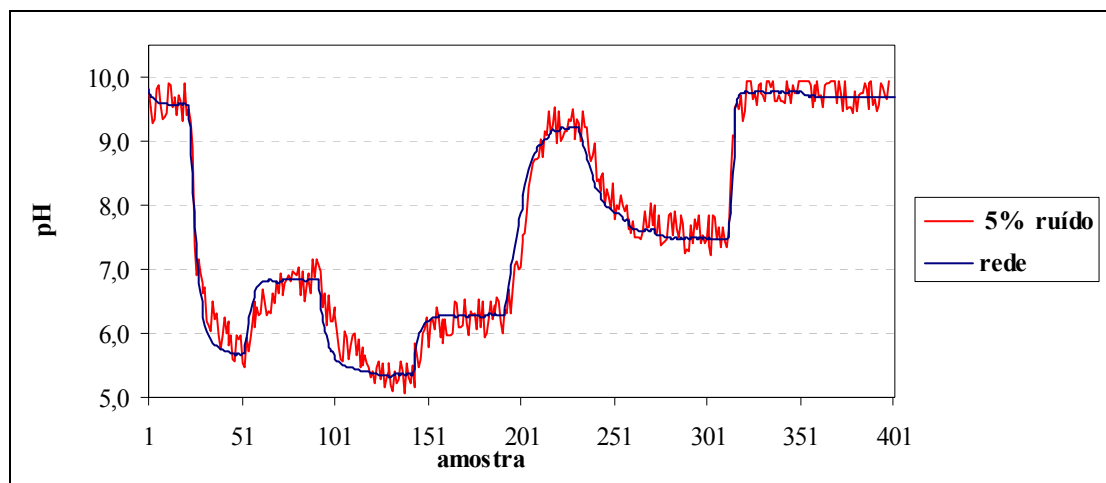


Figura 9-3 - Comparação previsão da rede em aberto x dados do teste

Na Figura 9-3 pode-se observar que a rede de funções de escala identificou bem os ganhos do processo e as transições de estado estacionário. O bom desempenho obtido mostra que esta metodologia de modelagem pode ser aplicada na previsão do comportamento de sistemas com ruído, mesmo que o preditor mais adequado fosse uma rede recorrente.

9.4 Identificação do Reator Biológico a partir de Dados Corrompidos com Ruído

Para geração de dados de treinamento com ruído, aplicou-se a metodologia descrita na seção 9.2 aos dados sem ruído mostrados nas Figuras 7-6 a 7-8 e, no caso do teste, para as Figuras 7-10 a 7-12. A variável de perturbação, que corresponde à vazão de alimentação, não foi alterada. Nas Figuras 9.4 e 9.5 mostram-se os grupos de dados de treinamento e teste obtidos com ruído a partir da situação sem ruído. Na Figura 9-6 mostra-se uma comparação das previsões obtidas com a rede atuando como um simulador e o grupo de dados de teste.

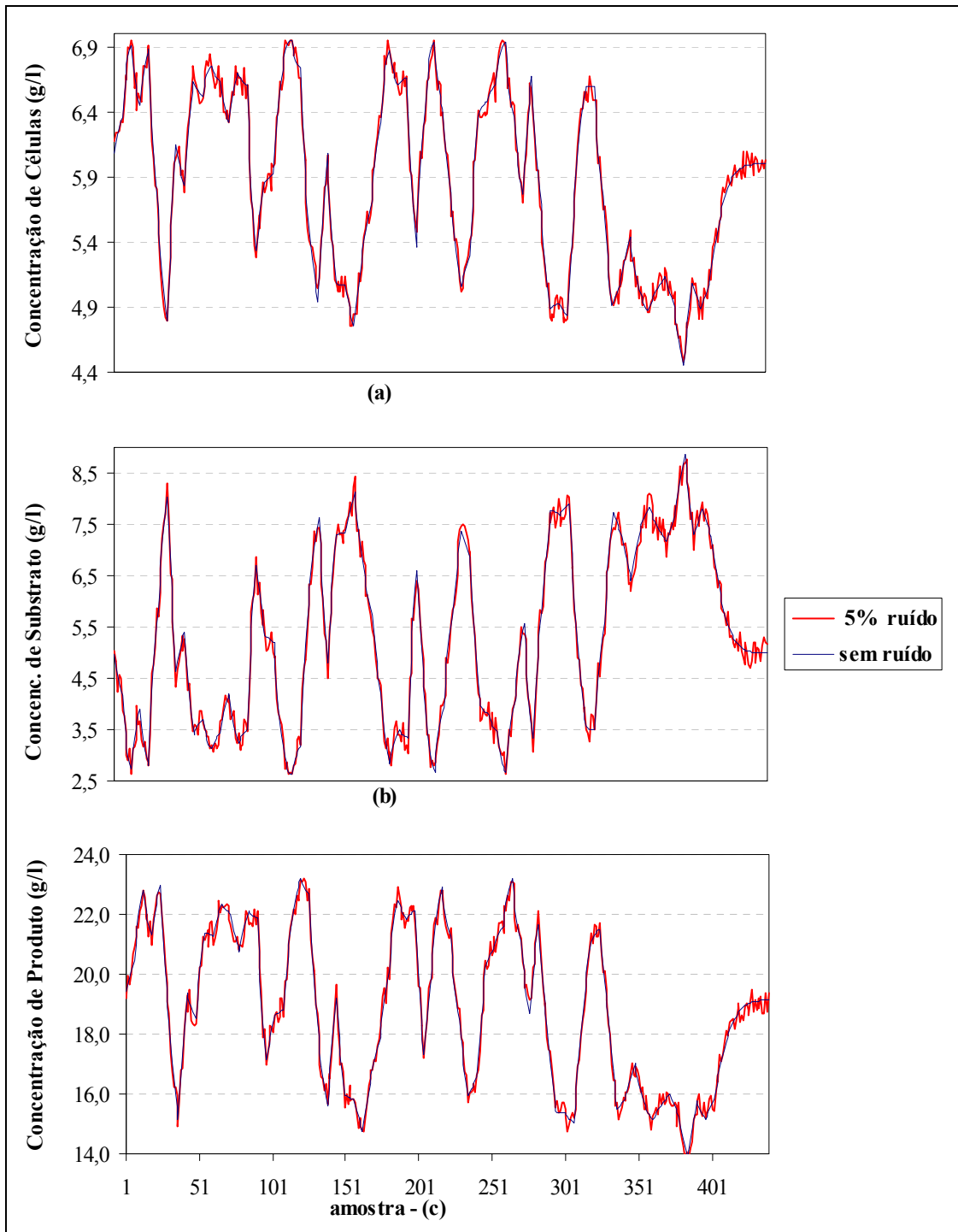


Figura 9-4 - Dados de treinamento com ruído obtidos a partir do caso sem ruído; (a) concentração de células; (b) concentração de substrato; (c) concentração de produto

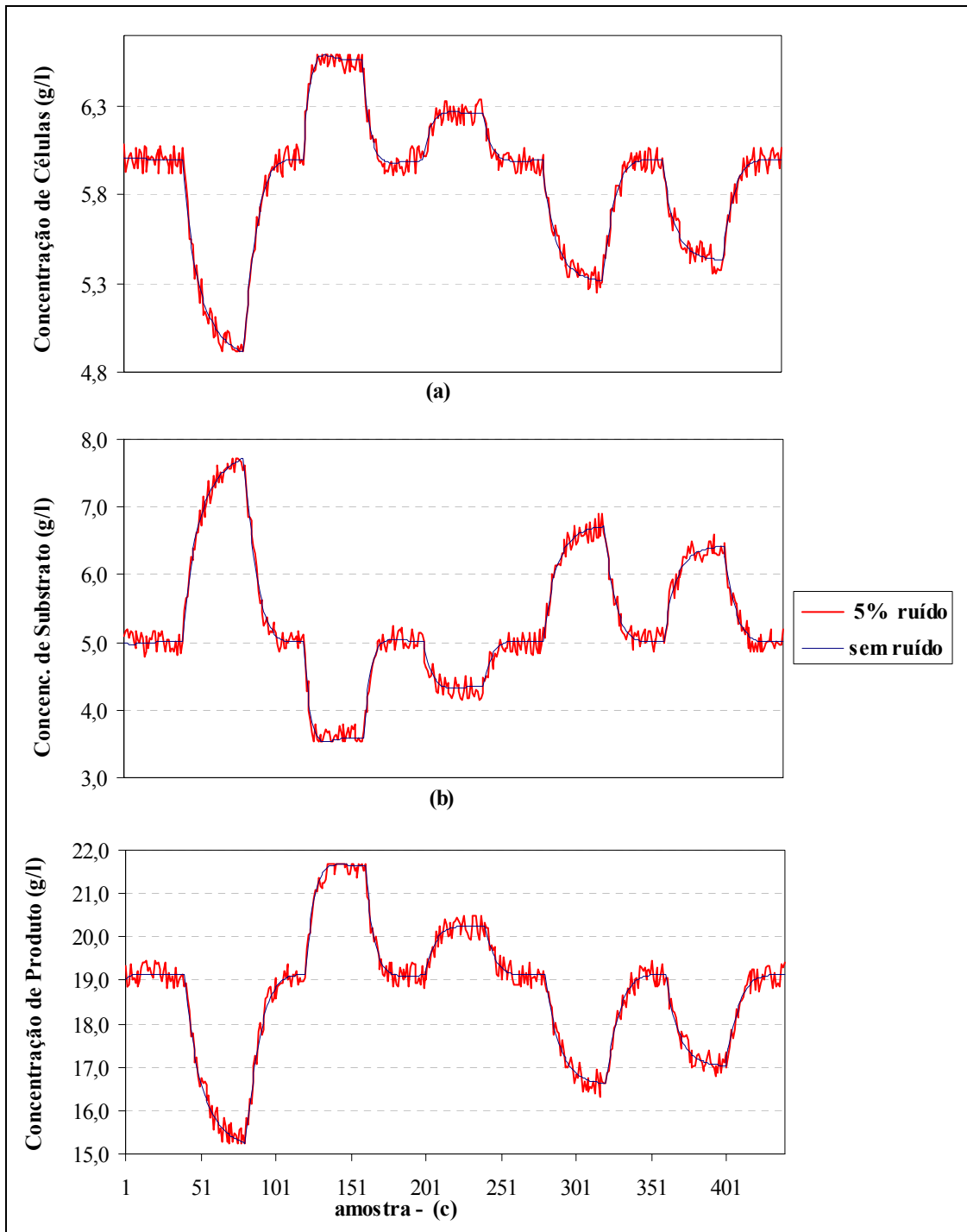


Figura 9-5 - Dados de teste com ruído obtidos a partir do caso sem ruído; (a) concentração de células; (b) concentração de substrato; (c) concentração de produto

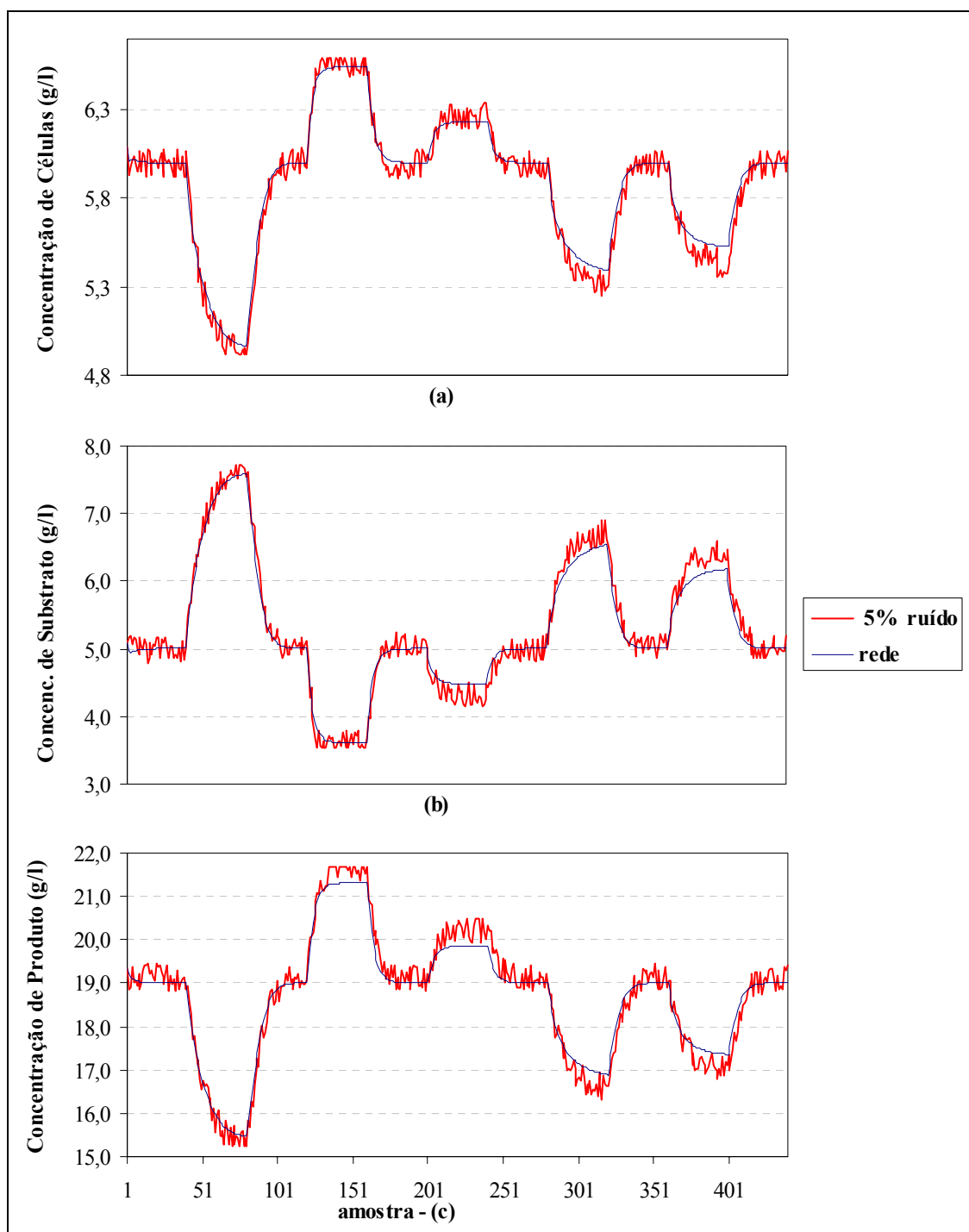


Figura 9-6 - Comparação rede atuando como simulador x dados do teste (a) concentração de células; (b) concentração de substrato; (c) concentração de produto

Nas Figuras 9.4 e 9.5 pode-se observar a grande quantidade de ruído adicionado aos dados originais. Uma rede de funções de escala, composta de 1 nível de resolução, foi treinada com o grupo de dados mostrado na Figura 9-4, para identificar as concentrações de células,

substrato e produto no instante $k+1$, como função dessas e da vazão de alimentação no instante k .

Na Figura 9-6 pode-se observar que a rede de funções de escala identificou bem as transições de estado estacionário e de forma regular os ganhos do processo. O resultado em termos de desempenho de modelagem é, sem dúvida, positivo quando se considera a dificuldade do teste aplicado.

9.5 Conclusões

O desempenho da rede de funções de escala foi validado utilizando essa como simulador de dois sistemas dinâmicos reconhecidos como *benchmarks* de identificação e controle não-linear: um reator biológico que apresenta não-linearidade de ganho e um tanque de pH , sendo que foi adicionado ruído aos dados de treinamento e teste. No caso do tanque de pH , a rede de funções de escala conseguiu identificar bem os ganhos do processo e as transições de estado estacionário. Para o reator biológico, o modelo neural proposto identificou bem os instantes em que ocorreram transições de estado estacionário e de forma regular os ganhos do processo.

A rede de funções de escala generalizou bem para os sistema dinâmicos utilizados, considerando a dificuldade do teste aplicado:

- 1) Foi adicionado ruído de grande amplitude às variáveis de entrada e saída dos processos;
- 2) A rede foi treinada para prever o comportamento do processo para apenas um passo a frente, no entanto, foi utilizada para prever um número arbitrário de passos em um grupo de dados de teste, diferente do utilizado no treinamento.
- 3) O preditor adequado para os processos aos quais foi inserido ruído seria uma rede recorrente, sendo que a rede de funções de escala proposta neste trabalho e utilizada foi do tipo estática.

10 CONTROLE PREDITIVO DO NÍVEL DE UM TANQUE COM ALTERAÇÃO DE DIÂMETRO

10.1 Descrição do Tanque de Nível

Como uma aplicação para a rede de funções de escala, foi efetuado experimentalmente o controle do nível de um tanque que apresenta alteração brusca de diâmetro. O tanque de nível utilizado neste trabalho faz parte de um conjunto de sistemas do Laboratório de Controle de Processos (LCP/ENQ/UFSC).

O tanque de nível, construído em PVC, é dividido em três seções: a inferior que é formada por um tubo com 0,82 m de altura e 0,15 m de diâmetro; a intermediária, localizada entre 0,82 e 0,94m que consiste de duas reduções em seqüência (0,15m \Rightarrow 0,10m) e (0,10m \Rightarrow 0,05m); a superior que consiste de um tubo com 0,05 m de diâmetro começando em 0,94 m e se estendendo até 1,65m. O processo também utiliza uma bomba centrífuga de 0,25 cv de potência, uma válvula de controle e válvulas manuais. O detalhamento do sistema é mostrado nas Figuras 10.1 e 10.2.

O sinal enviado pelo sistema de controle, responsável pela abertura da válvula de controle, pode variar entre 1 e 5 Volts. Os limites 1 a 5 Volts correspondem aos estados de completamente fechado e aberto da válvula, respectivamente.

O tanque de nível apresenta a não-linearidade de mudança brusca de diâmetro que varia em um fator de três, quando se considera a seção inferior e superior desse. O sistema também apresenta não-linearidade devido à válvula de controle, que é do tipo igual percentagem. Outra dificuldade do processo é que, durante o controle, o tempo de amostragem deve ser pequeno para evitar o vazamento de água pela parte superior. Esse tempo é adequado para o controle do nível da parte superior, no entanto, impõe restrições na identificação da seção inferior. As modificações no nível no caso desta última, entre instantes de amostragem, são relativamente pequenas e contaminadas por ruído, ou seja, a relação sinal/ruído é pequena.

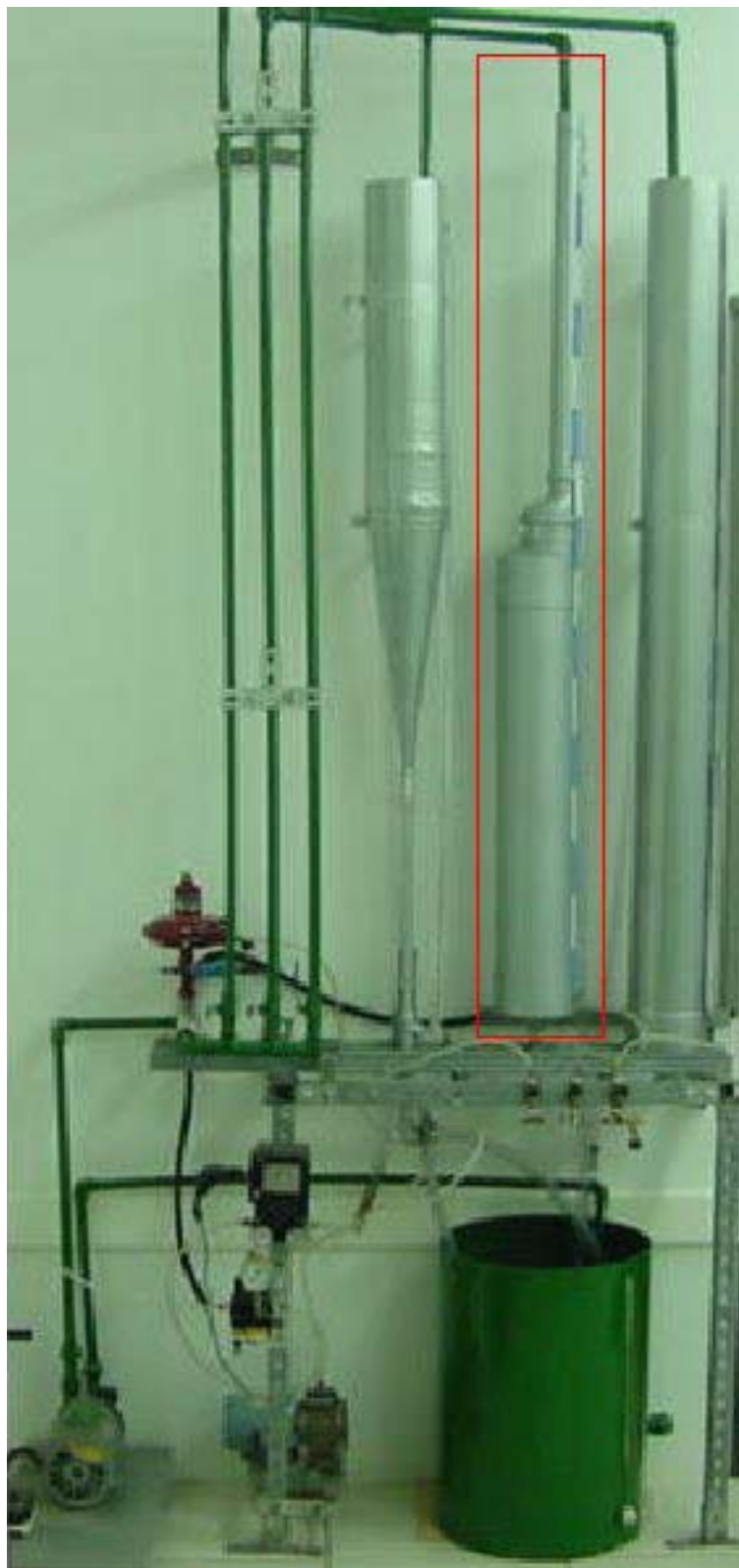


Figura 10-1 - Tanques de nível LCP; em detalhe o tanque com alteração brusca de diâmetro no qual foram realizados os experimentos de controle

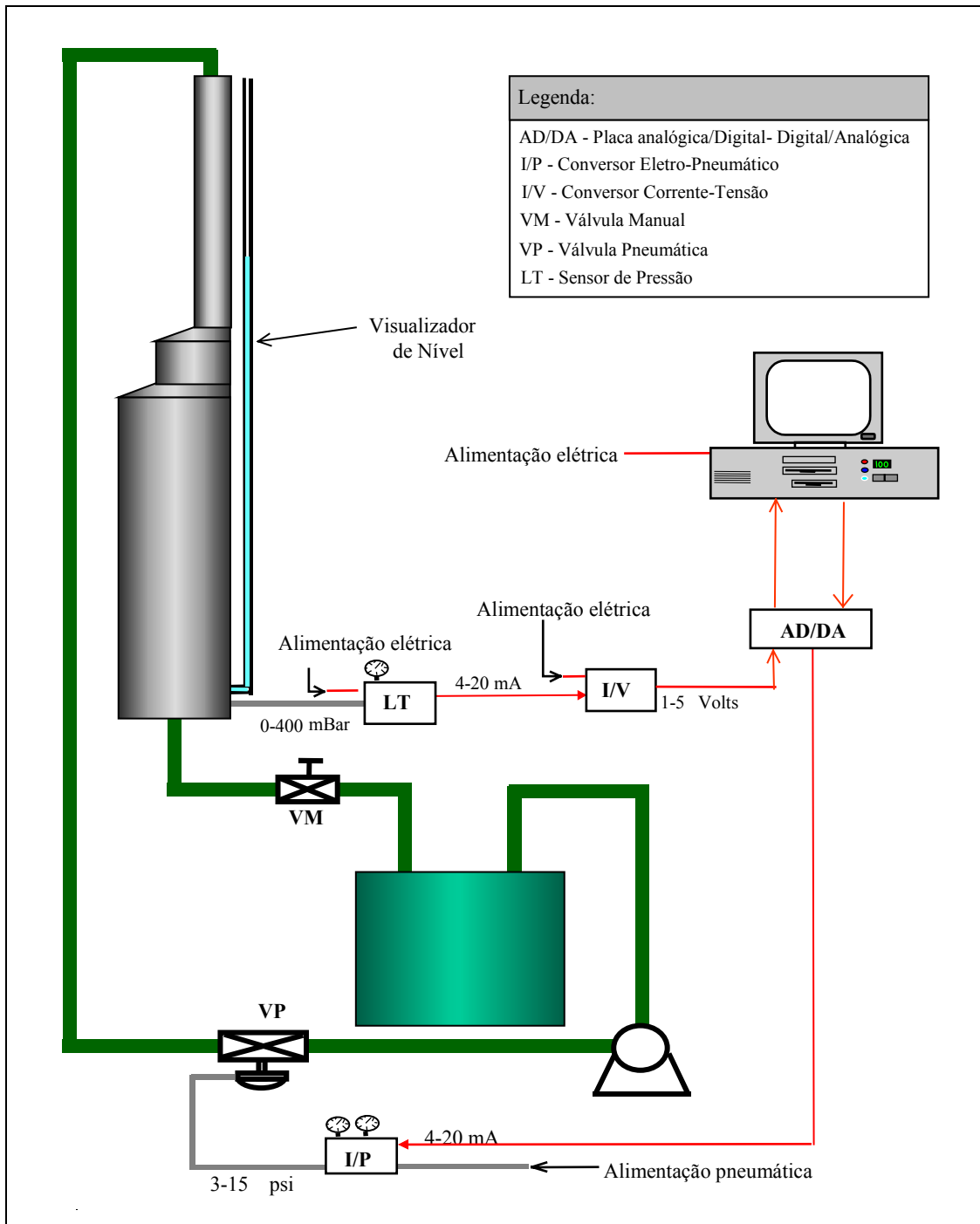


Figura 10-2 - Diagrama do sistema de controle do tanque de nível

10.2 Treinamento da Rede de Funções de Escala

Uma série temporal para identificação do processo foi obtida efetuando perturbações entre 1 e 5 Volts no sinal enviado pelo sistema de controle. Na Figura 10-4 mostra-se a evolução do nível do tanque para as perturbações aplicadas (Ver Figura 10-3).

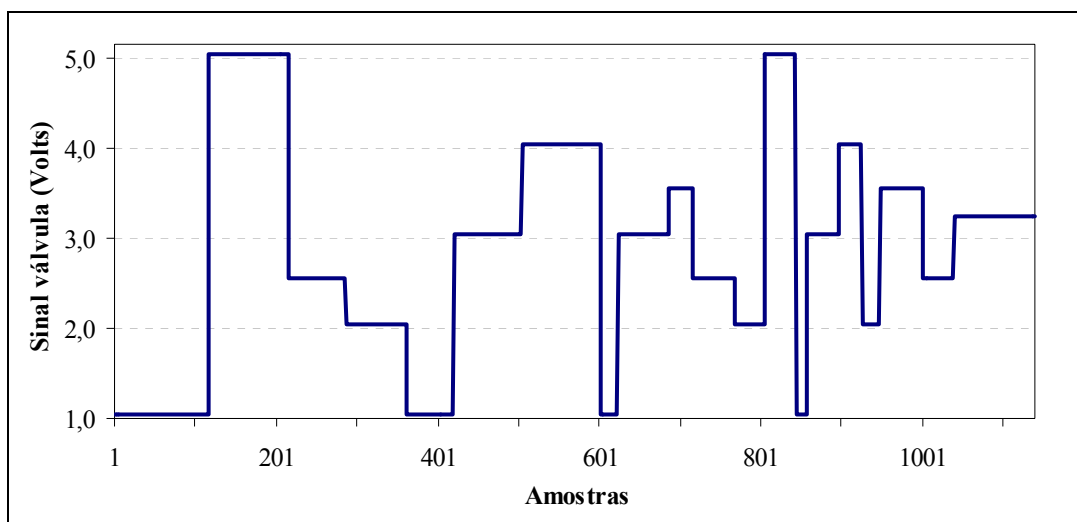


Figura 10-3 - Perturbações aplicadas no sinal enviado pelo sistema de controle (treinamento)

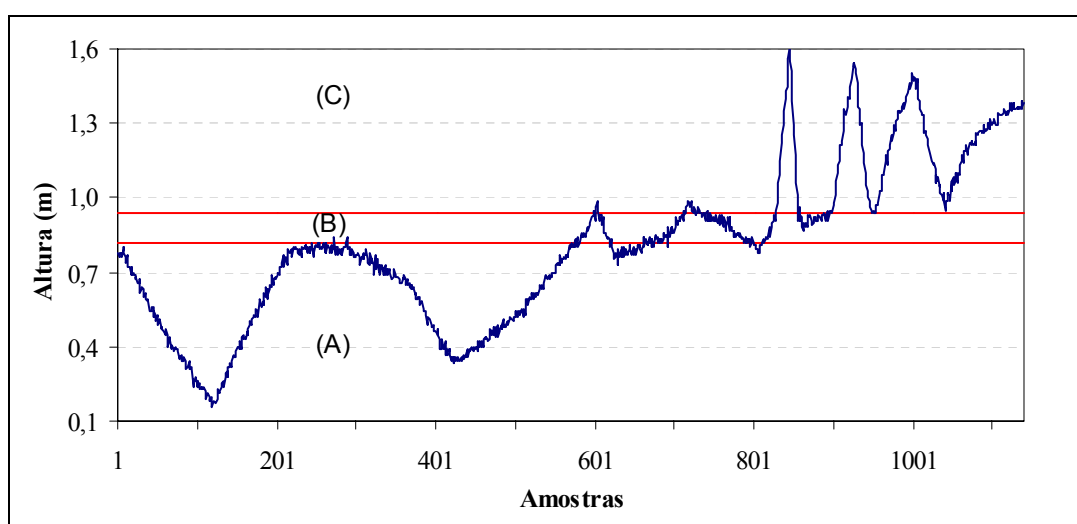


Figura 10-4 - Dados de altura (treinamento); (A) seção inferior; (B) seção intermediária; (C) seção superior tanque

A partir das informações coletadas do nível para as perturbações efetuadas, gerou-se o grupo de dados para treinamento da rede de funções de escala. O tempo de amostragem utilizado foi de 1 s. A partir de uma análise prévia, determinou-se que o processo poderia ser identificado com um modelo da forma $y_{(k+1)} = f[y_{(k)}, u_{(k)}]$, ou seja, o valor do nível no instante $(k+1)$ depende de maneira não-linear $f(\cdot)$ do nível e ação de controle no instante (k) . Dessa forma, o grupo de dados para treinamento da rede consistiu de uma saída $y_{(k+1)}$ e duas entradas $y_{(k)}$ e $u_{(k)}$. Também foi determinado que apenas 1 nível de resolução era suficiente para identificar o processo. A utilização de mais variáveis de entrada para a rede (atrasos no nível e

na variável de perturbação) e mais níveis de resolução não diminuíram significativamente o erro quadrático de aproximação.

A rede de funções de escala foi treinada *off-line* utilizando a regularização pela norma dos pesos descrita no capítulo 6. Na Figura 10-5 mostra-se uma comparação entre as previsões da rede e os dados de treinamento. O coeficiente de correlação da aproximação no treinamento foi de $R^2 = 99,4901$.

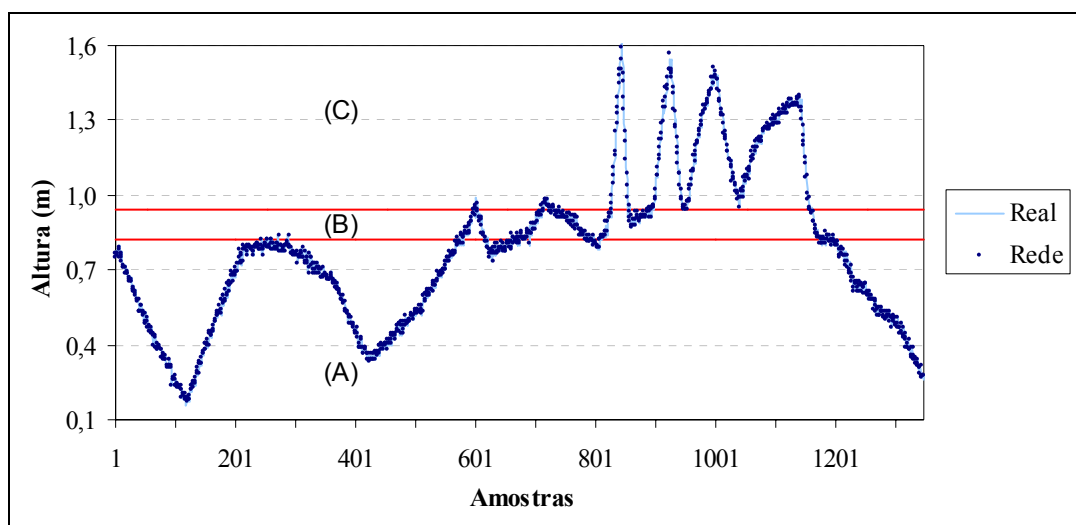


Figura 10-5 - Comparação dados reais x previsão rede (treinamento) (A) seção inferior; (B) seção intermediária; (C) seção superior tanque

10.3 Validação da Rede de Funções de Escala

O modelo, correspondente à rede de funções de escala, foi validado através de dois métodos: pela comparação da curva de simulação da rede para um grupo de dados de teste e por análise de resíduos.

10.3.1 Simulação

Foi gerado um grupo de dados para testes efetuando perturbações de 1 a 5 Volts no sinal enviado pelo sistema de controle (Ver Figura 10-6). Para estas perturbações, mostra-se, na Figura 10-7, a evolução do nível do processo e a curva de simulação da rede treinada.

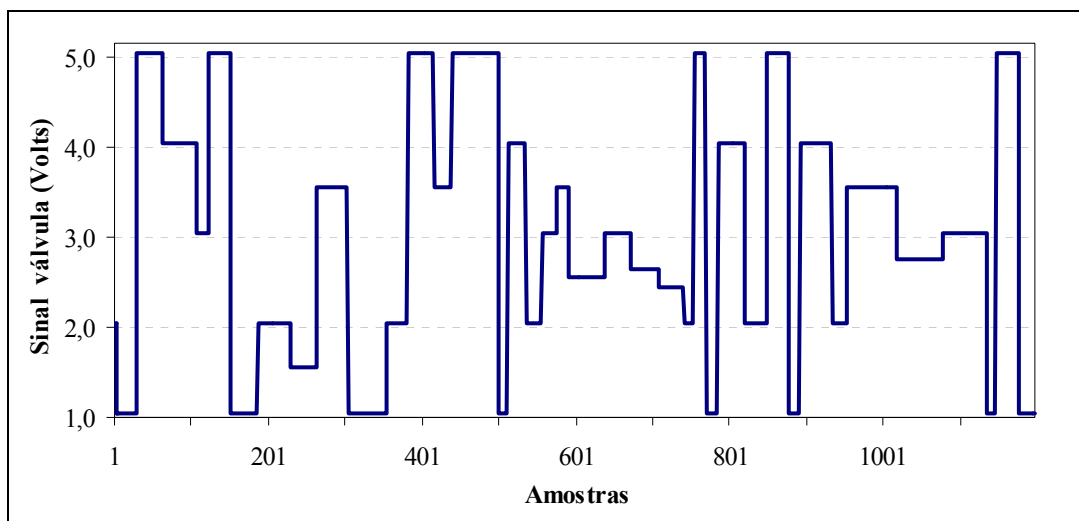


Figura 10-6 - Perturbações aplicadas no sinal enviado pelo sistema de controle (teste)

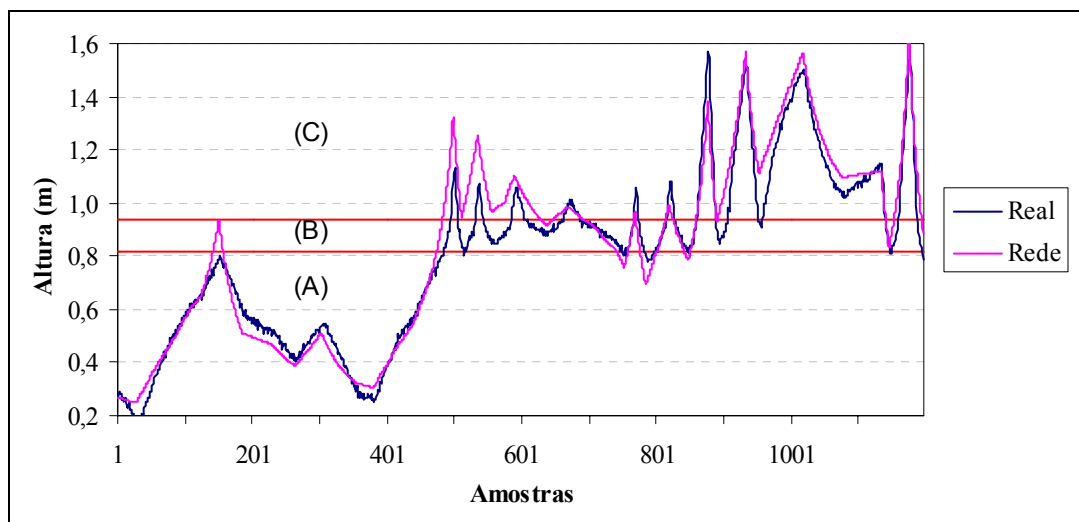


Figura 10-7 - Comparação dos dados teste x curva de simulação gerada pela rede
(A) seção inferior; (B) seção intermediária; (C) seção superior tanque

Pode-se observar na Figura 10-7 que a rede conseguiu identificar de forma razoável o comportamento do processo nas regiões inferior, intermediária e superior.

10.3.2 Análise de Resíduos de Treinamento

Os testes consistindo da função de auto-correlação dos resíduos e a correlação cruzada entre os resíduos e as entradas, foram originalmente desenvolvidos para modelos lineares. Para detectar tanto a presença de termos lineares quanto não-lineares, testes envolvendo a

utilização de correlações de alta ordem são necessários BILLINGS e VOON(1986). Os testes correspondentes à análise de resíduos, considerando que se disponha de um grupo de dados com um número arbitrário de pontos, consistem de:

$$\begin{aligned}
 \Phi(u^2)\varepsilon(\tau) &= 0 \quad \forall \tau \\
 \Phi(u^2)\varepsilon^2(\tau) &= 0 \quad \forall \tau \\
 \Phi\varepsilon\varepsilon u(\tau) &= 0 \quad \forall \tau \\
 \Phi u\varepsilon(\tau) &= 0 \quad \forall \tau \\
 \Phi\varepsilon\varepsilon(\tau) &= \delta(\tau) \quad \forall \tau
 \end{aligned} \tag{10.1}$$

onde

u : variável de perturbação;

ε : erro de previsão;

τ : deslocamento no tempo;

e considerando x e y variáveis da Equação (10.1) tem-se:

$$\Phi_{x,y}(\tau) = \frac{\sum_{k=\tau}^N (x_{(k)} - \bar{x}) \cdot (y_{(k)} - \bar{y})}{\sqrt{\sum_{k=1}^N (x_{(k)} - \bar{x})^2} \cdot \sqrt{\sum_{k=1}^N (y_{(k)} - \bar{y})^2}}$$

e

$$\Phi_{\varepsilon\varepsilon u}(\tau) = \frac{\sqrt{N} \cdot \sum_{k=\tau}^N \varepsilon_{(k)} \cdot \varepsilon_{(k-1)} \cdot (u_{(k-\tau)} - \bar{u})}{\sum_{k=1}^N \varepsilon^2_{(k)} \cdot \sqrt{\sum_{k=1}^N (u_{(k)} - \bar{u})^2}}$$

Para um grupo de dados com número finito de pontos, BILLINGS E VOON(1986) sugeriram que, com 95% de nível de confiança, se o valor de cada teste estatístico descrito for inferior a $\frac{1,96}{\sqrt{N}}$ (N : número de pontos do grupo de dados de treinamento) então não haverá estruturas lineares ou não-lineares não modeladas nos resíduos. Os resultados dos testes de correlação, correspondentes ao grupo de dados de treinamento, podem ser vistos na Figura 10-8.

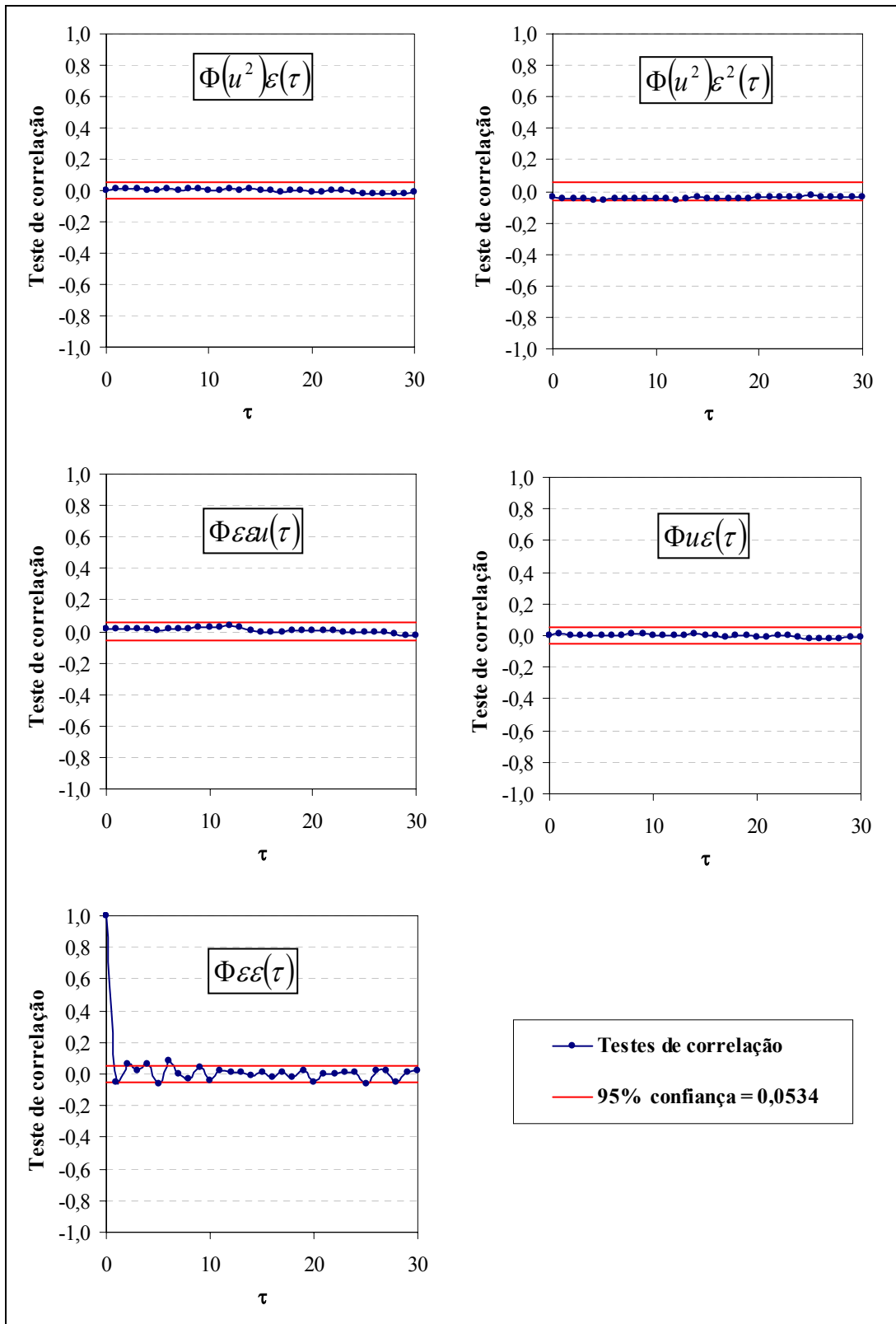


Figura 10-8 - Testes de correlação baseados em resíduos (95% confiança)

Na Figura 10-8 pode-se observar que a rede de funções de escala satisfaz os testes, para um nível de confiança de 95%. O valor limite $\frac{1,96}{\sqrt{N}}$ calculado para o grupo de dados de treinamento, correspondente a um nível de 95% de confiança, foi igual a 0,0534.

10.4 Princípios Básicos de Controle Preditivo

Nesta seção destacam-se apenas as idéias básicas de controle preditivo; uma descrição mais detalhada pode ser encontrada em GARCIA *et. al.*(1989) e CAMACHO e BORDONS(1994).

As idéias comuns aos diferentes esquemas de controle preditivo são:

- As saídas futuras de um processo entre os horizontes N_1 e N_2 , chamados de *horizontes mínimo e máximo de predição*, são previstas a cada instante de tempo k utilizando o modelo do processo. Estas saídas previstas $y_{(k+i|k)}$ para $i = N_1 \dots N_2$ dependem dos valores conhecidos no instante k (entradas e saídas passadas) e dos futuros sinais de controle $U_{j,(k+i)}$, com $j = 1 \dots N$ (número de variáveis manipuladas) e $i = 0 \dots N_u$ (*horizonte de controle*), que devem ser determinados;
- O conjunto de futuras ações de controle é calculado otimizando algum critério para manter o processo o mais próximo possível de uma trajetória de referência $y_{ref(k+i)}$ ($i = N_1 \dots N_2$), que pode ser o próprio *setpoint* ou, em geral, uma trajetória suave de mudança de *setpoint*;
- Normalmente aplica-se ao processo apenas o primeiro sinal de controle da seqüência calculada a cada intervalo de tempo. Esta estratégia é adotada pois a qualidade da previsão do modelo deteriora a medida que aumenta o *horizonte de predição*.

O critério de otimização mais empregado é o erro quadrático entre a saída prevista e a estipulada pela trajetória de referência. Em muitos casos, pode-se também incluir o esforço de controle na função objetivo CAMACHO e BORDONS(1994). O critério de otimização é mostrado na Equação (10.2).

$$J(N_1, N_2, N_u) = \sum_{i=N_1}^{N_2} \delta_{(i)} [y_{(k+i|k)} - y_{ref(k+i)}]^2 + \sum_{j=1}^N \sum_{i=1}^{N_u} \lambda_{(j,i)} [\Delta U_{j,(k+i-1)}]^2 \quad (10.2)$$

Com uma trajetória de referência definida pela Equação (10.3)

$$\begin{aligned} y_{ref(k)} &= y_{(k)} \\ y_{ref(k+i)} &= \alpha y_{ref(k+i-1)} + (1-\alpha)y_{sp} \quad (i=1\dots N_2) \end{aligned} \quad (10.3)$$

onde,

$\lambda_{(j,i)}$ - Fatores de penalização para $j=1..N$ e $i=1..N_u$;

α - Fator de ajuste da trajetória de referência;

y - Variável controlada;

y_{sp} - *Setpoint*, ponto ao qual o sistema deve ser conduzido.

O objetivo ao minimizar a Equação (10.2) é fazer que a saída futura $y_{(k+i|k)}$ siga a referência $y_{ref(k+i)}$ e, ao mesmo tempo, o esforço de controle $\Delta U_{j,(k+i-1)}$ seja minimizado. Esta equação possui alguns parâmetros ajustáveis (N_1 , N_2 , N_u , $\delta_{(i)}$, $\lambda_{(j,i)}$) que podem ser modificados para obter o comportamento desejado do sistema controlado. O significado de N_1 e N_2 é bastante intuitivo. Eles indicam o intervalo de tempo que é desejável que a saída siga a referência. Se N_1 é definido com um valor alto significa que os erros nos primeiros instantes não são importantes. Os coeficientes $\delta_{(i)}$ e $\lambda_{(j,i)}$ são seqüências de parâmetros que ponderam o comportamento futuro. No projeto de controladores preditivos é, em geral, utilizada uma trajetória de referência $y_{ref(k+i)}$ que não necessariamente coincide com o *setpoint*. Geralmente utiliza-se uma trajetória suave gerada a partir do valor atual da saída e do *setpoint* (Equação (10.3)).

10.5 Implementação Computacional do Controlador

A implementação computacional de uma estratégia de controle depende de duas ferramentas, *hardware* e *software* especiais dedicados. O *hardware* consistiu de um conjunto de duas placas, uma de terminais e uma de aquisição de dados (AD/DA). A placa AD/DA utilizada foi da marca DataTranslation possuindo uma resolução de 12 *bits*, 24 canais de

entrada, 16 digitais e 8 *single ended* e 2 canais de saída analógicos, com faixa de entrada e saída de 1-5 volts.

Para efetuar o controle de nível do tanque foi utilizado um *software* para ambiente *Windows*®, desenvolvido em Delphi 6,0®. Basicamente, o *software* de controle opera pela ação de um temporizador. Assim, inicialmente, faz-se a configuração dos dados de *hardware*, dos parâmetros do controlador e da curva de calibração. A curva de calibração é obtida em malha aberta e representa a relação linear entre as medidas realizadas em volts e as medidas reais do sistema. No caso do tanque de nível, isto é representado em um gráfico do tipo Volt x Metro e os valores dos coeficientes da equação da reta obtidos são inseridos no *software*.

Para a implementação das estratégias de controle propostas utilizou-se um microcomputador tipo *Celeron* 1 GHz, com 128 MBytes de memória RAM e 20 GBytes de disco rígido.

As especificações de todos os equipamentos utilizados encontram-se no apêndice 1 (seção 13.1) e a metodologia para curva de calibração do sensor de pressão é descrita no apêndice 2 (seção 13.2).

10.6 Projeto do Controlador

Partindo de um sistema que pode ser modelado pela equação $y_{(k+1)} = F(y_{(k)}, u_{(k)})$, treina-se uma rede neural para aproximar o mapeamento F , ou seja: $F(y_{(k)}, u_{(k)}) \approx H(y_{(k)}, u_{(k)})$, sendo que, $H(y_{(k)}, u_{(k)})$, é a saída prevista pela rede neural. A equação de previsão torna-se:

$$y_{(k+1|k)} = H(y_{(k)}, u_{(k)}) \quad (10.4)$$

Em um grupo de dados obtido a partir de perturbações aplicadas ao processo pode não haver informação suficiente para uma identificação adequada, sendo que uma rede neural treinada com tal grupo de dados poderá apresentar erros de previsão em determinadas regiões do processo. Dessa forma, a Equação (10.4) deveria ser modificada para levar em conta os erros de previsão, ou seja:

$$y_{(k+1)} = y_{(k+1|k)} + E_{(k+1)} = H(y_{(k)}, u_{(k)}) + E_{(k+1)} \quad (10.5)$$

onde $E_{(k+1)}$ é o erro de previsão no instante $k+1$, que é a diferença entre o valor real do processo no instante $k+1$ e a saída estimada pela rede.

Este termo é particularmente importante na correção de erros de previsão de estado estacionário e quando ignorado pode causar problemas de *offset*. O erro de previsão em $k+1$ não é conhecido no instante k , sendo que normalmente considera-se esse igual ao cometido no instante k , GARCIA *et. al.*(1989), SEBORG *et. al.*(1989).

No controle de nível implementado não se realimentou o erro de previsão, ou seja, foi efetuado controle em “malha aberta” do processo. Dessa forma, se o modelo apresentar erros de identificação de estado estacionário, a estrutura de controle utilizada não conseguirá eliminar o problema de *offset*.

Deve-se notar que a ênfase dessa aplicação está no teste da rede como modelo de previsão e não no projeto do controlador. Este último, por exemplo, não poderá rejeitar perturbações tipo carga, mesmo que se disponha de um modelo que represente perfeitamente o processo.

10.7 Resultados

O desempenho do controlador neural foi testado para perturbações tipo *setpoint*, sendo que seu desempenho foi comparado ao de um controlador **PI**, com relação à variável controlada e à manipulada.

Os parâmetros do **PI** foram inicialmente determinados pela técnica de Cohen-Coon ÅSTRÖM e HÄGGLUND (1995), sendo que foi efetuado o ajuste fino desse controlador diretamente no processo, com o objetivo de maximizar o desempenho nas regiões inferior, intermediária e superior do tanque de nível. Os parâmetros do controlador preditivo também foram ajustados experimentalmente (Ver Tabela 10.1).

Na Figura 10-9 mostram-se as transições de *setpoint* para o controlador neural e o **PI** na faixa de 0,5 a 1,3 m. Na Figura 10-10 podem ser observadas as respectivas ações de controle.

Para comparar adequadamente o desempenho dos controladores, em termos da velocidade de resposta às mudanças de *setpoint*, foi informado a esses, em cada instante de amostragem, apenas o *setpoint* atual e não os valores futuros desse. Assim sendo, no cálculo da ação de

controle, foi assumido que os valores futuros da referência eram iguais ao valor da referência definido para aquele instante de amostragem.

Tabela 10.1 - Parâmetros dos controladores

Controlador neural	PI
$\lambda = 0,0135$	
$\alpha = 0$	$K_c = 12 \text{ (V/m)}$
$N_u = 1$	$T_i = 32,5 \text{ (s)}$
$N_1 = 1$	
$N_2 = 15$	

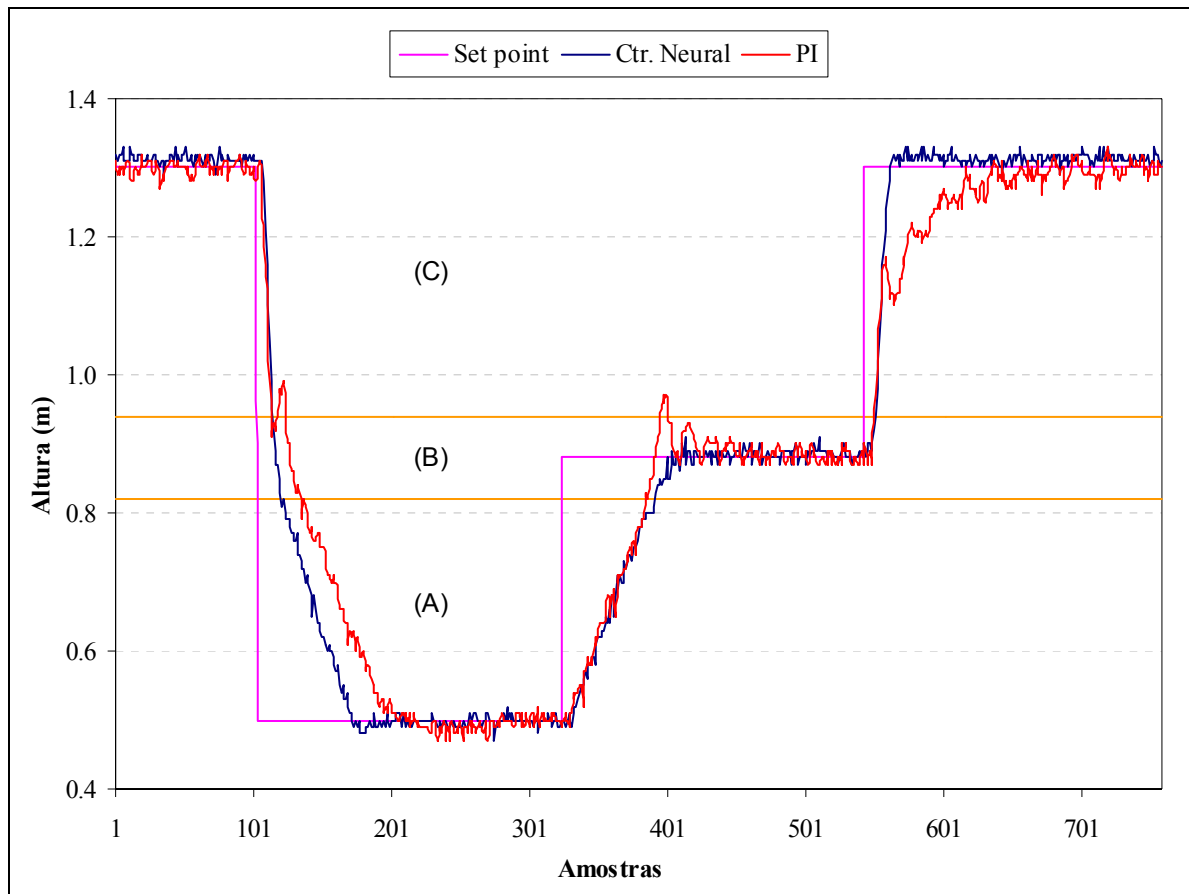


Figura 10-9 - Transições de *setpoint* para os controladores neural e PI. (A) seção inferior; (B) seção intermediária; (C) seção superior tanque

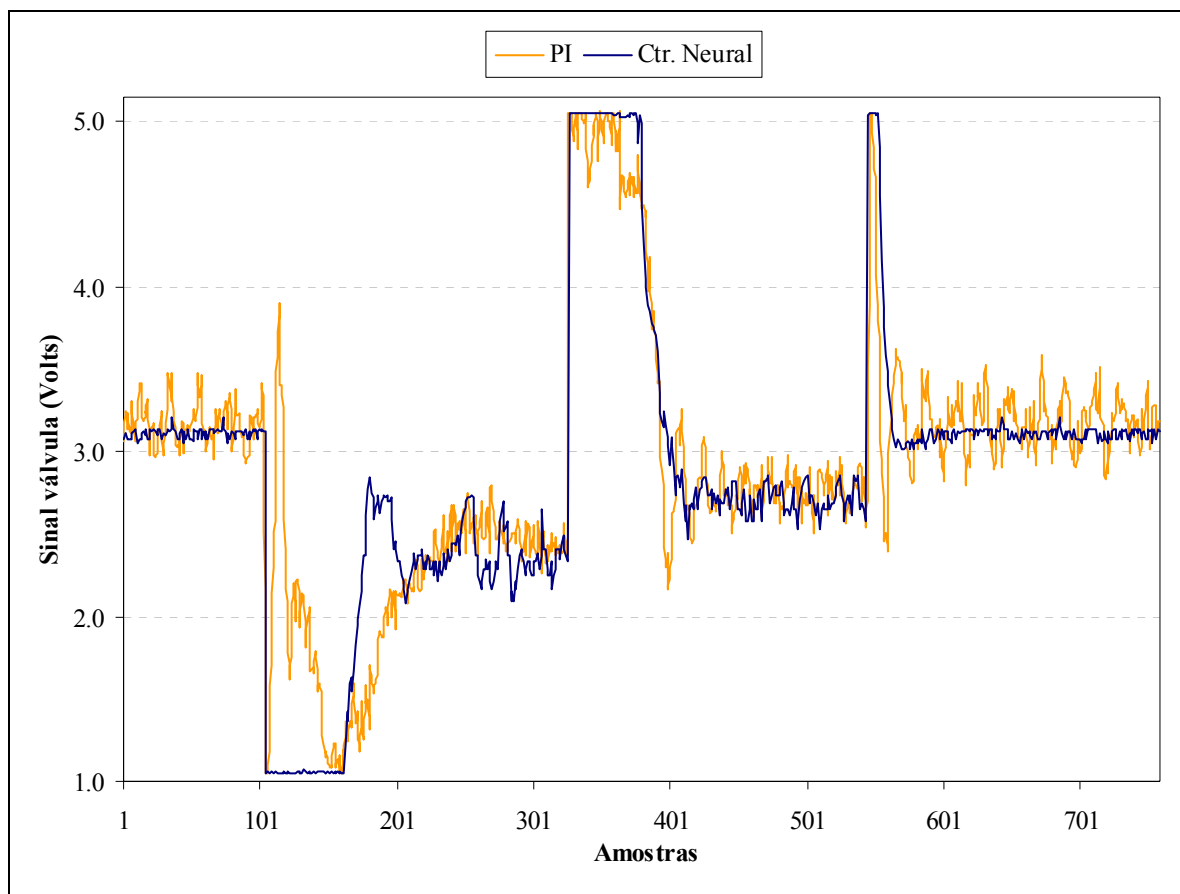


Figura 10-10 - Ações de controle tomadas pelos controladores neural e PI

Pode-se observar nas Figuras 10.9 e 10.10 que o controlador baseado em modelo neural apresentou apenas um pequeno *offset* na região superior do tanque e não apresentou *overshoot*. O **PI** apresentou *overshoot* na transição para a *setpoint* da região intermediária, oscilações no nível e variações bruscas da ação controle. Quando comparado ao **PI**, o controlador neural foi mais rápido e utilizou ações de controle mais suaves.

Deve-se notar que para $N_u = 1$, o problema de otimização da função custo do controlador é unidimensional. Este problema foi resolvido utilizando um método de pesquisa em linha. Como restrição à ação de controle, o valor aplicado ao atuador, em cada intervalo de amostragem, foi limitado ao intervalo 1 a 5 Volts. A restrição foi considerada explicitamente, ou seja, na minimização da função custo do controlador, a ação de controle foi limitada a assumir apenas valores no intervalo descrito em qualquer iteração do algoritmo de otimização.

Foi observado no controle do tanque de nível que o algoritmo de otimização sempre convergiu em um tempo menor que 0,1 s para os recursos computacionais utilizados. Dessa forma, não houve problemas na minimização da função custo do controlador uma vez que o tempo de amostragem utilizado foi de 1 s.

10.8 Conclusões

A capacidade de identificação da rede de funções de escala foi comprovada, no caso do tanque de nível, através de dois métodos: por comparação da curva de simulação obtida com a rede, para um determinado conjunto de dados de teste, e por análise de resíduos.

Além da não-linearidade de mudança de diâmetro do tanque de nível, a rede de funções de escala conseguiu identificar bem o processo na região inferior onde o sistema é lento e, portanto, apresenta pequena variação de nível, entre instantes de amostragem consecutivos, em relação ao ruído presente e para as ações de controle aplicadas.

A rede de funções de escala foi utilizada com sucesso como modelo de um controlador preditivo para o processo do tanque de nível.

O desempenho do controlador baseado na rede de funções de escala foi superior ao clássico **PI**, em relação ao comportamento das variáveis controlada e manipulada, na rejeição de perturbações tipo *setpoint*.

O projeto do controlador de nível baseado em rede neural não considerou o erro de previsão, ou seja, não foi utilizada nenhuma forma de realimentação desse erro. Nesta situação, caso o modelo apresentasse erros de identificação de estado estacionário, não seria possível eliminar o problema de *offset*. Na prática, foi observado apenas um pequeno *offset* no controle do nível na região superior do tanque, o que comprova a capacidade de previsão da rede de funções de escala.

11 CONTROLE DE UM TANQUE DE PH

Neste capítulo foi abordado o controle preditivo do tanque de pH simulado e descrito na seção 5.3. O processo possui três correntes de entrada - ácido, tampão e base - que são misturadas no tanque e o pH na saída desse é medido. Destas correntes, o pH foi a variável controlada, a vazão de base corresponde a variável manipulada e perturbações tipo carga foram aplicadas na vazão de ácido.

Foi efetuada uma comparação entre um controlador preditivo clássico, baseado em modelo linear, e outro que utiliza um modelo não-linear baseado na rede de funções de escala.

11.1 Identificação do Tanque de pH

Foram utilizados para identificação do processo os dados das perturbações aplicadas na vazão de base, mostrados na Figura 5-1, e os correspondentes dados de pH que podem ser vistos na Figura 5-2.

Conforme descrito anteriormente, a saída do processo no instante $(k+1)$, $pH_{(k+1)}$, pode ser identificada a partir das informações atuais $pH_{(k)}$ e $U_{(k)}$, sendo que $U_{(k)}$ é o valor da vazão de base no instante k . Os modelos empíricos linear e baseado em rede de funções de escala desenvolvidos para este sistema utilizaram, portanto, duas entradas $pH_{(k)}$ e $U_{(k)}$ e uma saída $pH_{(k+1)}$.

No caso da rede de funções de escala, foram utilizados três níveis de resolução para identificação do tanque de pH . Este modelo foi descrito e validado na seção 9.3 por simulação.

Para uma aproximação linear do tanque de pH , foi utilizado um modelo incremental linear, ou seja:

$$pH_{(k+1)} = a \cdot pH_{(k)} + b \cdot U_{(k)} + c \quad \Rightarrow \quad \Delta pH_{(k+1)} = a \cdot \Delta pH_{(k)} + b \cdot \Delta U_{(k)} \quad (\text{linear})$$

sendo os parâmetros a , b e c determinados por mínimos quadrados. A constante c , importante para fins de identificação uma vez que influencia o valor dos parâmetros a e b , é eliminada do controlador quando é utilizado um preditor incremental.

A constante c teve de ser utilizada na identificação uma vez que as médias das séries temporais de pH e U não são nulas.

O modelo ajustado aos dados do tanque de pH foi:

$$pH_{(k+1)} = 0,9045 \cdot pH_{(k)} + 0,0615 \cdot U_{(k)} - 0,2241$$

sendo que uma comparação entre os valores previstos pelo modelo incremental linear e os dados reais do processo pode ser vista na Figura 11-1.

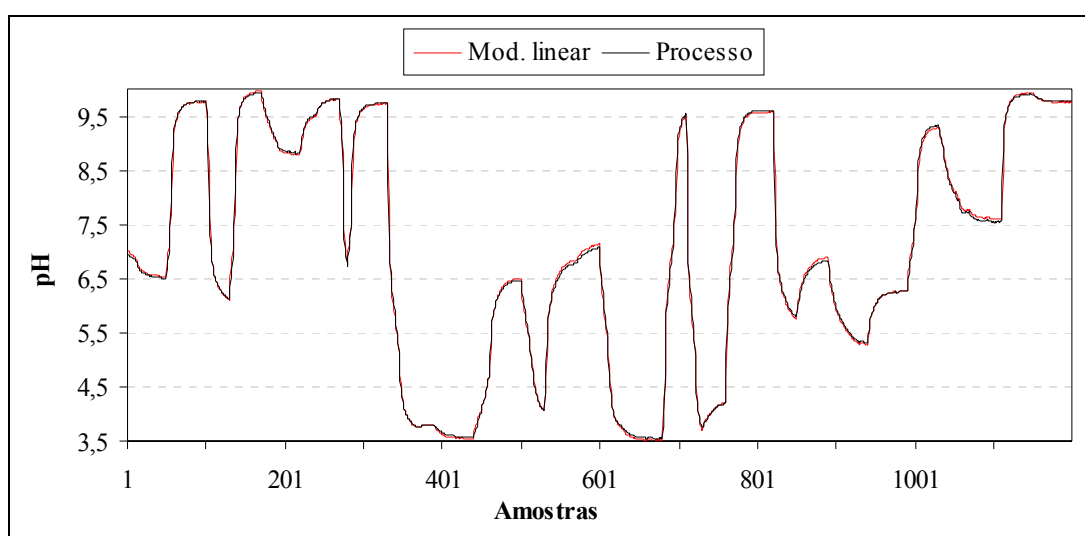


Figura 11-1 – Comparação dados reais processo x previsão do modelo linear

Analisando a Figura 11-1, tem-se a impressão de que o modelo incremental linear ajustou-se bem aos dados. No entanto, se for feita uma ampliação de uma determinada região desta figura, por exemplo, entre as amostras 530 e 630 percebe-se o erro de previsão é tendencioso, ou seja, os valores previstos pelo modelo incremental linear foram sempre superiores aos do processo (Ver Figura 11-2).

O comportamento tendencioso do erro de previsão do modelo incremental linear também foi observado em outras regiões do processo, além da descrita.

O modelo de um processo cujo erro de modelagem é tendencioso pode apresentar grandes desvios, em relação ao comportamento real do processo, quando utilizado para fazer previsões a vários passos à frente, uma vez que a saída deve ser realimentada e, portanto, os erros de previsão serão acumulados. Dessa forma, um controlador preditivo baseado em um modelo com tal característica pode apresentar desempenho insatisfatório.

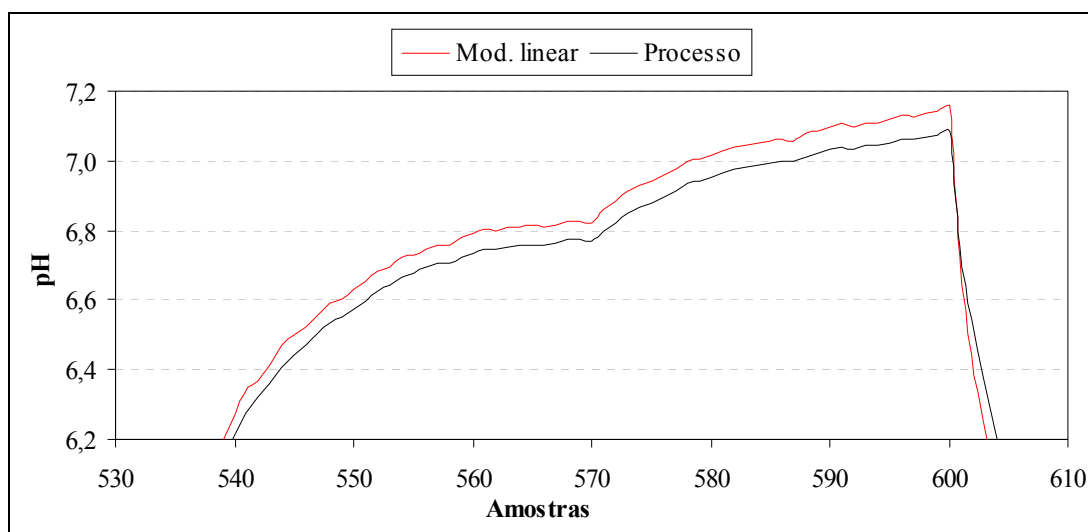


Figura 11-2 Comparação dados reais processo x previsão do modelo linear (correspondente às amostras entre 530 e 630 do grupo de dados)

11.2 Objetivos do Controlador

Os objetivos estabelecidos para o controlador foram efetuar rápidas e seguras transições de *setpoint*, sem *overshoot*, e rejeitar perturbações tipo carga.

O tempo de amostragem utilizado foi de 5 s, o mesmo da identificação dos modelos linear e da rede de funções de escala.

A ação de controle, correspondente à vazão de base, foi permitida variar no intervalo de 7,88 e 19,79 ml/s. Estes foram os limites de vazão de base utilizados na identificação. A restrição foi considerada explicitamente, ou seja, na minimização da função custo do controlador a ação de controle foi limitada a assumir apenas valores no intervalo descrito, em qualquer iteração do algoritmo de otimização..

A faixa de interesse de controle estabelecida foi *pH* entre 5 e 9.

11.3 Preditor Utilizado no Controlador

Seja:

$y_{(k+1)}$: Valor do processo no instante $(k+1)$;

$E_{(k+1)}$: Erro de predição cometido no instante $(k+1)$;

$y_{(k+1|k)}$: Predição do valor do processo, a partir de um modelo, para o instante $(k+1)$ utilizando-se os valores atuais e, em geral, atrasados desse.

Tem-se que:

$$y_{(k+1)} = y_{(k+1|k)} + E_{(k+1)}$$

O erro de predição em $(k+1)$ não é conhecido no instante k , sendo que normalmente considera-se esse igual ao erro de predição cometido no instante k , ou seja:

$$y_{(k+1)} = y_{(k+1|k)} + E_{(k+1)} \approx y_{(k+1|k)} + E_{(k)}$$

Assim sendo, um valor predito para $y_{(k+1)}$, possivelmente mais próximo desse que $y_{(k+1|k)}$, é dado por:

$$y_{(k+1)} \approx y_{(k+1|k)} + y_{(k)} - y_{(k|k-1)}$$

Este último preditor permite eliminar erros de modelagem de estado estacionário e rejeitar perturbações tipo carga, diferente do preditor utilizado no capítulo 10.

Exceto pelo preditor, todas as outras características do controlador permanecem inalteradas como as descritas no capítulo 10.

11.4 Resultados

Considerando o horizonte de controle $N_u = 1$ e a referência como sendo igual ao *setpoint*, fez-se um estudo da influência dos valores do horizonte de predição (N_2) e da penalização do esforço de controle (λ) para os casos de modelo linear e baseado em rede de funções de escala. Para este processo não há tempo morto, portanto, $N_1 = 1$.

Foi observado que, no caso do modelo linear ajustado, para pequenos valores de N_2 , não foi possível determinar um valor de λ que fornecesse um desempenho aceitável de controle para transições de *setpoint* na faixa de *pH* entre 5 e 9. Por exemplo, para $N_2 = 5$, mostra-se,

nas Figuras 11.3 e 11.4, o desempenho do controlador preditivo baseado em modelo linear para $\lambda=0,55$ e $\lambda=0,1$ respectivamente.

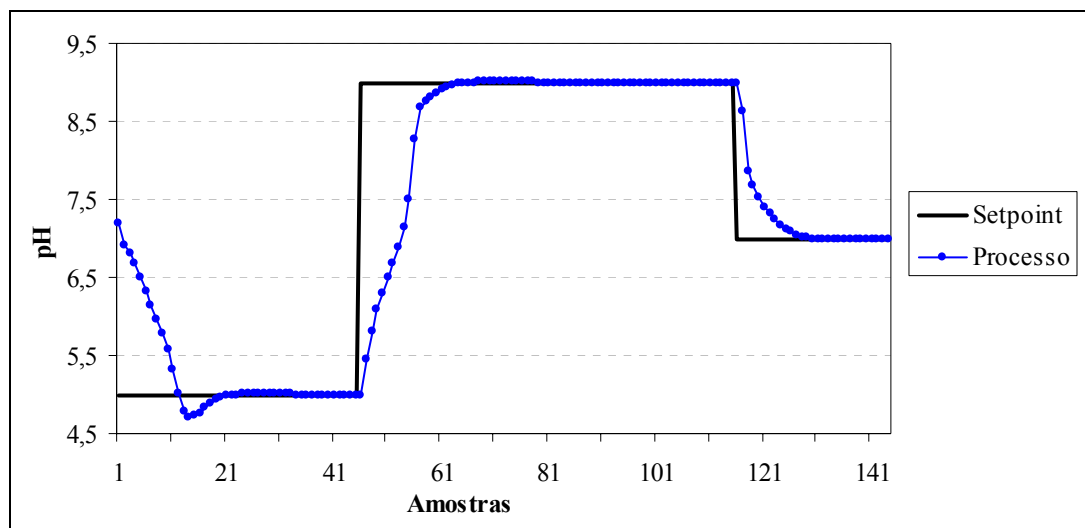


Figura 11-3 - Desempenho do controlador preditivo baseado em modelo linear ($N_2=5$, $\lambda=0,55$)

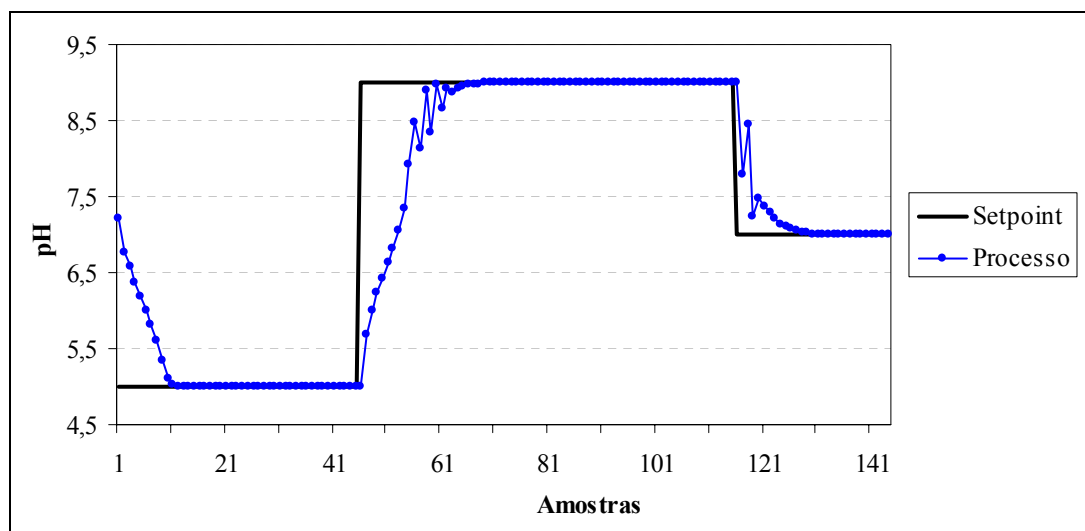


Figura 11-4 - Desempenho do controlador preditivo baseado em modelo linear ($N_2=5$, $\lambda=0,1$)

Na Figura 11-3, correspondente a $\lambda = 0,55$, nota-se que a penalização do esforço de controle é excessiva, sendo que o controlador não conseguiu evitar o *overshoot* para o *setpoint* correspondente ao $pH = 5$. Para $\lambda = 0,1$ (Ver Figura 11-4), o controlador foi mais rápido e não ocorreu *overshoot* nas transições de *setpoint*, no entanto, o comportamento transiente tornou-se oscilatório.

Assim sendo, no caso do controlador preditivo baseado em modelo linear, o valor de N_2 teve de ser aumentado para atender os objetivos de controle estabelecidos. Após algumas tentativas, foi determinado que estes objetivos eram satisfeitos para $N_2 = 20$.

Para o controlador preditivo baseado na rede de funções de escala, os requisitos de controle foram atendidos com apenas $N_2 = 2$.

Uma comparação de desempenho entre os controladores linear e baseado na rede de funções de escala, para perturbações tipo *setpoint*, pode ser vista na Figura 11-5. Os parâmetros utilizados para o primeiro foram $N_2 = 20$ e $\lambda = 11,5$ e para o último $N_2 = 2$ e $\lambda = 0,007$.

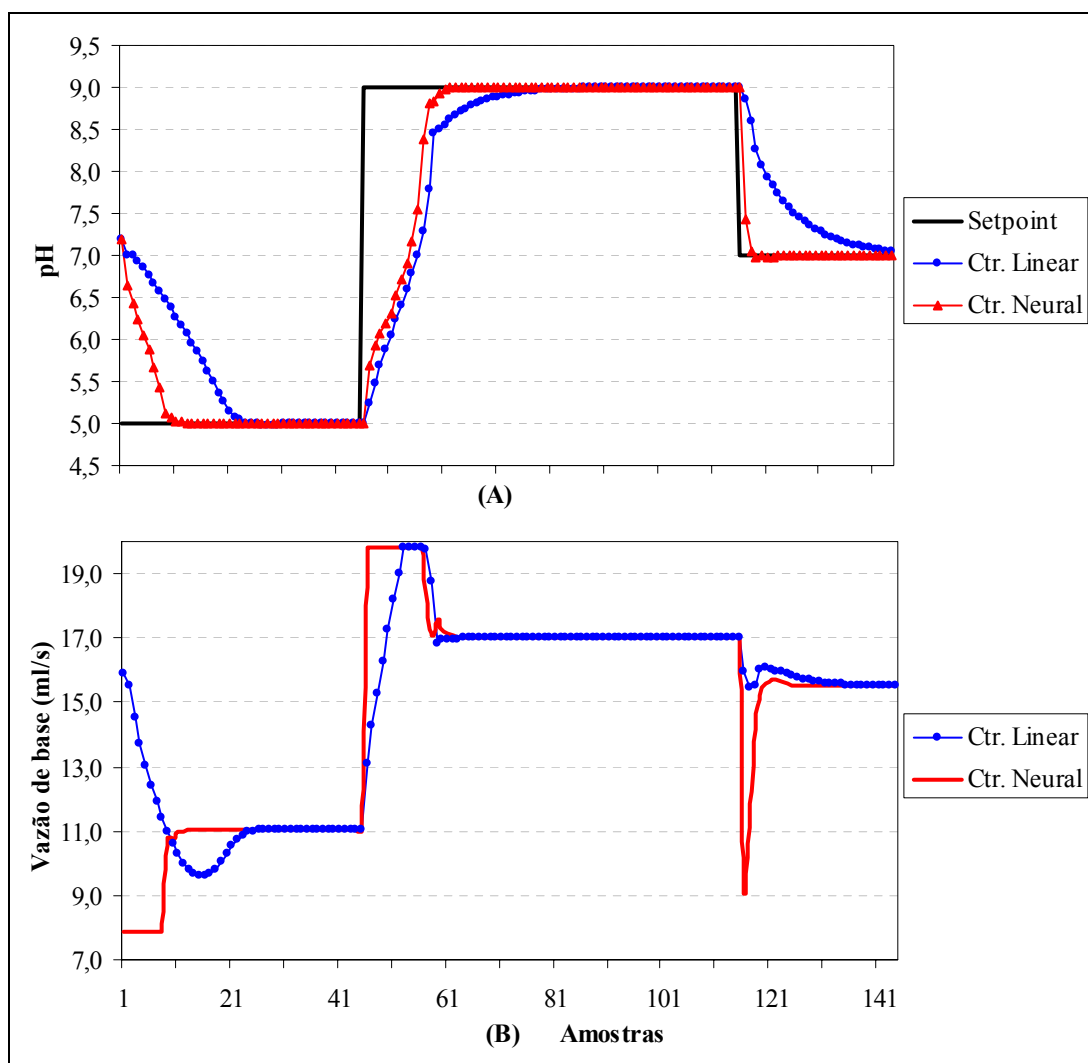


Figura 11-5 - Comparação entre o desempenho dos controladores preditivos baseados nos modelos linear e neural (perturbações *setpoint*) (A) *pH* (variável controlada); (B) vazão de base (variável manipulada)

Nota-se na Figura 11-5 que o controlador baseado na rede de funções de escala é bem menos conservativo, tomando rápidas e precisas ações de controle.

Os controladores preditivos baseados nos modelos linear e neural também foram comparados frente a perturbações tipos carga. Na Figura 11-6 mostra-se o desempenho destes para uma modificação brusca na vazão de ácido de 16,6 ml/s (valor nominal) para 12,0 ml/s e, após o sistema ter estabilizado, aplicou-se um degrau na vazão de ácido de 12 para 20 ml/s.

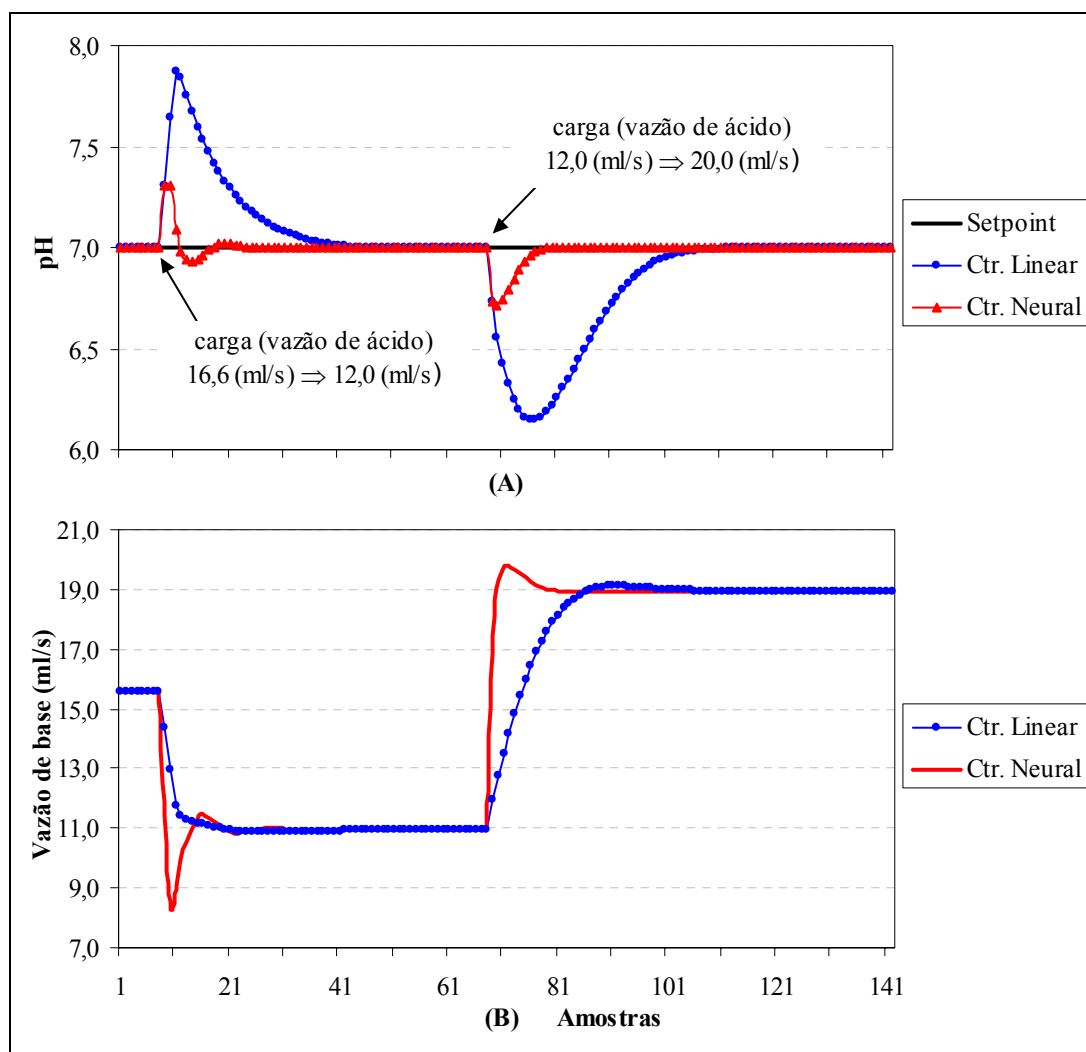


Figura 11-6 - Comparação de desempenho dos controladores preditivos baseados em modelo linear e neural (perturbações carga) (A) *pH* (variável controlada); (B) vazão de base (variável manipulada)

Nota-se na Figura 11-6 que o controlador baseado na rede de funções de escala conseguiu rejeitar com muito mais facilidade as cargas adicionadas, tomando, como no caso das perturbações tipo *setpoint*, rápidas e precisas ações de controle.

11.5 Conclusões

A rede de funções de escala foi utilizada com sucesso como modelo de um controlador preditivo para o tanque de pH .

Considerando o horizonte de controle como sendo igual a 1 e a referência como sendo igual ao *setpoint*, foi realizado um estudo para determinação dos valores ótimos de horizonte de predição e da penalização do esforço de controle, para os casos de modelo linear e baseado em rede funções de escala.

No controle do tanque de pH , para mudanças de *setpoint*, os controladores preditivos baseados em modelo não-linear (rede de funções de escala) e linear não apresentaram problemas de *offset* e *overshoot*, sendo que as ações de controle aplicadas foram suaves. No entanto, o primeiro foi bem menos conservativo, rejeitando perturbações tipo carga e efetuando transições de *setpoint* em um tempo significativamente menor. Basicamente, a diferença de velocidade de resposta pode ser atribuída ao valor do horizonte de predição. No caso não-linear, utilizou-se um horizonte de predição igual 2 e no caso linear, para atender aos objetivos de controle, foi necessário utilizar um valor 20 amostras.

12 CONCLUSÕES FINAIS

Uma função objetivo adequada à identificação de sistemas dinâmicos é formada pela soma de um termo relativo ao erro quadrático de aproximação e de outro correspondente a norma do vetor de pesos. A importância deste último, em relação ao primeiro, é controlada através de uma constante positiva denominada de parâmetro de regularização.

A minimização da norma dos pesos reduz as curvaturas presentes na superfície de modelos empíricos e, portanto, diminui o problema de sobre-treinamento em redes neurais.

O compromisso entre aproximar os dados e eliminar excessivas curvaturas da superfície de um modelo empírico impõe restrições no valor do parâmetro de regularização, sendo que a partir destas restrições foi determinado um valor ótimo para o parâmetro de regularização.

A rede neural proposta neste trabalho, denominada de rede de funções de escala e obtida a partir de modificações na *wavenet*, obteve sucesso na identificação e controle de processos não-lineares reconhecidos *benchmarks* nas áreas descritas.

A rede de funções de escala obteve melhor desempenho, se comparada com as redes de base radial e *feedforward*, na identificação de um atrator caótico e de um reator biológico. Os critérios utilizados para comparação foram o erro de aproximação de treinamento, teste e o critério de Akaike.

A rede de funções de escala foi validada com sucesso utilizando essa como simulador de processos, sendo efetuados testes com e sem ruído. Foram identificados vários processos não-lineares, entre eles um *CSTR* encamisado que apresenta multiplicidade de estados estacionários, um reator que apresenta resposta inversa variável (reator de Van de Vusse), um reator biológico e um tanque de *pH*. A rede de funções de escala generalizou muito bem pois, foi treinada para prever o comportamento dos processos para apenas um passo a frente, no entanto, foi capaz de prever um número arbitrário de passos de um grupo de dados de teste, diferente do utilizado no treinamento.

A rede de funções de escala foi utilizada na identificação e controle de um sistema experimental, um tanque de nível que apresenta alteração brusca de área. O desempenho do controlador baseado na rede de funções de escala foi superior ao clássico **PI**, em relação ao comportamento das variáveis controlada e manipulada, na rejeição de perturbações tipo *setpoint*.

O desempenho de um controlador preditivo baseado na rede de funções de escala e outro baseado em um modelo linear foi comparado para o tanque de *pH*. Ambos controladores não

apresentaram problemas de *offset* e *overshoot*, sendo que as ações de controle aplicadas foram suaves. No entanto, o primeiro foi bem menos conservativo, rejeitando perturbações tipo carga e efetuando transições de *setpoint* em um tempo significativamente menor.

Como continuação deste trabalho sugere-se:

- A rede de funções de escala, tal como a *wavenet*, apresenta crescimento exponencial do número de funções da base a medida que aumenta o número de entradas. Neste trabalho foram efetuadas modificações na *wavenet* de forma a diminuir a taxa de crescimento do número de funções. No entanto, outras simplificações deveriam ser efetuadas para que seja possível identificar processos com um grande número de variáveis de entrada.
- Algumas relações entre a(s) saída(s) e entrada(s) de um processo podem ser descritas através de restrições nos valores assumidos e/ou nas derivadas das variáveis significativas. Como a rede de funções de escala é facilmente diferenciável em relação aos parâmetros da camada de saída e às variáveis de entrada da rede, poderiam ser utilizadas outras formas de regularização, além da penalização da norma do vetor de pesos da rede.
- A identificação de um processo por uma rede de funções de escala pode ser interpretada como uma transformação de dados do domínio do tempo em projeções desses nos subespaços gerados pelas funções que constituem a rede. Assim sendo, a análise dos pesos e localização das respectivas funções de escala pode ser útil na detecção de características pouco observáveis no domínio do tempo.

13 APÊNDICE

13.1 Apêndice 1

Especificações dos equipamentos utilizados no controle do tanque de nível

Válvula de controle

Marca: Badger Meter - N °série 220194;

Características:

- Igual percentagem ar abre, falha fecha;
- Sinal de entrada 3 a 15 psi;

Conversores Eletropneumáticos

Marca: Hélix tipo P11-1111-2, N °série 530;

Sinal de entrada: 4 a 20 mA;

Sinal de saída: 3 a 15 psi;

Alimentação: 20 psi.

Filtros Reguladores de Pressão

Marca: Hélix - tipo F11-11;

Pressão máxima de Alimentação: 250 psi;

Saída: 20 psi.

Amplificador

Marca: Microquímica - modelo 308-A;

Características:

- Ganho programável: 0 a 9.000 vezes;
- Entrada: 0 -10 mV;
- Alimentação: 220 V.

Sensor de pressão

Marca: Contrisul;

Características:

- Entrada: 0 a 400 mBar;
- Saída: 4 a 20 mA;
- Alimentação 20 Vcc;
- Corpo em aço carbono, parte interna em aço inox 316.

Bomba Centrífuga

Marca: Schneider, modelo 02.01;

Potência ¼ cv, 3400 rpm;

Alimentação 220 V.

Microcomputador

Microcomputador tipo *Celeron* 1 GHz

Configuração:

- Unidade de disco rígido de 20 GBytes;
- Memória RAM de 128MBytes;
- Monitor SuperVGA color;
- Sistema operacional Windows™ 98.

Placa Analógica-Digital/Digital- Analógica

Interface de aquisição de dados para microcomputador tipo PC/XT/AT, marca DataTranslation, modelo DT2812, barramento ISA/EISA, com capacidade de transferência via DMA.

Características:

- 16 canais de entrada analógicos em modo comum ou 8 canais em modo diferencial;
- Resolução 12 *bits*;
- Taxa de amostragem de 60 KHz;
- Faixa de entrada de 0 a 1.25 V, 2,5 V, 5 V, 10 V, +/- 1,25 V, +/- 2.5 V, +/- 5 V, +/-10 V,
- 2 Canais de saída analógica;
- Conversão na faixa 10 µs a 3 min;
- 2 contadores/temporizadores de 16 *bits* dedicados a contagem de eventos e medidas de frequências;
- 1 contador/temporizador de 16 *bits* programável;
- 16 canais de saída digital;

- 16 canais de entrada digital;
- 3 canais compartilhados com contadores/temporizadores.

Conversor Tensão-Corrente

Conversor isolador de sinal marca ICI Instrumentação e Controle Industriais Ltda.;

Características:

- Configuração para termo-resistências, termopares e sinais padronizados;
- Sinal de saída: 0 a 20mA, 4 a 20 mA ou 0 a 10 Volts (optoisolado), configuráveis;
- 2 níveis de alarme configuráveis;
- Fonte auxiliar para transmissor: 5, 10, 12, 24 Vcc;
- Alimentação: 110 ou 220 Vac.

13.2 Apêndice 2 - Calibração do Sensor de Pressão

O objetivo da calibração dos elementos sensores é obter uma relação entre a variável a ser medida, no caso do tanque de nível esta é a altura, e o sinal enviado pelo sensor. A calibração obtida, realmente, não se refere ao sensor especificamente, mas sim ao conjunto de elementos de medição, ou seja, neste caso ao sensor de pressão e o conversor I/V. Desta forma, uma relação $\text{Voltagem}_x \text{Altura}$ é obtida.

13.2.1 Procedimento para Determinação da Curva de Calibração

O *software* de controle pode ser configurado para ler as informações da planta em volts e assim o *software* dever ser inicialmente configurado. Em seguida, a válvula Manual VM (Ver **Figura 10-2**) deve ser fechada e a válvula de controle aberta até encher completamente o tanque. Quando o tanque estiver cheio, a válvula de controle deve ser fechada e a válvula VM aberta e fechada em diferentes posições do tanque, anotando-se a altura da coluna de água e a respectiva voltagem. Dessa forma, obtém-se um grupo de dados de $\text{Altura(m)}_x \text{Voltagem(Volts)}$ com o qual procede-se uma aproximação linear. A equação da curva de

calibração obtida para o conjunto sensor de pressão-conversor I/V é mostrada abaixo (o coeficiente de correlação da reta obtido foi de 99,98%):

$$Altura(m) = -1,5356 + 0,9349 \cdot Volts$$

14 REFERÊNCIAS BIBLIOGRÁFICAS

- ALSBERG, B. K.; WOODWARD, A. M.; WINSON, M. K.; ROWLAND, J. J. e KELL D. B. - **Variable Selection in Wavelet Regression Models**. *Analytica Chimica Acta*, 368, p. 29-44, 1998.
- ANTONINI, M.; BARLAUD, M. e DAUBECHIES - **Image Coding Using Wavelet Transform**. *IEEE Transactions on Image Processing*, 1, p. 205-220, 1992.
- ÅSTRÖM, K. e HÄGGLUND, T. - **PID Controllers, Theory, Design, and Tuning**. Instrument Society of America, 1995.
- BAKSHI, B. R. e STEPHANOPOULOS. G - **Wave- Net: a Multiresolution, Hierarchical Neural Network with Localized Learning**. *AIChE J.*, 39, 1, p.57-81, 1993.
- BAZARAA, M. S.; SHERALI, H. D. e SHETTY C. M. - **Nonlinear Programming: Theory and Algorithms**, 2nd.ed., New York:John Wiley & Sons, 1993.
- BECERRA, V.M.; GALVAO, R.K.H.; CALADO, J.M.F. e SILVA, P.M. - **Linear Wavelet Models for Nonlinear Identification Applied to a Pressure Plant**. *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2002*, Vol. 3, p. 2180-2185, 2002.
- BILLINGS, S. A. e VOON, W. S. F. - **Correlation Based Model Validity Tests for Nonlinear Models**. *Int. J. Control*, Vol. 44, p-235-244., 1986.
- CAMACHO, E. F. e BORDONS, C. - **Model Predictive Control in the Process Industry**, 1994.
- CLAUMANN, C. A. - **Modelagem e Controle de Processos Não-lineares: Uma Aplicação de Algoritmos Genéticos no Treinamento de Redes Neurais Recorrentes**. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Química, UFSC, Florianópolis, SC, Brasil, 1999.
- CHEN, Z. e HAYKIN, S. - **A New View on Regularization Theory**. *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 3, p.1642-1647, 2001.
- CHENG, Y.; KARJALA, W. e HIMMELBLAU, D. M. - **Identification of Nonlinear Dynamic Process with Unknown and Variable Dead Time Using an Internal Recurrent Neural Network**. *Ind. Eng. Chem. Res.*, p. 1735-1742, 1995.
- CRISTEA, P.; TUDUCE, R. e CRISTEA, A. - **Time Series Prediction with Wavelet Neural Networks**. *Proceedings of the 5th Seminar on Neural Network Applications in Electrical Engineering, NEUREL 2000*, p. 5-10, 2000.

- DAUBECHIES, I. - **Orthonormal Bases of Compactly Supported Wavelets**. Comm. On Pure and Appl. Math., vol. XLI, p. 909, 1988.
- DAUBECHIES, I. - **Ten Lectures on Wavelets**, SIAM, 1992.
- DE JESUS, O.; PUKRITTAYAKAMEE, A. e HAGAN, M.T. - **A Comparison of Neural Network Control Algorithms**. Proceedings International Joint Conference on Neural Networks, IJCNN 2001, Vol. 1, p. 521–526, 2001.
- DELYON, B.; JUDISKY, A. e BENVENISTE A. - **Accuracy Analysis for Wavelet Approximations**. IEEE Trans. Neural Networks, 6(2), p. 332-348, 1995.
- DONGBING, G. e HUOSHENG, H. - **Wavelet Neural Network Based Predictive Control for Mobile Robots**. IEEE International Conference on Systems, Man, and Cybernetics, Vol. 5, p. 3544–3549, 2000.
- ELIAS-JUAREZ, A. e KANTOR, J. C. - **On the Application of Wavelets to Model Predictive Control**. American Control Conference, p. 1582-1586, 1992.
- EMBIRUÇU, M. - **Controle de Processos Não-lineares. Tese de Mestrado - COPPE/UFRJ**, 1993.
- FATEMI, O. e BOLOUKI, S. - **Wavelet Transform Architectures for Video Applications**. Canadian Conference on Electrical and Computer Engineering, IEEE CCECE. Vol. 2, p. 651–656, 2002.
- FAUSETT, L. - **Fundamentals of Neural Networks, Architectures, Algorithms and Applications**. Florida Institute of Technology. Prantice Hall International, 1994.
- GAO, L.; LU, L. e LI, Z. - **Prediction of Chaotic Time Series Based on Wavelet Neural Network**. OCEANS 2001 MTS/IEEE Conference and Exhibition, Vol. 4, p. 2046–2050, 2001.
- GARCIA, C. E.; PRETT, D. M. e MORARI, M. - **Model Predictive Control: Theory and Practice – a Survey**. Automatica, Vol. 25, p. 335-348, 1989.
- GHOSH, R. e VERMA, B. - **Least Square Method Based Evolutionary Neural Learning Algorithm**. Proceedings Joint International Conference on Neural Networks IJCNN 2001, Vol.4, p. 2596–2601, 2001.
- GIROSI, F., JONES, M. e POGGIO, T. - **Regularization Theory and Neural Networks Architectures**. Neural Computation, 7, p. 219-269, 1995
- GOLDBERG, D. - **Genetic Algorithms in Search, Optimization, and Machine Learning**. Addison-Wesley, New York, 1989.
- GROSSMANN, A. e MORLET, J. - **Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape**. SIAM J. Math Anal., 15, p.723-736, 1984.

- HAYKIN, S. - **Neural Networks – A Comprehensive Foundation**, 1999.
- HAROLD, H. S., TELFER, B. e KADAMBE, S. - **Neural Network Adaptive Wavelets for Signal Representation and Classification**. *Opt. Engng*, 31(9), p. 1907-1916, 1992.
- HARRIS, C. J. - **Advances in Intelligent Control**. Taylor & Francis, Great Britain, 1994.
- HENSON, M. A. e SEBORG, D. E. - **An Internal Model Control Strategy for Nonlinear Systems**. *AIChE J.*, Vol. 37, 7, p. 1065, 1991.
- HOLLAND, J. - **Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. MIT Press/ Bradford Books edition, 1975.
- HUANG, D. e JIN, Y. - **The Application of Wavelet Neural Networks to Nonlinear Predictive Control**. *International Conference on Neural Networks*, Vol. 2, p. 724–727, 1997.
- HUNG, C.; KIM, Y. e COLEMAN, T.L. - **A Comparative Study of Radial Basis Function Neural Networks and Wavelet Neural Networks in Classification of Remotely Sensed Data**. *Proceedings of the 5th Biannual World Automation Congress*. Vol. 13, p. 455-461, 2002.
- HUNT, K. J., SBARBARO, D., ZBIKOWSKI, R. e GAWTHROP, P. J. - **Neural Networks for Control Systems - A Survey**. *Automatica*, vol. 28, n° 6, p.1083-1112, 1992.
- JOHANSEN, T. A. - **On Tikhonov Regularization, Bias and Variance in Nonlinear System Identification**. *Automatica*, Vol. 33, p.441-446, 1997.
- KIM, S.; KIM Y.; SEO J. e JEON H. - **Design of the Scaling-Wavelet Neural Network Using Genetic Algorithm**. *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2002*, Vol. 3., p. 2174 –2179, 2002.
- KORENBERG, M. J.; BILLINGS, S. A.; LIU, P. Y. e MCILROY, P. J. - **Orthogonal Parameter Estimation Algorithm for Nonlinear Stochastic System**. *Int. J. Control*, 48, p.193-210, 1992.
- KOSANOVICH, K. A. e PIOVOSO, M. J. - **PCA of Wavelet Transformed Data Process for Monitoring**. *Intelligent Data Analysis 1*, p. 85-99, 1997.
- LEWIS, D. W. - **Matrix Theory**. World Scientific Publishing Co. Pte. Ltd, p. 105, 1991.
- LI, S. e XI, Y. - **Applications of Wavelet to Constrained Generalized Predictive Control**. *Proceedings of the IEEE International Symposium on Intelligent Control*, p. 247–252, 2000.

- LING, B. - **Neural Network Based Feedforward Adapter for Batch Process Control**. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Vol. 4, p. 305–310, 2000.
- LJUNG, L. - **System Identification**. Theory for the user. Prantice Hall, 1987.
- MALLAT, S. A. - **A Theory for Multiresolution Signal Decomposition: The Wavelet Representation**. IEEE Trans. Pat. Anal Mach. Intel., 11, 7, p. 674-693, 1989.
- MEYER, Y. - **Principe D'incertitude, Bases Hilbertiennes at Algebres D'operateurs**. Seminaire Bourbaki, nr. 662.
- MOTARD, R. L. e JOSEPH, B. - **Wavelet Applications in Chemical Engineering**. Kluwer Academic Publishers, 1994.
- NERRAND, O.; ROUSSEL-RAGOT, P.; URBANI, D.; PERSONNAZ, L. e DREYFUS, G. - **Training Recurrent Neural Networks: Why and How? An Illustration in Dynamical Process Modeling**. IEEE Transactions on Neural Networks, Vol. 5, n°2, março, p. 178-184, 1994.
- NIEMICE, M. e KRAVARIS, C. - **Controller Synthesis for Multivariable Nonlinear Nonminimum-phase Process**. Proceedings of the American Control Conference, Philadelphia, Pennsylvania, p. 2076-2080, 1998.
- O'SULLIVAN, F. - **A Statistical Perspective on Ill-posed Inverse Problems**. Statistical Science, 1, p. 502-527, 1986.
- PATI, Y. C. e KRISHNAPRASAD, P. S. - **Analysis and Systhesis of Feedforward Neural Networks Using Discrete Affine Wavelet Tranformations**. IEEE Trans. Neural Networks, 4(1), p.7385, 1993.
- PEDERSEN, M. W. - **Training recurrent networks**. Proceedings of the 1997 VII IEEE Workshop Neural Networks for Signal Processing, p. 355–364, 1997.
- PEREZ, H.; OGUNNAIKE, B. e DEVASIA, S. -**Output Tracking Between Operating Points for Nonlinear Process: Van de Vusse Example**. IEEE Transactions on Control Systems Technology, vol. 10, n° 4, julho, p. 611-617, 2002.
- POGGIO, T. e GIROSI, F. - **Networks for Approximation and Learning**. Proceedings of the IEEE, Volume: 78 , 9, p. 1481–1497, 1990.
- REN, Z., CHEN, J., TANG, X. e YAN, W. - **A Combined Method Based on Neural Network for Control System Fault Detection and Diagnosis**. Proceedings of the 2000 IEEE International Conference on Control Applications, p. 104–108, 2000.
- ROQUEIRO, N. - **Redes de Wavelets na Modelagem de Processos não-Lineares**. Tese de Doutorado - COPPE/UFRJ, 1995.

- RUMELHART, D. E. e MCCLELLAND, J. L. - **Parallel Distributed Processing: Explorations in the Microstructure of Cognition**. Vol 1, 1986.
- SAFAVI, A. A. e ROMAGNOLI, J. A. - **Application of Wave-nets to Modelling and Optimisation of a Multidimensional Chemical Process**. Proceedings IEEE International Conference on Neural Networks, Vol. 4, p. 1724 –1728, 1995.
- SAXENA, S.C.; KUMAR, V. e WAGHMARE, L.M. - **Temperature Control of CST Process Using Gaussian Neural Network with Adaptive Learning Rate**. Proceedings of the IEEE International Conference on Control Applications, p. 202 –207, 2000.
- SEBORG, D. E.; EDGAR, T. F. e MELLICHAMP, D. A. - **Process Dynamics and Control**, New York, ed. John Wiley & Sons, 1989.
- SHAO, R., JIA, F., MARTIN, E. B. e MORRIS A. J. - **Wavelets and Nonlinear Principal Component Analysis for Process Monitoring**. Control Engineering Practice, vol. 7, p. 865-879, 1999.
- SJOBERG, J. e VIBERG, M. - **Separable Nonlinear Least-squares Minimization - Possible Improvements for Neural Net Fitting**. Proceedings of the VII IEEE Workshop Neural Networks for Signal Processing, p. 345–354, 1997.
- STRANG, G. e NGUYEN, T. - **Wavelets e Filter Banks**, 1996.
- TAHUATA, T. - **Relatório Interno** - COPPE/UFRJ, 1992.
- THOMPSON, M. L. e KRAMER, M. A. - **Modeling Chemical Process Using Prior Knowledge and Neural Networks**. AIChE. J., 40, p. 1328-1340, 1994
- TIKHONOV, A. N. e ARSENIN V. Y. - **Solutions of Ill-posed Problems**. Wiston, Washington DC, 1977.
- TULLEKEN, H. J. A. F. - **Gray-box Modeling and Identification Using Physical Knowledge and Bayesian Techniques**. Automatica, 29, p. 285-308, 1993.
- VENKATARAMAN, S. e GONZ, R.S. - **Prediction of Polymer Insulating Material Degradation Using Daubechies Wavelet Transformation**. Annual Report Conference on Electrical Insulation and Dielectric Phenomena, p. 323–326, 2002.
- XIA, X.; HUANG, D. e JIN, Y. - **Nonlinear Adaptive Predictive Control Based on Orthogonal Wavelet Networks**. Proceedings of the 4th World Congress on Intelligent Control and Automation, Vol. 1, p. 305–311, 2002.
- XU, J. e HO, D.W.C. - **Adaptive Wavelet Networks for Nonlinear System Identification**. Proceedings of the American Control Conference, Vol. 5, p. 3472–3473, 1999
- XU, J. e TAN, Y. - **Nonlinear Adaptive Wavelet Control Using Constructive Wavelet Networks**. Proceedings of the American Control Conference, Vol. 1, p 624–629, 2001.

- WAHBA, G. - **Spline Models for Observational Data**. SIAM, Philadelphia, 1990.
- ZHANG, Q. e BENVENISTE, A. - **Wavelet Networks**. IEEE Trans. Neural Networks, 3,6, p. 889-898, 1992.
- ZHAO, J.; CHEN, B. e SHEN, J. - **Multidimensional Non-orthogonal Wavelet-sigmoid Basis Function Neural Network for Dynamic Process Fault Diagnosis**. Computers and Chemical Engineering, 23, p. 83-92, 1998.
- ZUPAN, J. e GASTEIGER, J. - **Neural Networks for Chemists**, 1993.
- YAMAGUCHI, C. - **Wavelet Analysis of Normal and Epileptic EEG**. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society Engineering in Medicine and Biology, EMBS/BMES, Vol. 1 , 23-26, p. 96 –97, 2002.