Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica e de Computação

IDENTIFICAÇÃO NÃO LINEAR USANDO UMA REDE *FUZZY WAVELET NEURAL NETWORK* MODIFICADA

José Medeiros de Araújo Júnior

Orientador: Fábio Meneghetti Ugulino de Araújo

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Automação e Sistemas) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Natal-RN, Março, 2014

UFRN / Biblioteca Central Zila Mamede.
Catalogação da Publicação na Fonte.

Araújo Júnior, José Medeiros de.

Identificação não linear usando uma rede fuzzy wavelet neural network modificada. / José Medeiros de Araújo Júnior. – Natal, RN, 2014.

109 f.: il.

Orientador: Fábio Meneghetti Ugulino de Araújo.

Tese (Doutorado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica.

1. Identificação de Sistemas – Tese. 2. Inferência – Tese. 3. Redes Neurais Artificiais – Tese. 4. Teoria *Wavelet* – Tese. 5. Redes *Wavelet Neural Network* – Tese. 6. Redes *Fuzzy Wavelet Neural Network* – Tese. I. Araújo, Meneghetti Ugulino de. II. Universidade Federal do Rio Grande do Norte. III. Título.

RN/UF/BCZM CDU
004.72

IDENTIFICAÇÃO NÃO LINEAR USANDO UMA REDE *FUZZY WAVELET NEURAL NETWORK* MODIFICADA

José Medeiros de Araújo Júnior

Tese de Doutorado aprovada em 24 de março de 2014 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Fábio Meneghetti Ugulino de Araújo (orientador) . . . . . DCA/UFRN

Prof. Dr. André Laurindo Maitelli . . . . . DCA/UFRN

Profª Drª Danielle Simone da Silva Casillo . . . . . EXA/UFERSA

Prof. Dr. Otacílio da Mota Almeida . . . . . CEE/UFPI

Prof. Dr. Takashi Yoneyama . . . . . IEE/ITA

Aos meus pais, José Medeiros e Francisca Alves, a minha esposa, Uiara, e aos meus irmãos, Marson, Patrícia e Kênia.

Agradecimentos

Agradeço primeiramente a Deus pela vida e por me dar a oportunidade de concluir este trabalho.

Agradeço a minha família, em especial aos meus pais, Francisca Alves Machado de Medeiros e José Medeiros de Araújo, pelo apoio dado em todos esses anos de estudo, pelo incentivo, pelo exemplo de honestidade, humildade e dedicação.

Aos meus irmãos, Marson Ricardo Machado de Medeiros, Patrícia Regina Machado de Medeiros e Kênia Machado de Medeiros, por terem me auxiliado em toda minha vida.

A minha esposa Uiara Oliveira Costa, pela compreensão, companhia, sugestões e apoio, que foram decisivos para a conclusão deste trabalho.

Ao meu orientador, Professor Fábio Meneghetti Ugulino de Araújo, pela confiança e entusiasmo, sempre me orientando de forma valiosa ao longo da elaboração deste trabalho e incentivando-me à pesquisa.

Aos professores Otacílio da Mota Almeida, Takashi Yoneyama, André Laurindo Maitelli e Danielle Simone Casillo, pelas sugestões e contribuições.

Ao colega e amigo Leandro Luttiane da Silva Linhares, pelo apoio na elaboração deste trabalho, sempre disposto a colaborar de forma decisiva nas situações mais difíceis.

Agradeço também aos colegas Carlos André Guerra Fonseca, Marconi Câmara Rodrigues, Marcelo Guerra, Daniel Guerra, José Soares, Luciano Júnior, Marcílio Onofre Filho, e tantos outros que contribuíram para o desenvolvimento deste trabalho.

Aos professores, demais colegas e funcionários da pós-graduação, pelos conhecimentos disponibilizados e pelo apoio.

Aos colegas do Curso de Engenharia Elétrica da Universidade Federal do Piauí, pelo apoio e compreensão.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.

A todos aqueles, enfim, que contribuíram e aprimoraram o conteúdo deste trabalho.

Resumo

Nas últimas décadas, as redes neurais têm se estabelecido como uma das principais ferramentas para a identificação de sistemas não lineares. Entre os diversos tipos de redes utilizadas em identificação, uma que se pode destacar é a rede neural *wavelet (*ou *Wavelet Neural Network* - WNN*)*. Esta rede combina as características de multirresolução da teoria *wavelet* com a capacidade de aprendizado e generalização das redes neurais, podendo fornecer modelos mais exatos do que os obtidos pelas redes tradicionais. Uma evolução das redes WNN consiste em combinar a estrutura *neuro-fuzzy* ANFIS (*Adaptive Network Based Fuzzy Inference System*) com estas redes, gerando-se a estrutura *Fuzzy Wavelet Neural Network* - FWNN. Essa rede é muito similar às redes ANFIS, com a diferença de que os tradicionais polinômios presentes nos consequentes desta rede são substituídos por redes WNN. O presente trabalho propõe uma rede FWNN modificada para a identificação de sistemas dinâmicos não lineares. Nessa estrutura, somente funções *wavelets* são utilizadas nos consequentes. Desta forma, é possível obter uma simplificação da estrutura com relação a outras estruturas descritas na literatura, diminuindo o número de parâmetros ajustáveis da rede. Para avaliar o desempenho da rede FWNN com essa modificação, é realizada uma análise das características da rede, verificando-se as vantagens, desvantagens e o custo-benefício quando comparada com outras estruturas FWNNs. As avaliações são realizadas a partir da identificação de dois sistemas simulados tradicionalmente encontrados na literatura e um sistema real não linear, consistindo de um tanque de multisseções e não linear. Por fim, a rede foi utilizada para inferir valores de temperatura e umidade no interior de uma incubadora neonatal. A execução dessa análise baseia-se em vários critérios, tais como: erro médio quadrático, número de épocas de treinamento, número de parâmetros ajustáveis, variância do erro médio quadrático, entre outros. Os resultados encontrados evidenciam a capacidade de generalização da estrutura modificada, apesar da simplificação realizada.

Palavras-chave: Identificação de Sistemas, Inferência, Redes Neurais Artificiais, Teoria *Wavelet*, Redes *Wavelet Neural Network*, Redes *Fuzzy Wavelet Neural Network*.

Abstract

In last decades, neural networks have been established as a major tool for the identification of nonlinear systems. Among the various types of networks used in identification, one that can be highlighted is the wavelet neural network (WNN). This network combines the characteristics of wavelet multiresolution theory with learning ability and generalization of neural networks usually, providing more accurate models than those ones obtained by traditional networks. An extension of WNN networks is to combine the neuro-fuzzy ANFIS (Adaptive Network Based Fuzzy Inference System) structure with wavelets, leading to generate the Fuzzy Wavelet Neural Network - FWNN structure. This network is very similar to ANFIS networks, with the difference that traditional polynomials present in consequent of this network are replaced by WNN networks. This paper proposes the identification of nonlinear dynamical systems from a network FWNN modified. In the proposed structure, functions only wavelets are used in the consequent. Thus, it is possible to obtain a simplification of the structure, reducing the number of adjustable parameters of the network. To evaluate the performance of network FWNN with this modification, an analysis of network performance is made, verifying advantages, disadvantages and cost effectiveness when compared to other existing FWNN structures in literature. The evaluations are carried out via the identification of two simulated systems traditionally found in the literature and a real nonlinear system, consisting of a nonlinear multi section tank. Finally, the network is used to infer values of temperature and humidity inside of a neonatal incubator. The execution of such analyzes is based on various criteria, like: mean squared error, number of training epochs, number of adjustable parameters, the variation of the mean square error, among others. The results found show the generalization ability of the modified structure, despite the simplification performed.

Keywords: System Identification, Inference, Artificial Neural Networks, Wavelets, Wavelet Neural Network, Fuzzy Wavelet Neural Network.

Sumário

Sumário	vii
Lista de Figuras	viii
Lista de Tabelas	x
Lista de Símbolos e Abreviaturas	xi
1. Introdução	1
2. Fundamentação Teórica	14
2.1. Redes Neurais	15
2.2. Funções *Wavelets*	17
2.3. *Wavelet Neural Network*	19
2.4. *Fuzzy Wavelet Neural Network*	25
2.5. Conclusões	28
3. Sistema de Identificação Proposto	29
3.1. *Fuzzy Wavelet Neural Network Modificada*	29
3.2. Conclusões	37
4. Identificação Não Linear	38
4.1. Conceitos Gerais	38
4.2. Estruturas de Modelagem	42
4.3. Conclusões	45
5. Metodologia e Estudos de Caso	46
5.1. Sistema Dinâmico Simulado 1	48
5.2. Sistema Dinâmico Simulado 2	52
5.3. Sistema Dinâmico Real 1 - Tanque Multisseções	56
5.4. Sistema Dinâmico Real 2 - Incubadora Neonatal	65
6. Conclusão e Perspectivas	84
Referências Bibliográficas	87

Lista de Figuras

Figura 2.1: Rede MLP totalmente conectada de três camadas 15

Figura 2.2: Estrutura básica de uma rede *wavelet* 20

Figura 2.3: *Wavelet* da família *Mexican Hat* 22

Figura 2.4: *Wavelets* da família *Mexican Hat* deslocadas 22

Figura 2.5: *Wavelets* da família *Mexican Hat* dilatadas 23

Figura 2.6: Estrutura da rede *Fuzzy Wavelet Neural Network* 26

Figura 3.1: Estrutura da rede *Fuzzy Wavelet Neural Network* proposta 30

Figura 3.2: Estrutura de um neurônio presente na camada consequente 33

Figura 4.1: Esquema de identificação de sistemas dinâmicos – Série-Paralelo 39

Figura 4.2: Procedimento básico de identificação 40

Figura 4.3: Estrutura do modelo NNARX 44

Figura 4.4: Estrutura do modelo NNFIR 45

Figura 5.1: Resultado da validação com a FWNN proposta para o sistema dinâmico simulado 1 51

Figura 5.2: Valor do RMSE obtido durante o treinamento e validação do sistema dinâmico simulado 1 51

Figura 5.3: Resultado da validação com a FWNN proposta para o sistema dinâmico simulado 2 54

Figura 5.4: Valor do RMSE obtido durante o treinamento e validação do sistema dinâmico simulado 2 55

Figura 5.5: Desenho esquemático do sistema de tanque multisseções 56

Figura 5.6: Visão geral do tanque multisseções. 57

Figura 5.7: Sinal PRS aplicado na bomba de água do tanque 58

Figura 5.8: Estrutura do modelo NNARX usando a rede FWNN 59

Figura 5.9: Histograma dos MSEs para: (a) FWNN – Proposta, (b) FWNN – Yilmaz & Oysal (2010). 62

Figura 5.10: Erro Médio Quadrático da rede FWNN proposta 63

Figura 5.11: Resposta da rede FWNN proposta 63

Figura 5.12: Resposta da rede FWNN proposta por Lu (2011) 64

Figura 5.13: Resposta da rede FWNN proposta por Abiyev & Kaynak (2008b) 64

Figura 5.14: Resposta da rede FWNN proposta por Yilmaz & Oysal (2012) 65

Figura 5.15: Incubadora neonatal do GPAR utilizada para validação dos algoritmos 68

Figura 5.16: Visão geral do modo de operação da incubadora neonatal 68

Figura 5.17: (a) Esquema da base do sistema de aquisição de dados, (b) Desenho da base do sistema de aquisição de dados 69

Figura 5.18: Estrutura da FWNN aplicada na inferência 71

Figura 5.19: Estrutura de treinamento das redes: (a) umidade, e (b) temperaturas 73

Figura 5.20: Sinais utilizados como entradas da rede: (a) Umidade no ponto de saída da incubadora - U_S, (b) Temperatura no ponto de saída da rede - P_S 73

Figura 5.21: Erro médio quadrático de treinamento da rede FWNN proposta 74

Figura 5.22: Histograma dos MSEs obtidos na inferência da umidade e temperatura no ponto A, para as redes (a) e (c) FWNN – Proposta, (b) e (d) FWNN – Yilmaz & Oysal (2010). 78

Figura 5.23: Resposta do melhor modelo obtido pela FWNN Proposta: (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E 79

Figura 5.24: Resposta do melhor modelo obtido pela FWNN presente em Lu (2011): (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E 80

Figura 5.25: Resposta do melhor modelo obtido pela FWNN presente em Abiyev & Kaynak (2008b): (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E 81

Figura 5.26: Resposta do melhor modelo obtido pela FWNN presente em Yilmaz & Oysal (2010): (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E 82

Lista de Tabelas

Tabela 5.1: Comparação do resultado da FWNN proposta com outras redes para o sistema dinâmico simulado 1 52

Tabela 5.2: Comparação do resultado da FWNN proposta com outras redes para o sistema dinâmico simulado 2 54

Tabela 5.3: Comparação entre a FWNN proposta e outras redes para tanque de nível multisseções 61

Tabela 5.4: Variáveis primárias da incubadora neonatal 72

Tabela 5.5: Resultados da análise das estruturas aplicadas na inferência 75

Tabela 5.6: Resultados da análise das estruturas aplicadas na inferência 76

Lista de Abreviaturas e Símbolos

ANFIS	*Adaptive Network Based Fuzzy Inference System*
AG	Algoritmo Genético
ARMAX	*AutoRegressive, Moving Average, eXogenous input*
ARX	*AutoRegressive, eXogenous input*
BIBO	*Bounded Input-Bounded Output*
BFGS	*Broyden-Fletcher-Goldfarb-Shanno*
CMOS	*Complementary Metal Oxide Semiconductor*
EMQ	Erro Médio Quadrático
ERNN	*Feedforward Neural Fuzzy System*
FIR	*Finite Impulse Response*
FWNN	*Fuzzy Wavelet Neural Network*
FWNNARX	*Fuzzy Wavelet Neural Network ARX*
FWNNFIR	*Fuzzy Wavelet Neural Network FIR*
FWNNPC	*Fuzzy Wavelet Neural Network Predictive Control*
HRNFN	*Hierarchical Recurrent Neural Fuzzy Networks*
IID	Independente e Identicamente Distribuídos
MLP	*Multilayer Perceptron*
MRA	*Multiresolution Analysis*
MSE	*Mean Squared Error*
NARMAX	*Non-linear Auto-Regressive, Moving-Average with eXogenous input model*
NNARMAX	*Neural Network ARMAX*
NNARX	*Neural Network ARX*

NNFIR	*Neural Network FIR*
NNOE	*Neural Network OE*
NNSSIF	*Neural Network SSIF*
OE	*Output Error*
PRBS	*Pseudo Random Binary Signal*
PRS	*Pseudo Random Signal*
PSO	*Particle Swarm Optimization*
PSS	*Power System Stabilizers*
RLS	*Recursive Least Squares*
RMSE	*Root Mean Square Error*
RFNN	*Recurrent Fuzzy Neural Network*
RNA	Rede Neural Artificial
RSONFIN	*Recurrent Self-Organizing Neural Fuzzy Inference Network*
SSIF	*State Space Innovations Form*
SVM	*Support Vector Machine*
TRFN-S	*Trapezoidal Fuzzy Numbers*
UTIN	Unidade de Terapia Intensiva Neonatal
WNN	*Wavelet Neural Network*
z_i	*i*-ésima saída da rede neural
x_i	Entradas da rede neural
θ	Vetor de parâmetros ajustáveis da rede
fa_i	Função de ativação dos neurônios da *i*-ésima camada de saída
fa_j	Função de ativação do *j*-ésimo neurônio da camada escondida
W_{ij}	Peso sináptico ligando o *j*-ésimo neurônio da camada escondida a *i*-ésima saída da rede neural

w_{jl}	Peso sináptico ligando *l*-ésima entrada ao *j*-ésimo neurônio da camada escondida
$\psi(t)$	Função *wavelet* no domínio do tempo
$\hat{\psi}(f)$	Transformada de Fourier da *wavelet* $\psi(t)$
C_{ψ}	Condição de Admissibilidade
t_i	Parâmetro de translação de cada *wavelet*
d_i	Parâmetro de abertura de cada *wavelet*
E	Erro entre a saída estimada da rede e a saída desejada
η	Coeficiente de aprendizagem
J	Valor da função custo
A_{qr}	*r*-ésima função de pertinência relativa ao *q*-ésimo sinal de entrada
a_{qr}	Abertura da função de pertinência
b_{qr}	Inclinação da função de pertinência
c_{qr}	Centro da função de pertinência
μ_m	Valor da *m*-ésima regra *fuzzy*
$\bar{\mu}_i$	*i*-ésima regra normalizada
g_j	Saída do *j*-ésimo neurônio da camada consequente
$\hat{y}$	Saída estimada do modelo
E_{av}	Média da energia do erro entre as saídas desejada e real
N	Número total de dados de treinamento da rede
$u(k)$	Sinal de excitação no *k*-ésimo instante de tempo
$e(k)$	Erro entre valor desejado para a saída da rede e o valor real no *k*-ésimo instante de tempo
$\varphi(t)$	Vetor contendo os valores de $y(k)$ e $u(k)$, e seus regressores
θ'	Vetor que contêm os parâmetros que realizam a ponderação dos sinais $y(k)$ e $u(k)$, e seus regressores
D	Atraso de transporte

α	Ordem da saída da rede neural
β	Ordem da entrada da rede neural
$F[x_i]$	Função matemática de x_i
N	Número de entradas da rede neural
M	Número de regras fuzzy
T_S	Valor da temperatura da incubadora fornecida pelo sensor SHT11
U_S	Valor da umidade da incubadora fornecida pelo sensor SHT11
U_A	Umidade da incubadora no ponto A
T_A	Temperatura da incubadora no ponto A
T_B	Temperatura da incubadora no ponto B
T_C	Temperatura da incubadora no ponto C
T_D	Temperatura da incubadora no ponto D
T_E	Temperatura da incubadora no ponto E
$\hat{U}_A$	Estimativa da umidade no ponto A
$\hat{T}_i$	Estimativas das temperaturas no pontos A, B, C, D e E
N_P	Número total de parâmetros ajustados na FWNN proposta
N_{in}	Número de entradas da rede
N_{fp}	Número de funções de pertinência por entrada
N_{pfp}	Número de parâmetros presentes em cada função de pertinência
N_r	Número de regras *fuzzy*
N_{pw}	Quantidade de parâmetros de cada *wavelet*

Capítulo 1

Introdução

Um dos grandes desafios na história da ciência tem sido obter sistemas análogos aos processos e fenômenos observados no universo. Por sistema análogo entende-se um sistema capaz de reproduzir algumas características de um fenômeno observado, assim como uma maquete reproduz as escalas, proporções, cores, etc. de uma construção real. Esses sistemas análogos são denominados de modelos, e podem ser obtidos pelo uso de técnicas de identificação (Aguirre *et al.*, 1998).

Com a crescente evolução e disponibilidade de computadores, o uso de modelos computacionais tem aumentado em praticamente todas as áreas do conhecimento humano, de modo que nos dias atuais o processo de modelagem de sistemas reais é de grande importância em quase todas as áreas da ciência. Na Engenharia, por exemplo, os modelos matemáticos são aplicados na análise de processos existentes, assim como no desenvolvimento de projetos de novos processos.

Normalmente, técnicas avançadas de projeto de controladores, otimização e supervisão de sistemas são baseadas em modelos dos processos. A qualidade do modelo tipicamente determina a qualidade da solução final de um determinado problema, dando assim à modelagem uma relevante importância no desenvolvimento do projeto. Como consequência, surge uma demanda por técnicas mais sofisticadas de modelagem (Nelles, 2001).

Na última década, tem-se verificado uma tendência geral que tornará o uso de técnicas de modelagem de sistemas desejável e até mesmo necessária em praticamente todas as áreas do conhecimento humano. Essa tendência se deve à inegável capacidade que hoje se tem de coletar

dados com informação sobre a dinâmica do sistema que está sendo observado (Guiraud & Poignet, 2009; Brockett & Glaser , 2001).

De forma geral, é possível agrupar as técnicas de modelagem em duas grandes categorias. A primeira categoria é baseada na modelagem pela física do processo, também denominada de relações fenomenológicas. Já a segunda é baseada na identificação a partir de dados. Esta categoria é normalmente conhecida por métodos de identificação de sistemas (Ljung, 1987). As diferenças entre as duas abordagens são muitas, bem como as diferenças no tipo de modelos obtidos em cada caso. Entretanto, o que provavelmente mais diferencia essas duas abordagens é a quantidade de conhecimento sobre o processo real utilizado na obtenção dos modelos. Na modelagem baseada nas leis físicas do processo, o modelo é desenvolvido a partir de toda informação disponível sobre o processo. Essa forma de identificação é chamada de identificação caixa-branca. Por outro lado, métodos de identificação de sistemas baseados em dados normalmente pressupõem pouco ou nenhum conhecimento prévio do sistema, justificando o nome "identificação caixa-preta" (Sjöberg *et al*., 1996).

A motivação para o estudo de técnicas de identificação de sistemas surge do fato de que frequentemente as equações envolvidas no funcionamento de um determinado sistema não são conhecidas ou elas são conhecidas, mas seria impraticável, por limitações de tempo e de recursos, levantar tais equações e estimar seus respectivos parâmetros.

Ultimamente, tem havido algum interesse em desenvolver métodos que permitam incorporar alguma informação que se tenha sobre o sistema durante a sua identificação (Lindskog & Ljung, 1994). Procedimentos com esta característica são denominados métodos de "identificação caixa-cinza" e são especialmente interessantes porque não exigem do usuário um profundo conhecimento "a priori" do processo, mas permitem a utilização de conhecimento prévio. Isso normalmente resulta na obtenção de melhores modelos.

Do ponto de vista de implementação, o problema de identificação de sistemas pode ser dividido em quatro etapas principais: (i) obtenção de dados de experimentação do sistema que se deseja modelar; (ii) escolha da estrutura do modelo que será utilizada para representar o sistema; (iii) estimação dos parâmetros do modelo; (iv) validação do modelo obtido (Ljung, 1987; Pinto, 2003).

No processo de identificação, deve-se levar em conta se o modelo a ser identificado é linear ou não linear. Nos sistemas lineares, um dos métodos mais utilizados é o método dos

mínimos quadrados. Existe o método de mínimos quadrados clássico, em que a identificação é feita em batelada, e o recursivo, em que a identificação é feita recursivamente, podendo ser utilizado *on line* (Coelho & Coelho, 2004).

A identificação de sistemas lineares é uma área da ciência muito importante uma vez que muitos dos sistemas podem ser descritos de forma aproximada por modelos lineares em um dado ponto de operação. Porém, pode-se observar que com o crescente interesse pela utilização de representações não lineares para caracterizar sistemas e fenômenos reais, tem ocorrido uma mudança no tipo de modelos utilizados. Na medida em que as representações lineares são substituídas em algumas aplicações por seus correspondentes não lineares mais acurados, torna-se possível analisar e reproduzir certos fenômenos e comportamentos dinâmicos mais complexos (Aguirre *et al.*, 1998).

Este crescente interesse pela utilização de representações não lineares se deve ao fato de que os sistemas dinâmicos encontrados na prática são, em última análise, não lineares. É bem verdade que em alguns casos aproximações lineares são suficientes para aplicações práticas. Entretanto, em uma série de exemplos, modelos lineares não serão satisfatórios e representações não lineares deverão ser usadas. Isso se deve ao fato de que uma grande quantidade dos sistemas reais são complexos e precisam ter suas não linearidades consideradas para que o modelo identificado seja mais exato para amplas faixas de operações. Dessa forma, a identificação de sistemas não lineares por meio de estruturas não lineares é uma área de pesquisa que vem cada vez mais ganhando importância (Yilmaz & Oysal, 2010; Banakar & Azeem, 2006; Kara & Eker, 2003).

Em face à escolha de que tipo de modelo utilizar, se linear ou não linear, é importante considerar: ao contrário do que possa parecer, melhorar a exatidão dos modelos não é a principal motivação para se usar modelos não lineares. De fato, há razões mais fortes para, em uma dada aplicação, optar por modelos não lineares como, por exemplo, o fato destes produzirem certos regimes dinâmicos que os modelos lineares não conseguem representar (Aguirre, 2007).

Há uma grande variedade de representações não lineares que, em principio, podem ser utilizadas na identificação de sistemas, tais como: série de *Volterra*, modelos de *Hammerstein* e de *Wiener*, modelos polinomiais contínuos, funções radiais de base, redes neurais artificiais, sistemas *fuzzy*, sistemas híbridos *neuro-fuzzy*, etc (Nelles, 2001; Efe, 2009; Narendra & Parthasarathy, 1990).

No caso das redes neurais artificiais (RNAs), as características que contribuem para que estas sejam uma das representações mais atrativas para essa aplicação é o fato de serem estruturas com capacidade de generalização e aprendizado (Jahangiri, 2010). Ao longo do tempo, as redes neurais artificiais vêm se consolidando como uma poderosa ferramenta a ser utilizada no processo de identificação de sistemas dinâmicos.

De acordo com Haykin (2001), as RNAs são estruturas paralelas, maciçamente distribuídas, constituídas por unidades simples de processamento, conhecidas como neurônios, que possuem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso. As redes do tipo Multilayer Perceptron (MLP) é uma das arquiteturas mais clássicas de redes neurais. Entre as propriedades citadas por Haykin (2001) que justificam a utilização de tais estruturas, destacam-se a sua característica intrínseca de não linearidade, as capacidades de generalização e adaptabilidade, a tolerância a falhas e a sua capacidade de aproximação universal, ou seja, uma rede pode aproximar qualquer função contínua para qualquer precisão desejada (Cybenko, 1989; Hornik, 1991; Hornik *et al.*, 1989). As redes neurais artificiais, através dessas características, conseguem obter representações dinâmicas adequadas dos sistemas a serem identificados com relativa simplicidade (Linhares, 2010).

A rede MLP é um tipo de rede neural *feedforward* (em que cada neurônio de uma camada recebe como entrada apenas as saídas de neurônios da camada imediatamente anterior) que possui a habilidade de aproximação universal (Haykin, 2001). Esta rede é inspirada em neurônios que possuem funções de ativação localmente sintonizadas ou neurônios seletivos, que respondem para determinadas faixas de sinais de entrada. Em princípio, as redes MLP podem ter multicamadas e terem funções de ativação na saída não lineares. Contudo, em problemas de identificação, essas redes têm tradicionalmente sido utilizadas com apenas três camadas (uma camada de nós de entrada e duas camadas de neurônios, uma intermediária e uma de saída).

Em termos de identificação, diversos trabalhos fazem o uso de redes MLP. Gorji & Menhaj (2008) utilizam uma rede MLP para modelar um sistema discreto não linear em espaço de estados. Efe (2009) realiza uma comparação entre três estruturas de identificação: ANFIS (*Adaptive Network Based Fuzzy Inference System*), MLP e SVM (*Support Vector Machine*). Esta tarefa foi realizada tendo como base a modelagem caixa-preta de um biorreator. Através dos resultados apresentados por este trabalho pode-se notar que as três estruturas apresentaram resultados satisfatórios. Entretanto, apesar do melhor desempenho obtido pela estrutura ANFIS, o

próprio autor reconhece que a MLP é uma estrutura de maior simplicidade e praticidade, apresentando também menor complexidade computacional.

Como pode ser visto, a modelagem de sistemas dinâmicos utilizando redes neurais MLP têm sido largamente estudada, gerando na literatura uma grande variedade de abordagens e estruturas derivadas aplicadas em várias áreas. Da mesma forma, a teoria *wavelet* (Chui, 1992; Meyer, 1993) tem sido extensivamente estudada e aplicada em diversas áreas da ciência e engenharia. Uma estrutura, inspirada pelas redes neurais e a teoria *wavelet*, doravante denominada de redes *Wavelets Neural Network* (WNN), tem-se apresentado como uma alternativa promissora às redes neurais tradicionais na modelagem de sistemas não lineares, classificação, predição e controle (Zhang & Benveniste, 1992; Billings & Wei, 2005; Lin *et al.,* 2005).

Este tipo de rede foi proposta inicialmente por (Zhang & Benveniste, 1992), podendo apresentar uma estrutura similar à rede MLP, com a diferença de apenas substituir as funções de ativação da camada escondida por funções *wavelets*. A integração das propriedades de localização das *wavelets* e as habilidades de aprendizado e generalização das redes neurais resultam em uma estrutura mais apropriada para ser utilizada na identificação de sistemas dinâmicos complexos do que as redes neurais tradicionais (Abiyev & Kaynak, 2008a).

Apesar de considerar que as WNNs foram popularizadas pelos trabalhos de Szu *et al*. (1992) e Zhang & Benveniste (1992), Billings & Wei (2005) afirmam que a origem das WNNs vem do trabalho de Daugman (1988), no qual *wavelets* são utilizadas na classificação e compressão de imagens. Song & Liu (2011) afirmam que essas redes são usadas principalmente na classificação de sinais. Entretanto, os pesquisadores têm aplicado as redes WNN com sucesso em diversas outras aplicações, tais como: aproximação de funções (Ling *et al.,* 2007), robótica (Yoo *et al.,* 2006), sistemas de energia (Huand & Huang, 2002), estimação não paramétrica (Zhang, 1997), identificação de sistemas não lineares (Billings & Wei, 2005), predição de séries temporais (Iyengar *et al.,* 2002), processamento de sinais (Erdol & Basbug, 1996), entre outras.

Esse sucesso da utilização de *wavelets* como funções de ativação em redes neurais pode ser comprovado pelo trabalho de Daubechies (1992), onde o autor mostra que as famílias de funções *wavelets* são aproximadores universais, o que fornece a base teórica para que estas venham a ser utilizadas no campo da aproximação de funções e de modelagem de processos. Para as funções *wavelets*, esta propriedade pode ser expressa da seguinte maneira: qualquer função

pode ser aproximada em qualquer nível desejado de exatidão por uma soma finita de *wavelets* (Oussar & Dreyfus, 2000).

Na análise de um sinal através de funções *wavelets*, o sinal original é representado por um conjunto de versões "deslocadas" e "escalonadas" de uma função *wavelet* mãe (*mother wavelet*) através da transformada *wavelet*. O termo "escalonadas" significa que a *wavelet* pode ser alongada ou comprimida através de seu parâmetro de dilatação, enquanto que o termo "deslocadas" significa que pode ser acrescentado um retardo a esta função por meio de seu parâmetro de translação. Esta flexibilidade permite que a análise do sinal possa ser realizada em frequências e resoluções diferentes (Rocha, 2008), utilizando o conceito de multiresolução.

Pode-se afirmar, portanto, que a idéia fundamental na analise de um sinal por meio dessas funções é que elas realizam essa analise de forma pontual e proporcional à escala do sinal, analisando cada ponto do sinal. Para isso, as *wavelets* utilizam funções de base curta, em altas frequências, e base longa, em baixas frequências. Isto permite, por exemplo, que uma descontinuidade no sinal possa ser isolada e analisada por funções de base muito curtas. Ao mesmo tempo, é possível obter-se uma análise frequencial detalhada através de funções de base muito longas (Lira, 2004). Desta forma, a metodologia de análise por *wavelets* é capaz de fornecer simultaneamente a representação de um sinal nos domínios do tempo e frequência.

A análise de um sinal por meio de funções *wavelets* constitui uma ferramenta que pode ser utilizada em um grande número de aplicações, como na análise e processamento de sinais, análise de imagens, visão computacional, etc (Araújo Júnior *et al*, 2013a). Ela foi desenvolvida como uma alternativa à análise de Fourier para solucionar o problema da resolução, pois esta última apresenta dificuldades em analisar detalhes locais, acarretando erros em sinais que contêm descontinuidades e variações bruscas (Lira, 2004).

Devido às propriedades das *wavelets* e das redes neurais, a ideia de combinar estas duas técnicas se apresenta quase que naturalmente (Tan *et al*., 2000). Segundo Song & Liu (2011), a rede WNN é a prova de que a teoria de análise *wavelet* combina perfeitamente com a teoria de redes neurais artificiais. As WNNs herdam as vantagens fornecidas por estas duas estruturas, incluindo as características de análise de sinais em diferentes frequências e resoluções fornecidas pelas *wavelets* e as características de adaptação, robustez, tolerância a falhas e generalização das redes neurais. Uma vez que as *wavelets* apresentam excelente desempenho na análise de sinais não estacionários e na modelagem de funções não lineares, as redes neurais aplicadas em

conjunto com funções *wavelets* oferecem taxas de convergência e desempenho melhores do que as tradicionais redes neurais (Jain *et al*., 2010).

Outra característica importante é a de como são tratados os parâmetros internos dos neurônios onde estão presentes as *wavelets*, ou seja, como são ajustados os parâmetros de dilatação e de translação das funções de ativação *wavelets* destes neurônios. Segundo Billings & Wei (2005), de acordo com o ajuste destes parâmetros, as redes WNN podem ser classificadas em WNNs de grade fixa e WNNs adaptativas. Nas redes WNNs de grade fixa, os parâmetros que descrevem as funções de ativação são fixos (pré-determinados) e apenas os pesos sinápticos precisam ser ajustados durante o treinamento da rede. Nas redes WNN adaptativas, os parâmetros de dilatação e translação também são parâmetros desconhecidos da rede, ou seja, precisam ser ajustados durante o processo de treinamento da rede.

Com relação à utilização das redes WNN, podem-se encontrar diversos trabalhos na literatura, voltados tanto para aplicações, como também para diferentes formas de treinamento dessas redes. Hassouneh *et al*. (2008) utilizam as redes neurais *wavelets* para identificar um servomecanismo não linear. Gao *et al*. (2007) projetam um sensor virtual aplicado em um processo de tratamento de esgoto. Para atingir este objetivo, é identificado um modelo não linear obtido a partir de uma WNN treinada pelo algoritmo do gradiente descendente. Oussar *et al*. (1998) propõem algoritmos alternativos para o treinamento de redes WNN utilizadas na identificação de sistemas dinâmicos não lineares. Além disso, o autor apresenta um mecanismo de inicialização dos parâmetros das *wavelets* que leva em conta a localização destas funções. Para a obtenção de resultados, são identificados alguns modelos simulados e um atuador hidráulico de um braço robótico real.

Moura (2003) utiliza as redes WNN para modelar uma coluna de destilação responsável por separar hidrocarbonetos em um processo contínuo. Cordova & Yu (2012) aplicam as redes neurais *wavelets* na identificação de sistemas discretos não lineares na forma de modelos NARMAX (*Non-linear Auto-Regressive, Moving-Average with eXogenous input model*). Os autores propõem um método para determinar a estrutura das redes *wavelets* e diferentes algoritmos para definir os valores dos parâmetros dessas funções.

Da mesma forma que nas tradicionais redes neurais, para se treinar uma rede WNN, do tipo adaptativa, um dos algoritmos mais utilizados é o algoritmo *backpropagation* (Hong-yan *et al*., 2011; Li *et al*., 2010; Yang & Deb., 2009). Assim, para treinar uma rede WNN utilizando este

algoritmo, os gradientes com respeito a todos os parâmetros desconhecidos devem ser expressos explicitamente. Entretanto, o cálculo destes gradientes pode ser não trivial em alguns casos, especialmente quando se deseja identificar modelos de altas dimensões. Além disto, muitos dos algoritmos baseados em gradiente são bastante sensíveis às condições iniciais. Ou seja, a inicialização dos parâmetros da rede neural *wavelet* é extremamente importante para se obter bons resultados e uma rápida convergência (Oussar & Dreyfus, 2000; Billings & Wei, 2005).

Uma evolução das redes WNN consiste em combinar a estrutura *neuro-fuzzy* ANFIS (*Adaptive Network Based Fuzzy Inference System*) com *wavelets*, gerando-se a estrutura *Fuzzy Wavelet Neural Network*- FWNN (Yılmaz & Oysal, 2009). Esta estrutura é análoga à tradicional estrutura ANFIS (Jang, 1993), com a diferença de que os consequentes das regras *fuzzy* passam a ser redes WNN. Neste caso, em vez das tradicionais funções Takagi-Sugeno de saída, são as redes WNN que serão interpoladas pelas regras *fuzzy*. Assim, cada nó presente na camada consequente consiste de uma rede WNN. E a soma de todos esses nós resultam na saída da rede FWNN. Apesar dessa forma de combinar essas estruturas ser a mais tradicional, podem-se encontrar trabalhos em que a estrutura FWNN utiliza as *wavelets* como funções de pertinência, e mantém os tradicionais polinômios na camada consequente.

Independente das duas possíveis formas de combinação, diversos pesquisadores têm feito o uso das redes FWNN para desenvolver um sistema com capacidade de aprendizado rápido que possa descrever sistemas não lineares que são caracterizados por incertezas. Essas redes são utilizadas, de forma geral, nas mesmas áreas de aplicação das redes WNN, como por exemplo, previsão de séries temporais e identificação de sistemas dinâmicos não lineares (Abiyev, 2006; Amina & Kodogiannis, 2011; Araújo Júnior & Araújo, 2011). O que diferencia as pesquisas utilizando essa estrutura, em geral, é a sua aplicabilidade e a forma como essas técnicas são combinadas e estruturadas.

A estrutura FWNN apresentada por Zhang *et al*. (2010), utiliza *wavelets* como funções de pertinência cujas saídas formam um conjunto de regras que interpolam modelos locais representados por funções Takagi-Sugeno de saída. Da mesma forma, o trabalho apresentado em Song & Shi (2011), utiliza funções *wavelet* como funções de pertinência de uma rede FWNN. No entanto, a fim de melhorar a capacidade de generalização do modelo, os parâmetros presentes nas partes premissa e consequente são ajustados separadamente, utilizando, respectivamente, o

algoritmo Levenberg-Marquardt e o método dos mínimos quadrados recursivos (*Recursive Least Squares* - RLS).

Ainda utilizando *wavelets* como funções de pertinência, tem-se o trabalho de Guo *et al.* (2005a), em que os autores propõem uma FWNN para o diagnóstico de falhas em maquinas elétricas. As *wavelets*, do tipo Morlet, têm suas formas ajustadas por um algoritmo *on-line*, de maneira que o sistema adquira uma melhor capacidade de aprendizagem. Além disso, esse algoritmo é aplicado para construir automaticamente o sistema, ou seja, determinar o número de regras, número de funções de pertinência, etc. A vantagem desse algoritmo é que ele converge mais rapidamente e as regras *fuzzy* são obtidas com maior precisão. Os resultados apresentados comprovam que esse sistema possui rápida aprendizagem e um boa capacidade de diagnóstico de falhas.

Os mesmos autores do trabalho citado acima (Guo *et al.,* 2005b) propuseram um outro trabalho aplicando FWNN no diagnostico de falhas. Nele, eles citam a rede FWNN como sendo uma nova direção de pesquisa de monitoramento inteligente para diagnóstico de falhas. As *wavelets* têm suas formas ajustadas também *on-line*, de tal modo que a rede tem melhor capacidade de treinamento e de adaptação. Os resultados das simulações mostram que a FWNN tem como vantagens uma capacidade de rápida aprendizagem e alta precisão no diagnóstico. O algoritmo de treinamento é baseado no gradiente descendente, com a taxa de aprendizagem adaptativa, de modo a garantir a rapidez no processo de aprendizagem.

No entanto, a maioria das estruturas FWNN encontradas na literatura utiliza funções *wavelets* na parte consequente das regras *fuzzy*. Entre os vários trabalhos, alguns deles podem ser descritos em detalhe como se segue. A FWNN proposta em Abiyev & Kaynak (2008b), utiliza um somatório de versões dilatadas e deslocadas de funções *wavelets* presentes nas redes WNN, situadas na parte consequente das regras *fuzzy*, para identificação e controle de um sistema dinâmico. A FWNN apresentada em Lu (2011) é bastante semelhante, mas a normalização das regras *fuzzy* é realizada antes da camada consequente, e não no final da rede.

A FWNN, com base na análise multiresolução (*Multiresolution Analysis* - MRA) da transformada *wavelet* e nos conceitos *fuzzy*, é utilizada para aproximar funções não lineares em Ho *et al.* (2001). Nesta estrutura, cada consequente é representado por uma sub-WNN. Assim, a rede FWNN pode ser vista como um processo de combinação de várias redes WNN, interpoladas por um conjunto de regras *fuzzy*.

Uma FWNN para o controle de sistemas não lineares foi desenvolvido em Zekri *et al.* (2008). Considerando-se as vantagens apresentadas por essa rede, um modelo FWNN foi escolhido para representar a dinâmica de um sistema controlado por uma outra rede FWNN, projetada com base em um controlador preditivo (FWNNPC).

Um algoritmo de aprendizado híbrido aplicado no treinamento de redes FWNN é apresentado em Davanipoor *et al.* (2012). O algoritmo fornece os parâmetros iniciais pela técnica de agrupamento *K-Means* e, em seguida, atualiza-os por meio de uma combinação do *backpropagation* com o método dos mínimos quadrados recursivos. Os parâmetros são atualizados na direção da descida mais íngreme do gradiente do erro, com uma taxa de aprendizagem adaptativa que é diferente para cada época de treinamento e apenas depende do sinal de erro do gradiente. Alguns resultados de simulação demonstram que a velocidade de convergência é superior à conseguida com outros métodos de treinamento utilizados em redes FWNN.

Um novo método para projetar Sistemas Estabilizadores de Potência (*Power System Stabilizers* - PSS) usando FWNN é apresentado em Ganjefar & Tofighi (2013). No método proposto, um Algoritmo Genético (AG) é utilizado para ajustar adequadamente todos os parâmetros de uma FWNN. Os resultados apresentados demonstram as capacidades da FWNN proposta, estabilizando o amortecimento das oscilações da energia de um sistema elétrico.

O algoritmo *Particle Swarm Optimization* (PSO) é utilizado em Esmaili *et al.* (2013), para ajustar os parâmetros de dilatação, de translação, pesos sinápticos e os parâmetros das funções de pertinência, a fim de melhorar a capacidade e a precisão do modelo identificado por uma rede FWNN. Nos dois casos de identificação de sistemas dinâmicos simulados, utilizados como estudos de caso, o desempenho da rede ajustada pelo PSO é superior aos métodos de treinamento existentes, tais como AG e o algoritmo *Broyden-Fletcher-Goldfarb-Shanno* (BFGS).

No presente trabalho, é proposta uma estrutura FWNN modificada que é semelhante as apresentadas por Abiyev & Kaynak (2008b), Lu (2011), e, principalmente, Yilmaz & Oysal (2010). A principal diferença entre as redes apresentadas em Yilmaz & Oysal (2010) e Abiyev & Kaynak (2008b) está relacionada com a geração das regras *fuzzy*. O número de regras em Abiyev & Kaynak (2008b) é definido de acordo com o número de funções de pertinência atribuídos às entradas da FWNN, enquanto que em Yilmaz & Oysal (2010), cada combinação possível das funções de pertinência representa uma regra. Além disso, em Yilmaz & Oysal (2010) as saídas

dos neurônios da camada consequente são calculadas de uma forma diferente, uma vez que cada função *wavelet* presente em cada um desses neurônios é multiplicada por um peso sináptico, enquanto que em Abiyev & Kaynak (2008b), a saída de cada neurônio é que é multiplicada por um peso.

A estrutura apresentada em Lu (2011) é muito semelhante à usada em Abiyev & Kaynak (2008b). No entanto, em Lu (2011) a etapa de normalização responsável pelo cálculo da contribuição de cada regra na interpolação dos modelos presentes na parte consequente é realizado antes desta camada. Ambos os passos de normalização e formação das regras *fuzzy* da FWNN proposta neste trabalho são realizados da mesma forma como em Yilmaz & Oysal (2010). O que diferencia a FWNN proposta das estruturas Yilmaz & Oysal (2010), Abiyev & Kaynak (2008b) e Lu (2011) está relacionado mais especificamente em como os consequentes são ponderados.

Como pode ser visto, apesar da diferença na forma de gerar as regras, todas as redes FWNN citadas têm a característica de possuir como consequente redes do tipo WNN. Dessa forma, este trabalho propõe uma rede FWNN alternativa, em que os consequentes são formados apenas por funções *wavelets*. Essas funções serão ponderadas apenas pelos pesos de ativação das regras *fuzzy*, de modo que os modelos locais da estrutura proposta são unicamente representados por um conjunto dessas funções. O presente trabalho irá investigar os possíveis ganhos com essa modificação, tais como a possibilidade de uma diminuição no número de parâmetros a serem ajustados no treinamento da rede, reduzindo a complexidade de cálculo e do algoritmo, com a expectativa de que não exista perda de desempenho significativa da nova rede FWNN quando utilizada na identificação de sistemas dinâmicos não lineares.

O algoritmo utilizado para realizar o treinamento da rede, a principio, será o *backpropagation*. O treinamento fará o ajuste de todos os parâmetros das funções de pertinência e das *wavelets*. O tipo de *wavelet* a ser utilizado será a *Mexican Hat*, pelo fato de ser o tipo de função *wavelet* mais encontrada nos principais trabalhos a respeito de FWNN, e pelo fato de essa função ser continuamente diferenciável, possibilitando o uso do algoritmo de treinamento citado.

A FWNN proposta terá seu desempenho avaliado na identificação da dinâmica de quatro estudos de caso. Em todos eles, uma comparação com outras redes FWNN presentes na literatura será realizada. Também serão utilizadas nessa comparação, redes clássicas, como as redes ANFIS, WNN e MLP. Portanto, os resultados obtidos na identificação de sistemas dinâmicos

complementam a fundamentação teórica que descreve a evolução da rede FWNN, desde a estrutura mais tradicional, que é a rede MLP, até a nova rede FWNN proposta.

Os dois primeiros estudos de caso consistem em duas plantas simuladas, tradicionalmente encontradas na literatura em estudos envolvendo redes FWNN, pelo fato de serem não lineares e também, por servirem como *Benchmarks*. Os dois últimos estudos de caso consistem em sistemas não lineares e reais, que são um sistema de tanque de multisseções e uma incubadora neonatal.

O tanque se caracteriza por possuir múltiplas seções e ser um sistema não linear com ampla faixa de operação. Esse sistema didático tem como objetivo principal o estudo de sistemas de identificação e controle (Fonseca, 2012), através da manipulação do nível de seu tanque. Portanto, a rede FWNN realizará a identificação de sua dinâmica, e a mesma será avaliada a partir de diferentes configurações, alterando-se o número de funções de pertinência e o número de regras.

Por fim, o último estudo de caso consiste em uma incubadora neonatal. A motivação para o uso desse equipamento como estudo de caso se deve ao fato de que ele é um sistema não linear, também com ampla faixa de operação, e, além disso, tem-se a possibilidade de uma aplicação prática não acadêmica da rede proposta. Isso ocorre pelo fato de que esse equipamento deve prover um ambiente seguro para recém nascidos, e por isso, deve apresentar valores de temperatura e umidade em seu interior de acordo com valores especificados em uma norma de segurança. Para verificar se a temperatura e a umidade em seu interior estão dentro de uma faixa adequada, a rede FWNN será utilizada para inferir os valores dessas grandezas.

Organização do trabalho

No próximo capítulo, são resumidamente apresentados alguns conceitos teóricos que dão base às redes FWNN. Inicialmente, é feita uma breve descrição das redes neurais MLP. Em seguida, é feito um breve resumo a respeito da teoria *wavelet*, mostrando suas principais características, de modo a ajudar no entendimento da união dessas duas técnicas. Por fim, são mostrados os principais conceitos das redes *neuro-fuzzy* ANFIS, e como se dá a sua combinação

com a rede WNN, gerando-se a rede FWNN, a qual também tem seus principais conceitos explicados.

No Capítulo 3, é apresentado de forma detalhada a estrutura da rede FWNN proposta, procurando-se evidenciar as principais diferenças em relação as redes FWNN já presentes na literatura, e mostrando as vantagens da modificação realizada. Todos os aspectos do processo de treinamento da rede também foram apresentados, incluindo, por exemplo, o tipo de função de ativação e de *wavelet* utilizados neste trabalho.

Como a rede proposta será avaliada na identificação de sistemas dinâmicos, no capítulo 4 são listados alguns conceitos desta área do conhecimento, apresentando algumas da estruturas neurais de identificação propostas por Nørgaard *et al*. (2001). Por fim, no Capítulo 5, são apresentados os estudos de caso utilizados, de forma detalhada, e os resultados obtidos. Inicialmente a rede foi avaliada em dois estudos de caso simulados, e em seguida, para melhor consolidação dessa avaliação, a rede foi utilizada em dois estudos de caso reais. Finalmente, no Capítulo 6, são listadas as conclusões obtidas e as perspectivas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Em alguns casos, quando o sistema é não linear, complexo e a aplicação a ser desenvolvida exige um modelo que represente com fidelidade o comportamento deste sistema, as suas características não lineares devem ser consideradas neste modelo. Devido a este fato, o interesse científico e prático pela utilização de estruturas não lineares no processo de identificação de sistemas dinâmicos não lineares vem ganhando cada vez mais espaço. Além das estruturas tradicionais já citadas na introdução deste trabalho, ultimamente têm surgido novas estruturas na área de identificação, tais como as redes *Wavelet Neural Network, Adaptive Network Based Fuzzy Inference System* e *Fuzzy Wavelet Neural Network* (Zhang & Benveniste, 1992; Jang, 1993; Yılmaz & Oysal, 2009).

Baseado na teoria das Redes Neurais Artificiais (RNAs) e na teoria *wavelet*, a rede WNN vem sendo bastante utilizada em diversos tipos de aplicações (Cordova & Yu, 2012; Hassouneh *et al*., 2008). De modo geral, esta rede apresenta uma estrutura similar à apresentada, por exemplo, pelas redes MLP, diferenciando apenas pelo fato de que as funções de ativação normalmente utilizadas (gaussiana, sigmoide, tangente hiperbólica) são substituídas por funções *wavelets*. Assim, a integração das propriedades de localização das *wavelets* e as habilidades de aprendizado e generalização das redes neurais resultam em uma estrutura mais apropriada para ser utilizada na identificação de sistemas dinâmicos complexos do que as redes neurais tradicionais (Jahangiri, 2010).

Uma evolução das redes WNN são as redes *Fuzzy Wavelet Neural Network* (Yılmaz & Oysal, 2009). Essa estrutura é análoga à tradicional estrutura ANFIS (Jang, 1993), com a diferença de que os consequentes das regras *fuzzy* passam a ser redes WNN. Neste caso, em vez

dos tradicionais modelos polinomiais, um conjunto dessas redes é que será interpolado pelas regras *fuzzy*.

2.1 Redes Neurais

As RNAs são utilizadas em diversos tipos de aplicações (Nørgaard *et al*., 2001; Linhares, 2013), como por exemplo, a classificação de padrões, filtragem de sinais, análise de imagens, controle e identificação de sistemas dinâmicos (Linhares, 2010), sendo esta última a aplicação alvo deste trabalho. A justificativa se dá pelo fato de que muitos dos sistemas físicos reais são complexos, o que inviabiliza a utilização de modelos fenomenológicos.

Existem várias arquiteturas de redes neurais, tais como redes MLP, funções de base radial, redes de Kohonen, etc. Um dos principais tipos são as redes MLP (Figura 2.1). A arquitetura básica dessa rede possui os seus neurônios dispostos em camadas, em que cada neurônio de uma camada recebe como entrada apenas as saídas dos neurônios da camada imediatamente anterior ou das entradas da rede (Nørgaard *et al*., 2001).

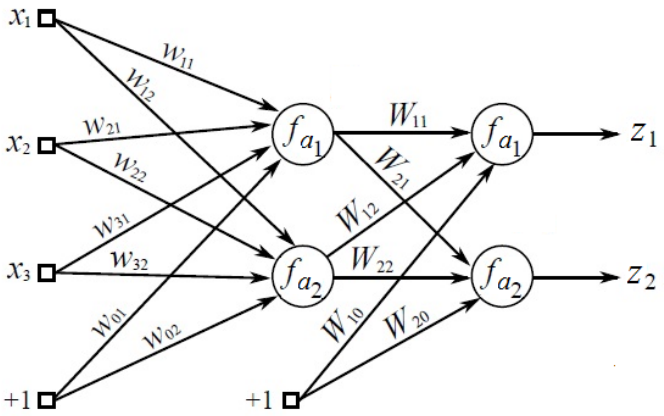


Figura 2.1: Rede MLP totalmente conectada de três camadas.

Uma rede neural tem, pelo menos, uma camada de nós de entrada e uma camada composta pelos neurônios de saída. No entanto, para a rede aprender problemas de complexidade mínima, ela deve ter pelo menos uma camada intermédia de neurônios, denominada camada escondida. Cybenko (1989) demonstrou que qualquer função contínua pode ser aproximada por

uma RNA com uma única camada de neurônios ocultos formados por funções de ativação sigmóide e uma camada de saída formada por neurônios lineares. Portanto, conclui-se que não há necessidade de utilizar mais do que uma camada oculta. No entanto, a utilização de arquiteturas mais sofisticadas pode ser mais apropriada em casos de mapeamento de maior complexidade, como por exemplo, em problemas de classificação de padrões (Parekh *et al.*, 2000).

No caso da rede MLP mostrada na Figura 2.1, pode-se perceber que a mesma é totalmente conectada, possuindo três camadas (camada de entrada, uma única camada intermediária de neurônios não lineares e uma camada de neurônios lineares de saída). A última camada é conhecida como camada de saída, pois fornece as saídas da rede neural. A segunda camada é conhecida como camada oculta, intermediária ou escondida, por estar, de certa forma, escondida entre as entradas externas da rede (x_1, x_2, x_3) e a camada de saída. Além disso, todos os nós, em todas as camadas estão ligados a todos os nós das camadas vizinhas através de conexões que têm os parâmetros ajustáveis, pesos sinápticos, que são responsáveis pela capacidade de armazenamento de informação de uma RNA.

As conexões de pesos sinápticos entre as camadas da MLP são ajustadas através do processo chamado de aprendizagem ou treinamento, que normalmente é feito de modo supervisionado. Neste processo, os pesos sinápticos, que são parâmetros livres, devem ser adaptados com base em um conjunto de observações de pares de entrada-saída e uma função custo. Dessa forma, é dado um conjunto de exemplos de entrada e saída, e o objetivo é o de minimizar o valor da função custo, tal que a saída da rede se aproxime o máximo possível da saída desejada (Haykin, 2001).

A Equação 2.1 expressa matematicamente o funcionamento de uma rede neural MLP de três camadas, onde z_i é a i-ésima saída da rede (Nørgaard *et al*., 2001). Facilmente, a idéia contida nesta equação pode ser expandida para redes MLPs com maiores números de camadas.

$$z_i(x,\theta) = fa_i\left[\sum_{j=1}^{n_2} W_{ij}\, fa_j\left(\sum_{l=1}^{n_1} w_{jl}x_l + w_{j0}\right) + W_{i0}\right] \tag{2.1}$$

O vetor de parâmetros θ contém todos os parâmetros ajustáveis da rede: pesos sinápticos e *bias* ($w_{j,l}$,$W_{i,j}$). Neste caso, o *bias* pode ser interpretado como um peso sendo aplicado a uma entrada fixa de valor unitário. O número de neurônios da camada escondida e o número de

entradas da rede são, respectivamente, n_2 e n_1, enquanto fa_i e fa_j são as funções de ativação dos neurônios das camadas de saída e escondida. As funções de ativação mais comumente utilizadas nas redes *perceptron* de múltiplas camadas são a linear, a sigmóide ou logística, e a tangente hiperbólica. Essas funções são dadas pelas Equações 2.2, 2.3 e 2.4, respectivamente.

$$fa(x) = kx \tag{2.2}$$

$$fa(x) = \frac{1}{1 + e^{-ax}}, \quad a > 0 \tag{2.3}$$

$$fa(x) = \tanh(x) = \frac{1 - e^{-\gamma x}}{1 + e^{-\gamma x}}, \quad \gamma > 0 \tag{2.4}$$

2.2 Funções *Wavelets*

As funções *wavelets* têm despertado interesse em várias áreas, e o rápido avanço de sua teoria se deve, basicamente, à sua origem interdisciplinar, que tem despertado o interesse de pesquisadores de diferentes áreas do conhecimento, e à forma simples e versátil com que certos conceitos são abordados (Cândido, 2008).

Wavelets são funções matemáticas que analisam os dados em diferentes componentes de frequências, e estudam cada componente com uma resolução correspondente à sua escala. Essas funções foram desenvolvidas independentemente do campo de aplicação e atualmente vêm sendo utilizadas em diversas áreas de conhecimento: visão computacional e humana, radar e sonar, computação gráfica, bancos de filtros, compressão de imagens, etc (Oliveira, 2003).

As *wavelets* utilizam a idéia de aproximação através do conceito de superposição de funções. Esta idéia tem origem no trabalho de Joseph Fourier, que no século XIX descobriu que seria possível utilizar senos e cossenos para representar outras funções. Porém, embora estas funções apresentem comportamento tipicamente suave e previsível, as *wavelets* diferem pelo fato de se apresentarem de forma irregular e assimétrica. É possível realizar também uma analogia

entre a análise de *wavelets* e a análise de Fourier. Na análise de Fourier, o sinal original é decomposto em ondas senoidais de várias frequências através da transformada de Fourier. De acordo com Veitch (2005), esta metodologia funciona satisfatoriamente quando o sinal em estudo é estacionário. Entretanto, quando a frequência varia ao longo do tempo, ou existem algumas singularidades, a análise de Fourier não será mais suficiente.

Na análise *wavelet*, o sinal original é representado por versões de uma função *wavelet* base, ou *wavelet* mãe, através da transformada *wavelet*. Neste caso, a *wavelet* base pode ser comprimida ou alongada através de seu parâmetro de dilatação e/ou sofrer um retardo, por meio de seu parâmetro de translação. Esta flexibilidade permite que a análise do sinal possa ser realizada em frequências e resoluções diferentes, o que caracteriza uma forma de análise conhecida por Análise de Multirresolução (Rocha, 2008).

Desta forma, a metodologia da análise *wavelet* é capaz de fornecer simultaneamente informações de tempo e frequência de um sinal, obtendo a representação da série nos domínios de tempo-frequência. Esta característica distingue as *wavelets* de outros métodos baseados em séries ortogonais, tais como nas usuais análises de Fourier, em que a representação da série baseia-se somente em funções dependentes da frequência. Nestes casos, a análise de sinais desta série será prejudicada caso ocorram pequenas alterações no domínio da frequência, uma vez que estas acarretam em distorções no domínio do tempo.

As *wavelets* buscam formular funções matemáticas capazes de ampliar o intervalo de dados, de modo a separar os diferentes componentes de frequência existentes na série, possibilitando assim a análise de cada uma delas em sua escala correspondente. Tal propriedade permite que diferentes frequências sejam relevantes para distintos subintervalos do tempo, resultando em uma melhor adaptação temporal (Rocha, 2008). Sendo assim, é possível afirmar que a análise *wavelet* processa dados em diferentes escalas e resoluções, permitindo que sejam vistos tanto características globais quanto os detalhes locais de um sinal (Graps, 1995).

Em termos matemáticos, para que uma função seja considerada do tipo *wavelet*, ela tem de satisfazer a certas propriedades matemáticas (Graps, 1995). Essas propriedades são citadas a seguir:

a) A integral da função *wavelet* deve ser zero, isto é:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{2.5}$$

b) A integral do quadrado da função *wavelet* deve ser unitária:

$$\int_{-\infty}^{+\infty} \psi^2(t)dt = 1 \tag{2.6}$$

c) A energia da função *wavelet* deve ser finita:

$$\int_{-\infty}^{+\infty} |\psi(t)|^2 dt < \infty \tag{2.7}$$

d) Se $\hat{\psi}(f)$ é a transformada de Fourier da *wavelet* $\psi(t)$, a seguinte condição de admissibilidade deve ser satisfeita:

$$C_\psi = \int_0^\infty \frac{|\hat{\psi}(f)|^2}{f} df < \infty \tag{2.8}$$

2.3 *Wavelet Neural Network*

Devido às características citadas anteriormente, as funções *wavelets* vêm sendo utilizadas como funções de ativação nas redes neurais tradicionais, como por exemplo, a rede MLP, que passa a ser denominada de rede *Wavelet Neural Network* (WNN), Figura 2.2. A justificativa para a utilização das *wavelets* como função de ativação decorre do fato de elas possuírem habilidade de detectar de forma rápida e mais precisa as singularidades dos sinais a serem analisados (Pan & Xia, 2008), permitindo que a WNN tenha bom desempenho na identificação de sistemas dinâmicos não lineares (Jahangiri, 2010).

Segundo Li *et al*. (2010), essas redes neurais possuem melhor capacidade de aproximação, de tolerância a falhas e de previsão do que as redes neurais tradicionais. Além disso, também apresentam convergência de treinamento mais rápida. Outra motivação para o uso das redes WNN é o fato de que estas estruturas são aproximadores universais, capazes de realizar representações de funções com boa precisão. A comprovação deste fato tem origem nos trabalhos de Hornik *et al.* (1989) e Kreinovich *et al*. (1994).

Uma rede neural WNN geralmente consiste numa rede neural do tipo *feedforward* com, normalmente, uma única camada escondida, cujos neurônios da camada oculta apresentam funções de ativação obtidas de uma família ortonormal de *wavelets*. Estes neurônios são também conhecidos como *wavelons* (Veitch, 2005).

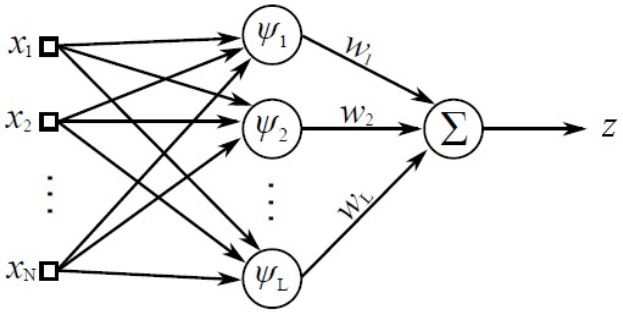


Figura 2.2: Estrutura básica de uma rede *wavelet*.

Nessa rede, as entradas são enviadas diretamente para a camada escondida. Ao ser calculado o valor da função *wavelet* para cada entrada, as saídas dos neurônios presentes nessa camada são multiplicadas por pesos sinápticos, e então, enviadas para a última camada, em que, por meio de um neurônio contendo uma função de ativação do tipo linear, a saída da rede será calculada. Assim, na rede WNN, como as funções de ativação são funções *wavelets*, a saída da rede é dada pela equação (Abiyev & Kaynak, 2008b):

$$z(x/\theta) = \sum_{k=1}^{L} w_k \psi_k \left(\frac{x - t_k}{d_k} \right) \tag{2.9}$$

Onde θ são os parâmetros adaptativos da rede: [w_k; d_k; t_k], em que w_k são os pesos sinápticos presentes na camada de saída, d_k é o parâmetro de dilatação e t_k é a translação da *wavelet*.

Como citado na introdução, essas redes podem ser do tipo grade fixa ou adaptativas. No caso de ser do tipo grade fixa, os parâmetros das *wavelets* são previamente determinados, por meio de algum algoritmo, e o processo de treinamento da rede realizará apenas o ajuste dos pesos sinápticos. No caso de a rede ser do tipo adaptativa, todos os parâmetros da rede são ajustados, incluindo os parâmetros das *wavelets*, mesmo que estes parâmetros sejam inicializados de forma ótima.

Existem diversas famílias de *wavelets*. No entanto, como a WNN normalmente utilizada é do tipo adaptativa, e seu treinamento ocorre por meio do algoritmo *backpropagation (*Abiyev & Kaynak, 2008b*)*, que envolve o uso de derivadas, deve-se escolher uma família que seja continuamente diferenciável. Dessa forma, no presente trabalho foi escolhida a *wavelet* da família *Mexican Hat*, mostrada na Figura 2.3, e que é representada pela Equação 2.10.

$$\psi_k = \left(1 - \left(\frac{x - t_k}{d_k} \right)^2 \right) \exp\left(-0.5 \left(\frac{x - t_k}{d_k} \right)^2 \right) \frac{1}{\sqrt{d_k}} \tag{2.10}$$

O parâmetro t_k indica que a função $\psi_k(x)$ foi transladada de uma distância equivalente a t_k, sendo então um parâmetro de translação. Já variando-se o parâmetro d_k é possível alongar ou comprimir a forma da função. Por isto o parâmetro d_k é conhecido como parâmetro de dilatação.

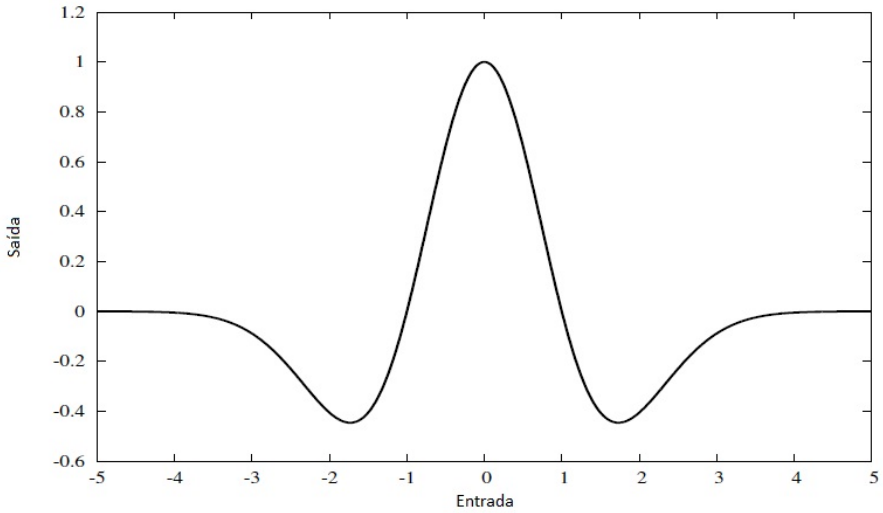


Figura 2.3: *Wavelet* da família *Mexican Hat*.

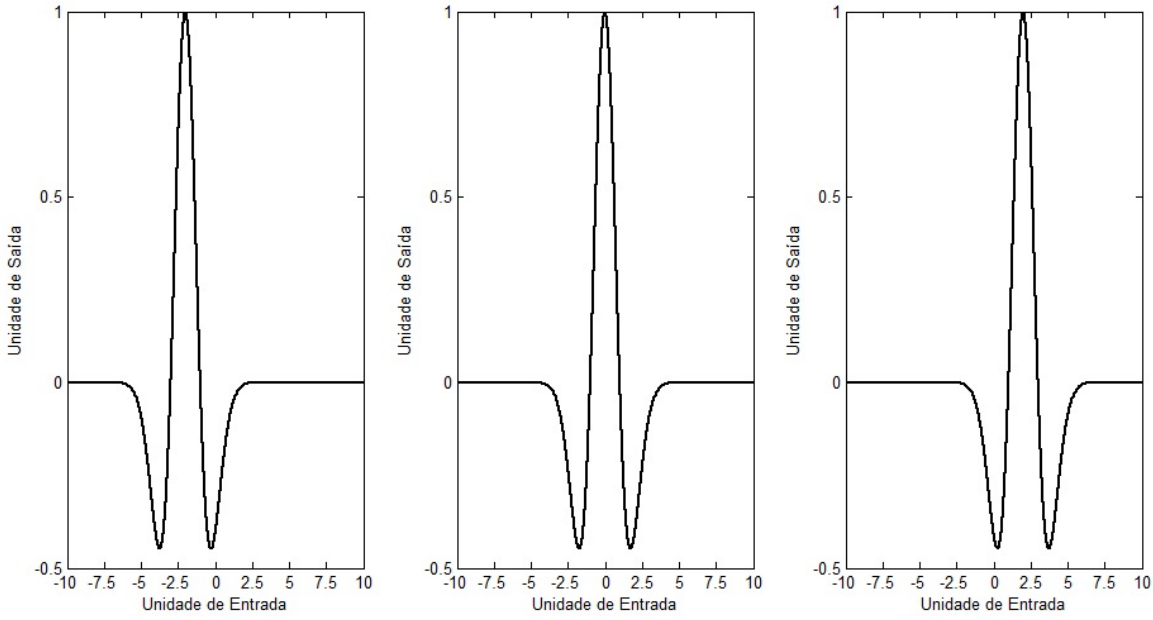


Figura 2.4: *Wavelets* da família *Mexican Hat* deslocadas.

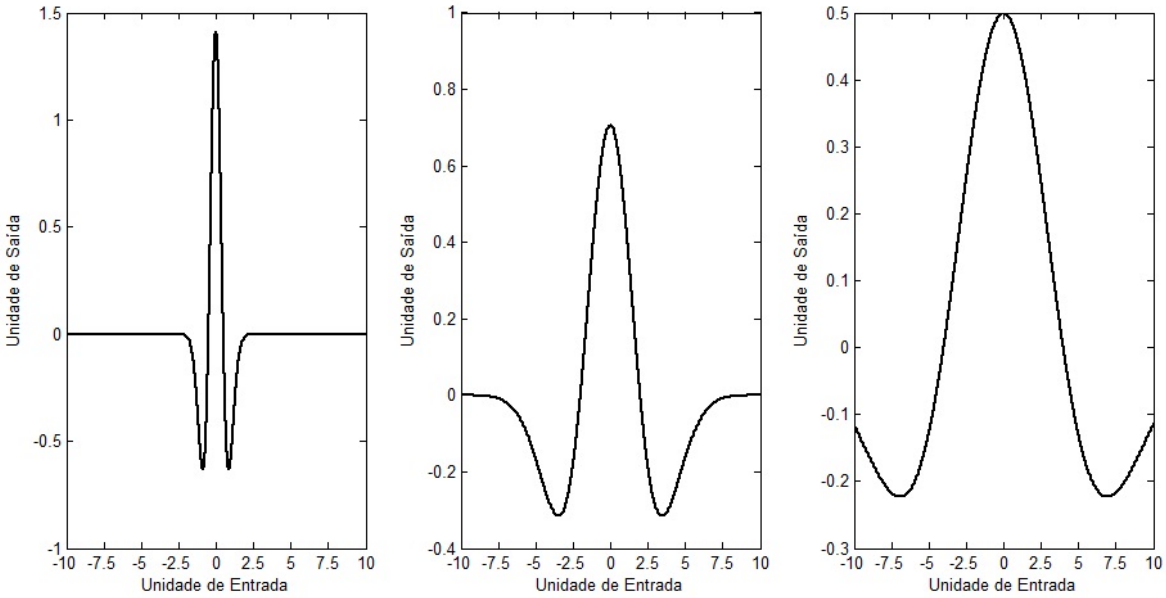


Figura 2.5: *Wavelets* da família *Mexican Hat* dilatadas.

O algoritmo *backpropagation* consiste em dois passos de computação. O primeiro passo é conhecido como passo para frente (*forward*), ou propagação, e o segundo é chamado de passo para trás (*backward*), ou retropropagação. No passo para frente, os pesos sinápticos se mantêm inalterados em toda a rede e os sinais de saída dos neurônios da rede são calculados individualmente. Já no passo para trás, para aplicação do algoritmo *backpropagation*, é necessária a utilização da derivada parcial da *wavelet* em relação aos seus parâmetros ajustáveis. Assim, o procedimento de treinamento da WNN envolve a utilização das derivadas do erro em relação aos seus parâmetros ajustáveis, que são os pesos da camada de saída, e os parâmetros de dilatação e translação das *wavelets*. Essas derivadas são apresentadas nas Equações 2.11 a 2.13.

$$\frac{\partial E}{\partial d_{i,k}} = -(z_d - z)w_k \left[\exp\left(-0.5\left(\frac{x_i - t_{i,k}}{d_{i,k}} \right)^2 \right) \left(3\left(\frac{x_i - t_{i,k}}{d_{i,k}} \right)^2 - \left(\frac{x_i - t_{i,k}}{d_{i,k}} \right)^4 \right) \right] \frac{1}{\sqrt{d_{i,k}^3}} \qquad (2.11)$$

$$\frac{\partial E}{\partial t_{i,k}} = -(z_d - z)w_k \left[\exp\left(-0.5\left(\frac{x_i - t_{i,k}}{d_{i,k}} \right)^2 \right) \left(3\left(\frac{x_i - t_{i,k}}{d_{i,k}} \right) - \left(\frac{x_i - t_{i,k}}{d_{i,k}} \right)^3 \right) \right] \frac{1}{\sqrt{d_{i,k}^3}} \tag{2.12}$$

$$\frac{\partial E}{\partial w_k} = -(z_d - z) \left[\left(1 - \left(\frac{x_i - t_{i,k}}{d_{i,k}} \right)^2 \right) \exp\left(-0.5\left(\frac{x_i - t_{i,k}}{d_{i,k}} \right)^2 \right) \frac{1}{\sqrt{d_{i,k}}} \right] \tag{2.13}$$

Onde z_d é a saída desejada para a rede, z corresponde a saída da rede, x_i é o vetor de variáveis de entradas, $t_{i,k}$ e $d_{i,k}$ são os parâmetros de translação e dilatação da *i*-ésima *wavelet* presente no *k*-ésimo neurônio da camada intermediária.

Com essas derivadas, pode-se partir para a atualização dos parâmetros, utilizando as Equações 2.14 a 2.16:

$$d_{i,k}(n+1) = d_{i,k}(n) - \eta \frac{\partial E}{\partial d_{i,k}} \tag{2.14}$$

$$t_{i,k}(n+1) = t_{i,k}(n) - \eta \frac{\partial E}{\partial t_{i,k}} \tag{2.15}$$

$$w_k(n+1) = w_k(n) - \eta \frac{\partial E}{\partial w_k} \tag{2.16}$$

Onde o n se refere a n-ésima época de treinamento. Assim, o algoritmo *backpropagation* deverá minimizar a função custo dada, por exemplo, pela Equação 2.17 até que J atinja um valor pré-determinado para um número máximo de iterações:

$$J = \min\left(\frac{1}{2} \sum_{j=1}^{N} \left(z_d - z \right)^2 \right) \tag{2.17}$$

2.4 *Fuzzy Wavelet Neural Network*

Uma evolução das redes WNN consiste em combinar a estrutura *neuro-fuzzy* ANFIS, baseada no sistema *fuzzy* Takagi-Sugeno, com as redes WNN, gerando-se a estrutura *Fuzzy Wavelet Neural Network* - FWNN (Davanipoor *et al.,* 2012). Nesse caso, em vez das tradicionais funções Sugeno de saída, são as redes WNN que serão interpoladas pelas regras *fuzzy*.

A estrutura ANFIS é uma técnica inteligente híbrida, mais especificamente uma técnica *neuro-fuzzy*, que infere conhecimentos utilizando uma estrutura com lógica *fuzzy*, fácil de tratar, e acrescenta à esta estrutura a possibilidade de aprendizagem inerente às redes neurais artificiais. Essa rede também pode ser vista como uma rede neural artificial de seis camadas, em que cada uma é responsável por uma operação que resultará em uma saída equivalente a encontrada por um sistema *fuzzy* do tipo Takagi-Sugeno (Jang *et al*., 1997; Jang & Sun, 1995).

O fluxo de dados no ANFIS pode ser descrito em camadas. Na camada de número um estão dispostas as entradas da rede. Na camada dois, os valores das entradas são operados por funções de pertinência que indicam o grau de compatibilidade de cada entrada nos conjuntos *fuzzy* de entrada. Na camada três, as regras *fuzzy* são geradas, e a normalização destas ocorre na camada quatro. Já na camada cinco, os valores das regras normalizadas são utilizadas para multiplicar polinômios Takagi-Sugeno, cujos valores são calculados em função das entradas. Por fim, as saídas desses neurônios da camada consequente são somadas na camada seis.

No ANFIS, o ajuste dos parâmetros ocorre em duas camadas: na camada número dois e na camada número cinco. O motivo é o fato de que os neurônios destas camadas possuem funções com parâmetros variáveis que podem ser ajustados para minimizar uma função custo, da mesma forma que as redes neurais artificiais, com aplicação de métodos de aprendizagem.

Como citado anteriormente, quando os modelos locais (neurônios com funções Sugeno) presentes na camada consequente da estrutura ANFIS são substituídos por redes WNN, tem-se a estrutura FWNN. Dessa forma, a rede consiste uma estrutura híbrida que combina a teoria *wavelet*, lógica *fuzzy* e redes neurais. As *wavelets* fornecem a capacidade de analisar sinais não estacionários e descobrir detalhes locais em alta frequência; a lógica *fuzzy* tem a capacidade de lidar com incertezas e reduzir a complexidade dos dados; já as redes neurais contribuem com as suas características de aprendizagem e generalização. Esta combinação permite projetar modelos

FWNN para descrever sistemas dinâmicos não lineares caracterizados por incertezas, utilizando um rápido mecanismo de aprendizagem (Abiyev & Kaynak, 2008b).

As Redes FWNN podem ser encontradas na literatura de diferentes maneiras, conforme citado no Capítulo 1. Algumas diferenciam das redes *neuro-fuzzy* ANFIS apenas pelo fato de que as funções de pertinência são *wavelets*. Nesse caso, a camada consequente se mantém exatamente igual a presente na rede ANFIS. Mas a maioria das redes FWNN utilizam essas funções na camada consequente, atuando como modelos locais. A Figura 2.6 mostra um diagrama esquemático desse tipo de estrutura FWNN, utilizada, como por exemplo, no trabalho de Yilmaz & Oysal (2010). A descrição de seu fluxo de dados pode ser feita camada por camada, tornando mais clara a compreensão de seus mecanismos de funcionamento.

Nessa estrutura, os consequentes presentes na camada cinco são formados por funções *wavelets*, que por serem ponderadas por pesos sinápticos, se caracterizam como sendo redes WNN, fazendo o papel tradicionalmente desempenhado por polinômios, como citado anteriormente.

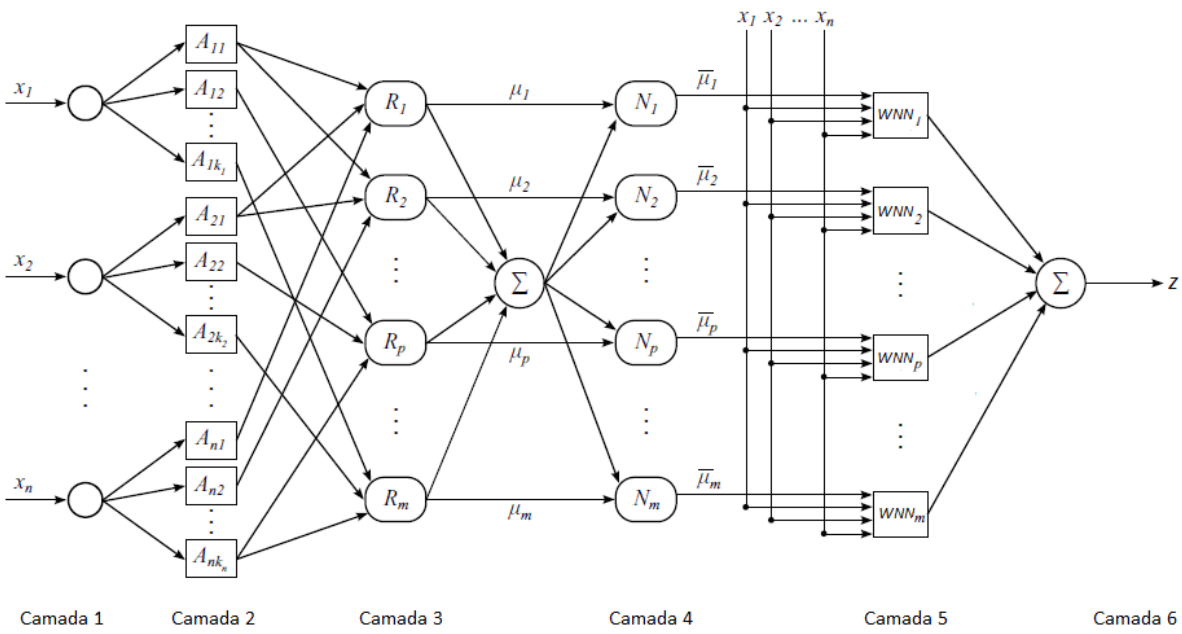


Figura 2.6: Estrutura da rede *Fuzzy Wavelet Neural Network*.

Essa ponderação ocorre tradicionalmente de duas formas. Na primeira, conforme realizado no trabalho de Abiyev & Kaynak (2008b), as *wavelets* presentes em cada neurônio da camada consequente são somadas, e em seguida, essa soma é multiplicada por um peso sináptico. Isso seria o equivalente a uma rede neural MLP com a existência de peso sináptico apenas no neurônio da camada de saída.

Na segunda forma de ponderação, cada uma das *wavelets* presentes em um neurônio consequente é multiplicada por um peso sináptico, conforme é apresentado no trabalho de Yilmaz & Oysal (2010). Portanto, nas estruturas FWNN, funções *wavelets* são interpoladas pelas regras *fuzzy* e ponderadas por pesos sinápticos. Assim, têm-se estruturas WNN fazendo o papel dos consequentes, em vez de modelos polinomiais Takagi-Sugeno.

Com relação ao treinamento, o processo ocorre de forma similar a rede ANFIS, consistindo de uma etapa *forward* e uma *backward*, em que os parâmetros de dilatação e de translação das *wavelets* são ajustados ao mesmo tempo que os pesos sinápticos e os parâmetros das funções de pertinência. Um ponto importante a ser comentado com relação às FWNN é que, independente da forma com que essas funções sejam ponderadas, o número de *wavelets* presentes em cada neurônio consequente será determinado pela quantidade de entradas da rede (n), e o número de neurônios *wavelets* (m), normalmente, é definido de acordo com a quantidade de entradas da rede e do número de funções de pertinência para cada uma dessas entradas, ou seja, esse número é determinado de acordo com o número de regras.

Assim, definida a forma como serão ponderados os consequentes, definido a dimensão da estrutura da rede e tendo em mãos os dados a serem utilizados na geração do modelo, os passos apresentados abaixo devem ser executados, de modo a ser realizado o treinamento:

- Passo 1: Inicializar os parâmetros da rede. Uma observação a ser feita é que, em termos práticos, a inicialização dos parâmetros das *wavelets* é um dos maiores problemas das redes neurais *wavelets*. Thuillard (2000) afirma que uma boa inicialização é extremamente importante para se obter uma rápida convergência e sucesso durante o treinamento.

- Passo 2: Apresentar à rede as entradas de treinamento e suas correspondentes saídas desejadas;

- Passo 3: Calcular a saída da rede;

- Passo 4: Calcular o valor do gradiente do erro para cada um dos parâmetros livres da rede através das equações apresentadas anteriormente;

- Passo 5: Atualizar os parâmetros adaptativos da rede;

- Passo 6: Validar o modelo obtido;

- Passo 7: Repetir os procedimentos 2-6 até que o modelo seja validado.

Esse mesmo algoritmo será utilizado no processo de treinamento da rede FWNN proposta, a ser apresentada no próximo capítulo. Além disso, para melhor esclarecimento, derivadas e expressões matemáticas necessárias a atualização dos parâmetros da rede são apresentadas, de modo a deixar mais claro o processo de implementação da rede proposta.

2.5 Conclusões

Neste capítulo, buscou-se detalhar as técnicas clássicas, que são as redes MLP, funções *wavelets*, redes WNN e as redes *neuro-fuzzy* ANFIS, que servem de base para as redes FWNN. As principais características dessas redes foram apresentadas, bem como a motivação de se combinar essas teorias. Buscou-se também apresentar uma estrutura geral da rede FWNN, apresentando suas principais características. Assim, a partir do exposto neste capítulo, pode-se ter uma clara noção, por meio da forma como as estruturas clássicas são combinadas, de como as redes FWNN são estruturadas, além de estabelecer uma fundamentação teórica para o melhor entendimento da rede FWNN proposta.

Capítulo 3

Sistema de Identificação Proposto

Este trabalho propõe uma FWNN que se diferencia de outras existentes na literatura, como por exemplo as apresentadas por Yilmaz & Oysal (2010), Abiyev & Kaynak (2008b) e Lu (2011), pelo fato de que os consequentes são constituídos apenas por funções *wavelets*, ponderadas apenas pelos pesos de ativação das regras *fuzzy*, não existindo uma segunda ponderação, que é ocasionada pela multiplicação das *wavelets* presentes por pesos sinápticos. Isso difere das estruturas FWNN presentes na literatura, em que as *wavelets* existentes em cada nó da camada consequente são multiplicadas por pesos sinápticos, fazendo com que cada modelo local seja ponderado duas vezes, pelas regras *fuzzy* e pelos pesos, tornando a quantidade de parâmetros ajustáveis maior do que a estrutura a ser apresentada neste trabalho.

3.1 *Fuzzy Wavelet Neural Network Modificada*

A FWNN proposta é uma alternativa às existentes na literatura, apresentando uma estrutura de menor complexidade matemática e de mais fácil compreensão e implementação. Isso ocorre, principalmente, quando a rede é utilizada na identificação de sistemas de maior complexidade, o que exige um conjunto de regras maior, aumentando, portanto, o número de consequentes.

Esta FWNN será treinada pelo algoritmo *backpropagation*. Como já explicado no capítulo 2, pelo fato de este algoritmo envolver o uso de derivadas, uma família de *wavelets* continuamente diferenciáveis deve ser adotada como consequentes da rede (este mesmo motivo

serve de justificativa para a escolha da função *bell shaped* como função de pertinência dos antecedentes). Dessa forma, no presente trabalho foi escolhida a *wavelet* da família *Mexican Hat*, Figura 2.3, conforme representada pela Equação 2.10.

O fluxo de dados desta estrutura, cuja representação gráfica é igual a Figura 3.1, pode ser analisado camada a camada, facilitando a compreensão de seus mecanismos de funcionamento e da modificação adotada.

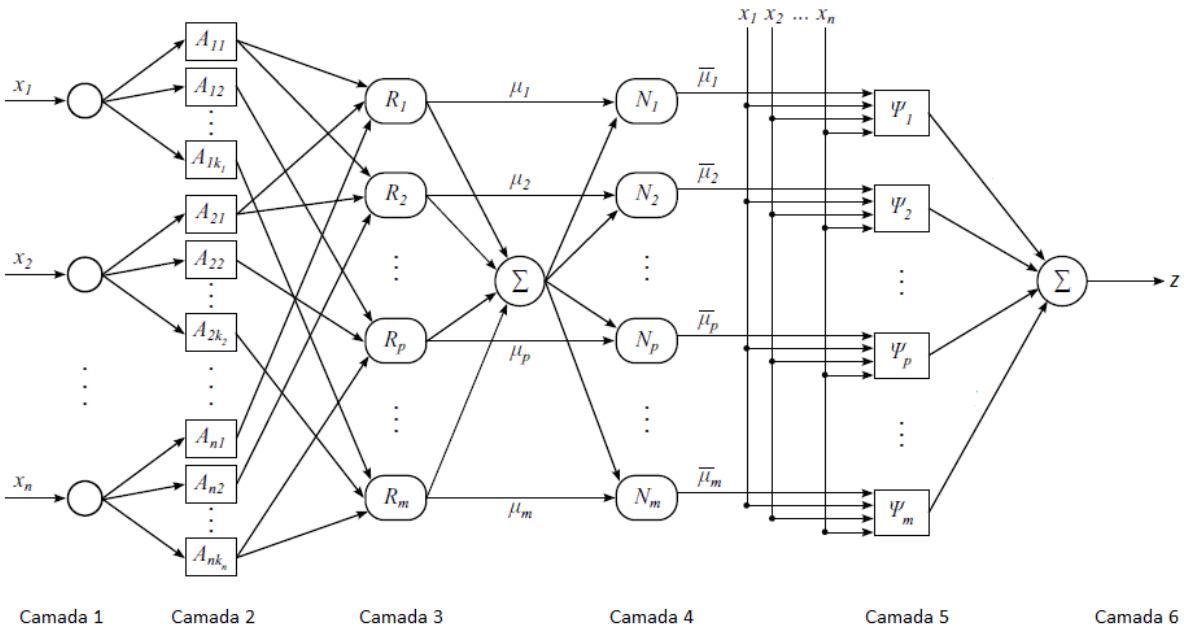


Figura 3.1: Estrutura da rede *Fuzzy Wavelet Neural Network* Proposta.

As camadas da rede são descritas a seguir:

Camada 1: Esta corresponde à camada de entrada da rede. Os sinais de entrada do vetor $x = [x_1, x_2, \ldots, x_n]$, contendo os dados a serem utilizados no treinamento, são transferidos para a segunda camada da rede através dos nós de entrada.

Camada 2: Esta é a camada de *fuzzificação*, consistindo de nós adaptativos que geram graus de pertinência baseados nos sinais de entrada utilizando alguma função de pertinência parametrizada, como por exemplo a função *bell shaped,* definida de acordo com a Equação 3.1.

$$A_{qr}\left(x_q\right)=\frac{1}{1+\left|\frac{x_q-c_{qr}}{a_{qr}}\right|^{2b_{qr}}} \tag{3.1}$$

onde q = 1, 2, . . . , n e r = 1, 2, . . . , k_q. A_{qr} corresponde à r-ésima função de pertinência relativa ao q-ésimo sinal de entrada da camada 1. As variáveis a_{qr}, b_{qr} e c_{qr} são, respectivamente, a abertura, a inclinação e o centro da função de pertinência. Cada um dos sinais de entrada possuem k_q funções de pertinência.

Camada 3: Esta é a camada onde ocorre a geração das regras da rede. Os nós dessa camada são nós fixos, cujos valores representam o peso de ativação de cada regra. A saída de cada nó é o resultado de uma operação do tipo T-norma, de todos os sinais de entrada dessa camada. As saídas μ_i dos nós desta camada, R_i, i = 1, 2, . . . ,m, são calculadas de acordo com as Equações de 3.2.

$$\begin{aligned}
\mu_1 &= A_{11}(x_1)A_{12}(x_2)....A_{n1}(x_n)\\
\mu_2 &= A_{11}(x_1)A_{21}(x_2)....A_{n2}(x_n)\\
&\vdots\\
\mu_m &= A_{1k_1}(x_1)A_{2k_2}(x_2)....A_{nk_n}(x_n)
\end{aligned} \tag{3.2}$$

De acordo com o apresentado em (Yilmaz & Oysal, 2010), a saída do p-ésimo nó da camada de regras é dada por:

$$\mu_p=\prod_{q=1}^{n}A_{qj}(x_q) \qquad p=1,2,....,m \qquad j=1,2,....,k_q \tag{3.3}$$

onde A_{qj} é a j-ésima função de pertinência do sinal de entrada q. Neste caso, j representa o número da única função de pertinência de x_q que fornece entrada para a regra μ_p. Portanto, o sinal de entrada fornece apenas um grau de pertinência para uma determinada regra. A partir disso, pode-se concluir que cada possível combinação de graus de pertinências provenientes da camada 2 representa uma regra *fuzzy*. Logo, o número total de regras é dado por:

$$m = \prod_{i=1}^{n} k_i \tag{3.4}$$

Apesar de ocorrer a combinação de todas as funções de pertinência presentes em todas as entradas, o número de regras acaba não se tornando excessivo, pois com um número limitado de funções de pertinência em cada entrada já é possível formar uma ampla base de regras.

Camada 4: As saídas desta camada representam a normalização dos pesos de ativação das regras, ou seja, a normalização das saídas dos neurônios da camada anterior. O fator de normalização μ_i da i-ésima regra é calculado por:

$$\bar{\mu}_i = \frac{\mu_i}{\sum_{i=1}^{m} \mu_i} \quad i = 1,2,....,m \tag{3.5}$$

Camada 5: Os nós dessa camada possuem funções *wavelets*. Assim, a partir das entradas, das regras normalizadas e dos parâmetros ajustáveis dessas funções, cada nó terá sua saída g_j calculada de acordo com as Equações 3.6 e 3.7.

$$g_j = \bar{\mu}_j \Psi_j \quad j = 1,2,.....,n \tag{3.6}$$

$$g_j = \bar{\mu}_j \sum_{i=1}^{m} \left(1 - \left(\frac{x - t_{ij}}{d_{ij}}\right)^2\right) e^{-0.5\left(\frac{x-t_{ij}}{d_{ij}}\right)^2} \frac{1}{\sqrt{d_{ij}}} \quad j = 1,2,....m \tag{3.7}$$

A variável *n* corresponde ao número de funções *wavelets* em um nó *wavelet* da camada 5. A Figura 3.2 apresenta com maiores detalhes a estrutura de um nó *wavelet*, ilustrando suas entradas, as *wavelets* e a ponderação proveniente das regras *fuzzy*.

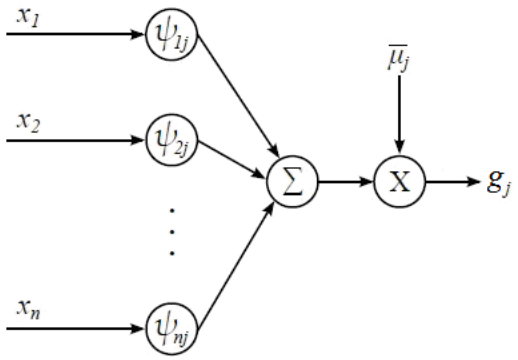


Figura 3.2: Estrutura de um neurônio presente na camada consequente.

Camada 6: Nessa camada, todas as saídas dos neurônios da camada 5 são somadas, resultando assim na saída da FWNN.

$$z = \sum_{j=1}^{m} g_j(\cdot) \tag{3.8}$$

A partir da rede apresentada na Figura 3.1 e representada pelas equações 3.1 a 3.8, pode-se identificar claramente que os neurônios que necessitam de aprendizado estão presentes na segunda e quinta camadas, pois na camada 2 estão localizadas as funções de pertinência de entrada e na camada 5, as *wavelets*, que definem as implicações das regras.

Para a execução do procedimento de aprendizagem da FWNN por meio do algoritmo *backpropagation*, é necessário obter as derivadas parciais da função custo adotada em relação aos parâmetros ajustáveis rede. Desta forma, o procedimento de treinamento envolve a utilização de derivadas parciais em relação aos parâmetros de dilatação d_{ij} e translação t_{ij} das *wavelets* e de abertura a_{qr}, inclinação b_{qr} e centro c_{qr} das funções de pertinência A_{qr}.

Uma vez que a FWNN proposta possui apenas *wavelets* nos consequentes, os pesos sinápticos presentes nas redes WNN deixam de ser encarados como parâmetros ajustáveis,

reduzindo a etapa *backward* do treinamento. Com relação ao número total de parâmetros ajustados nesta etapa de treinamento da rede proposta, o mesmo pode ser calculado a partir da seguinte equação:

$$N_p = \left(N_{in} N_{fp} N_{pfp}\right) + \left(N_r N_{in} N_{pw}\right) \tag{3.9}$$

onde o primeiro termo consiste no número de parâmetros dos antecedentes, sendo N_{in} o número de entradas da rede, N_{fp} o número de funções de pertinência por entrada, e N_{pfp} é o número de parâmetros presentes em cada função de pertinência. O segundo termo consiste no número total de parâmetros presentes nos consequentes das regras *fuzzy*. Neste segundo termo, N_r é o número de regras *fuzzy* e N_{pw} é a quantidade de parâmetros de cada *wavelet*.

Assim como apresentado por Haykin (2001), considere a função custo utilizada pelo algoritmo *backpropagation* para realizar o ajuste de parâmetros após a apresentação de um exemplo de treinamento como sendo a energia do erro (E) definida em 3.10, onde z_d e z são a saída desejada fornecida por um exemplo de treinamento e a saída estimada pela rede, respectivamente. Assim, as derivadas de E em relação aos parâmetros ajustáveis das funções *wavelets* são apresentadas nas equações 3.11 e 3.12.

$$E = \frac{1}{2} e^2 = \frac{1}{2} (z_d - z)^2 \tag{3.10}$$

$$\frac{\partial E}{\partial d_{i,l}} = -(z_d - z)\mu_i \left[e^{-0.5T_{ij}^2} \left(3\left(\frac{x - t_{ij}}{d_{ij}} \right)^2 - \left(\frac{x - t_{ij}}{d_{ij}} \right)^4 \right) \right] \frac{1}{\sqrt{d_{i,l}^3}} \tag{3.11}$$

$$\frac{\partial E}{\partial t_{i,l}} = -(z_d - z)\mu_i \left[e^{-0.5T_{ij}^2} \left(3\left(\frac{x - t_{ij}}{d_{ij}} \right) - \left(\frac{x - t_{ij}}{d_{ij}} \right)^3 \right) \right] \frac{1}{\sqrt{d_{i,l}^3}} \tag{3.12}$$

O ajuste dos parâmetros das funções de pertinência do sistema necessita das derivadas parciais das mesmas em relação aos seus parâmetros:

$$\frac{\partial E}{\partial a_{qr}} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial \Psi}\frac{\partial \Psi}{\partial \bar{\mu}}\frac{\partial \bar{\mu}}{\partial \mu}\frac{\partial \mu}{\partial A}\frac{\partial A}{\partial a_{qr}} \tag{3.13}$$

$$\frac{\partial E}{\partial b_{qr}} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial \Psi}\frac{\partial \Psi}{\partial \bar{\mu}}\frac{\partial \bar{\mu}}{\partial \mu}\frac{\partial \mu}{\partial A}\frac{\partial A}{\partial b_{qr}} \tag{3.14}$$

$$\frac{\partial E}{\partial c_{qr}} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial \Psi}\frac{\partial \Psi}{\partial \bar{\mu}}\frac{\partial \bar{\mu}}{\partial \mu}\frac{\partial \mu}{\partial A}\frac{\partial A}{\partial c_{qr}} \tag{3.15}$$

Onde:

$$\frac{\partial A_{qr}}{\partial a_{qr}} = \frac{2b_{qr}}{a_{qr}} A_{qr}(1 - A_{qr}) \tag{3.16}$$

$$\begin{cases} \dfrac{\partial A_{qr}}{\partial b_{qr}} = -2\ln\left|\dfrac{x_q - c_{qr}}{a_{qr}}\right| A_{qr}(1 - A_{qr}), se: x_q \neq c_{qr} \\ 0, \qquad\qquad se: x_q = c_{qr} \end{cases} \tag{3.17}$$

$$\begin{cases} \dfrac{\partial A_{qr}}{\partial c_{qr}} = \dfrac{2b_{qr}}{x_q - c_{qr}} A_{qr}(1 - A_{qr}), se: x_q \neq c_{qr} \\ 0, \qquad\qquad se: x_q = c_{qr} \end{cases} \tag{3.18}$$

A partir do cálculo das derivadas da energia do erro em relação aos parâmetros ajustáveis, pode-se realizar a atualização destes parâmetros a partir do principio da regra delta (Haykin, 2001), utilizando as equações abaixo:

$$d_{i,j}(k+1) = d_{i,j}(k) - \eta \frac{\partial E}{\partial d_{i,j}} \tag{3.19}$$

$$t_{i,j}(k+1) = t_{i,j}(k) - \eta \frac{\partial E}{\partial t_{i,j}} \tag{3.20}$$

$$a_{i,j}(k+1) = a_{i,j}(k) - \eta \frac{\partial E}{\partial a_{i,j}} \tag{3.21}$$

$$b_{i,j}(k+1) = b_{i,j}(k) - \eta \frac{\partial E}{\partial b_{i,j}} \tag{3.22}$$

$$c_{i,j}(k+1) = c_{i,j}(k) - \eta \frac{\partial E}{\partial c_{i,j}} \tag{3.23}$$

Época após época, o algoritmo *backpropagation* deverá ajustar os parâmetros da estrutura FWNN com o objetivo de minimizar a média da energia do erro (E_{av}) de todos os exemplos de treinamento apresentados ao algoritmo, até que seja atingido um determinado número de épocas. A média da energia do erro é representada matematicamente por:

$$E_{av} = \frac{1}{N}\sum_{i=1}^{N} E_i = \frac{1}{2N}\sum_{i=1}^{N}\left(z_i^d - z_i\right)^2 \tag{3.24}$$

em que z_i^d e z_i são, respectivamente, a saída desejada e a saída da rede FWNN para o *i*-ésimo exemplo de treinamento e N é o número total de pares entrada-saída do sistema a ser identificado utilizado durante a etapa de treinamento do modelo. Uma informação importante é que se deve dar especial atenção à inicialização dos parâmetros da *wavelet*. Uma inicialização inadequada pode comprometer o treinamento, e consequentemente, a qualidade do resultado final. Uma boa escolha dos parâmetros de dilatação e translação incrementa a velocidade de treinamento e resulta em rápida convergência.

3.2 Conclusões

O objetivo deste capítulo foi o de mostrar, de forma detalhada, a rede FWNN proposta. Uma breve comparação da mesma em relação às principais redes FWNN encontradas na literatura foi apresentada, tendo como objetivo evidenciar a contribuição da rede proposta. Para tornar claro como essa rede é implementada, a sua estrutura foi descrita detalhadamente, camada a camada, bem como as operações realizadas em cada uma de suas camadas.

Além disso, as equações matemáticas envolvidas no funcionamento da rede FWNN proposta e as derivadas necessárias ao ajuste de seus parâmetros foram apresentadas. Explicações foram dadas para justificar a escolha das funções que contém esses parâmetros, que são as funções de pertinência (*bell shaped*) e as funções *wavelets* (*Mexican Hat*).

De acordo com o já apresentado, o presente trabalho investiga a utilização da rede FWNN proposta na identificação de sistemas dinâmicos. Para um melhor entendimento da forma como essa rede pode ser aplicada para identificar a dinâmica dos sistemas utilizados como estudo de caso neste trabalho, o próximo capítulo apresentará em detalhes os principais conceitos relacionados à teoria de identificação de sistemas.

Capítulo 4

Identificação Não Linear

Como citado na introdução, na modelagem caixa-preta, também conhecida por modelagem experimental ou empírica, pouco ou nenhum conhecimento é exigido das leis da natureza que regem o comportamento dos sistemas reais a serem modelados. Na literatura este tipo de modelagem também é referenciado por identificação de sistemas (Narendra & Parthasarathy, 1990; Aguirre, 1998). Assim, este trabalho utilizará a rede proposta na identificação de sistemas. Dessa maneira, o comportamento dinâmico dos sistemas a serem identificados será estimado a partir de medições realizadas na entrada e na saída da planta, ou seja, a partir de pares de valores entrada-saída.

4.1 Conceitos Gerais

Segundo Nørgaard *et al*. (2001), identificar um sistema é a tarefa de inferir um modelo que represente a sua dinâmica a partir de uma série de medições de suas variáveis. A simulação, predição, detecção de falhas e o desenvolvimento de sistemas de inferência e controle são aplicações típicas da identificação de sistemas dinâmicos (Araújo Júnior, 2007).

No Capítulo 1, foi visto que se a identificação baseia-se exclusivamente em dados obtidos do sistema, assumindo nenhum ou pouco conhecimento sobre os princípios físicos do sistema, o processo de identificação é conhecido como modelagem caixa-preta. Um dos objetivos deste trabalho é analisar a aplicação da rede FWNN na identificação de sistemas dinâmicos não lineares. Sabe-se que, de maneira geral, as redes neurais são estruturas capazes de adquirir

conhecimentos a partir de um processo denominado de treinamento, que é realizado a partir de um conjunto de dados de entrada-saída. Desta forma, claramente, nota-se que a tarefa de identificar um processo a partir da modelagem caixa-preta pode ser realizada por meio de uma rede FWNN.

Segundo Nelles (2001), modelar e identificar sistemas não lineares são tarefas desafiadoras, pois os processos não lineares são únicos no sentido de que eles não compartilham muitas propriedades. Portanto, uma importante meta a ser atingida por qualquer esquema de identificação para sistemas não lineares é a generalidade, isto é, possuir a capacidade de descrever uma ampla classe de sistemas estruturalmente diferentes.

A Figura 4.1 ilustra a tarefa de identificar um sistema. Por uma questão de simplicidade, assume-se que o processo possui apenas uma única saída. Uma vez que se tem a formulação para uma única saída, é possível estender essa formulação para casos com múltiplas saídas. Um modelo deve representar o mais próximo o comportamento de um processo. Isto é tipicamente quantificado em termos de uma função de erro entre a saída do processo e saída do modelo. É através do erro observado que os parâmetros do modelo são ajustados.

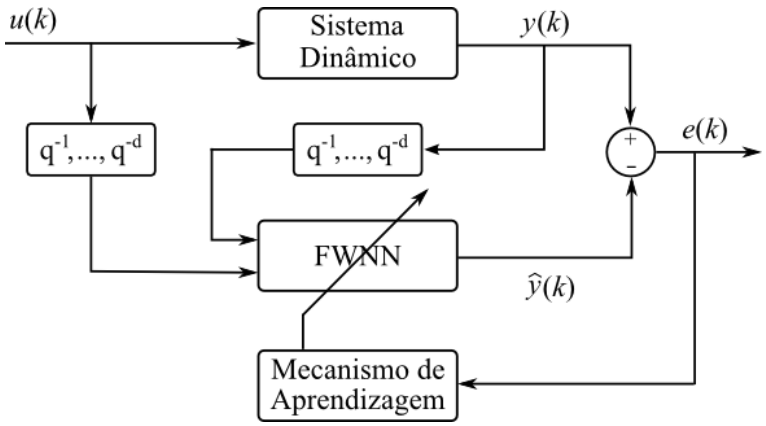


Figura 4.1: Esquema de identificação de sistemas dinâmicos – Série-Paralelo.

De forma geral, as entradas do modelo, de acordo com esse esquema, são valores presentes e passados do sinal de excitação $u(k)$ e a saída do sistema $y(k)$, assim como seus valores

passados. Nesta figura, *y(k)* é a resposta desejada do sistema e $\hat{y}(k) = z(k)$ é a saída estimada pela rede. Os parâmetros da FWNN são atualizados de acordo com a previsão de erro $e(k)$.

O procedimento básico para a realização da identificação do tipo caixa-preta assemelha-se bastante aos passos a serem seguidos no desenvolvimento de projetos utilizando redes neurais. As características das redes neurais e essa semelhança fazem dessas estruturas uma das técnicas mais utilizadas nesse tipo de identificação.

A Figura 4.2 ilustra, através de diagrama, o procedimento básico da modelagem do tipo caixa-preta. Na etapa de coleta de dados são realizadas intervenções no sistema com o objetivo de obter conjuntos de dados que descrevam o comportamento do sistema dentro de sua amplitude de operação. A ideia consiste em variar os sinais de entrada do sistema e observar a influência dessa variação em suas saídas. Os conjuntos de dados obtidos serão posteriormente utilizados para inferir o modelo do sistema. Em muitos casos, para intervir no processo e coletar os dados necessários à identificação, são aplicados ao sistema sinais de entrada ou de excitação ricos em frequências. Os sinais PRS (*Pseudo Random Signal*) e sua variações, como os sinais PRBS (*Pseudo Random Binary Signal*), são muito utilizados na prática, por possuírem esta característica.

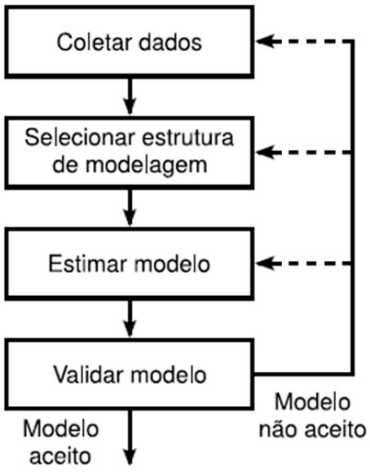


Figura 4.2: Procedimento básico de identificação.

Alguns cuidados importantes devem ser levados em consideração na coleta de dados, principalmente quando está se tratando com processos reais. Geralmente, as variáveis de um processo possuem diferentes magnitudes, dependendo das unidades de medida adotadas e da natureza do processo. Isso pode fazer com que as variáveis de maiores magnitudes tornem-se dominantes sobre as variáveis de menores magnitudes durante o processo de identificação. Portanto, nesses casos, faz-se necessária a normalização dos dados coletados. De acordo com Fortuna *et al*. (2007), as normalizações *min-max* e *zscore* são os métodos mais comumente utilizados.

Com relação à próxima etapa, diversas são as estruturas de modelagem existentes, cada uma com as suas vantagens e desvantagens em relação a determinados tipos de problemas. De acordo com Nørgaard *et al*. (2001), de uma forma geral, o problema de seleção destas estruturas pode ser dividido em duas partes:

1. Selecionar uma "família" de estruturas de modelagem apropriada para descrever o sistema, por exemplo, estruturas de modelos lineares, redes *perceptron* de múltiplas camadas, redes de funções de base radial, *wavelets*, etc.

2. Selecionar um subconjunto da família selecionada de estruturas de modelagem.

Após selecionar a estrutura de modelagem, o próximo passo é selecionar um modelo específico desta estrutura que possa melhor representar o sistema de acordo com algum critério, como o erro médio quadrático (EMQ). Este processo é conhecido na literatura estatística como estimação. Entretanto, este mesmo processo quando envolve modelos neurais é usualmente chamado de treinamento ou aprendizado. É nesta etapa que são utilizados os conjuntos de dados obtidos na etapa inicial do processo de identificação. Através dos dados experimentais e do algoritmo de treinamento, uma rede neural tem os seus parâmetros livres ajustados de forma a melhor representar a dinâmica do sistema a ser identificado.

Depois de o modelo ter sido estimado ou treinado, ele deve ser avaliado com o objetivo de analisar a sua capacidade de representar o sistema em estudo. Esta etapa é chamada de validação. Se o modelo satisfizer os requisitos do projeto, ele é aceito e estará pronto para ser utilizado para

os fins a que foi desenvolvido. Em caso contrário, deve-se retornar para alguma das etapas anteriores, buscando aprimorar o modelo.

4.2 Estruturas de Modelagem

As estruturas de modelagem baseadas em redes neurais apropriadas para identificação de sistemas não lineares são generalizações das estruturas de modelagem linear. Elas são caracterizadas pelo seu vetor de regressão, ou seja, pelo vetor que contém os valores passados das variáveis usadas para se estimar a saída do sistema. Algumas estruturas de modelagem linear são: FIR (*Finite Impulse Response*), ARX (*AutoRegressive, eXogenous input*), ARMAX (*AutoRegressive, Moving Average, eXogenous input*), OE (*Output Error*) e SSIF (*State Space Innovations Form*). Dependendo da escolha do vetor de regressão, diferentes estruturas de modelo neural emergem. Se o vetor de regressão for selecionado de forma similar à utilizada pelo modelo ARX, a estrutura de modelagem neural será chamada NNARX (*Neural Network ARX*). Do mesmo modo, temos as estruturas NNFIR, NNARMAX, NNOE e NNSSIF (Lucena, 2005).

As estruturas neurais citadas são muito bem descritas por Nørgaard *et al*. (2001), apresentando as vantagens e desvantagens de cada uma delas. Pode-se descrever resumidamente cada uma destas estruturas de acordo com seus vetores de regressão. No modelo NNFIR, o vetor de regressão é formado apenas por medições passadas das variáveis de entrada do processo. No modelo NNARX, são acrescentados ao vetor de regressão valores de medições passadas das variáveis de saída do processo. Segundo Nørgaard *et al*. (2001), estes dois modelos são sempre estáveis, pois representam puramente relações algébricas entre as variáveis do modelo. O mesmo não pode ser dito das demais estruturas de modelagem neurais.

A estrutura de modelagem NNARMAX adiciona ao vetor de regressão do modelo NNARX, informações sobre o erro de predição do modelo. Portanto, exige uma comparação entre o valor estimado e o valor real da variável de saída do modelo. O modelo NNSSIF utiliza uma abordagem um pouco diferente dos demais, adicionando ao modelo informações relativas ao espaço de estados do sistema. Portanto, este modelo é de potencial interesse no projeto de controladores em espaço de estados. Um outro detalhe a se destacar é que, de acordo com Sørensen (1993), o modelo NNSSIF pode ser considerado equivalente a um filtro de Kalman estendido para sistemas não lineares.

Os modelos de identificação que serão adotados como base para ser utilizados no decorrer deste trabalho são os modelos NNARX e o NNFIR, que, como citado anteriormente, são representações dos modelos ARX e FIR aplicadas às redes neurais.

Em termos matemáticos, a explicação para a obtenção do modelo NNARX se dá a partir da seguinte equação:

$$\hat{y}(k,\theta') = f[\varphi(k,\theta'),\theta'] \tag{4.1}$$

Onde θ' denota os parâmetros que serão ajustados no modelo. Sendo que φ e θ' são dados por:

$$\varphi(k) = [y(k-1),...,y(k-\alpha);u(k-d),...,u(k-d-\beta)]^T \tag{4.2}$$

$$\theta' = [-a_1,...,-a_\alpha;b_0,...,b_\beta]^T \tag{4.3}$$

No entanto, *f*[.] é uma função não linear, aproximada por uma estrutura específica, como por exemplo, uma rede MLP, WNN ou FWNN. Dessa forma, de maneira geral, a equação que define a estimação obtida por meio dessas estruturas não lineares pode ser representada pela Equação 4.4.

$$\hat{y}(k) = f(y(k-1),...,y(k-\alpha),u(k-d),...,u(k-d-\beta)) \tag{4.4}$$

sendo os valores de *y(k-i)* e *u(k-j)* dados por regressores, utilizados como entrada da rede. Como a rede proposta neste trabalho, quando aplicada na identificação do primeiro estudo de caso real, baseia-se no modelo NNARX, então essa estrutura pode ser chamada de FWNNARX.

A Figura 4.3 apresenta o diagrama dessa estrutura, onde $\hat{y}(k)$ é a saída estimada da planta, *d* é o atraso de transporte, α é a ordem da saída, β é a ordem da entrada, *y* é a saída e *u* a entrada da planta. Pode-se notar a aplicação de regressores neste modelo, fazendo com que a saída presente da rede esteja relacionada com valores passados de entrada e saída do sistema a ser identificado. A utilização de regressores é de fundamental importância, uma vez que a estrutura é projetada com o intuito de identificar a dinâmica existente em um sistema físico.

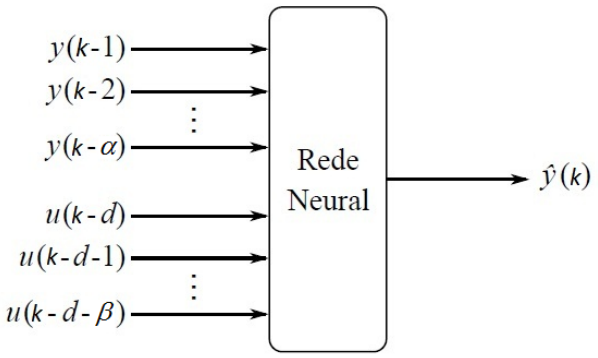


Figura 4.3: Estrutura do modelo NNARX.

No caso do modelo NNFIR (*Neural Network With Finite Impulse Response*), a determinação matemática é dada, também, a partir da Equação 4.1. Sendo que, neste caso, o vetor φ é definido por:

$$\varphi = [u(k-d),...u(k-d-\beta)]^T \tag{4.5}$$

Correspondentemente, o vetor de parâmetros θ' é composto por:

$$\theta' = [b_0, b_1, ..., b_\beta]^T \tag{4.6}$$

No entanto, da mesma forma que no caso do modelo NNARX, como o sistema a ser identificado é não linear, onde $f[.]$ pode ser aproximado por redes MLP, WNN ou FWNN, a expressão matemática do modelo não linear se dá a partir também da Equação 4.7.

$$\hat{y}(k) = f(u(k-d), u(k-d-1), ..., u(k-d-\beta)) \tag{4.7}$$

A estrutura adotada no último estudo de caso deste trabalho baseia-se na NNFIR, portanto, nesse caso, a rede proposta pode ser chamada de FWNNFIR.

De forma geral, o esquema correspondente ao modelo NNFIR é representado na Figura 4.4:

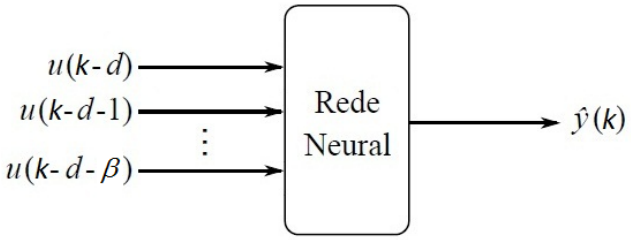


Figura 4.4: Estrutura do modelo NNFIR.

O modelo NNFIR, assim como o NNARX, pode ser definido como estável no sentido BIBO, considerando que o vetor de regressores e os pesos sinápticos têm valores finitos para qualquer instante *k*. Esta é uma característica importante para a análise de estabilidade de sistemas não lineares, devido a estes sistemas terem um comportamento mais complexo do que os sistemas lineares (Gabriel Filho, 2004).

4.3 Conclusões

Neste capítulo, o esquema de identificação a ser utilizado no presente trabalho foi apresentado. Também foram descritos os passos necessários no procedimento de identificação, bem como alguns critérios que devem ser analisados no decorrer deste processo, de tal forma que o sistema a ser modelado seja satisfatoriamente identificado. Os principais tipos de estruturas não lineares utilizadas na identificação de sistemas foram apresentados. Um maior destaque foi dado às estruturas NNARX e NNFIR, pois estas são as utilizadas no processo de treinamento da rede proposta, como poderá ser visto no próximo capitulo.

Maiores detalhes da utilização prática dessas estruturas por meio da rede FWNN proposta e também por outras redes implementadas, bem como os aspectos relacionados à implementação, definição dos regressores, coleta de dados, etc. são apresentados no próximo capítulo.

Capítulo 5

Metodologia e Estudos de Caso

A fim de avaliar o desempenho da FWNN proposta, quatro estudos foram realizados. Os dois primeiros são exemplos de sistemas dinâmicos simulados retirados da literatura. A principal razão para a utilização destes sistemas é que eles são funções não lineares complexas e todos eles são conhecidos por serem estáveis no sentido BIBO, ou seja, para uma entrada limitada, a saída também é limitada (Weigned *et al.,* 1990; Kim & Kim, 1997). Além disso, a utilização de tais sistemas torna mais fácil a comparação da FWNN proposta com outras estruturas, já que resultados com esses sistemas são facilmente encontrados na literatura, para diversos tipos de redes, incluindo outras redes do tipo FWNN.

Além da utilização desses modelos, o presente trabalho procurou aplicações práticas para testar a rede FWNN e avaliar o seu desempenho em sistemas reais. Inicialmente, um sistema dinâmico experimental consistindo de um tanque para armazenamento de líquidos com múltiplas seções é analisado. Neste último caso, por consistir de um experimento real, todas as estruturas de modelagem utilizadas na comparação tiveram de ser implementadas. Por motivos de comodidade, todas as redes foram treinadas pelo algoritmo *backpropagation* tradicional, de modo que uma comparação justa pudesse ser estabelecida. O objetivo da aplicação da técnica no sistema de tanques com múltiplas seções será o de identificar o comportamento do nível do tanque quando a bomba é submetida a várias tensões distintas. Para cumprir o objetivo, três entradas foram utilizadas na estrutura do identificador: o nível do tanque no instante atual, nível no instante anterior e o valor da tensão aplicada à bomba no instante atual.

Por fim, um segundo sistema dinâmico experimental foi utilizado no estudo. Este último estudo de caso consiste em uma incubadora neonatal. A motivação para aplicar a rede neural

proposta nesse sistema é o fato de que esse equipamento, por ser utilizado no apoio ao tratamento de recém nascidos, deve seguir sérias restrições no que diz respeito ao seu procedimento de calibração, estabelecidas pela norma IEC-60601-2-19 (ANSI/AAMI/IEC, 2009). Além disso, deve-se sempre otimizar ao máximo a sua utilização, diminuindo o número de paradas da mesma para calibração, de modo a atender o maior número possível de recém-nascidos. Assim, a rede FWNN proposta será utilizada para inferir temperaturas e a umidade presentes em vários pontos da incubadora, de modo a auxiliar no melhor funcionamento da mesma. Neste estudo de caso, a rede proposta também é comparada com outras redes neurais, sendo, todas elas, treinadas pelo algoritmo *backpropagation*.

Todas as redes implementadas neste trabalho foram desenvolvidas na linguagem de programação C++, utilizando o compilador Builder C++, versão 6.5, instalados em um computador com sistema operacional Windows 7, de 32 bits. De forma geral, para a aplicação da rede proposta nos quatro estudos de caso citados, foi realizada inicialmente uma etapa de comunicação. Essa etapa é responsável pela comunicação com o sistema para que seja efetuada a coleta tanto de dados para o procedimento de treinamento das redes quanto de dados para validação. Estes últimos dados servem para testar se os modelos encontrados possuem boa capacidade de generalização, representando bem as dinâmicas dos sistemas.

No caso dos sistemas simulados, representados por modelos matemáticos, essa etapa consiste apenas em coletar dados de forma direta, sem a necessidade de comunicação. No estudo de caso do tanque multisseções, drivers de uma placa de aquisição da *National Instruments* foram instalados no computador para estabelecer a comunicação entre o computador e a placa de aquisição. Já para o estudo de caso da incubadora, todo o processo de comunicação da incubadora com um computador foi feito via microcontroladores PIC, programados em *Matlab*.

Para o treinamento e validação das redes utilizadas no presente trabalho, foi utilizada a estrutura de identificação série-paralelo, apresentada em (Narendra & Parthasarathy, 1990), e ilustrado na Figura 4.1. Sendo assim, essa estrutura é adotada para identificar as relações dinâmicas dos estudos de caso utilizando a FWNN proposta e as outras redes utilizadas para efeito de comparação.

Um ponto importante a ser comentado no que diz respeito ao processo de treinamento das redes, é que todos os parâmetros das *wavelets* das redes implementadas foram inicializados de forma aleatória, com valores entre 0 e 1. Essa forma, embora não seja a mais eficiente, é bastante

utilizada pela sua simplicidade. Li *et al*. (2010), por exemplo, inicializa os parâmetros da rede de forma aleatória. Com relação aos parâmetros das funções de pertinência, estas foram inicializadas de forma a ficarem igualmente distribuídas e com pelo menos uma função de pertinência ativa em toda região de atuação da planta. Para isso, utilizou-se a estratégia em que o valor da pertinência no ponto de cruzamento de duas funções de pertinência deve ser igual a 0,5 para ambas. Maiores detalhes dessa forma de inicialização podem ser encontrados no trabalho de Rodrigues (2010).

Outro ponto a ser comentado, é que todos os parâmetros das redes FWNN desenvolvidas durante este trabalho foram ajustados simultaneamente pelo algoritmo *backpropagation*, tanto os parâmetros das funções de pertinência quanto os parâmetros de translação e dilatação das *wavelets*.

5.1 Sistema Dinâmico Simulado 1

Neste primeiro estudo de caso, um sistema dinâmico não linear simulado é adotado para avaliar o desempenho da estrutura FWNN proposta. Este sistema também é utilizado em Yilmaz & Oysal (2010); Abiyev & Kaynak (2008b); Juang (2002); Juang & Lin (1999). O seu comportamento dinâmico é dado pela seguinte equação:

$$y(k) = 0{,}72y(k-1) + 0{,}025y(k-2)u(k-1) + 0{,}01u^2(k-2) + 0{,}2u(k-3) \qquad (5.1)$$

Para efeito de ajuste dos parâmetros dos modelos dinâmicos da estrutura FWNN proposta e das outras FWNN presentes na literatura que serão utilizadas para verificação de desempenho, é preciso dispor de um número finito de exemplos de treinamento dados na forma de pares de vetores $[x(k);y(k)]$, onde $k = 1, 2,..., N$. Dessa forma, pode-se representar os N exemplos de treinamento como sendo:

$$\{[x(1), y(1)];[x(2), y(2)];...;[x(N), y(N)]\} \qquad (5.2)$$

onde $x(k)$ é o k-ésimo vetor de entradas do treinamento e $y(k)$ a respectiva saída desejada para o modelo.

Desta forma, os vetores $x(k)$ e $y(k)$ devem ser formados de acordo com os regressores a serem utilizados como entrada da rede. Portanto, assume-se que estes vetores estão relacionados segundo uma função *F[.]*, tal que:

$$d(k) = F[x(k)], \qquad k = 1,2,.....,N. \tag{5.3}$$

E justamente o mapeamento *F[x(k)]* que se deseja conhecer. Para isto pode-se utilizar as estruturas que serão analisadas neste trabalho para gerar uma aproximação de *F[x(k)]*, denotada por $\hat{F}[x(k)]$, tal que: $\hat{y} = \hat{F}[x(k)]$, onde $\hat{y}$ é o vetor de saídas geradas ou estimadas pela rede, que, espera-se, sejam muito próximas das saídas reais desejadas $y(k)$.

A partir disso, a fim de treinar a FWNN proposta, o conjunto de dados de treinamento foi obtido, contendo 900 entradas, similares às presentes nos trabalhos de Yilmaz & Oysal (2010), Juang (2002) e Juang & Lin (1999). Metade das entradas é gerada de forma Independente e Identicamente Distribuídas (IID) em uma sequência uniforme ao longo do conjunto [-2, 2], enquanto que a outra metade é formada por um sinal senoidal, dada por $1{,}05 sen(\pi k/45)$. Como pode ser visto na Equação 5.1, a saída do sistema depende de dois valores de saída anteriores e três valores de entrada anteriores. No entanto, apenas *u(k-1)* e *y(k)* são utilizados como entradas para o FWNN proposta para prever *y(k+1)*. Essas entradas são adotadas para estabelecer uma comparação mais justa com os resultados apresentados em Yilmaz & Oysal (2010) e Abiyev & Kaynak (2008b). Pelo mesmo motivo, duas funções de pertinência são usadas para cada entrada do modelo FWNN proposto. Assim, a terceira camada da estrutura da rede proposta é constituída por quatro regras *fuzzy*.

A rede proposta foi treinada usando as equações necessárias para atualização dos parâmetros, presentes no capítulo 3. Esse treinamento ocorreu em 200 épocas como em Yilmaz & Oysal (2010), e a sua curva de aprendizagem é apresentada na Figura 5.2. Como critério de desempenho, foi utilizada a raiz do erro médio quadrático (RMSE), dado pela equação 5.4, usando $N = 900$.

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y - \hat{y}\right)^2} \tag{5.4}$$

Após o procedimento de treinamento, o mesmo sinal de teste, *u(k)*, apresentado em Abiyev & Kaynak (2008b) e utilizado para avaliar o desempenho das outras redes, como encontrado na literatura, foi adotado para analisar a capacidade de generalização da FWNN proposta. Esse sinal é dado pela equação:

$$u(k) = \begin{cases} sen(\pi k/25), & k < 250 \\ 1.0, & 250 \le k < 500 \\ -1.0, & 500 \le k < 750 \\ 0{,}3 \cdot sen(\pi k/25) + 0{,}1 \cdot sen(\pi k/32) + 0{,}6 \cdot sen(\pi k/10) & 750 \le k < 1000. \end{cases} \quad (5.5)$$

Esse sinal de teste foi utilizado no procedimento de validação das redes. No presente trabalho, a validação é realizada de época em época, logo após o ajuste dos parâmetros da rede. Na validação, um conjunto de dados diferente do utilizado no procedimento de treinamento é apresentado à rede, com o objetivo de verificar se a estrutura neural é capaz de estimar satisfatoriamente o comportamento dinâmico do sistema. Com isso, é possível verificar, época após época, se a rede esta ganhando ou perdendo desempenho. Além disso, nesse processo de validação, quando o MSE obtido a partir dos dados de validação começa a aumentar continuamente de valor, o treinamento da rede é encerrado. Esta forma de realizar a validação foi utilizada tanto no presente como nos demais estudos de caso utilizados neste trabalho.

A Figura 5.1 mostra as saídas desejada e estimada do sistema dinâmico simulado 1 para a rede FWNN proposta. Como pode ser visto, o sinal de saída do sistema possui características bem diferentes em 3 etapas. Na primeira, o sinal apresenta características de um sinal do tipo senoidal, na segunda, o sinal assemelha-se a dois degraus, com amplitudes de +1 e -1; por fim, na terceira etapa o sinal apresenta características bem oscilatórias. Em todo o domínio de excursão do sinal, nota-se que a rede FWNN conseguiu sempre acompanhar a saída desejada com boa exatidão.

Para uma avaliação mais completa em relação às outras redes, foi montada a Tabela 5.1, que mostra os resultados de outros trabalhos encontrados na literatura. Nessa tabela, pode ser visto que o modelo proposto apresenta um melhor desempenho do que os outros modelos de redes neurais nomeadamente *Recurrent Self-Organizing Neural Fuzzy Inference Network* -

RSONFIN - (Juang & Lin, 1999), *Feedforward Neural Fuzzy System* - ERNN - (Elman, 1990), e *Trapezoidal Fuzzy Numbers* - TRFN-S - (Juang, 2002), mesmo essas redes utilizando um número maior de parâmetros.

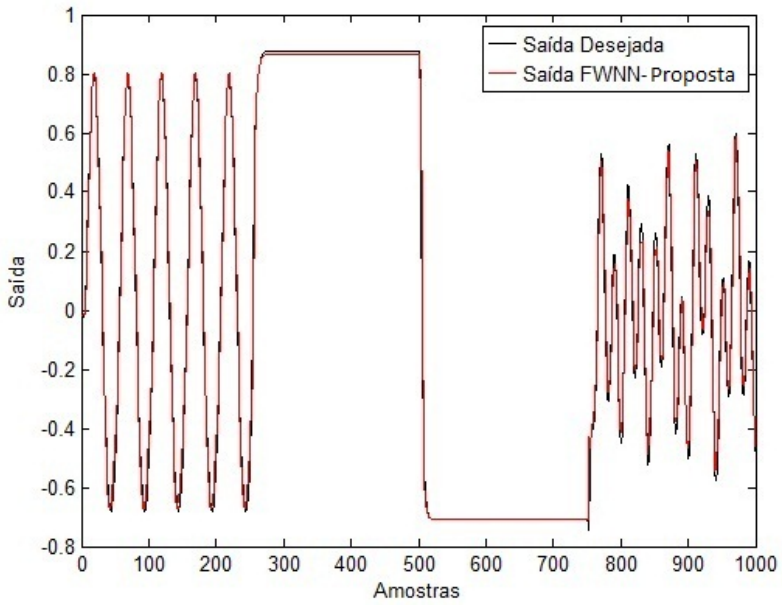


Figura 5.1: Resultado da validação com a FWNN proposta para o sistema dinâmico simulado 1.

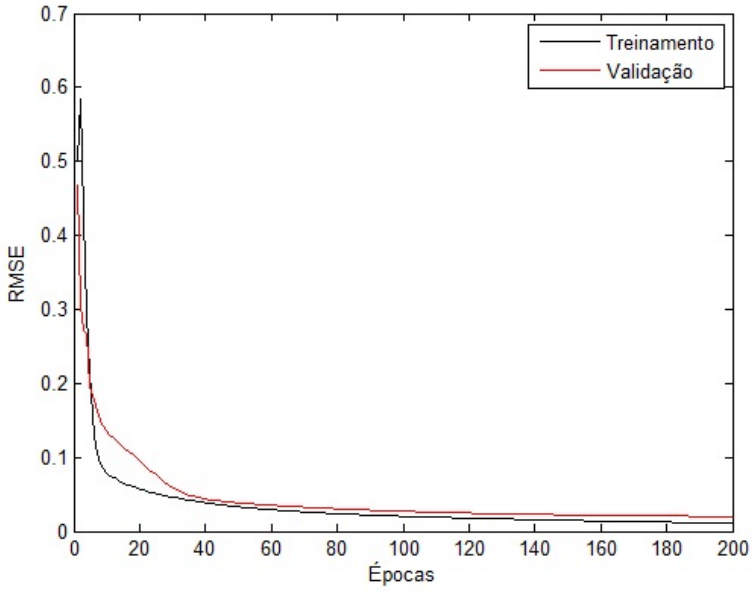


Figura 5.2: Valor do RMSE obtido durante o treinamento e validação do sistema dinâmico simulado 1.

Modelos	Número de Parâmetros	Número de Regras	RMSE de Validação
ERNN	54	-	$7{,}8 \times 10^{-2}$
RSONFIN	49	-	$6{,}0 \times 10^{-2}$
TRFN-S	33	-	$3{,}13 \times 10^{-2}$
FWNN-Abiyev 1	27	3	$2{,}2609 \times 10^{-2}$
FWNN-Abiyev 2	43	5	$2{,}0169 \times 10^{-2}$
FWNN-Yilmaz 1	32	4	$2{,}2226 \times 10^{-2}$
FWNN-Yilmaz 2	28	4	$2{,}2204 \times 10^{-2}$
FWNN-Yilmaz 3	32	4	$2{,}1342 \times 10^{-2}$
FWNN Proposta	24	4	2.3075×10^{-2}

Tabela 5.1: Comparação do resultado da FWNN proposta com outras redes para o sistema dinâmico simulado 1.

Com relação às redes FWNN, percebe-se que o desempenho da FWNN proposta (*RMSE* = $2{,}3075 \times 10^{-2}$) é próximo aos apresentados pelas redes presentes em Yilmaz & Oysal (2010) e Abiyev & Kaynak (2008b), embora a rede FWNN proposta utilize menos parâmetros de aprendizagem. Com relação aos parâmetros, percebe-se que, embora a redução de parâmetros não tenha sido significativa, deve-se atentar para o fato de que as estruturas analisadas são pequenas, possuindo um número pequeno de regras.

5.2 Sistema Dinâmico Simulado 2

Outro sistema dinâmico simulado é utilizado como um segundo estudo de caso de identificação. O mesmo processo também é usado em Yilmaz & Oysal (2010), Abiyev & Kaynak (2008b), Juang (2002), Juang & Lin (1999), Lee & Teng (2000), Wang & Chen (2008) e é expressado pela seguinte equação a diferenças:

$$y(k+1) = f(y(k), y(k-1), y(k-2), u(k), u(k-1)) \tag{5.6}$$

Onde:

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2} \tag{5.7}$$

Como pode ser visto, a saída atual desse sistema depende de dois valores atrasados da entrada e de três valores atrasados da saída. A fim de comparar os resultados obtidos com os indicados em Yilmaz & Oysal (2010) e Abiyev & Kaynak (2008b), apenas *y(k)* e *u(k)*, foram usados para prever *y(k+1)*. Para treinar a rede FWNN proposta, 900 amostras de treinamento foram geradas, assim como no estudo de caso anterior. Além disso, duas funções de pertinência também foram utilizadas neste exemplo. Essas especificações foram definidas de tal forma que a comparação fosse a mais justa possível com os resultados apresentados no trabalho de Yilmaz & Oysal (2010). A fim de validar o modelo obtido por meio do treinamento, o sinal de entrada apresentado na Equação 5.5 foi usado.

As saídas estimada e desejada obtidas na validação da rede proposta são apresentadas na Figura 5.3. A dinâmica desse sistema é bastante similar à apresentada pelo estudo de caso anterior, no entanto, o mesmo apresenta algumas oscilações no inicio do primeiro degrau, fato este que o torna mais difícil de ser identificado do que o do 1º estudo de caso. Apesar disso, percebe-se pela Figura 5.3 que a FWNN proposta apresentou um resultado bastante satisfatório, tendo a sua saída permanecida próxima da saída desejada.

Apesar do resultado gráfico mostrar o bom desempenho da rede proposta, para que se tenha uma real avaliação dessa performance, a Tabela 5.2 apresenta resultados obtidos a partir de outras redes, e encontrados facilmente na literatura. Diante desses resultados numéricos, percebe-se que a rede proposta apresentou resultados melhores do que as cinco primeiras redes presentes na Tabela 5.2, propostas respectivamente por Lee & Teng (2000), Juang & Lin (1999), Juang (2002) e Wang & Chen (2008). Em relação às redes FWNN utilizadas na comparação, pode-se verificar claramente que o desempenho da rede proposta apresentou resultados bastante próximos, tendo um RMSE igual a 3,7901 x 10^{-2}, enquanto os melhores RMSEs apresentados pelas outras redes são 3,2116 x 10^{-2} (Yilmaz & Oysal, 2010) e 3,0125 x 10^{-2} (Abiyev & Kaynak, 2008b).

Assim, tal como no primeiro estudo de caso, o desempenho da FWNN proposta é próximo do obtido por outros modelos FWNN. Ela também apresenta um desempenho melhor do que os outros modelos de redes neurais, com exceção de RFNN (Alieva *et al.*, 2009). Mas a RFNN tem

96 parâmetros a serem ajustados durante a fase de treinamento, o que corresponde a um número muito maior do que a rede proposta, que possui 24.

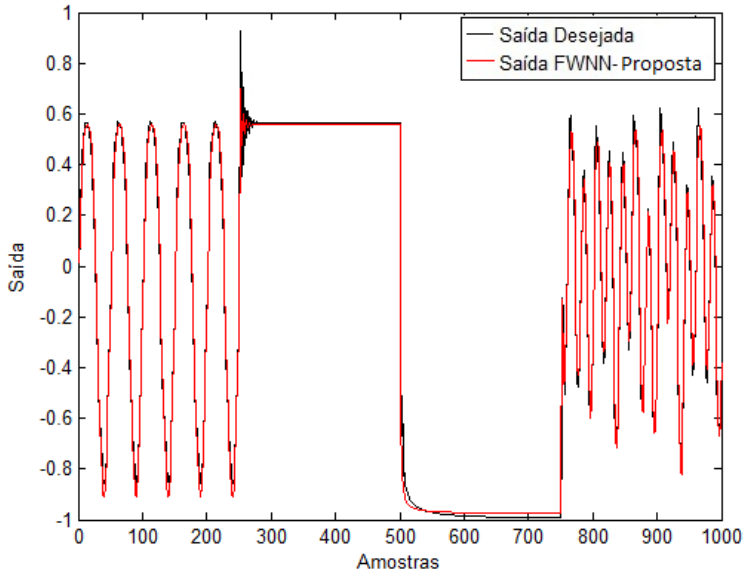


Figura 5.3: Resultado da validação com a FWNN proposta para o sistema dinâmico simulado 2.

Modelos	Número de Parâmetros	Número de Regras	RMSE de Validação
RFNN - Lee & Teng (2000)	112	-	$5{,}75\ x\ 10^{-2}$
RSONFIN	36	-	$7{,}80\ x\ 10^{-2}$
Feedforw. Neur. Fuz	48	-	$5{,}21\ x\ 10^{-2}$
TRFN-S	33	-	$3{,}46\ x\ 10^{-2}$
HRNFN	21	-	$4{,}93\ x\ 10^{-2}$
RFNN – Alieva (2009)	96	-	$6{,}4\ x\ 10^{-3}$
FWNN-Abiyev 1	27	3	$3{,}1212\ x\ 10^{-2}$
FWNN-Abiyev 2	43	5	$3{,}0125\ x\ 10^{-2}$
FWNN-Yilmaz 1	32	4	$3{,}3724\ x\ 10^{-2}$
FWNN-Yilmaz 2	28	4	$3{,}2116\ x\ 10^{-2}$
FWNN-Yilmaz 3	32	4	$3{,}3327\ x\ 10^{-2}$
FWNN Proposta	24	4	$3{,}7901\ x\ 10^{-2}$

Tabela 5.2: Comparação do resultado da FWNN proposta com outras redes para o sistema dinâmico simulado 2.

As curvas do erro RMSE de treinamento como uma função do número de épocas são mostrados na Figura 5.4.

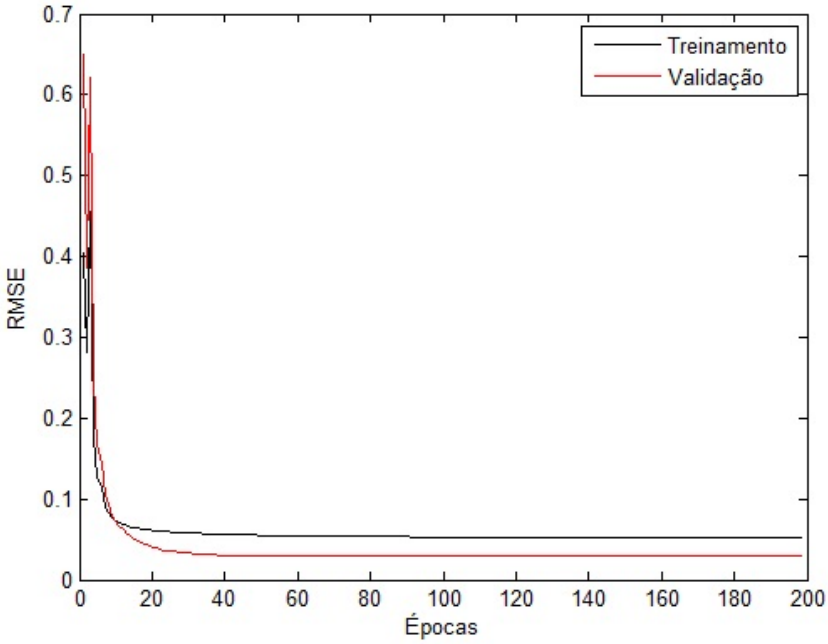


Figura 5.4: Valor do RMSE obtido durante o treinamento e validação do sistema dinâmico simulado 2.

Diante dos resultados apresentados, pode-se concluir que a rede proposta identificou com sucesso a dinâmica do sistema presente neste segundo estudo de caso. Além disso, como as FWNNs analisadas neste estudo de caso possuem uma estrutura pequena, com poucas funções de pertinencia e poucas regras, percebe-se que a redução de parâmetros não foi significativa.

Uma observação que deve ser feita é que as outras FWNN utilizadas na comparação, tanto no presente estudo de caso como no anterior, utilizam algoritmos de treinamento com alguma modificação, de modo a apresentar uma performance ainda melhor. No caso da FWNN proposta, o treinamento ocorreu somente pela utilização do algoritmo *backpropagation* em sua forma clássica, sem utilização de ferramentas de otimização do processo de treinamento, como por exemplo, eta adaptativo.

5.3 Sistema Dinâmico Real 1 - Tanque Multisseções

Após utilizar a rede FWNN proposta em dois estudos de caso simulados, o próximo passo do presente trabalho foi aplicar esta mesma rede na identificação de sistemas reais. Com este intuito, primeiramente foi utilizado um tanque em acrílico contendo múltiplas seções (Fonseca, 2012). Na Figura 5.5 é possível visualizar a estrutura do tanque multisseções por meio de seu desenho esquemático.

O tanque é constituído por três seções, cada uma delas formada por um paralelepípedo. A base do tanque, onde estão localizados o orifício de escoamento e o sensor de nível, possui 10 cm de altura e sua área de seção transversal é de aproximadamente 25 cm^2. As demais seções possuem áreas de 36 cm^2 e 49 cm^2. Em termos de capacidade de armazenamento, o tanque comporta um total de 1198 cm^3 (1,2 litros). Considerando que no presente trabalho foi utilizado somente 5 cm de altura da seção maior, o volume de armazenamento nessas circunstancias é de 855 cm^3.

Em termos de tempo de escoamento do liquido presente no interior do tanque, o mesmo demora 37 segundos para esvaziar por completo, considerando uma coluna de liquido de 25 cm em seu interior. Deste período de tempo, 7 segundos são necessários para esvaziar a seção superior (diminuição do nível de 25 cm para 20 cm), 14 segundos para a seção intermediária, e, por fim, 16 segundos para que a seção inferior e o tanque encontrem-se vazios.

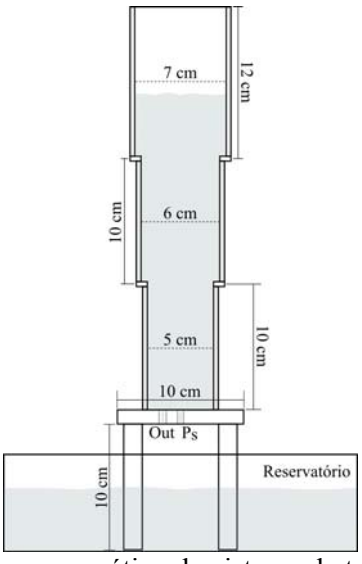


Figura 5.5: Desenho esquemático do sistema de tanque multisseções.

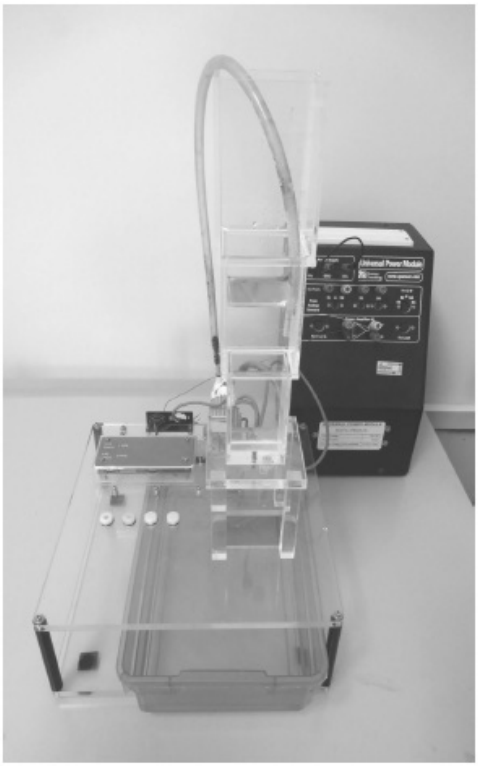


Figura 5.6: Visão geral do tanque multisseções.

O funcionamento e operação desse sistema é relativamente simples. Uma bomba de sucção, que admite tensões de 0V a +15V e apresenta vazão máxima de 100 cm^3/seg, eleva a água contida em um reservatório inferior através de uma mangueira de borracha até a parte superior do tanque. A água, então, escoa pelas diferentes seções do tanque até passar pelo seu orifício de saída e retornar ao reservatório.

O tanque multisseções utilizado para analisar a FWNN proposta é um exemplo de sistema não linear. Entre as não linearidades contidas no sistema, pode ser citada a diferença nas áreas de seção transversal dos paralelepípedos que constituem o tanque. Cada uma das seções do tanque apresenta um comportamento dinâmico diferente. Ainda tratando das não linearidades presentes neste estudo de caso, um ponto importante a ser comentado é que a transição entre uma seção e

outra do tanque multisseções ocorre de maneira abrupta, apresentando um comportamento não diferenciável. Isto ocorre devido ao formato de paralelepípedo das seções do tanque.

Neste estudo de caso, todas as redes utilizadas para efeito de comparação foram implementadas. A fim de se ter uma comparação mais justa, todas as redes implementadas foram treinadas pelo mesmo algoritmo de treinamento utilizado na rede proposta.

O processo de identificação deste sistema teve início com a coleta de dados, tendo como objetivo obter amostras experimentais que descrevam o comportamento dinâmico do processo dentro de toda sua faixa de operação. Para a obtenção das amostras a serem utilizadas nas etapas de treinamento e validação das redes analisadas neste trabalho, foi aplicado um sinal PRS (*Pseudo Random Signal*) para variar a tensão de alimentação da bomba responsável por transferir água do reservatório para o tanque de multisseções.

O sinal PRS estende os PRBS (Pseudo sequência binária aleatória) para diferentes amplitudes. A Figura 5.7 mostra o sinal PRS usado no presente estudo de caso.

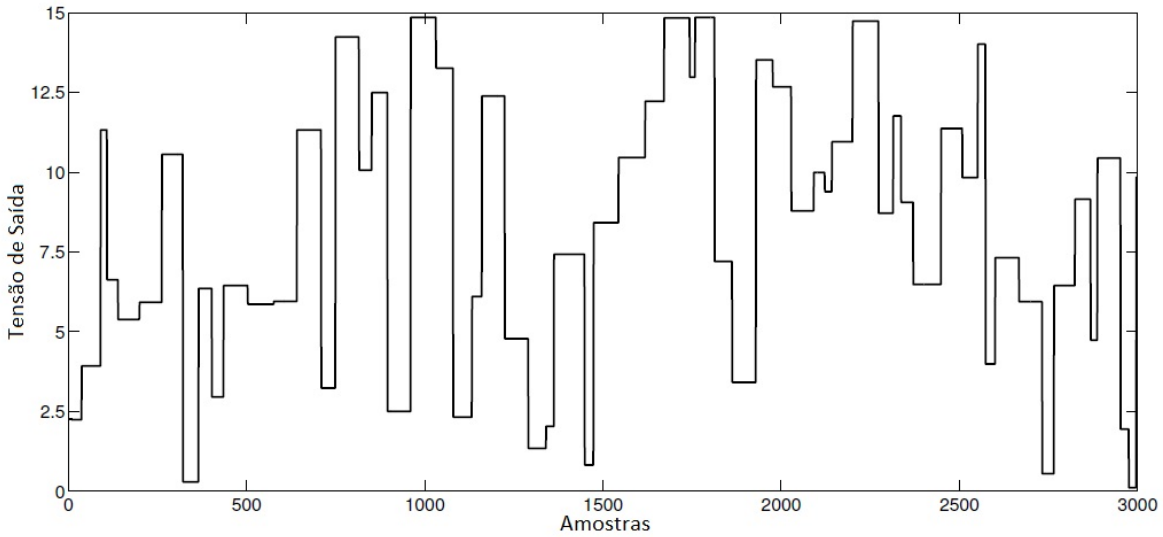


Figura 5.7: Sinal PRS aplicado na bomba de água do tanque.

O período de amostragem utilizado durante este procedimento foi de 0,5 s. Os dados coletados foram divididos em dois conjuntos, um de treinamento e um de validação. Assim,

depois da excitação do sistema, os dados recolhidos foram divididos em dois grupos que compreendem cerca de 80% e 20% do total. O primeiro grupo foi utilizado no treinamento das estruturas neurais de identificação e o segundo foi utilizado na fase de validação. Tanto no treinamento como na validação, as entradas da rede e a saída desejada foram normalizadas entre 0 e 1, a fim de evitar problemas numéricos na fase de treino.

As redes foram implementadas tendo como estrutura o modelo NNARX de 1ª ordem. Considerando a Figura 5.8, as entradas dos modelos FWNN são definidas com $n = m = 1$ e $d = 0$. Assim, *u(k)*, *u(k - 1)* e *y(k-1)* são definidas como sendo as entradas das redes, de modo a se estimar o valor de *y(k)*.

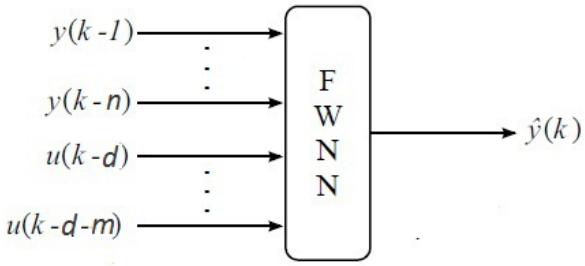


Figura 5.8: Estrutura do modelo NNARX usando a rede FWNN.

Após o modelo dinâmico do sistema ser estimado pela rede, é necessário investigar se este possui uma capacidade de generalização satisfatória. Esta investigação é realizada durante a etapa de validação do modelo, a qual deve estar, preferencialmente, relacionada com a utilização prática do modelo estimado. Nesta etapa são apresentados ao modelo obtido dados experimentais diferentes dos utilizados no ajuste de seus parâmetros. Assim, é possível avaliar a capacidade de generalização da rede em relação à dinâmica do sistema em estudo.

O desempenho da FWNN proposta é comparado com as redes FWNN apresentadas pelos autores Yilmaz & Oysal (2010), por Abiyev & Kaynak (2008b) e por Lu (2011). Além disso, para efeito de melhor avaliação com algumas estruturas neurais clássicas, as redes ANFIS (Jang, 1993), WNN (Araújo Júnior *et al*, 2013a) e MLP (Haykin, 2001) também serão utilizadas na comparação.

Para gerar os resultados, cada estrutura de identificação foi treinada 20 vezes. Alguns critérios utilizados para a comparação estão relacionados às dimensões das estruturas de identificação, como o número de parâmetros a serem ajustados, número de funções de pertinência para cada entrada do modelo e número de regras *fuzzy*. Em relação ao processo de treinamento, o critério será a média do número de épocas de treinamento.

Os critérios de desempenho considerados neste trabalho são a média do erro médio quadrático médio com relação aos dados de validação e a variância da sequência desses erros. Os resultados obtidos neste estudo de caso estão resumidos na Tabela 5.3.

O primeiro critério nesta tabela, Número de Parâmetros, mostra a quantidade de parâmetros a serem ajustados durante o processo de treinamento. Como esperado, o número de parâmetros da rede FWNN proposta, quando utilizada com duas funções de pertinencia, é menor do que o apresentado pelas outras redes FWNN, devido ao fato de a mesma não possuir pesos sinápticos após a camada de inferência. A diferença maior ocorre em relação à rede FWNN proposta por Yilmaz & Oysal (2010), que apresenta a estrutura mais parecida com a rede proposta, em que todas as funções de pertinência são combinadas, de modo que o número de regras é maior. Com relação às redes presentes nos trabalhos de Abiyev & Kaynak (2008b) e Lu (2011), percebe-se que as mesmas, com dez funções de pertinência têm menos parâmetros do que a FWNN proposta, quando treinada com três funções de pertinência. No entanto, a FWNN proposta com apenas duas funções de pertinência para cada entrada do modelo já apresentou resultados satisfatórios. Além disso, nessas duas redes, não há uma combinação de todas as funções de pertinência, fato este que acaba gerando uma base de regras menor, e, consequentemente, um número de neurônios menor na camada consequente.

Continuando a análise, os dois critérios seguintes indicados na Tabela 5.3 são o número de funções de pertinência e o número de regras, que é obtido de acordo com a forma como as funções de pertinência são combinadas. A rede proposta é testada com oito e vinte e sete regras, assim como a FWNN presente em Yilmaz & Oysal (2010), uma vez que, como citado acima, a associação das funções de pertinência nestas estruturas ocorre de maneira idêntica. As redes presentes em Abiyev & Kaynak (2008b) e por Lu (2011) usam menos regras, cinco, sete e dez, devido ao fato de que o seu número de regras é igual ao número de funções de pertinência existente em cada uma das entradas destas estruturas.

Modelos	Número de Parâmetros	Número de Funções de Pertinência	Número de Regras	Número Médio de Épocas	Média do MSE de Validação	Variância do MSE
FWNN-Lu 1	65	5	5	181,0	$8{,}023\ x\ 10^{-4}$	$3{,}091\ x\ 10^{-7}$
FWNN-Lu 2	91	7	7	183,3	$7{,}320\ x\ 10^{-4}$	$2{,}386\ x\ 10^{-7}$
FWNN-Lu 3	130	10	10	189,0	$5{,}617\ x\ 10^{-4}$	$1{,}160\ x\ 10^{-7}$
FWNN- Abiyev 1	65	5	5	191,1	$8{,}542\ x\ 10^{-4}$	$3{,}040\ x\ 10^{-7}$
FWNN- Abiyev 2	91	7	7	196,9	$6{,}039\ x\ 10^{-4}$	$1{,}905\ x\ 10^{-7}$
FWNN- Abiyev 3	130	10	10	189,7	$4{,}061\ x\ 10^{-4}$	$2{,}263\ x\ 10^{-8}$
FWNN- Yilmaz 1	84	2	8	187,3	$5{,}799\ x\ 10^{-4}$	$2{,}559\ x\ 10^{-7}$
FWNN- Yilmaz 2	261	3	27	195,3	$2{,}915\ x\ 10^{-4}$	$4{,}823\ x\ 10^{-9}$
ANFIS 1	42	2	4	191,0	$1{,}323\ x\ 10^{-3}$	$1{,}689\ x\ 10^{-6}$
ANFIS 2	108	3	9	199,5	$3{,}560\ x\ 10^{-4}$	$5{,}087\ x\ 10^{-10}$
WNN	60	6 Neurônios	-	247,3	$1{,}217\ x\ 10^{-3}$	$3{,}858\ x\ 10^{-7}$
MLP	24	6 Neurônios	-	400,0	$3{,}8\ x\ 10^{-3}$	$3{,}0\ x\ 10^{-6}$
FWNN-Proposta 1	60	2	8	150,0	$7{,}769\ x\ 10^{-4}$	$8{,}202\ x\ 10^{-8}$
FWNN- Proposta 2	180	3	27	172,0	$2{,}972\ x\ 10^{-4}$	$2{,}207\ x\ 10^{-8}$

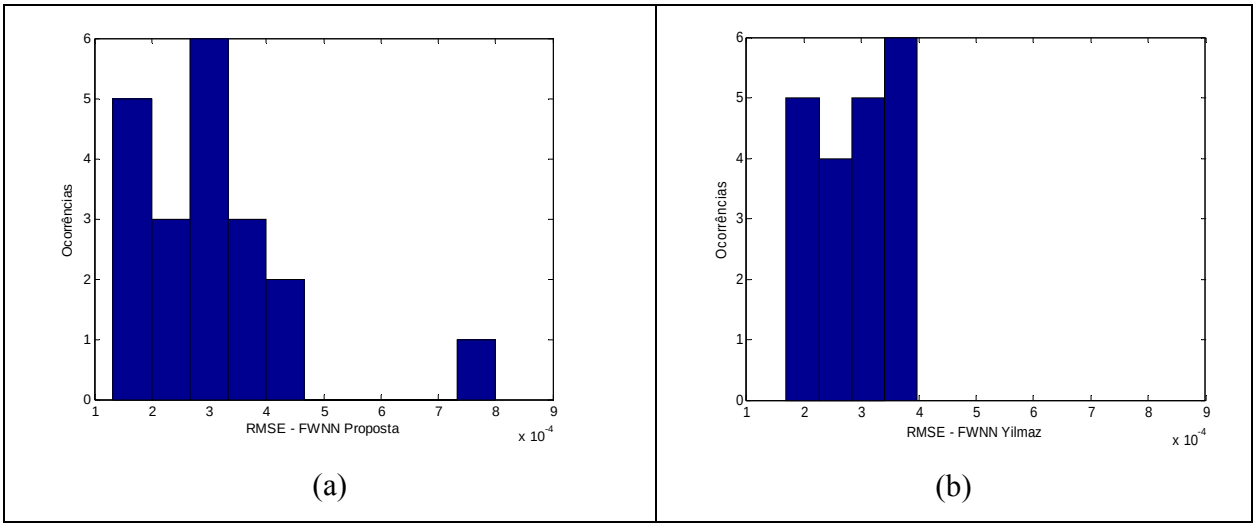
Tabela 5.3: Comparação entre a FWNN proposta e outras redes para tanque de nível multisseções.

Sobre o critério relacionado com o processo de treinamento da rede, o número médio de épocas (obtido após cada estrutura ter sido treinada 20 vezes), pode ser visto que, para este estudo de caso, o procedimento de aprendizagem da FWNN proposta é mais rápido do que o das estruturas FWNN restantes. É importante ressaltar que o final de cada processo de treinamento foi determinado a partir do momento em que o valor do MSE de validação, após estabilizar em um valor satisfatório, começava a dar indícios, através do aumento de seu valor, de que a rede encontrada até o momento iria perder desempenho em termos de generalização.

Finalmente, as duas últimas colunas da Tabela 5.3 mostram a média e a variância dos MSEs obtidos para cada uma das estruturas avaliadas no processo de identificação após cada estrutura ter sido treinada 20 vezes. Pode-se ver que o valor médio do MSE apresentado pela rede proposta, que é $2{,}972\ x\ 10^{-4}$, é próximo ao apresentado por outras redes. No entanto, o MSE médio indica que a FWNN presente em Yilmaz & Oysal (2010) apresenta valores melhores para este critério, com um valor de $2{,}915\ x\ 10^{-4}$, sendo seguida pela rede proposta.

Com relação à variância, pode-se verificar que o resultado obtido pela FWNN proposta e pelas outras redes FWNN tiveram pouca diferença, apresentando valores muito pequenos, mostrando que a convergência das redes é sempre satisfatória. O melhor resultado obtido foi o da rede ANFIS, treinada com 3 funções de pertinência. Isso ocorre pelo fato de que a rede ANFIS não possui parâmetros inicializados de forma aleatória (Rodrigues, 2006), como é o caso das redes FWNNs.

A comparação dos resultados referentes a média do MSE e da variância obtidos pelas redes FWNN Proposta e a presente em Yilmaz & Oysal (2010), cuja variância foi a melhor dentre as FWNNs apresentadas, com valor de 4,823 x 10^{-9}, pode ser verificada também a partir dos histogramas do MSE, apresentados na Figura 5.9.



(a) (b)

Figura 5.9: Histograma dos MSEs para (a) FWNN – Proposta, (b) FWNN – Yilmaz & Oysal (2010).

A curva mostrada na Figura 5.10 apresenta o comportamento do erro quadrático médio (MSE) de treinamento e de validação da rede proposta.

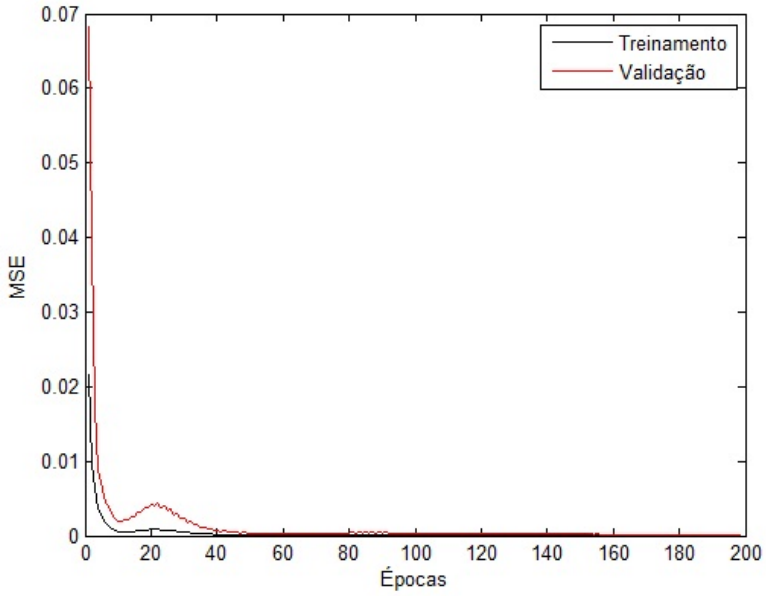


Figura 5.10: Erro Médio Quadrático da rede FWNN proposta.

As Figuras 5.11 a 5.14 mostram graficamente o resultado da saída das redes FWNNs utilizadas na comparação em relação à saída desejada.

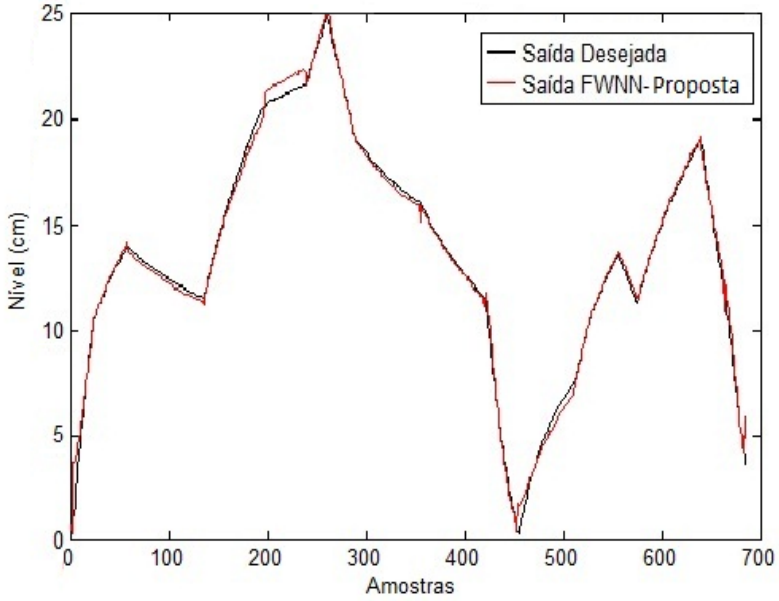


Figura 5.11: Resposta da melhor rede FWNN proposta.

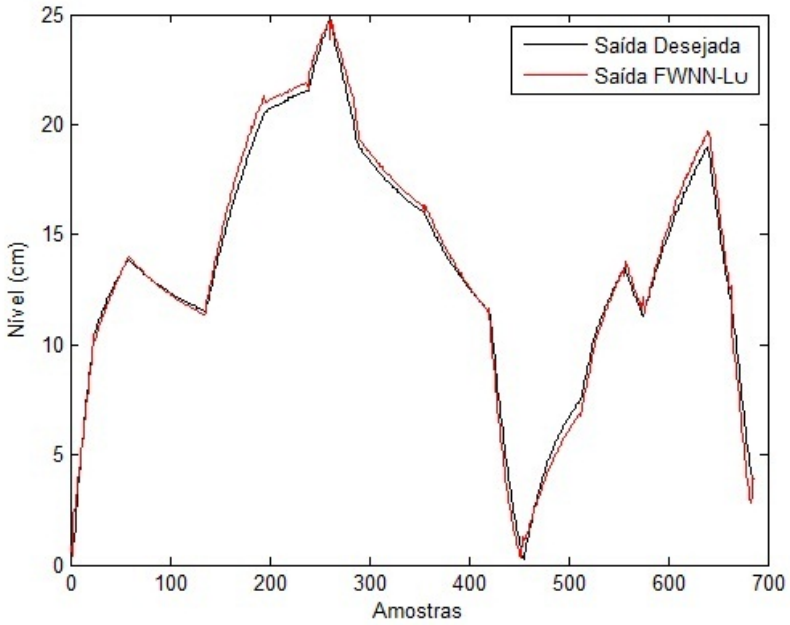


Figura 5.12: Resposta da melhor rede FWNN - Lu (2011).

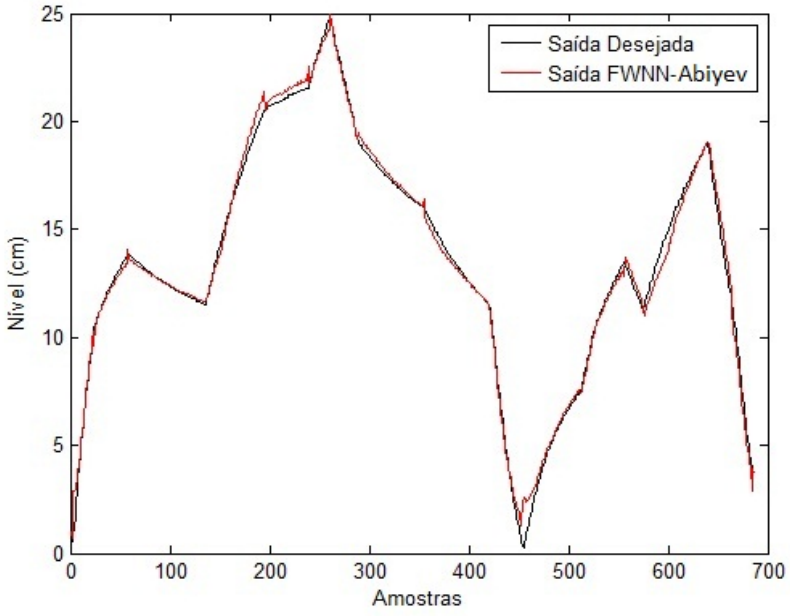


Figura 5.13: Resposta da melhor rede FWNN - Abiyev & Kaynak (2008b).

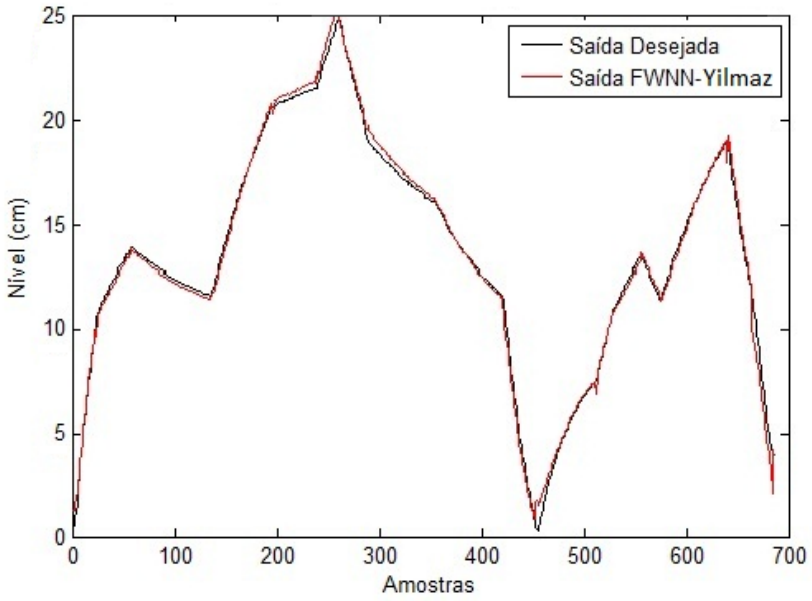


Figura 5.14: Resposta da melhor rede FWNN - Yilmaz & Oysal (2010).

Especificamente na Figura 5.11, percebe-se o bom desempenho da rede proposta, e comparando-se com as Figuras 5.12, 5.13 e 5.14, nota-se que o desempenho foi praticamente o mesmo.

5.4 Sistema Dinâmico Real 2 - Incubadora Neonatal

Por fim, a performance da rede FWNN proposta será avaliada em um sistema de inferência de temperatura e umidade de uma incubadora neonatal. Os dados utilizados neste trabalho foram disponibilizados pelo Grupo de Pesquisa em Automação e Robótica (GPAR) do programa de Pós Graduação em Engenharia Elétrica da Universidade Federal do Ceará (Albuquerque, 2012).

A incubadora neonatal é um dos equipamentos de uma Unidade de Terapia Intensiva Neonatal (UTIN) mais importantes. Neste período de início da vida, as incubadoras são ferramentas fundamentais para reduzir os riscos de mortalidade e doenças. A incubadora proporciona um microclima adequado para recém-nascidos em seus primeiros dias de vida, controlando a temperatura interna e umidade relativa (Araújo Júnior *et al.,* 2013b).

A importância desse equipamento, de modo geral, vem do fato de que recém-nascidos com complicações de saúde e parto prematuro têm grande dificuldade para regular a temperatura do corpo devido a algumas razões, tais como a alta taxa de metabolismo causada por uma condição de doença, o baixo peso e a alta taxa entre a superfície e o volume do seu corpo, o que implica em uma elevada quantidade de energia perdida por quilo quando comparado a um adulto. Neste contexto, uma incubadora neonatal ajuda a cuidar da saúde do recém-nascido. A incubadora neonatal deve apresentar um ambiente térmico de modo a proporcionar condições favoráveis que garantam o mínimo de gasto de energia do recém-nascido, mantendo a temperatura do corpo dentro de uma faixa de segurança (Araújo Júnior *et al.,* 2013b).

Bem como qualquer outro equipamento médico elétrico, uma incubadora neonatal deve ser calibrada periodicamente, pois seu mau funcionamento pode causar sérios danos à saúde do recém-nascido ou até mesmo, em uma situação extrema, levá-lo à morte. A norma técnica IEC 60601-2-19 estabelece especificações operacionais para as incubadoras neonatais, de modo que um ambiente seguro possa ser oferecido para os recém-nascidos. Estas especificações são verificadas através da realização de vários ensaios, incluindo a aplicação de sinais de entrada para o fluxo de temperatura, umidade do ar e a análise do comportamento destas variáveis em pontos específicos no interior da incubadora.

Para realizar os ensaios mencionados, a incubadora é retirada de serviço e um sistema de calibração é instalado. Uma vez que para executar a calibração da incubadora, seu funcionamento deve ser interrompido, e o número de incubadoras é limitado na maioria dos hospitais brasileiros, uma estratégia de manutenção que minimize o tempo de parada da incubadora é bastante útil (Araújo Júnior *et al.,* 2013b).

A ideia básica no presente estudo de caso é o de utilizar a FWNN proposta para inferir os valores de temperatura e umidade apresentados por sensores posicionados no interior do equipamento de acordo com a norma IEC 60601-2-19, apenas usando os sensores já incluídos de fábrica nas incubadoras neonatais comerciais, e verificar sua aplicabilidade como uma nova forma de calibração. Essa inferência possui quatro vantagens significativas sobre o processo de calibração padrão:

- Enquanto que o processo de calibração padrão requer a instalação de cinco sensores de temperatura e um sensor de umidade extras, um procedimento utilizando uma rede neural

para inferir os valores medidos por esses sensores faz com que não seja mais necessária a instalação dos mesmos;

- O processo para verificar se a incubadora está calibrada pode ocorrer com a mesma estando em operação;

- Com o uso dos sistemas de inferência, somente as informações de temperatura e umidade fornecidas pelo componente SHT11, que já se encontra presente na incubadora tradicional, são necessárias para a realização da calibração deste equipamento neonatal;

- A necessidade de calibração é diagnosticada em tempo real e a presença de um profissional especializado não é necessária.

Em termos estruturais, uma incubadora neonatal, Figura 5.15, consiste em uma caixa rígida construída em aço e acrílico, em que um recém nascido pode ser mantido em um ambiente controlado para o cuidado médico. O dispositivo inclui um aquecedor de corrente alternada, um ventilador elétrico para circular o ar aquecido, um recipiente de água utilizada para adicionar umidade no ambiente interno da incubadora, um filtro mecânico, através do qual flui o oxigênio e uma porta de acesso para os serviços de cuidados de enfermagem.

O motor elétrico faz circular o ar para a região interna da incubadora neonatal através de uma entrada de ar na parte inferior do equipamento. Este fluxo de ar influencia o nível de temperatura e umidade no interior da cúpula de incubadora, bem como o nível de oxigênio. O ar é renovado por um exaustor e uma entrada de ar.

Sendo assim, a incubadora consiste em um sistema multivariável, caracterizado pelas não linearidades e interação das variáveis temperatura e umidade. Essas variáveis são reguladas por meio do controle da corrente elétrica que flui através de dois resistores, um responsável por gerar calor e o outro por gerar vapor. Os valores de temperatura e umidade são medidos por um sensor situado na parte inferior da cúpula, perto da entrada de ar. A Figura 5.16 mostra um diagrama esquemático da operação global de uma incubadora neonatal.

Figura 5.15: Incubadora neonatal do GPAR utilizada para validação dos algoritmos.

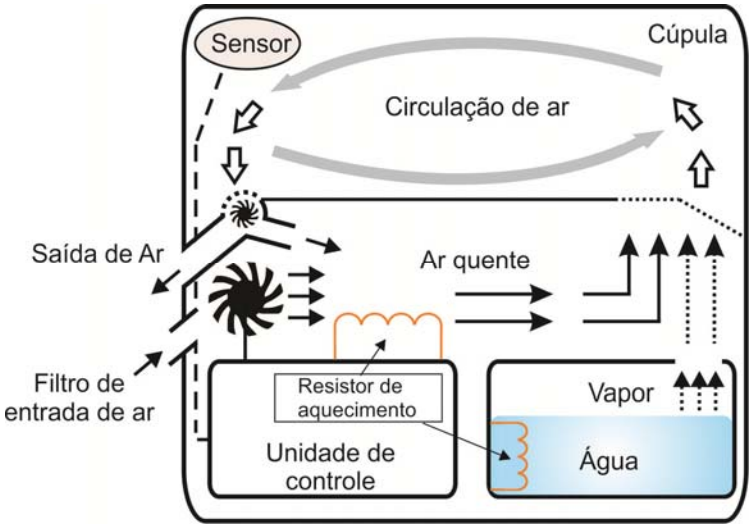


Figure 5.16: Visão geral do modo de operação da incubadora neonatal.

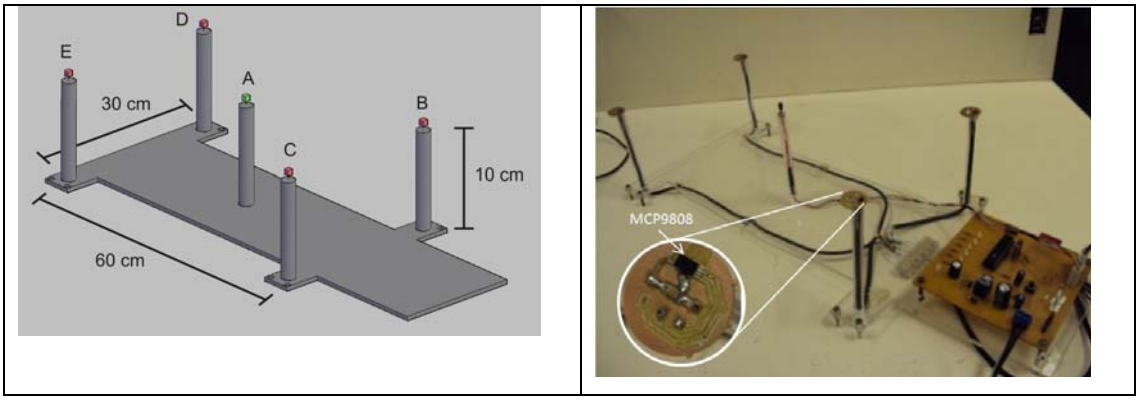
O sensor utilizado para realizar as medições de temperatura e umidade é o SHT11, e é colocado no interior da incubadora. Esse sensor, internamente, integra dois sensores e um dispositivo para processamento de sinal, todos contidos em um micro circuito. Os dois sensores presentes neste dispositivo são: um sensor capacitivo para a medição da umidade relativa e um sensor *band-gap* para medir a temperatura. De acordo com a Sensirion, (2011), fabricante do

SHT11, a tecnologia CMOS aplicada nestes sensores garante uma excelente confiabilidade e estabilidade a longo prazo. Ambos os sensores são acoplados a um conversor analógico/digital de 14 bits e um circuito de interface serial. Isto resulta na geração de um sinal de qualidade superior, com um rápido tempo de resposta e insensibilidade a perturbações externas.

Este sensor já está presente, de fábrica, na incubadora, e no caso do processo de inferência, os valores de temperatura e umidade medidos por ele serão utilizados no treinamento da rede proposta. Portanto, de modo a se realizar a coleta de dados para se treinar as redes, tanto a FWNN proposta como as outras redes utilizadas na comparação, foi feita inicialmente a implementação da estrutura necessária para realizar o procedimento de calibração padronizado e de um sistema digital.

Este sistema é composto por uma plataforma de aquisição de dados com base em um microprocessador PIC18LF2620, e um link de rede sem fio para comunicação de dados com base no protocolo ZigBee. O principal objetivo do sistema embarcado é proporcionar a aquisição de dados e um meio de comunicação do PC com a incubadora.

Para calibrar a incubadora utilizando o sistema embarcado citado, devem-se seguir as especificações determinadas pela norma IEC 60601-2-19. Esta norma padrão especifica o comportamento da temperatura em pontos específicos dentro da incubadora. Estes pontos são apresentados na Figura 5.17 e denominados como pontos A, B, C, D e E. Além da temperatura, a umidade relativa deve também ser avaliada no ponto central A.



(a) (b)

Figura 5.17: (a) Esquema da base do sistema de aquisição de dados, (b) Desenho da base do sistema de aquisição de dados.

A parte mecânica do sistema de calibração é composta por uma base rígida feita de acrílico, cinco suportes para sensores dispostos de acordo com a recomendação da norma e uma placa de controle. No pontos B, C, D e E são posicionados os sensores de temperatura de tipo MCP9808 (Microchip, 2011). O ponto A integra um sensor de temperatura e umidade do tipo SHT11.

A base de acrílico com os sensores para fins de calibração deve ser posicionada no interior da incubadora durante os ensaios e removida da incubadora quando esta entra em funcionamento. O sensor SHT11, que integra a estrutura original incubadora comercial, é mantido durante todo o teste.

Neste processo de calibração, as correntes elétricas que passam pelos resistores utilizados para gerar calor e umidade no interior da incubadora são ajustadas até que a umidade no ponto A e as temperaturas em todos os pontos - de A a E - estejam com seus valores de acordo com o especificado pela norma.

Neste estudo de caso, a rede FWNN deve fazer a inferência da umidade e das temperaturas nos pontos especificados pela norma. Os sistemas inferenciais (ou *soft sensors*) representam uma abordagem atrativa para estimar variáveis primárias do processo, particularmente quando os sensores convencionais não estão disponíveis ou quando o seu elevado custo ou limitações técnicas dificultam a sua utilização. Estes sistemas fazem uso de variáveis facilmente disponíveis para estimar variáveis primárias de interesse (Brosilow & Joseph, 2002).

Os *soft sensors* são algoritmos matemáticos que têm sido usados em uma ampla gama de aplicações na indústria, tais como a supervisão, controle e otimização de processos, devido às suas vantagens em relação a problemas de medição apresentados por alguns sensores físicos. Esses sistemas fornecem medidas mais rápidas, com maior precisão e confiabilidade a um custo mais baixo de desenvolvimento, implementação e manutenção.

De acordo com Warne *et al*. (2004), apesar da utilização ampla de modelos inferenciais em aplicações práticas, apenas poucas técnicas são discutidas na literatura. De modo geral, existem essencialmente três abordagens para a construção de modelos de inferência: modelagem mecanicista, regressão estatística e modelagem por meio de técnicas de inteligência artificial.

Neste contexto, espera-se que os sistemas de inferência possam ser utilizados para a avaliação das condições de incubação e a calibração de um modo otimizado, permitindo que esse sistema possa substituir o tradicional mecanismo de calibração baseado em sensores instalados no interior da incubadora.

Para o processo de inferência, a rede FWNN, assim como as outras redes utilizadas com o intuito comparativo, deve ter como entradas os valores de umidade e temperatura (variáveis secundárias) que são instantaneamente medidos pelo sensor SHT11. Assim, após o treinamento, as redes devem ser capazes de estimar as variáveis primárias do sistema, que são: umidade no ponto A e a temperatura dos pontos A, B, C, D e E.

A estrutura geral da rede FWNN quando aplicada na inferência neste estudo de caso é mostrada na Figura 5.18, em que *Ts* e *Us* são as medições de temperatura e umidade, respectivamente, fornecidas pelo sensor SHT11 já presente na incubadora (localizado em um ponto chamado de ponto de saída), *d* é o atraso do sistema de transporte e, finalmente, m_1 e m_2 são os regressores aplicados às variáveis secundárias.

Na saída da estrutura, $\hat{y}_i$ é a estimativa da *i*-ésima variável primária fornecida pelo sistema de inferência, de acordo com a Tabela 5.4.

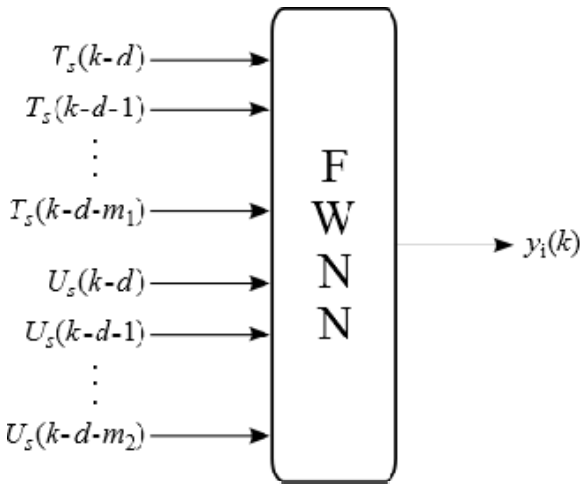


Figura 5.18: Estrutura da FWNN aplicada na inferência.

Índice *i*	Variável Primária y_i	Descrição
1	U_A	Umidade no ponto A
2	T_A	Temperatura no ponto A
3	T_B	Temperatura no ponto B
4	T_C	Temperatura no ponto C
5	T_D	Temperatura no ponto D
6	T_E	Temperatura no ponto E

Tabela 5.4: Variáveis primárias da incubadora neonatal.

Para o treinamento e validação da rede proposta, foram coletadas 900 amostras de temperatura e umidade nos pontos A, B, C, D e E, com período de amostragem de 24 segundos. Os dados coletados foram divididos em dois grupos, um para treinamento, contendo 750 amostras, e 150 amostras para validação. As amostras foram coletadas através da alteração da corrente elétrica aplicada em dois resistores responsáveis pela geração de calor e vapor na parte interna da incubadora.

Os valores de umidade (U_S) e da temperatura (T_S) foram usados como dados de entrada na rede, com o objetivo de inferir os valores medidos pelos sensores de umidade no ponto A (U_A) e temperatura presente nos pontos A, B, C, D e E (T_A, T_B, T_C, T_D, T_E).

As estruturas neurais utilizadas nos sistemas de inferência deste estudo de caso estão presentes na Figura 5.19. Na Figura 5.19 (a), utilizou-se $T_S(k)$, $U_S(k)$ e $U_S(k-1)$ para a estimativa da umidade $\hat{U}_A(k)$. Na Figura 5.19 (b), são utilizadas $U_S(k)$, $T_S(k)$, e $T_S(k-1)$, para a estimativa da temperatura $\hat{T}_i(k)$, em que *i* se refere às temperaturas no pontos A, B, C. D e E.

Esses valores, mostrados na Figura 5.20, são usados no treinamento da rede, realizado por meio do algoritmo de *backpropagation*, e foram utilizados também no procedimento de validação do modelo neural obtido, a fim de verificar a possibilidade de generalização do modelo obtido.

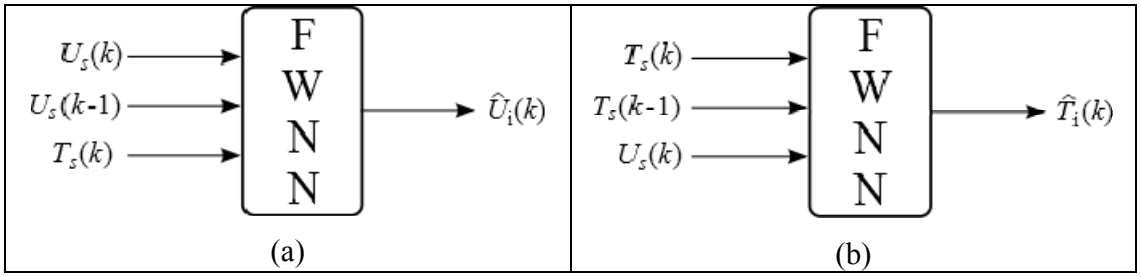


Figura 5.19: Estrutura de treinamento das redes: (a) umidade, e (b) temperaturas.

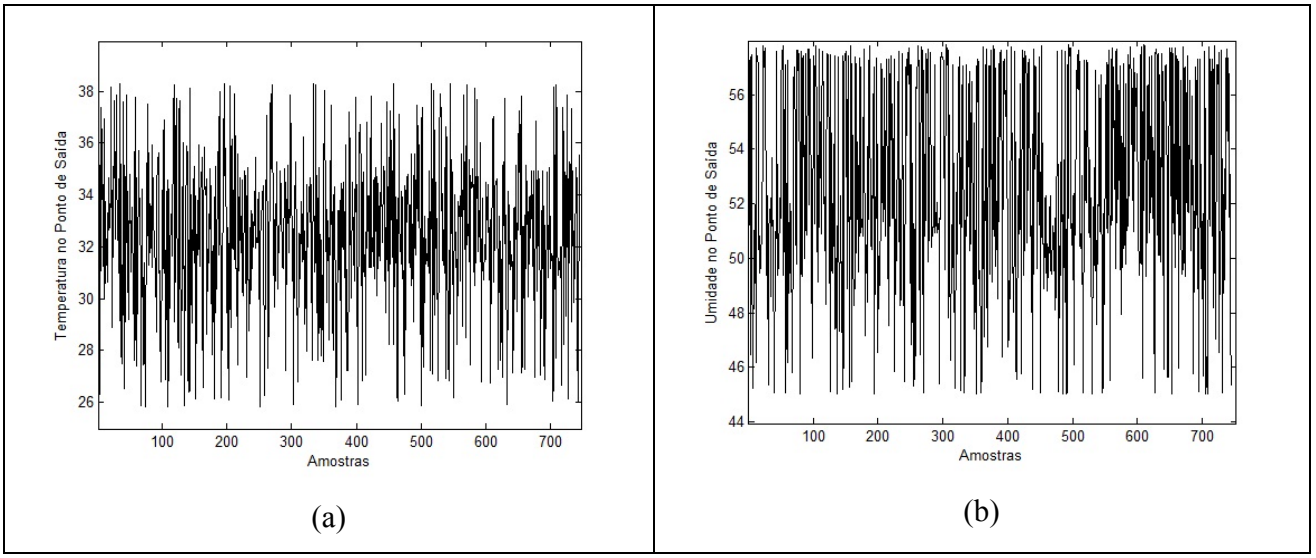


Figura 5.20: Sinais utilizados como entradas da rede: (a) Temperatura no ponto de saída da incubadora – *US*, (b) Umidade no ponto de saída da rede - *TS*.

Pode-se observar também, com base na Figura 5.21, que o erro médio quadrático segue a tendência de baixar e estabilizar em um valor bastante próximo a zero, mostrando uma boa estabilidade na convergência do treinamento, apesar da inicialização aleatória dos parâmetros adaptativos da rede.

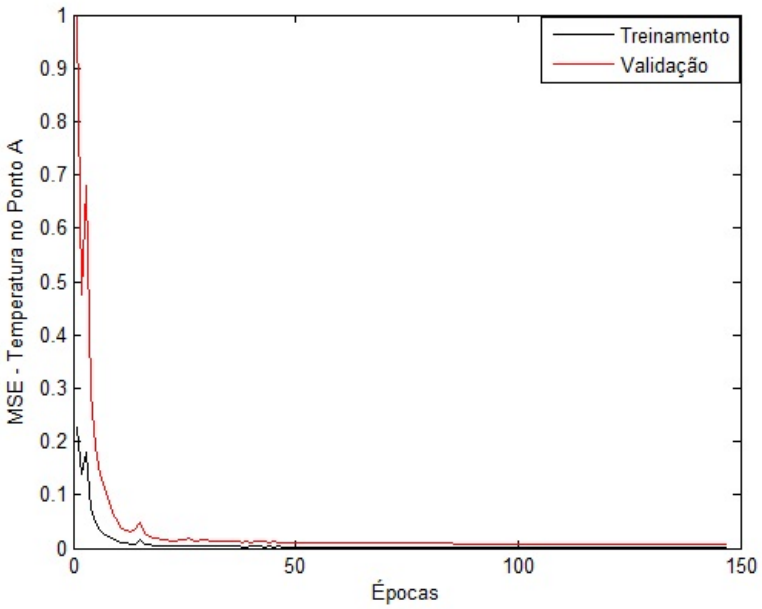


Figura 5.21: Erro médio quadrático de treinamento da rede FWNN-P.

Após o processo de treinamento e validação tanto da rede FWNN proposta como das outras redes utilizadas na comparação, o próximo objetivo é o de avaliar o desempenho das redes em termos dos vários parâmetros que estão sendo utilizados para avaliar do desempenho das redes.

Em termos numéricos, para melhor avaliação do desempenho da rede, foram montadas as Tabelas 5.5 e 5.6. Estas tabelas são similares à apresentada no estudo de caso do tanque, e mostram o número de parâmetros, os quais indicam a dimensão da rede, o número médio de épocas e os resultados das validações em termos de média e variância do MSE. Assim como no estudo de caso do tanque de múltiplas seções, a variância e os valores médios de épocas de treinamento e MSE apresentados foram obtidos a partir da realização de 20 treinamentos. Estes valores podem ser visualizados nas duas ultimas colunas das Tabelas 5.5 e 5.6.

Esses resultados foram conseguidos realizando-se o treinamento de todas as redes com as mesmas configurações testadas no estudo de caso anterior. Ou seja, variou-se o número de funções de pertinência e o número de regras. Esses valores, assim como o número de parâmetros ajustados, foram os mesmos apresentados na Tabela 5.3.

Variáveis Inferidas	Modelos	Número de Parâmetros	Número de Funções de Pertinência	Número de Regras	Número Médio de Épocas	Média do MSE de Validação	Variância do MSE
U_A	FWNN Lu 1	65	5	5	172	$7{,}87 \times 10^{-3}$	$3{,}25 \times 10^{-7}$
	FWNN Lu 2	91	7	7	168	$7{,}262 \times 10^{-3}$	$2{,}58 \times 10^{-6}$
	FWNN Lu 3	130	10	10	165	$6{,}681 \times 10^{-3}$	$3{,}51 \times 10^{-7}$
	FWNN Abiyev 1	65	5	5	184	$6{,}595 \times 10^{-3}$	$9{,}19 \times 10^{-7}$
	FWNN-Abiyev 2	91	7	7	184	$5{,}844 \times 10^{-3}$	$9{,}92 \times 10^{-7}$
	FWNN-Abiyev 3	130	10	10	184	$5{,}11 \times 10^{-3}$	$1{,}27 \times 10^{-7}$
	FWNN-Yilmaz 1	84	2	8	180	$6{,}94 \times 10^{-3}$	$6{,}75 \times 10^{-7}$
	FWNN-Yilmaz 2	261	3	27	173	$5{,}1 \times 10^{-3}$	$1{,}77 \times 10^{-7}$
	ANFIS 1	42	2	4	185	$1{,}068 \times 10^{-2}$	$2{,}37 \times 10^{-9}$
	ANFIS 2	108	3	9	185	$6{,}81 \times 10^{-3}$	$6{,}79 \times 10^{-8}$
	WNN	60	6 Neurônios	-	200	$1{,}04 \times 10^{-2}$	$3{,}45 \times 10^{-6}$
	MLP	24	6 Neurônios	-	215	$1{,}896 \times 10^{-2}$	$5{,}32 \times 10^{-5}$
	FWNN Proposta 1	60	2	8	182	$7{,}707 \times 10^{-3}$	$2{,}77 \times 10^{-7}$
	FWNN Proposta 2	180	3	27	170	$5{,}4 \times 10^{-3}$	$2{,}04 \times 10^{-7}$
T_A	FWNN Lu 1	65	5	5	171	$7{,}157 \times 10^{-3}$	$2{,}65 \times 10^{-6}$
	FWNN Lu 2	91	7	7	170	$6{,}66 \times 10^{-3}$	$1{,}9 \times 10^{-6}$
	FWNN Lu 3	130	10	10	170	$6{,}06 \times 10^{-3}$	$5{,}87 \times 10^{-7}$
	FWNN Abiyev 1	65	5	5	185	$6{,}28 \times 10^{-3}$	$1{,}59 \times 10^{-6}$
	FWNN-Abiyev 2	91	7	7	183	$5{,}566 \times 10^{-3}$	$2{,}28 \times 10^{-6}$
	FWNN-Abiyev 3	130	10	10	183	$4{,}855 \times 10^{-3}$	$8{,}53 \times 10^{-7}$
	FWNN-Yilmaz 1	84	2	8	177	$6{,}58 \times 10^{-3}$	$1{,}86 \times 10^{-6}$
	FWNN-Yilmaz 2	261	3	27	178	$5{,}045 \times 10^{-3}$	$3{,}02 \times 10^{-7}$
	ANFIS 1	42	2	4	184	$1{,}17 \times 10^{-2}$	$9{,}76 \times 10^{-8}$
	ANFIS 2	108	3	9	185	$6{,}191 \times 10^{-3}$	$1{,}0 \times 10^{-10}$
	WNN	60	6 Neurônios	-	200	$9{,}69 \times 10^{-3}$	$6{,}23 \times 10^{-6}$
	MLP	24	6 Neurônios	-	211	$1{,}3 \times 10^{-2}$	$1{,}828 \times 10^{-5}$
	FWNN Proposta 1	60	2	8	173	$6{,}723 \times 10^{-3}$	$1{,}1 \times 10^{-6}$
	FWNN Proposta 2	180	3	27	167	$5{,}011 \times 10^{-3}$	$6{,}35 \times 10^{-7}$
T_B	FWNN Lu 1	65	5	5	164	$2{,}716 \times 10^{-3}$	$3{,}77 \times 10^{-7}$
	FWNN Lu 2	91	7	7	182	$1{,}984 \times 10^{-3}$	$3{,}5 \times 10^{-7}$
	FWNN Lu 3	130	10	10	180	$1{,}702 \times 10^{-3}$	$8{,}88 \times 10^{-8}$
	FWNN Abiyev 1	65	5	5	185	$1{,}849 \times 10^{-3}$	$3{,}14 \times 10^{-7}$
	FWNN-Abiyev 2	91	7	7	177	$1{,}394 \times 10^{-3}$	$3{,}07 \times 10^{-8}$
	FWNN-Abiyev 3	130	10	10	179	$1{,}307 \times 10^{-3}$	$2{,}21 \times 10^{-8}$
	FWNN-Yilmaz 1	84	2	8	180	$1{,}991 \times 10^{-3}$	$4{,}08 \times 10^{-7}$
	FWNN-Yilmaz 2	261	3	27	173	$1{,}299 \times 10^{-3}$	$1{,}94 \times 10^{-8}$
	ANFIS 1	42	2	4	185	$2{,}92 \times 10^{-3}$	$6{,}0 \times 10^{-10}$
	ANFIS 2	108	3	9	182	$1{,}4 \times 10^{-3}$	$8{,}57 \times 10^{-10}$
	WNN	60	6 Neurônios	-	200	$3{,}541 \times 10^{-3}$	$9{,}24 \times 10^{-7}$
	MLP	24	6 Neurônios	-	215	$3{,}426 \times 10^{-2}$	$7{,}63 \times 10^{-4}$
	FWNN Proposta 1	60	2	8	175	$1{,}951 \times 10^{-3}$	$1{,}59 \times 10^{-7}$
	FWNN Proposta 2	180	3	27	159	$1{,}711 \times 10^{-3}$	$1{,}3 \times 10^{-7}$

Tabela 5.5: Resultados da análise das estruturas aplicadas na inferência.

Variáveis Inferidas	Modelos	Número de Parâmetros	Número de Funções de Pertinência	Número de Regras	Número Médio de Épocas	MSE de Validação	Variância do MSE
T_C	FWNN Lu 1	65	5	5	175	$2{,}79 \times 10^{-3}$	$1{,}72 \times 10^{-6}$
	FWNN Lu 2	91	7	7	165	$2{,}50 \times 10^{-3}$	$5{,}54 \times 10^{-7}$
	FWNN Lu 3	130	10	10	174	$2{,}021 \times 10^{-3}$	$1{,}33 \times 10^{-7}$
	FWNN Abiyev 1	65	5	5	172	$2{,}063 \times 10^{-3}$	$2{,}1 \times 10^{-6}$
	FWNN-Abiyev 2	91	7	7	180	$1{,}654 \times 10^{-3}$	$1{,}56 \times 10^{-7}$
	FWNN-Abiyev 3	130	10	10	177	$1{,}447 \times 10^{-3}$	$9{,}03 \times 10^{-8}$
	FWNN-Yilmaz 1	84	2	8	183	$1{,}681 \times 10^{-3}$	$7{,}12 \times 10^{-8}$
	FWNN-Yilmaz 2	261	3	27	170	$1{,}419 \times 10^{-3}$	$1{,}51 \times 10^{-8}$
	ANFIS 1	42	2	4	185	$4{,}25 \times 10^{-3}$	$3{,}22 \times 10^{-10}$
	ANFIS 2	108	3	9	185	$1{,}698 \times 10^{-3}$	$1{,}76 \times 10^{-7}$
	WNN	60	6 Neurônios	-	200	$3{,}265 \times 10^{-3}$	$1{,}82 \times 10^{-6}$
	MLP	24	6 Neurônios	-	215	$6{,}7 \times 10^{-3}$	$6{,}43 \times 10^{-5}$
	FWNN Proposta 1	60	2	8	180	$1{,}929 \times 10^{-3}$	$1{,}87 \times 10^{-7}$
	FWNN Proposta 2	180	3	27	177	$1{,}465 \times 10^{-3}$	$2{,}25 \times 10^{-8}$
T_D	FWNN Lu 1	65	5	5	175	$1{,}29 \times 10^{-3}$	$7{,}47 \times 10^{-8}$
	FWNN Lu 2	91	7	7	160	$1{,}096 \times 10^{-3}$	$6{,}65 \times 10^{-8}$
	FWNN Lu 3	130	10	10	184	$9{,}69 \times 10^{-4}$	$1{,}16 \times 10^{-8}$
	FWNN Abiyev 1	65	5	5	180	$1{,}593 \times 10^{-3}$	$3{,}22 \times 10^{-7}$
	FWNN-Abiyev 2	91	7	7	162	$9{,}23 \times 10^{-4}$	$1{,}98 \times 10^{-8}$
	FWNN-Abiyev 3	130	10	10	183	$8{,}7 \times 10^{-4}$	$5{,}72 \times 10^{-9}$
	FWNN-Yilmaz 1	84	2	8	177	$1{,}03 \times 10^{-3}$	$6{,}05 \times 10^{-8}$
	FWNN-Yilmaz 2	261	3	27	180	$7{,}0 \times 10^{-4}$	$3{,}41 \times 10^{-9}$
	ANFIS 1	42	2	4	185	$2{,}349 \times 10^{-3}$	$6{,}6 \times 10^{-10}$
	ANFIS 2	108	3	9	185	$6{,}65 \times 10^{-4}$	$2{,}5 \times 10^{-11}$
	WNN	60	6 Neurônios	-	200	$4{,}812 \times 10^{-3}$	$6{,}91 \times 10^{-6}$
	MLP	24	6 Neurônios	-	214	$7{,}367 \times 10^{-3}$	$7{,}68 \times 10^{-5}$
	FWNN Proposta 1	60	2	8	184	$1{,}11 \times 10^{-3}$	$6{,}86 \times 10^{-8}$
	FWNN Proposta 2	180	3	27	173	$8{,}27 \times 10^{-4}$	$2{,}28 \times 10^{-8}$
T_E	FWNN Lu 1	65	5	5	165	$1{,}76 \times 10^{-3}$	$8{,}97 \times 10^{-8}$
	FWNN Lu 2	91	7	7	173	$1{,}361 \times 10^{-3}$	$3{,}9 \times 10^{-8}$
	FWNN Lu 3	130	10	10	184	$1{,}297 \times 10^{-3}$	$2{,}85 \times 10^{-8}$
	FWNN Abiyev 1	65	5	5	184	$1{,}267 \times 10^{-3}$	$1{,}24 \times 10^{-8}$
	FWNN-Abiyev 2	91	7	7	180	$1{,}346 \times 10^{-3}$	$6{,}13 \times 10^{-8}$
	FWNN-Abiyev 3	130	10	10	183	$1{,}194 \times 10^{-4}$	$1{,}46 \times 10^{-8}$
	FWNN-Yilmaz 1	84	2	8	178	$1{,}229 \times 10^{-3}$	$1{,}81 \times 10^{-8}$
	FWNN-Yilmaz 2	261	3	27	172	$9{,}84 \times 10^{-4}$	$1{,}37 \times 10^{-8}$
	ANFIS 1	42	2	4	173	$3{,}393 \times 10^{-3}$	$2{,}47 \times 10^{-8}$
	ANFIS 2	108	3	9	182	$1{,}022 \times 10^{-3}$	$2{,}21 \times 10^{-9}$
	WNN	60	6 Neurônios	-	200	$5{,}82 \times 10^{-3}$	$5{,}43 \times 10^{-6}$
	MLP	24	6 Neurônios	-	212	$5{,}947 \times 10^{-3}$	$5{,}13 \times 10^{-5}$
	FWNN Proposta 1	60	2	8	180	$1{,}285 \times 10^{-3}$	$1{,}45 \times 10^{-8}$
	FWNN Proposta 2	180	3	27	173	$1{,}054 \times 10^{-3}$	$1{,}0 \times 10^{-8}$

Tabela 5.6: Resultados da análise das estruturas aplicadas na inferência.

Os resultados que mostraram valores mais elevados em termos de média do MSE foram os obtidos a partir da inferência da temperatura e da umidade no ponto A. A umidade neste ponto apresenta uma faixa de operação mais ampla que os valores de temperatura tanto neste mesmo ponto como nos demais. Com relação à temperatura no ponto A, esta varia dentro de uma faixa de valores menor que o das demais redes, variando de aproximadamente 25ºC a 33ºC, contendo, no entanto, alguns picos com valores acima de 36ºC. Neste ponto, a rede proposta apresentou MSE médio para temperatura de 5,011 x 10^{-3}, e para a umidade apresentou o valor de 5,4 x 10^{-3}.

Com relação aos valores de temperatura nos demais pontos, as temperaturas variam, de forma bastante uniforme, dentro da faixa de 25ºC a 39ºC, e nesses pontos todas as redes apresentaram valores de MSE menor que no ponto A. Especificamente a respeito do desempenho da rede FWNN proposta nos pontos B, C, D e E, a mesma teve desempenho bastante satisfatório, com a média do erro médio quadrático variando entre 8,27 x 10^{-4} (T_D) e 1,711 x 10^{-3} (T_B). Esse desempenho foi conseguido com a rede sendo treinada com 27 regras.

Os resultados para as demais redes FWNN também podem ser considerados satisfatórios. O melhor valor foi apresentado pela presente em Yilmaz & Oysal (2010), com 27 regras, na inferência da temperatura T_C, com a média do MSE sendo 7,0 x 10^{-4}. Já o pior valor foi obtido pela rede proposta por Lu (2011), com 5 regras, quando aplicada na inferência da temperatura T_C, cujo MSE médio foi de 2,79 x 10^{-3}.

Nas Tabelas 5.5 e 5.6, também são mostradas a variância do erro das respostas de todas as redes. Como pode ser visto nestas tabelas, a variância apresentada pela rede ANFIS foi menor em boa parte das variáveis inferidas. No entanto, essa rede não possui nenhum parâmetro inicializado aleatoriamente, como já citado no estudo de caso anterior, o que contribui para que haja uma menor variação dos valores do MSE obtidos em cada treinamento.

A variância apresentada pela rede FWNN proposta, como por exemplo, 2,04 x 10^{-7} relativo à umidade no ponto A, é razoavelmente próxima da variância apresentada pelas outras redes FWNN para a umidade neste ponto. Esse mesmo comportamento ocorreu para a variância do MSE das temperaturas estimadas nos pontos de A até E.

Em termos destes parâmetros, a rede FWNN proposta apresentou resultados muito próximos dos obtidos pela rede proposta por Yilmaz & Oysal (2010). A Figura 5.22 mostra os histogramas para o MSE, obtidos a partir dos 20 treinamentos realizados com essas duas redes, para a umidade no ponto A e a temperatura neste mesmo ponto. Os histogramas para as

temperaturas inferidas nos demais pontos (B, C, D e E), não são apresentados, pois estes seguem um padrão de comportamento similar aos ilustrados na Figura 5.22.

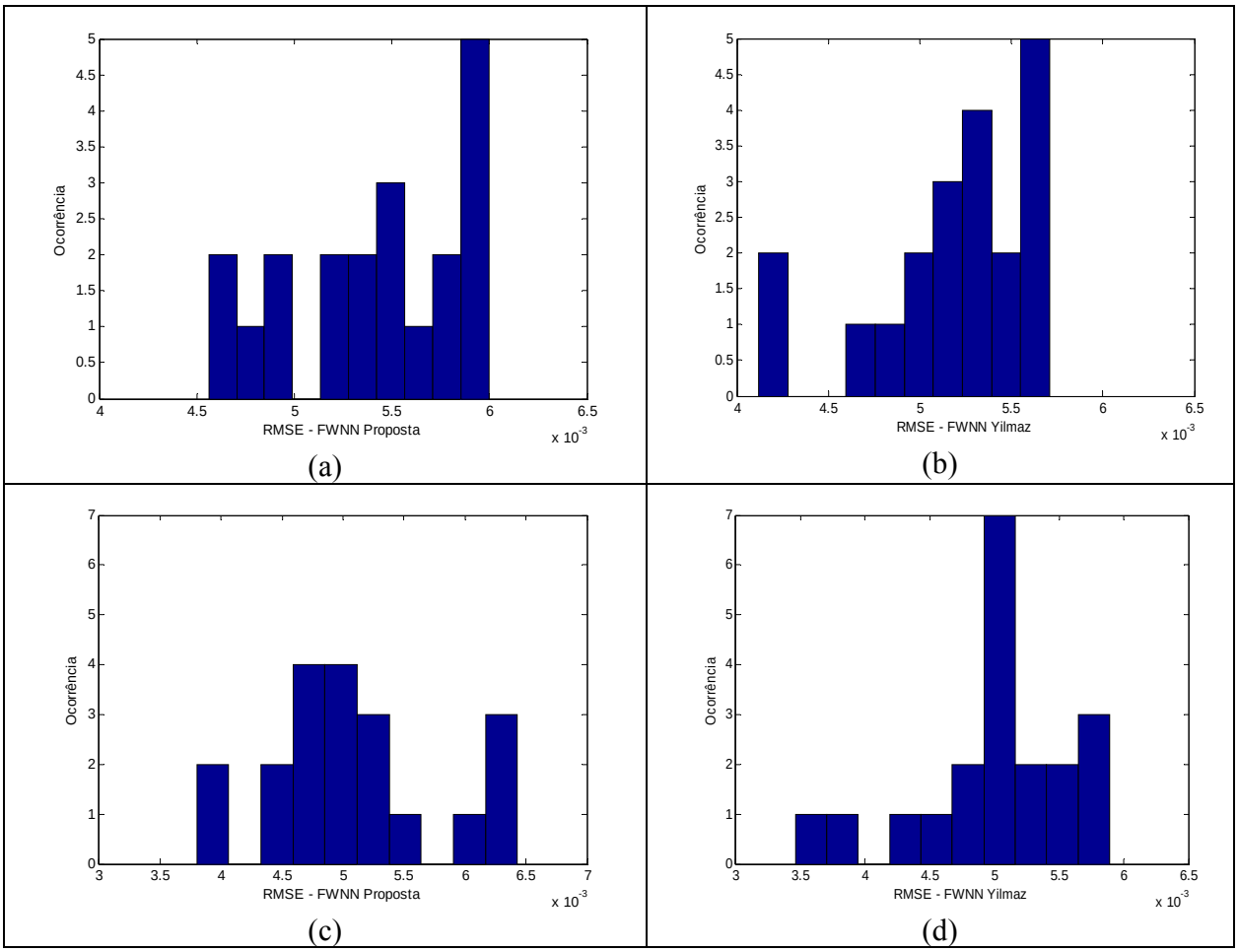


Figura 5.22: Histograma dos MSEs obtidos na inferência da umidade e temperatura no ponto A, para as redes (a) e (c) FWNN – Proposta, (b) e (d) FWNN – Yilmaz & Oysal (2010).

.

Por fim, no geral, os melhores resultados foram obtidos pelas redes propostas em Abiyev & Kaynak (2008b) e por Lu (2011), utilizando 10 regras, e pela rede presente em Yilmaz & Oysal (2010) e pela FWNN proposta, utilizando 27 regras. Os resultados gráficos das validações da rede FWNN proposta e das redes presentes em Yilmaz & Oysal (2010), por Abiyev & Kaynak (2008b) e por Lu (2011), são apresentados nas Figuras 5.23 (a-f) até 5.26 (a-f).

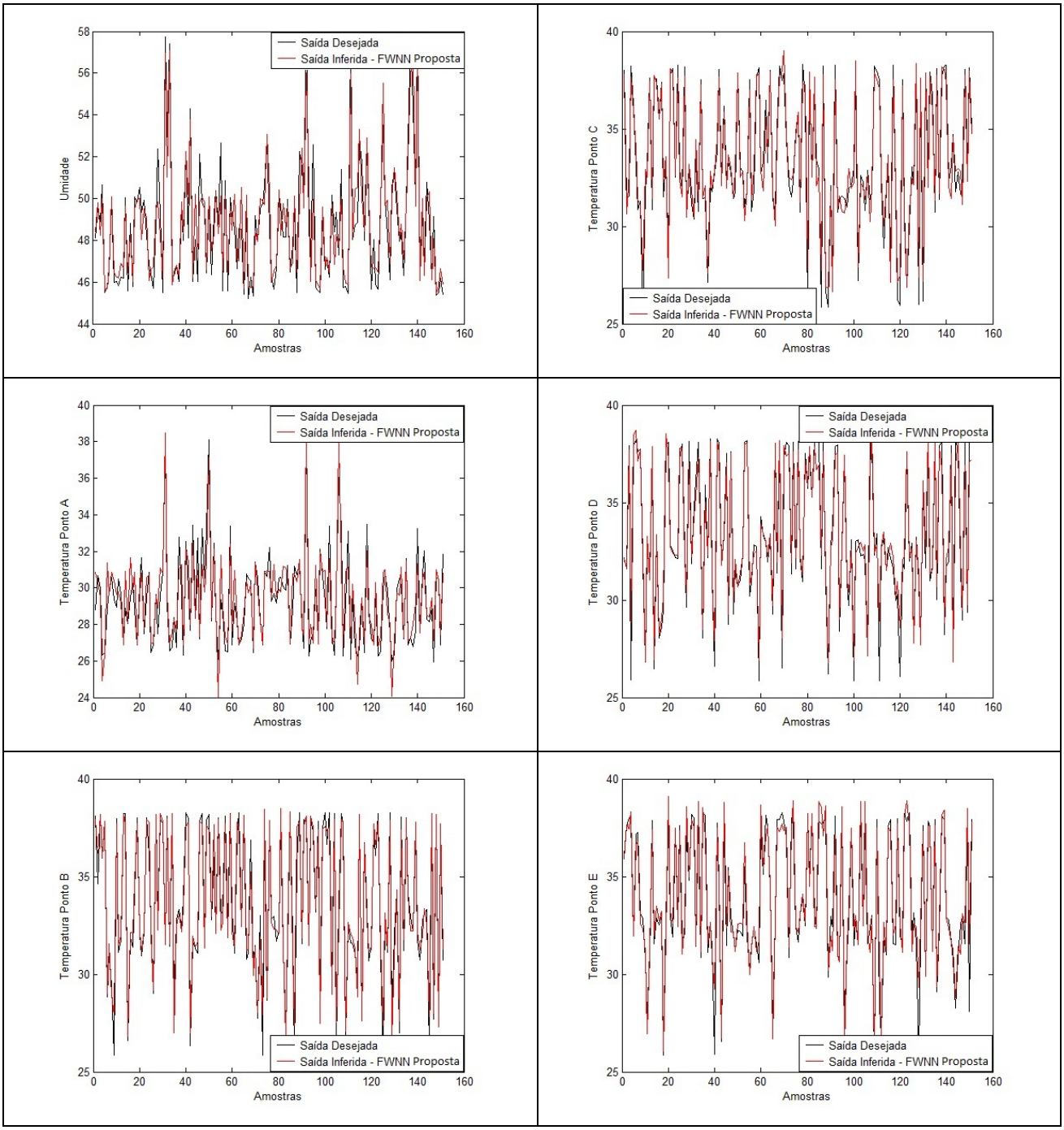


Figura 5.23: Resposta do melhor modelo obtido pela FWNN Proposta: (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E.

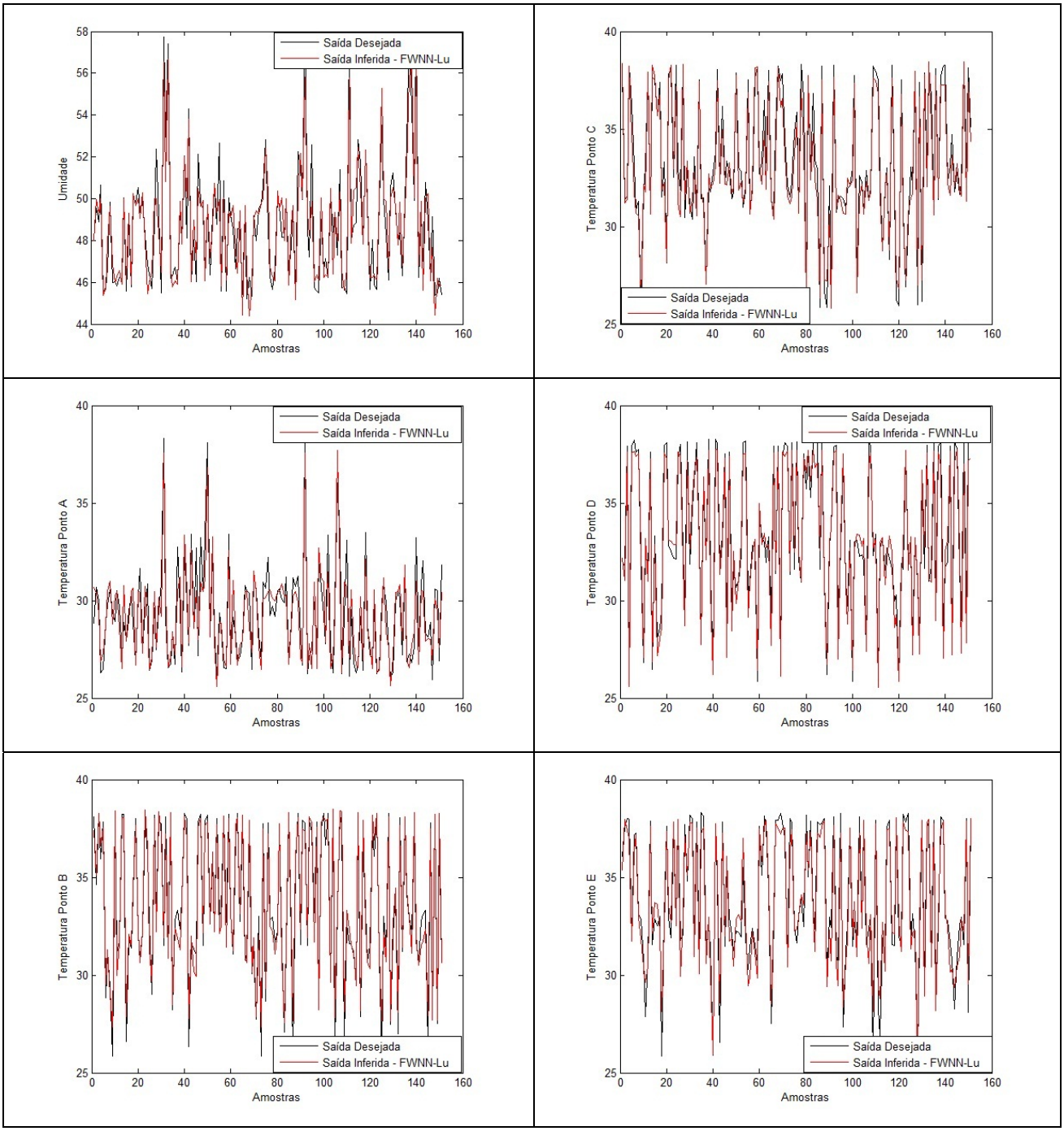


Figura 5.24: Resposta do melhor modelo obtido pela FWNN presente em Lu (2011): (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E.

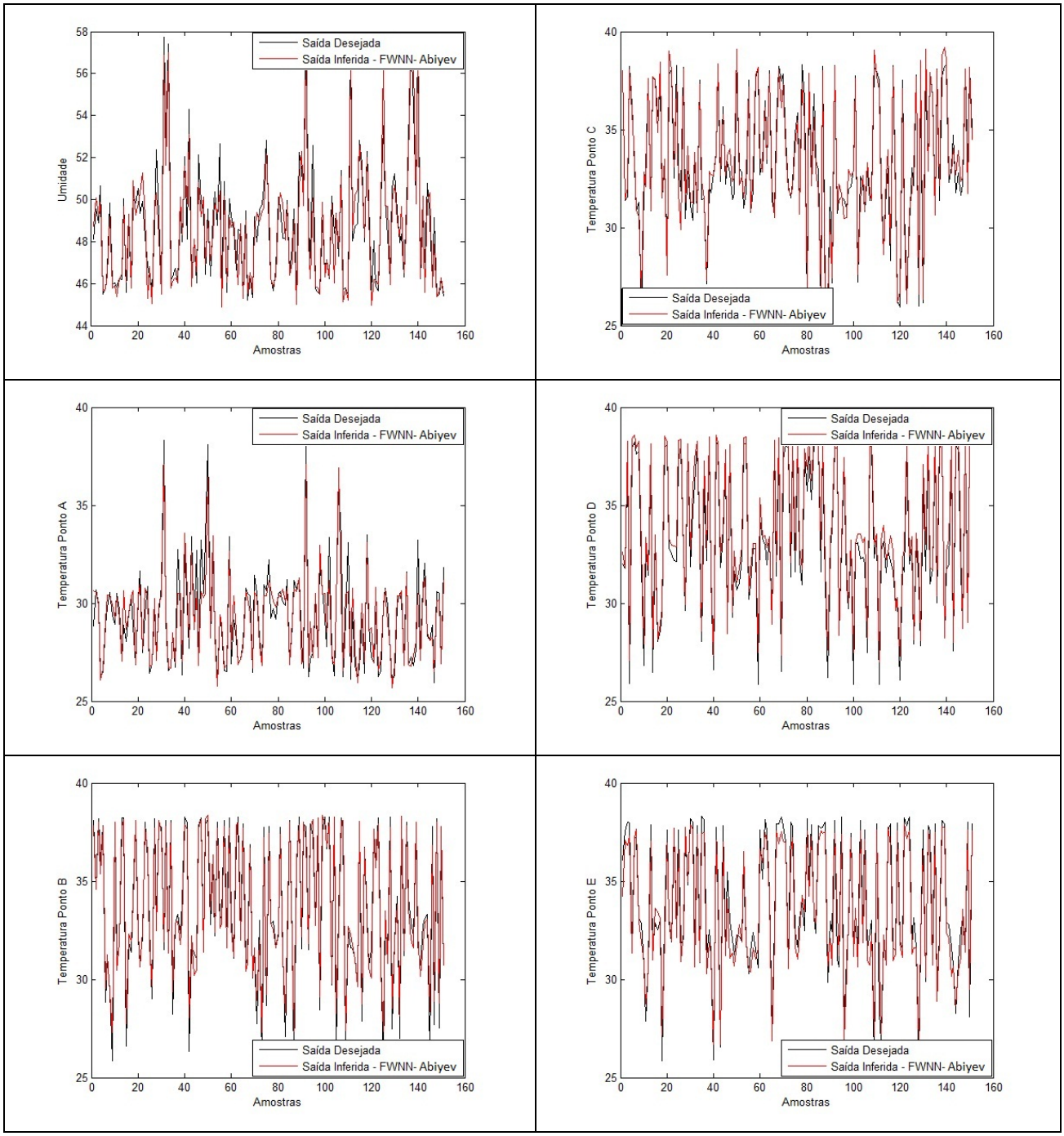


Figura 5.25: Resposta do melhor modelo obtido pela FWNN presente em Abiyev & Kaynak (2008b): (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E.

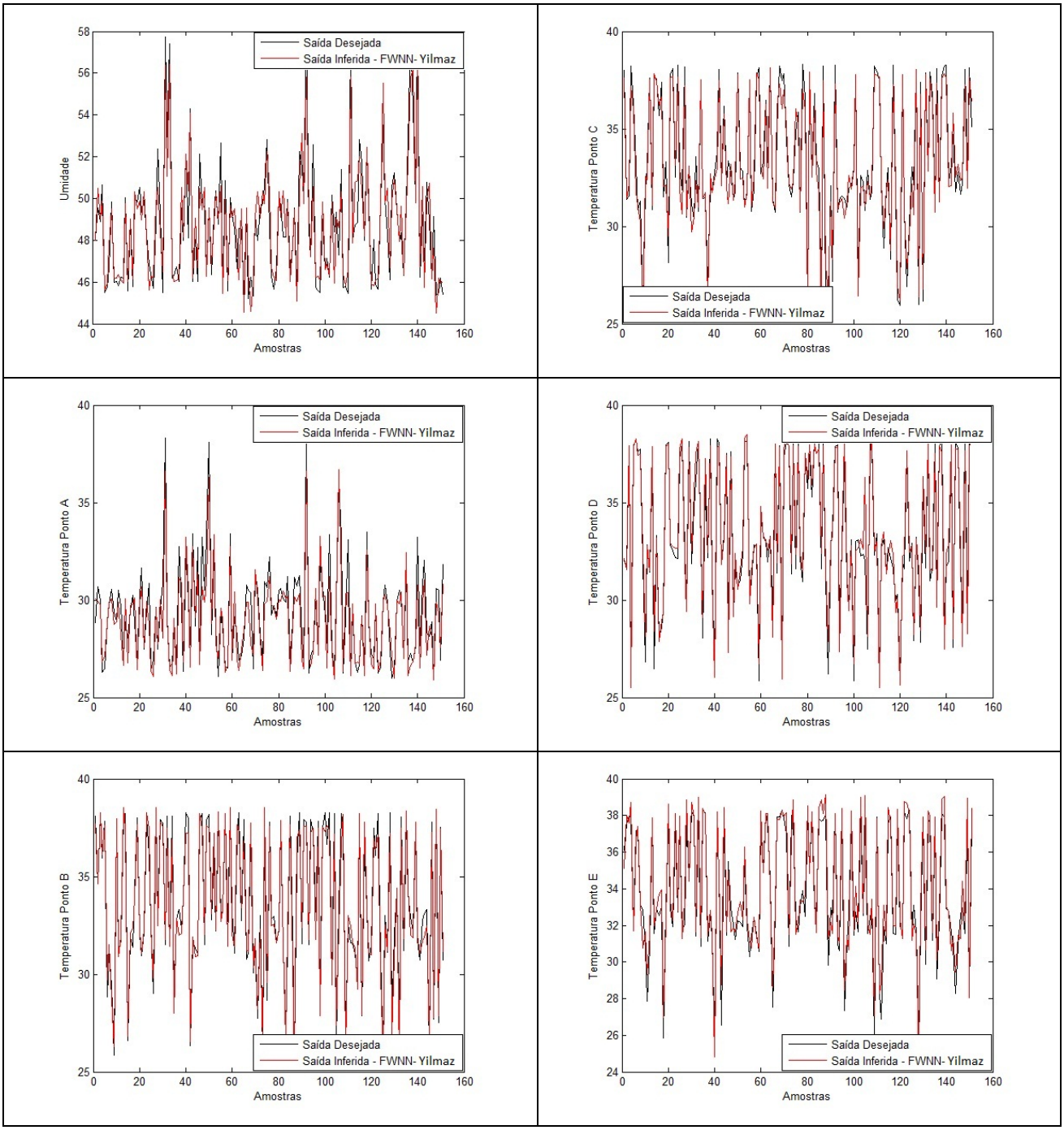


Figura 5.26: Resposta do melhor modelo obtido pela FWNN presente em Yilmaz & Oysal (2010): (a) umidade no ponto A, (b) temperatura no ponto A, (c) temperatura no ponto B, (d) temperatura no ponto C, (e) temperatura no ponto D e (f) temperatura no ponto E.

A partir destes resultados, pode-se verificar que todas as redes FWNN conseguiram inferir adequadamente os valores desejados para todas as variáveis a serem inferidas. Em se tratando da rede proposta, a mesma, apesar de ser mais simples que as demais, conseguiu inferir adequadamente a umidade e todas as temperaturas dadas pelos sensores posicionados em todos os pontos especificados pela norma. Os valores inferidos ficaram sempre próximos dos valores desejados, mesmo nas extremidades. Esse fato é importante por que mostra que a rede conseguiu generalizar bem a dinâmica da umidade e das temperaturas. Dessa forma, pode-se utilizar a mesma na inferência das variáveis relacionadas ao ambiente interno da incubadora.

Capítulo 6

Conclusões e Perspectivas

Neste trabalho foi apresentada uma técnica alternativa para identificação de sistemas não lineares. Esta nova técnica pode ser considerada uma variação das redes *Fuzzy Wavelet Neural Network* e foi obtida por meio de uma modificação, visando tornar a sua estrutura mais simples em relação às demais redes FWNN encontradas na literatura.

No desenvolvimento deste trabalho, inicialmente, procurou-se informar, passo a passo, as principais características da rede proposta, de modo a deixar claras as diferenças para as demais redes FWNN, e visando também, facilitar o processo de implementação da mesma. Além disso, para ajudar na melhor compreensão da alteração proposta, o trabalho mostrou a evolução das redes FWNN, desde uma arquitetura mais clássica, no caso a rede MLP, passando pelas redes WNN e ANFIS. Em seguida, foi sugerido um algoritmo de aprendizagem clássico para buscar a configuração ótima para a estrutura proposta.

Em termos práticos, a fim de investigar se a rede proposta mantém a boa capacidade de generalização apresentada pelas outras redes FWNN encontradas na literatura, extensos testes de identificação de sistemas não lineares foram realizados. Para isso, foram utilizadas quatro plantas: foi feita a identificação de duas plantas simuladas e, em seguida, visando aplicações práticas da rede proposta, foi feita a identificação de duas plantas reais.

As duas plantas simuladas consistem em exemplos não lineares simulados, amplamente utilizados na literatura como *benchmarks*. Já os dois sistemas reais, consistem num tanque multisseções, cujo objetivo foi o de identificar a sua dinâmica, e uma incubadora neonatal, em que se procurou inferir a umidade e a temperatura no interior da mesma, de modo a investigar a aplicabilidade da rede proposta no processo de calibração da incubadora.

Os resultados apresentados na identificação das duas plantas simuladas mostraram que a rede proposta obteve bons resultados, apresentando uma perda de desempenho muito pequena em relação às demais redes FWNN. Além disso, pode-se concluir que, quando a base de regras é pequena, como nos casos das plantas simuladas, a redução do número de parâmetros não é significativa. Porém, deve-se ressaltar que o algoritmo de treinamento ficou mais simples e, que de fato, houve uma redução no número de operações matemáticas da rede.

Os resultados da identificação do tanque multisseções mostram que esta técnica aproxima de maneira satisfatória a dinâmica do tanque e, quando comparado com o resultado obtido por meio das outras redes FWNN utilizadas na comparação, percebe-se que praticamente não houve perda de desempenho da rede proposta. Já com relação aos resultados obtidos no estudo de caso da incubadora, conclui-se que a rede obteve êxito na inferência da temperatura e da umidade no interior da mesma, mostrando ser uma estrutura viável em aplicações práticas que exigem respostas bastante precisas, como no caso de um processo de calibração.

Além disso, nesses dois últimos estudos de caso, pode-se concluir também que, em caso onde seja necessária a utilização de uma base de regras maior, a redução no número de parâmetros, principalmente em relação à rede da Yilmaz & Oysal (2010) é significativa. Além disso, como o trabalho mostrou a evolução dessa rede a partir de redes clássicas, as comparações realizadas com estas estruturas mostraram a superioridade da rede proposta, apresentando erros de validação bem inferiores, significando a obtenção de um modelo mais adequado, e de forma mais rápida.

De forma geral, os resultados demonstraram que não existe, de fato, perda da capacidade de generalização, se comparado com outras FWNNs utilizadas para fins de comparação, e que a rede FWNN proposta pode ser de grande valor em uma série de aplicações, tais como identificação, inferencia, controle e aplicações industriais.

Trabalhos futuros

Durante o processo de testes, tanto com a rede proposta, como com as demais redes FWNN, percebeu-se que a questão da inicialização dos parâmetros das redes, principalmente os parâmetros das *wavelets*, tem bastante influência no desempenho do treinamento deste tipo de

rede. Portanto, trabalhos futuros deverão investigar maneiras eficientes e sistemáticas de se inicializar esses parâmetros.

Além disso, no presente estudo, um algoritmo baseado em gradiente foi utilizado para estimar os parâmetros livres da nova estrutura FWNN. Esse algoritmo foi utilizado para realizar o ajuste de todos os parâmetros da rede. Embora tenham sido obtidos resultados satisfatórios, o estudo de outros algoritmos de aprendizagem podem ser investigados, buscando-se definir critérios para sua escolha, de maneira mais apropriada às características da rede proposta, tanto de forma geral e no caso específico de cada sistema. Também se pode pesquisar uma forma de obtenção automática do número de regras a serem utilizadas para compor a base de regras da rede FWNN, bem como a sugestão de métodos para a escolha do formato das funções de pertinência.

Referências Bibliográficas

Abiyev, R. H. (2006), *Time Series Prediction Using Fuzzy Wavelet Neural Network Model*, *Lecture Notes in Computer Sciences, Germany: Springer- Verlag*, vol. 4132, nº 1, pp. 191–200.

Abiyev, R. H. & Kaynak, O. (2008a), *Identification* e *Control of Dynamic Plants Using Fuzzy Wavelet Neural Networks*, *IEEE Int. Symp. Intell. Control (ISIC)*, pp. 1295–1301, San Antonio.

Abiyev, R. H. & Kaynak, O. (2008b), *Fuzzy Wavelet Neural Networks for Identification and Control of Dynamic Plants—A Novel Structure and Comparative Study, IEEE Trans. Ind. Electron.*, vol. 55, nº 8, pp. 3133–3140.

Aguirre, L. A., Rodrigues, G. G. e Jacome, C. R. F. (1998), Identificação de Sistemas Não Lineares Utilizando Modelos NARMAX Polinomiais, SBA Controle & Automação, vol. 9.

Aguirre, L. A. (2007), *Introdução à Identificação de Sistemas: Técnicas Lineares e Não Lineares Aplicadas a Sistemas Reais*, 3 ed., Editora UFMG, Belo Horizonte.

Albuquerque, A. A. M. (2012), Controle de Temperatura e Umidade de uma Incubadora Neonatal Conforme a Norma NBR IEC 60.601-2/19, Dissertação de Mestrado, Universidade Federal do Ceará (UFC), Fortaleza/CE – Brasil.

Alieva, R. A., Guirimov, B. G., Fazlollahi, B., e Aliev, R. R. (2009), *Evolutionary Algorithm-Based Learning of Fuzzy Neural Networks. Part 2: Recurrent Fuzzy Neural Networks, Fuzzy Sets Syst*., vol. 160, nº 17, pp. 2553–2566.

Amina, M. & Kodogiannis, V. S. (2011), *Load Forecasting Using Fuzzy Wavelet Neural Networks, IEEE Int. Conf. Fuzzy Syst.*, Taipei, Taiwan, pp. 1033–1040.

Araújo Júnior, J. M. (2007), Técnicas de Inteligência Artificial para a Geração Dinâmica de Set Points para uma Coluna de Destilação, Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte (UFRN), Natal/RN - Brasil.

Araújo Júnior, J. M., Linhares, L. L. S. e Araújo, F. M. U. (2013a), Identificação de Sistema Biomecânico a Partir de Redes Neurais *Wavelets* - Simpósio Brasileiro de Automação Inteligente – SBAI, Fortaleza, Brasil.

Araújo Júnior, J. M. & Araújo, F. M. U. (2011), *Identification Nonlinear by Fuzzy Wavelet Neural Networks, 21st Brazilian Congress of Mechanical Engineering*, Natal, RN, Brasil.

Araújo Júnior, J. M., Menezes Júnior, J. M. P., Albuquerque, A. A. M., Almeida, O. M., e Araújo, F. M. U. (2013b), *Assessment Certification of the Neonatal Incubator Through Sensors Neural Network Inferential, Sensors,* vol. 13, pp.: 15613-15632.

Association for the Advancement of Medical Instrumentation, ANSI/AAMI/IEC 60601-2-19. Medical Electrical Equipment - Part 2-19: Particular Requirements for Safety of Baby Incubator, 2009.

Banakar, A. & Azeem, M. F. (2006), *Identification and Prediction of Nonlinear Dynamical Plants Using TSK and Wavelet Neuro-Fuzzy Models*, 3 *International IEEE Conference Intelligent Systems,* pp.: 617-620.

Billings, S. A. & Wei, H. (2005), *A New Class of Wavelet Networks for Nonlinear System Identification, IEEE Trans. Neural Netw.*, vol. 16, nº 4, pp. 862–874.

Brockett, R.W. & Glaser, S. (2001), *Optimal Input Design for NMR System Identification. Proceedings of the Conference on Decision and Control*, vol. 5, pp.: 4128-4133.

Brosilow, C. & Joseph, B. (2002), *Techniques of Model Based Control*, 2 ed.; Prentice Hall: New York, USA.

Cândido, C. K. S. S. (2008), Classificação de Distúrbios na Rede Elétrica Usando Redes Neurais e Wavelets, Tese de Doutorado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN, Brasil.

Chui, C. K. (1992), *An Introduction to Wavelets*, New York: Academic.

Coelho, A. A. R. & Coelho, L. S. (2004). Identificação de Sistemas Dinâmicos Lineares, Florianópolis: Editora da UFSC.

Cordova, J. & Yu, W. (2012), *Two Types of Haar Wavelet Neural Networks for Nonlinear System Identification*, *Neural Processing Letters,* vol. 35, nº 3, pp.: 283–300.

Cybenko, G. (1989), *Approximation by Superpositions of a Sigmoidal Function*, *Mathematics of Control Signals, and Systems*, vol 2, nº 4, pp.: 303–314.

Daubechies, I. (1992), *Ten Lectures on Wavelets*, *CBMS-NSF Regional Series in Applied Mathematics (SIAM)*.

Daugman, J. G. (1988), *Complete Discrete 2-d Gabor Transforms by Neural Networks for Image Analysis and Compression*, *IEEE Transactions on In Acoustics, Speech and Signal Processing*, vol. 36, nº 7, pp.: 1169–1179.

Davanipoor, M., Zekri, M. e Sheikholeslam, F. (2012), *Fuzzy Wavelet Neural Network With an Accelerated Hybrid Learning Algorithm,* IEEE Transactions on Fuzzy Systems, vol. 20, nº 3, pp.: 463-470.

Efe, M. O. (2009), *A Comparison of ANFIS, MLP and SVM in Identification of Chemical Processes, IEEE International Symposium on Intelligent Control,* pp.: 689-694, Saint Petersburg, Rússia.

Elman, J. L. (1990), *Finding Structure in Time*, *Cognit. Sci*., vol. 14, nº 2, pp. 179–211.

Erdol, N. & Basbug, F. (1996), *Wavelet Transform Based Adaptive Filters: Analysis and New Results*, *IEEE Transactions on Signal Processing*, vol. 44, nº 9, pp.: 2163–2169.

Esmaili, A., Shahbazian, M. e Moslemi, B. (2013), *Nonlinear Process Identification Using Fuzzy Wavelet Neural Network Based on Particle Swarm Optimization Algorithm, Journal of Basic and Applied Scientific Research*, vol. 3, pp.: 302-309.

Fonseca, C. A. G. (2012), Estrutura ANFIS Modificada para Identificação e Controle de Plantas com Ampla Faixa de Operação e Não Linearidade Acentuada", Tese de Doutorado, Universidade Federal do Rio Grande do Norte (UFRN), Natal/RN - Brasil.

Fortuna, L., Giannone, P., Graziani, S. e Xibilia, M. G. (2007), *Virtual Instruments Based on Stacked Neural Networks to Improve Product Quality Monitoring in a Refinery*, *IEEE Transactions on Instrumentation and Measurement*, vol. 56, nº 1, pp.: 95–101.

Gabriel Filho, O. (2004), Contribuições à Análise de Robustez de Sistemas de Controle Usando Redes Neurais, Tese de Doutorado, Universidade Federal do Rio Grande do Norte (UFRN), Natal/RN - Brasil.

Ganjefar, S. & Tofighi, M. (2013), *A Fuzzy Wavelet Neural Network Stabilizer Design Using Genetic Algorithm for Multi-Machine Systems*, *Electrotechnical Review*, vol. 2013, nº 5, pp. 19–25.

Gao, M., Tian, J. e Li, K. (2007). *The Study of Soft Sensor Modeling Method Based on Wavelet Neural Network for Sewage Treatment*, *International Conference on Wavelet Analysis and Pattern Recognition*, vol. 2, pp.: 721-726.

Gorji, A. A. & Menhaj, M. B. (2008), *Identification of Nonlinear State Space Models Using an MLP Network Trained by the EM Algorithm*, *IEEE International Joint Conference on Neural Netowrks (IJCNN)*, pp. 53–60.

Graps, A. L. (1995). An Introduction to Wavelets, *IEEE Computational Sciences and Engineering,* vol. 2, pp.: 50–61.

Guiraud, M. B. & Poignet, D. (2009), *Physiological Musculoskeletal Model Identification for the Lower Limbs Control of Paraplegic Under Implanted FES, The IEEE/RSJ International Conference on Intelligent Robots and Systems*, Louis - USA.

Guo, Q-J., Yu, H-B., e Xu, A-D. (2005a), *An Online Self-Constructing Wavelet Fuzzy Neural Network for Machine Condition Monitoring*, *Proceedings of International Conference on Machine Learning and Cybernetics,* vol. 7.

Guo, Q-J., Yu, H-B., e Xu, A-D. (2005b), *Wavelet Fuzzy Neural Network for Fault Diagnosis Proceedings, International Conference on Communications, Circuits and Systems*, vol. 2.

Hassouneh, W., Dhaouadi, R. e Al-Assaf, Y. (2008). Dynamic Modeling of Nonlinear Systems Using Wavelet Networks, *Journal of Robotics and Mechatronics*, vol. 20, nº 1, pp.: 178–187.

Haykin, S., 2001, *Neural Networks*, 2nd ed., Prentice Hall, Upper Saddle River, New Jersey.

Ho, D. W. C., Zhang, P.-A. e Xu, J. (2001), *Fuzzy Wavelet Networks for Function Learning, IEEE Trans. Fuzzy Syst*., vol. 9, nº 1, pp.: 200–211.

Hong-yan, W., Lin, J., Bin, L. e Lilong, L. (2011), *The Deformation Time Series Prediction Based on Wavelet and Neural Network, International Conference on Electric Information and Control Engineering (ICEICE)*, pp.: 6242–6245.

Hornik, K., Stinchcombe, M. e White, H. (1989), *Multilayer Feedforward Networks are Universal Approximators, Neural Networks,* vol. 2, nº 2, pp.: 359–366.

Hornik, K. (1991), Approximation capabilities of multilayer feedforward networks. Neural Networks, v. 4, p. 251–257.

Huand, Y. C. & Huang, C. M. (2002), *Evolving Wavelet Networks for Power Transformer Condition Monitoring*, *IEEE Transactions on Power Delivery*, vol. 17, nº 2, pp.: 412–416.

Iyengar, S. S., Cho, E. C. e Phoha, V. V. (2002), *Foundation of Wavelet Networks and Applications*, *Chapman and Hall/CRC Press*, Boca Raton, FL.

Jang, J-S. R., Sun, C-T., e Mizutani, E. (1997), *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learn and Machine Intelligence*, 1ª edição, Prentice-Hall.

Jang, J-S. R. & Sun, C-T. (1995), Neuro-Fuzzy Modeling and Control, *Proceedings of the IEEE*, vol. 83, nº 3, pp.: 378–406.

Jain, T., Singh, S. N. e Srivastava, S. C. (2010), *Adaptive Wavelet Neural Network-Based Fast Dynamic Available Transfer Capability Determination*, *IET Generation, Transmission & Distribution*, vol. 4, nº 4, pp.: 519–529.

Jahangiri, F. (2010), *Identification of Twing-Tanks Dynamics Using Adaptive Wavelet Differential Neural Networks, International Joint conference on Neural Networks*, pp.:1 - 5.

Jang, J.-S. R. (1993), "*Anfis : Adaptive Network Based Fuzzy Inference System*, *IEEE Trans. Syst., Man, Cybern.*, vol. 23, nº 3, pp.: 665–685.

Juang, C.-F. (2002), *A TSK-Type Recurrent Fuzzy Network for Dynamic Systems Processing by Neural Network and Genetic Algorithms*, *IEEE Trans. Fuzzy Syst.*, vol. 10, nº 2, pp.: 155–170.

Juang, C.-F. & Lin, C.-T. (1999), *A Recurrent Self-Organizing Neural Fuzzy Inference Network, IEEE Trans. Neural Netw.*, vol. 10, nº 4, pp.: 828–845.

Kara, T. & Eker, T. (2003), *Experimental Nonlinear Identification of a Two Mass System, Proceedings of 2003 IEEE Conference on Control Applications*, pp.: 66-71.

Kim, D. & Kim, C. (1997), *Forecasting Time Series With Genetic Fuzzy Predictor Ensembles*, *IEEE Trans. Fuzzy Syst.*, vol. 5, nº 4, pp.: 523–535.

Kreinovich, V., Sirisaengtaksin, O. e Cabrera, S. (1994), *Wavelet Neural Networks are Asymptotically Optimal Approximators for Functions of One Variable*, *IEEE International Conference on Neural Networks*, vol. 1, pp.: 299–304.

Lee, C.-H. & Teng, C.-C. (2000), *Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks, IEEE Trans. Fuzzy Syst*., vol. 8, nº 4, pp.: 349–366.

Li, C., Minming, T. & Haibo, D. (2010), *Predicting Water Irruption Quantity From Coal Floor Based on Wavelet Neural Network*, *International Forum on Information Technology and Applications*, vol. 3, pp.: 154-157.

Lin, F., Shen, P., e Kung, Y. (2005), *Adaptive Wavelet Neural Network Control for Linear Synchronous Motor Servo Drive*, *IEEE Trans. Magn.*, vol. 41, nº 12, pp.: 4401–4412.

Lindskog, P. & Ljung, L. (1994), *Tools for Semi-Physical Modeling, Department of Electrical Engineering, Linköping University*.

Ling, S. H., Yeung, C.W., Chan, K. Y., Lu, H. H. C. e Leung, F. H. F. (2007), *A New Hybrid Particle Swarm Optimization With Wavelet Mutation for Neural Network Training*, *Proceedings of IEEE Congress on Evolutionary Computation (CEC2007)*, pp.: 1977–1984.

Linhares, L. L. S. (2010), Sistema Híbrido de Inferência Baseado em Análise de Componentes Principais e Redes Neurais Artificiais Aplicado a Plantas de Processamento de Gás Natural, Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN, Brasil.

Linhares, L. L. S., Araújo Júnior, J. M. e Araújo, F. M. U. (2013), *Nonlinear System Identification of a Lower Limb Model by Fuzzy Wavelet Neural Networks*, IECON - 39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria.

Lira, M. M. S., (2004), *Wavelets* na Compactação e Processamento de Sinais de Distúrbios em Sistemas de Potência para Classificação via Redes Neurais Artificiais, Tese de Doutorado, Universidade Federal Pernambuco (UFPE), Recife/PE – Brasil.

Ljung, L. (1987), *System identification - Theory for the User*. Prentice-Hall International, New Jersey, U.S.A.

Lu, C. (2011), *Wavelet Fuzzy Neural Networks for Identification and Predictive Control of Dynamic Systems*, *IEEE Trans. Ind. Electron*., vol. 58, nº 7, pp.: 3046–3058.

Lucena, P. B. (2005), Análise de um Controlador Baseado no Jacobiano Estimado da Planta Através de uma Rede Neural, Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.

Meyer, Y. (1993), *Wavelets: Algorithms and Applications*, Philaelphia, PA: SIAM.

Microchip. 0.5◦C Maximum Accuracy Digital Temperature Sensor. Octuber 2011. Retrieved in http://ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf.

Moura, L. G. (2003), Modelagem Empírica de Colunas de Destilação Utilizando Redes Neurais de Wavelets para Otimização e Controle de Processos, Dissertação de Mestrado, Universidade Federal de Santa Catarina, UFSC, Florianópolis, SC, Brasil.

Narendra, K. & Parthasarathy, K. (1990), *Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions On Neural Networks*. vol. 1, Nº 1.

Nelles, O. (2001), *Nonlinear System Identification*, Springer-Verlag, Berlin, Heidelberg, New York.

Nørgaard, M., Ravn, O., Poulsen, N. K. e Hansen, L. K. (2001), *Neural Networks for Modeling and Control of Dynamic Systems*, Springer-Verlag London Limited, Londres, Inglaterra.

Oliveira, H. M. (2003), Análise de Sinais para Engenheiros: Uma Abordagem Via *Wavelets*, Universidade Federal de Pernambuco, Departamento de Eletrônica e Sistemas, Editora Manole.

Oussar, Y. & Dreyfus, G. (2000), *Initialization by Selection for Wavelet Network Training*, *Neurocomputing,* vol. 34, pp.: 131–143.

Oussar, Y., Rivals, I., Personnaz, L. e Dreyfus, G. (1998), *Training Wavelet Networks for Nonlinear Dynamic Input-Output Modeling*, *Neurocomputing*, vol. 20, nº 1-3, pp.: 173–188.

Pan, H. & Xia, L.Z. (2008), *Efficient Object Recognition Using Boundary Representation and Wavelet Neural Network*, *IEEE Transactions on Neural Networks*, vol. 19, nº. 12, pp.: 2132-2149.

Pinto, L. D. (2003), Identificação de Sistemas Contínuos Utilizando o Método dos Momentos de Poisson, Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul.

Parekh, R., Yang, J. e Honavar, V., (2000), *Constructive Neural-Network Learning Algorithms for Pattern Classification,* IEEE Transactions on Neural Networks, vol. 11, nº 2, pp.: 436-451.

Rocha, V. B. (2008), Uma Abordagem de *Wavelets* Aplicada à Combinação de Previsões: Uma Análise Teórica e Experimental, Dissertação de Mestrado, Universidade Federal do Paraná, UFPR, Curitiba, PR, Brasil.

Rodrigues, M. C. (2006), Técnicas Inteligentes Híbridas para o Controle de Sistemas Não Lineares, Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte (UFRN), Natal/RN – Brasil.

Rodrigues, M. C. (2010), Identificação Fuzzy-Multimodelos para Sistemas Não Lineares, Tese de Doutorado, Universidade Federal do Rio Grande do Norte (UFRN), Natal/RN – Brasil.

Sensirion. Datasheet SHT1x (SHT10, SHT11, SHT15) Humidity and Temperature Sensor. 2011. http://www.sensirion.com/fileadmin/userupload/customers/sensirion/Dokumente/Humidity/Sensirion Humidity SHT1x Datasheet V5.pdf.

Sjöberg, J., Zhang, Q., Ljung, L., Beneviste, A., Deylon, B., Glorennec, P.Y., Hjalmarsson, H. e Juditsky, A. (1996), *Nonlinear Black-Box Modeling in System Identification: A Unified Overview*. *Automatica*, pp.: 1691-1724.

Song, Q. & Liu, L. (2011), *Wavelet Neural Network Based on Improved Particle Swarm Algorithm*, *The 6th International Forum on Strategic Technology (IFOST)*, vol. 2, pp.: 1000-1004.

Song, J.-R. & Shi, H.-B. (2011), *Dynamic System Modeling Based on Wavelet Recurrent Fuzzy Neural Network, Seventh International Conference on Natural Computation,* pp.: 766-770.

Sørensen, O. (1993), *Neural Networks Performing System Identification for Control Applications*, *International Conference on Artificial Neural Networks*, pp. 172–176.

Szu, H. H., Telfer, B., Kadambe, S. e Shubha, L. (1992), *Neural Network Adaptive Wavelets for Signal Representation and Classification*, *Optical Engineering*, vol. 31, pp.: 1907–1916.

Tan, Y., Dang, X., Liang, F. e Su, C, Y. (2000), *Dynamic Wavelet Neural Network for Non Linear Dynamic System Identification*, *Proceedings of the IEEE International Conference on Control Applications*, Anchorage, Alaska, EUA.

Thuillard, M. (2000), *A Review of Wavelet Networks, Wavenets, Fuzzy Wavenets and their Applications*, *Applications, ESIT 2000*, pp.: 5–16.

Veitch, D. (2005), *Wavelet Neural Networks and Their Application in the Study of Dynamical Systems*, Dissertação de Mestrado, University of York, UK.

Zekri, M., Sadri, S., e Sheikholeslam, F. (2008), *Adaptive Fuzzy Wavelet Network Control Design for Nonlinear Systems," Fuzzy Sets Syst*., vol. 159, nº 20, pp. 2668–2695.

Zhang, Q. & Benveniste, A. (1992), *Wavelet Networks*, *IEEE Trans. Neural Netw.*, vol. 3, nº 6, pp. 889–898.

Zhang, Q. (1997), *Using Wavelet Network in Nonparametric Estimation*, *IEEE Transactions on Neural Networks*, vol. 8, nº 3, pp.: 227–236.

Zhang, L. J., Yang, J. e Chu, Y. (2010), *Software Reliability Growth Model based on Fuzzy Wavelet Neural Network, 2nd International Conference on Future Computer and Communication (ICFCC)*, vol. 1, pp.: 664 – 668.

Yang, X. S. & Deb, S. (2009), *Cuckoo Search Via Lévy Flights*, *World Congress on Nature & Biologically Inspired Computing, NaBIC 2009*, pp.: 210–214.

Yılmaz, S. & Oysal, Y. (2009), *A Fuzzy Wavelet Neural Network Model for System Identification, Ninth International Conference on Intelligent Systems Design and Applications*, pp.: 1284 – 1289.

Yilmaz, S. & Oysal, Y. (2010), *Fuzzy Wavelet Neural Network Models for Prediction and Identification of Dynamical Systems, IEEE Transactions on Neural Networks*, vol. 21, nº 10, pp.: 1599-1609.

Yoo, S., Park, J. e Choi, Y. (2006), *Adaptive Dynamic Surface Control of Flexible-Joint Robots Using Self-Recurrent Wavelet Neural Networks*, *IEEE Transactions on Systems,Man and Cybernetics: Part B: Cybernetics*, vol. 36, nº 6, pp.: 1342–1355.

Wang, J.-S. & Chen, Y.-P. (2008), *A Hammerstein Recurrent Neuro-Fuzzy Network With an Online Minimal Realization Learning Algorithm, IEEE Trans. Fuzzy Syst*., vol. 16, nº 6, pp.: 1597–1612.

Warne, K., Prasad, G., Rezvani, S., e Maquire, L. (2004), *Statistical and Computational Intelligence Techniques for Inferential Model Development: A Comparative Evaluation and a Novel Proposition for Fusion. Engineering Applications of Artificial Intelligence*, vol. 17, pp.: 871–885.

Weigned, A. S., Rumelhart, D. E. e Huberman, B. A. (1990), *Predicting the Future: A Connectionist Approach, PDP Research Group*, Stanford, CA, Tech. Rep., Stanford-PDP-90-01 or PARC-SSL-90-20.