



UNIVERSIDADE FEDERAL DO PIAUÍ  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Heinrich Hertz Silva

**Identificação Multivariável de um Sistema de  
Tanques Acoplados Utilizando *Wavelet Neural  
Network***

Teresina

Outubro de 2019



Heinrich Hertz Silva

**Identificação Multivariável de um Sistema de Tanques  
Acoplados Utilizando *Wavelet Neural Network***

Dissertação submetida ao programa de pós-graduação em Engenharia Elétrica da Universidade Federal do Piauí como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Dr. José Medeiros de Araújo Júnior

Teresina

Outubro de 2019

FICHA CATALOGRÁFICA  
Universidade Federal do Piauí  
Biblioteca Comunitária Jornalista Carlos Castello Branco  
Serviço de Processamento Técnico

S586i Silva, Heinrich Hertz.  
Identificação Multivariável de um Sistema de Tanques  
Acoplados Utilizando *Wavelet Neural Network* / Heinrich Hertz  
Silva. – 2019.  
142 f.

Dissertação (Mestrado em Engenharia Elétrica) –  
Universidade Federal do Piauí, Teresina, 2019.  
“Orientador: Dr. José Medeiros de Araújo Júnior”.

1. Funções Wavelets. 2. Identificação de Sistemas.  
3. Redes Neurais Artificiais. I. Título.

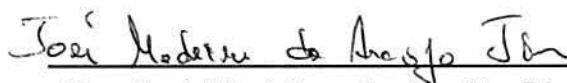
CDD 621.3

Heinrich Hertz Silva

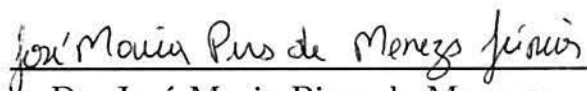
## Identificação Multivariável de um Sistema de Tanques Acoplados Utilizando *Wavelet Neural Network*

Dissertação submetida ao programa de pós-graduação em Engenharia Elétrica da Universidade Federal do Piauí como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Teresina, Setembro de 2019.

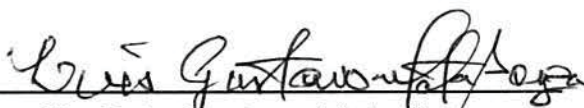


Dr. José Medeiros de Araújo Júnior  
Orientador - UFPI



Dr. José Maria Pires de Menezes  
Júnior

Avaliador Interno - UFPI



Dr. Luis Gustavo Mota Souza  
Avaliador Interno - UFPI



Dr. Marcelo Roberto Bastos Guerra  
Vale

Universidade Federal Rural do Semi-Árido -  
UFERSA



*Dedico este trabalho aos meus pais,  
Ernandes José Silva e Aucione Barbosa  
da Silva, pelo apoio e dedicação à  
minha formação, pois sem eles meus  
sonhos estariam mais distantes. A  
minha amada esposa Bruna Maria de  
Moura Soares, pelo seu amor e apoio  
ao longo destes 2 anos de Mestrado.*





# Agradecimentos

A Deus, pela vida, e por ter me dado saúde e força para superar as dificuldades ao longo de minha vida.

Aos meus pais, Ernandes José Silva e Aucione Barbosa da Silva, por sempre acreditarem e depositarem confiança em todos os meus sonhos e objetivos.

Ao Prof. Dr. José Medeiros de Araújo Júnior, pela confiança, paciência, ensinamentos, e principalmente pelo incentivo, auxiliando-me nos momentos mais difíceis da execução deste trabalho.

Aos professores participantes da banca examinadora, Prof. Dr. José Maria Pires de Menezes Júnior, Prof. Dr. Luis Gustavo Mota Souza e Prof. Dr. Marcelo Roberto Bastos Guerra Vale pelas colaborações, sugestões e pelo tempo dedicado.

Aos grandes amigos Francisco Victor Lemos e Andrei Carvalho pelo grande apoio e ajuda durante todo o curso.

A Universidade Federal do Piauí, pela oportunidade de fazer a pós-graduação.



*“Não vos amoldeis às estruturas deste mundo,  
mas transformai-vos pela renovação da mente,  
a fim de distinguir qual é a vontade de Deus:  
o que é bom, o que Lhe é agradável, o que é perfeito.  
(Bíblia Sagrada, Romanos 12, 2)*



# Resumo

Fundamentado na teoria de identificação de sistemas, redes neurais artificiais e funções *Wavelets*, o presente trabalho apresenta a identificação de um sistema de nível com tanques acoplados com características não lineares, utilizando como ferramenta a *Wavelet Neural Network* (WNN), com o intuito de verificá-la como uma alternativa para identificação de sistemas dinâmicos. Para melhor avaliar as potencialidades não lineares da WNN, optou-se por utilizar um sistema de nível, que por si só já possui natureza não linear. Além disso, o sistema possui acoplamento de sinais, ruídos de medição e foi construído de forma artesanal. Tais características aumentam ainda mais a não linearidade do sistema e, conseqüentemente, a complexidade para modelagem. Para excitar o sistema utilizou-se o sinal PRBS (*Pseudo Random Binary Signal*) e, uma vez excitado o sistema, coletou-se dados de entradas e saída a fim de realizar a modelagem utilizando a estrutura de identificação NNARX (*Neural Network AutoRegressive with eXogenous input*). Na etapa seguinte realizou-se a estimação e validação do modelo neural obtido por meio da *Wavelet Neural Network*. Os resultados obtidos validam a aplicabilidade das Redes Neurais *Wavelets* em sistemas não lineares nos quais ocorrem acoplamentos de sinais e que estão susceptíveis a ruídos de medição, com isso atingindo o objetivo em estudo.

**Palavras-chave:** funções *wavelets*, identificação de sistemas, redes neurais artificiais.



# Abstract

Based on the theory of systems identification, artificial neural networks and Wavelets functions, the present work presents the identification of a level system with coupled tanks with nonlinear characteristics, using as a tool the Wavelet Neural Network (WNN), in order to verify it as an alternative for identifying dynamic systems. To better evaluate the nonlinear potentialities of WNN, we chose to use a level system, which in itself already has a nonlinear nature. In addition, the system has signal coupling, measurement noise and is handcrafted. Such features further increase the nonlinearity of the system and, consequently, the complexity for modeling. The Pseudo Random Binary Signal (PRBS) signal was used to excite the system and, once the system was excited, input and output data were collected in order to perform the modeling using the NNARX (Neural Network AutoRegressive with eXogenous) identification structure. In the next step, the neural model obtained through the Wavelet Neural Network was estimated and validated. The results validate the applicability of Wavelet Neural Networks in nonlinear systems where signal couplings occur and are susceptible to measurement noise, thus achieving the objective under study.

**Keywords:** wavelets functions, systems identification, artificial neural networks.





# Lista de ilustrações

Figura 1 – Neurônio Artificial. . . . .	38
Figura 2 – Função de Ativação Logística. . . . .	39
Figura 3 – Função de Ativação Tangente Hiperbólica. . . . .	39
Figura 4 – Função de Ativação Linear (ou Identidade). . . . .	40
Figura 5 – Exemplo de rede <i>feedforward</i> de camadas múltiplas. . . . .	41
Figura 6 – Ilustração de rede <i>Multilayer Perceptron</i> . . . . .	43
Figura 7 – Função Cosseno. . . . .	46
Figura 8 – Função <i>Wavelet</i> . . . . .	47
Figura 9 – Função <i>Wavelet Mexican Hat</i> . . . . .	50
Figura 10 – Função <i>Wavelet Morlet</i> . . . . .	51
Figura 11 – Função Densidade de Probabilidade Gaussiana. . . . .	51
Figura 12 – Funções <i>Wavelets</i> Gaussianas. . . . .	52
Figura 13 – Funções <i>Wavelets Meyer</i> . . . . .	53
Figura 14 – Estrutura da <i>Wavelet Neural Network</i> . . . . .	54
Figura 15 – Neurônio <i>Wavelet</i> . . . . .	55
Figura 16 – <i>Wavelon</i> com Função de Ativação <i>Wavelet</i> Multidimensional. . . . .	55
Figura 17 – <i>Wavelets</i> da Família <i>Mexican Hat</i> Transladadas. . . . .	56
Figura 18 – <i>Wavelets</i> da Família <i>Mexican Hat</i> Dilatadas. . . . .	57
Figura 19 – Ilustração representativa de acoplamento de sinais. . . . .	60
Figura 20 – Procedimento básico de identificação caixa preta. . . . .	61
Figura 21 – Diagrama de Blocos - Identificação de processos. . . . .	62
Figura 22 – Estrutura NNARX para identificação de sistemas. . . . .	63
Figura 23 – Sinal utilizado para excitação do sistema em estudo. . . . .	64
Figura 24 – Esquema do sistema de tanques acoplados. . . . .	66
Figura 25 – Sistema de tanques de nível acoplados utilizado na pesquisa. . . . .	66
Figura 26 – Sensor de Pressão Utilizado. . . . .	67
Figura 27 – Placa SanUSB. . . . .	68
Figura 28 – Sinal PRBS Aplicado à Bomba Hidráulica associada ao Tanque 1. . . . .	69
Figura 29 – Sinal PRBS Aplicado à Bomba Hidráulica associada ao Tanque 2. . . . .	70
Figura 30 – Resposta do Sistema: Nível do Tanque 1. . . . .	70
Figura 31 – Resposta do Sistema: Nível do Tanque 2. . . . .	71
Figura 32 – Ruídos de Medição. . . . .	72
Figura 33 – Estrutura NNARX para identificação do sistema de nível desacoplado. . . . .	74
Figura 34 – Estrutura NNARX para identificação do sistema de nível acoplado. . . . .	75
Figura 35 – Representação da MLP para o estudo de caso multivariável. . . . .	78
Figura 36 – Representação da WNN para o estudo de caso multivariável. . . . .	80

Figura 37 – Configuração com Tanques de Nível Desacoplados: Sinal PRBS utilizado para excitar as bombas hidráulicas dos Tanques 01 e 02 para obtenção dos dados de treinamento. . . . .	86
Figura 38 – Configuração com Tanques Desacoplados: Sinal PRBS utilizado para validação dos modelos dos Tanques. . . . .	86
Figura 39 – Configuração com Tanques de Nível Desacoplados: Comparação das dinâmicas entre os tanques de nível 01 e 02. . . . .	87
Figura 40 – Configuração com Tanques de Nível Desacoplados (MLP): Médias dos MSEs para validação do Tanque de Nível 01. . . . .	88
Figura 41 – Configuração com Tanques de Nível Desacoplados (MLP): Variâncias dos MSEs para validação do Tanque de Nível 01. . . . .	89
Figura 42 – Configuração com Tanques de Nível Desacoplados (MLP): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 01). . . . .	90
Figura 43 – Configuração com Tanques de Nível Desacoplados (MLP): MSE (206.760 Épocas de Treinamento). . . . .	90
Figura 44 – Configuração com Tanques de Nível Desacoplados (WNN): Médias dos MSEs para validação do Tanque de Nível 1. . . . .	92
Figura 45 – Configuração com Tanques de Nível Desacoplados (WNN): Variâncias dos MSEs para validação do Tanque de Nível 1. . . . .	92
Figura 46 – Configuração com Tanques de Nível Desacoplados (WNN): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 1). . . . .	94
Figura 47 – Configuração com Tanques de Nível Desacoplados (WNN): MSE (163 Épocas de Treinamento). . . . .	94
Figura 48 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para excitar a bomba hidráulica do Tanque 1 para obtenção dos dados de treinamento. . . . .	95
Figura 49 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para excitar a bomba hidráulica do Tanque 2 para obtenção dos dados de treinamento. . . . .	96
Figura 50 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para Validação do modelo Tanque 1. . . . .	96
Figura 51 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para Validação do modelo Tanque 2. . . . .	97
Figura 52 – Configuração com Tanques de Nível Acoplados (MLP): Médias dos MSEs para Validação do Tanque de Nível 1. . . . .	98
Figura 53 – Configuração com Tanques de Nível Acoplados (MLP): Variância dos MSEs para Validação do Tanque de Nível 1. . . . .	98
Figura 54 – Configuração com Tanques de Nível Acoplados (MLP): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 01). . . . .	99

Figura 55 – Configuração com Tanques de Nível Acoplados (MLP):Validação do modelo para saída 2 do sistema dinâmico (Tanque de Nível 02). . . . .	100
Figura 56 – Configuração com Tanques de Nível Acoplados (MLP): MSE – (354.614 Épocas). . . . .	100
Figura 57 – Configuração com Tanques de Nível Acoplados (WNN): Médias dos MSEs para Validação do Tanque de Nível 01. . . . .	101
Figura 58 – Configuração com Tanques de Nível Acoplados (WNN): Variância dos MSEs para Validação do Tanque de Nível 01. . . . .	102
Figura 59 – Configuração com Tanques de Nível Acoplados (WNN): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 01). . . . .	103
Figura 60 – Configuração com Tanques de Nível Acoplados (WNN): Validação do modelo para saída 2 do sistema dinâmico (Tanque de Nível 02). . . . .	104
Figura 61 – Configuração com Tanques de Nível Acoplados (WNN): MSE (91 Épocas). 104	
Figura 62 – Análise de Resíduos para o Modelo WNN do Tanque de Nível 1: (a) Histograma dos Resíduos; (b) Autocorrelação dos resíduos; (c) Correlação cruzada linear entre os sinais de resíduos e entrada; (d) Correlação não linear entre os sinais de resíduos e entrada. . . . .	107
Figura 63 – Análise de Resíduos para o Modelo WNN do Tanque de Nível 2: (a) Histograma dos Resíduos; (b) Autocorrelação dos resíduos; (c) Correlação cruzada linear entre os sinais de resíduos e entrada; (d) Correlação não linear entre os sinais de resíduos e entrada. . . . .	108
Figura 64 – Aproximação das Amostras de Resíduos com a Função Gaussiana para o Modelo do Tanque de Nível 1 - Estudo de Caso Multivariável. . . . .	109
Figura 65 – Análise de Resíduos para o Modelo MLP do Tanque de Nível 1: (a) Histograma dos Resíduos; (b) Autocorrelação dos resíduos; (c) Correlação cruzada linear entre os sinais de resíduos e entrada; (d) Correlação não linear entre os sinais de resíduos e entrada. . . . .	110
Figura 66 – Convergência dos Modelos neurais WNN e MLP para 100, 500 e 1000 épocas. . . . .	111



# Lista de tabelas

Tabela 1 – Parâmetros da Função <i>indinput</i> : Tanques Acoplados . . . . .	69
Tabela 2 – Parâmetros da Função <i>indinput</i> : Tanques Desacoplados . . . . .	71
Tabela 3 – Dados de Parametrização - Estudo de caso com desacoplamento . . . . .	76
Tabela 4 – Dados de Parametrização - Estudo de caso com acoplamento . . . . .	77
Tabela 5 – Análise da MLP: Erro como Critério de Parada . . . . .	88
Tabela 6 – Análise da WNN: Erro como Critério de Parada . . . . .	91
Tabela 7 – Coeficientes de Translação - <i>Wavelets</i> : Tanques Desacoplados . . . . .	93
Tabela 8 – Coeficientes de Dilatação - <i>Wavelets</i> : Tanques Desacoplados . . . . .	93
Tabela 9 – Análise da MLP Multivariável: Erro como Critério de Parada . . . . .	97
Tabela 10 – Análise da WNN Multivariável: Erro como Critério de Parada . . . . .	101
Tabela 11 – Coeficientes de Translação - <i>Wavelets</i> : Tanques Acoplados . . . . .	103
Tabela 12 – Coeficientes de Dilatação - <i>Wavelets</i> : Tanques Acoplados . . . . .	103
Tabela 13 – Comparação entre os MSEs . . . . .	105



# Lista de abreviaturas e siglas

AMR	Análise Multirresolucional
A/D	Analógico Digital
ARX	<i>AutoRegressive eXogenous input</i>
ARMAX	<i>AutoRegressive eXogenous input</i>
CC	Corrente Contínua
CPU	<i>Central Processing Unit</i>
FIR	<i>Finite Impulse Response</i>
GSP	<i>Gyro-stabilized platform</i>
MCA	Metro de Coluna de Água
MIMO	<i>Multiple-Input / Multiple-Output</i>
MLP	<i>Multilayer Perceptron</i>
MSE	<i>Mean-Squared Error</i>
NARMAX	<i>Nonlinear AutoRegressive Moving Average with eXogenous input</i>
NARX	<i>Nonlinear AutoRegressive models with eXogenous input</i>
NNARX	<i>Neural Network AutoRegressive eXogenous input</i>
NNFIR	<i>Neural Network Finite Impulse Response</i>
OE	<i>Output Error</i>
PIC	<i>Peripheral Intergrated Controller</i>
PID	Proporcional, Integral e Derivativo
PWM	<i>Pulse-Width Modulation</i>
PRBS	<i>Pseudo Random Binary Signal</i>
RLS	<i>Recursive Least Squares</i>
RNA	Rede Neural Artificial

SISO	<i>Single-Input / Single-Output</i>
SMF	Sistema de Manobra Flexível
SSIF	<i>State Space Innovations Form</i>
TPD	Transmissor de pressão diferencial
TCW	Transformada Continua <i>Wavelet</i>
TDW	Transformada Discreta <i>Wavelet</i>
UFPI	Universidade Federal do Piauí
VRFT	<i>Virtual Reference Feedback Tuning e Q-Learning</i>
WNN	<i>Wavelet Neural Network</i>



# Lista de símbolos

$\%$	Porcentagem
$A$	Ampere
$V$	Volt
$\copyright$	Copyright
$x_n$	Sinais de Entrada
$w$	Peso Sináptico
$\mathbf{W}^{(L)}$	Matriz de pesos sinápticos
$w_{ji}$	Peso Sináptico (escalar)
$\theta$	Parâmetros adaptativos da WNN
$g$	Função de ativação
$h$	Função de Ativação
$d$	Atraso de Transporte
$n_y$	Ordem de Saída
$n_x$	Ordem de entrada
$z^{-1}$	Operador de Atraso
$e$	Erro
$u$	Potencial de Ativação
$\beta$	Constante real relacionada com a inclinação das funções de ativação logística e tangente hiperbólica
$Y_M$	Saída de uma RNA
$\delta$	Gradiente local dos neurônios de uma RNA
$\psi$	Função <i>wavelet</i>
$\lambda$	Parâmetro de dilatação <i>wavelet</i>
$t$	Parâmetro de translação <i>wavelet</i>

- $\Psi$  Saída *wavelon* multidimensional
- $\Theta$  Limiar de ativação de neurônios artificiais

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>27</b>
1.1	Estado da Arte	29
1.2	Justificativa	33
1.3	Hipóteses e Objetivos	33
1.4	Organização da Dissertação	34
1.5	Trabalhos Publicados e Submetidos	35
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>37</b>
2.1	Redes Neurais Artificiais	37
2.1.1	Considerações Gerais	37
2.1.2	<i>Multilayer Perceptron</i>	42
2.2	Funções <i>Wavelets</i>	46
2.3	<i>Wavelet Neural Network</i>	53
2.4	Sistemas Dinâmicos	58
2.4.1	Sistemas dinâmicos e suas classificações	58
2.4.1.1	Sistemas lineares e não lineares	59
2.4.1.2	Sistema MIMO e acoplamento de sinais	59
2.4.2	Identificação de sistemas dinâmicos	60
2.4.3	Modelos neurais de identificação	62
2.5	<i>Pseudo Random Binary Signal</i>	63
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>65</b>
3.1	Equipamentos e Operação: Sistema de Tanques de Nível	65
3.2	Coleta de Dados	68
3.3	Pré-Processamento de Dados	71
3.4	Seleção da Estrutura de Identificação	72
3.5	Estimação dos Modelos do Sistema de Tanques	77
3.5.1	Modelo Neural: <i>Multilayer Perceptron</i>	77
3.5.2	Modelo Neural: <i>Wavelet Neural Network</i>	79
3.6	Validação dos Modelos do Sistema de Tanques	82
3.6.1	Simulação dos Modelos: Saída Estimada vs Saída Medida	82
3.6.2	Análise de Resíduos	83
<b>4</b>	<b>RESULTADOS</b>	<b>85</b>
4.1	Resultados da Validação: Tanques Desacoplados	85
4.1.1	Modelo Neural MLP	87

4.1.2	Modelo Neural WNN . . . . .	91
<b>4.2</b>	<b>Resultados da Validação: Tanques Acoplados . . . . .</b>	<b>93</b>
4.2.1	Modelo Neural MLP . . . . .	95
4.2.2	Modelo Neural WNN . . . . .	99
<b>4.3</b>	<b>Análise das Hipóteses do Trabalho . . . . .</b>	<b>104</b>
4.3.1	Hipótese 1: Viabilidade da WNN aplicada na identificação de tanques de nível multivariável . . . . .	105
4.3.2	Hipótese 2: Exatidão e capacidade de generalização da WNN . . . . .	105
4.3.3	Hipótese 3: WNN possui melhor convergência que a MLP . . . . .	110
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>113</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>115</b>
	<b>APÊNDICES . . . . .</b>	<b>121</b>
	<b>APÊNDICE A – SCRIPT PARA COLETA DE DADOS . . . . .</b>	<b>123</b>
	<b>APÊNDICE B – SCRIPT PRINCIPAL . . . . .</b>	<b>127</b>
	<b>APÊNDICE C – FUNÇÕES UTILIZADAS NO SCRIPT . . . . .</b>	<b>133</b>

# 1 Introdução

Desde o surgimento da matemática, os homens tentam estudar e compreender sistemas e fenômenos que os auxiliem a solucionar problemas do seu cotidiano. Para isso faz-se necessário uma representação matemática desses sistemas, ou seja, um modelo matemático capaz de reproduzir algumas características de um fenômeno observado, assim como uma maquete reproduz as escalas, proporções, cores, etc. de uma construção real (AGUIRRE, 2015).

Atualmente, modelos matemáticos são utilizados por cientistas e engenheiros para projetar sistemas com o intuito de controlar e simular fenômenos do mundo real, podendo ser dois tipos: estático e dinâmico. Modelos estáticos relacionam variáveis sem quantificar sua dependência temporal e são normalmente descritos por equações algébricas. No entanto, se a evolução temporal de um sistema é desejada, modelos dinâmicos devem ser usados (AGUIRRE, 2015).

Assim, um sistema dinâmico pode ser definido como um sistema com memória, no qual o estado presente depende dos estados passados. Matematicamente, trata-se de um conjunto de equações, lineares ou não, envolvendo as mesmas variáveis que sofrem uma determinada evolução ao longo do tempo, podendo ser representadas por meio de conjunto de equações diferenciais, função de transferência ou espaço de estados (FRANKLIN; POWELL; EMAMI-NAEINI, 2013).

Genericamente, a modelagem de sistemas dinâmicos pode ter duas abordagens. Quando a matemática por trás do fenômeno é conhecida e esse conhecimento é utilizado para estimação do modelo, tal abordagem é chamada de Modelagem por Princípios Básicos ou Fundamentos. No entanto, quando o conhecimento matemático não existe, mas existem dados observados do fenômeno em questão, esses dados podem ser usados para obter o modelo. Este segundo método é chamado de Modelagem Conduzido por Dados ou Identificação de Sistemas (FRANKLIN; POWELL; EMAMI-NAEINI, 2013).

Assim, identificação de sistemas é o nome dado a área de conhecimento dedicada a construção de modelos matemáticos a partir de dados observados em sistemas dinâmicos reais, em que por meio de testes no processo, mede-se sua resposta a determinadas entradas e utiliza-se os dados obtidos para construir um modelo analítico (KEESMAN, 2011).

As raízes de projetos para identificação de sistemas só começaram a se consolidar na década de 1960, quando os modelos paramétricos explícitos começaram a ocasionar interesse na comunidade de controle. No entanto, o desenvolvimento dos conceitos estatísticos essenciais usados em sistemas de identificação começaram em 1809. Entre os desenvolvimentos deste período, 1809 a 1960, pode-se citar: regressão linear, mínimos

quadrados, método da máxima-verossimilhança, aproximação estocástica, análise por série temporal e o método da variável instrumental (COELHO; COELHO, 2016).

Somente em 1985, com a visão estatística de sistemas de identificação já madura e sedimentada, as estruturas clássicas de estimação de parâmetros alcançaram relativo sucesso. O desenvolvimento de ferramentas computacionais poderosas e o aparecimento maciço de pacotes de *softwares* comerciais também contribuíram para o aprimoramento da área de identificação (COELHO; COELHO, 2016).

Atualmente, entre os tópicos com desenvolvimentos significativos na área de identificação de sistemas, cita-se: identificação para controle; propriedades de rejeição e tratamento de ruído; análise de dados no domínio da frequência e modelos não lineares (área de redes neurais artificiais, modelos *fuzzy*, transformada *wavelet*, etc) e aplicações (COELHO; COELHO, 2016).

O crescente interesse pela utilização de modelos não lineares se deve ao fato dos sistemas dinâmicos encontrados na prática serem, em última análise, não lineares. É bem verdade que em alguns casos aproximações lineares são suficientes para aplicações práticas (AGUIRRE, 2015). Entretanto, a 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, estruturas não lineares de identificação vem ganhando expressiva importância nas pesquisas atuais, tornando-se ferramentas usuais (JUNIOR, 2014).

Há uma grande variedade de representações não lineares que, pelo menos em princípio, podem ser utilizadas na identificação de sistemas. Dentre essas representações, tem-se a *Série de Volterra* utilizada para sistemas com menor grau de não linearidade. A grande desvantagem dessa aplicação é o grande número de parâmetros a determinar, e isso é consequência da *Série de Volterra* tentar explicar a saída apenas em função da entrada (AGUIRRE, 2015).

Outros dois tipos de modelos não lineares, que foram bastante populares até final da década de 70, são os modelos de *Hammerstein* e de *Wiener*. Ambos os modelos são uma composição de um modelo dinâmico linear em cascata com uma função estática não linear. Já os modelos NARX (do termo em inglês *Nonlinear AutoRegressive models with eXogenous input*) são modelos discretos no tempo que explicam o valor de saída em função de valores prévios dos sinais de saída e de entrada. Quando se deseja evitar polarização de parâmetros é comum incluir termos de ruído no modelo. Quando isso é feito, o modelo passa a ser um modelo NARMAX (do inglês *Nonlinear AutoRegressive Moving Average with eXogenous input*) (AGUIRRE, 2015).

As Redes Neurais Artificiais (RNAs) vêm sendo empregadas em diversos problemas de engenharia, por possuírem a capacidade de aquisição e manutenção do conhecimento.

Mais especificamente, as características mais atrativas das RNAs, fazendo destas ferramentas poderosas para aplicação na identificação de sistemas não lineares, consistem de suas habilidades em aprender, adaptar, generalizar, realizar processamento paralelo e ser tolerante a falhas (SILVA; SPATTI; FLAUZINO, 2010).

Dentre as RNAs que possuem as características supracitadas, tem-se a rede *Multi-Layer Perceptron* (MLP) que é uma das arquiteturas mais clássicas de redes neurais. É uma rede do tipo *feedforward* (em que cada neurônio de uma camada recebe como entrada apenas as saídas de neurônios da camada imediatamente anterior), possuindo a habilidade de aproximação universal de função (HORNIK; STINCHCOMBE; WHITE, 1989).

Outra RNA que tem-se apresentado como uma alternativa promissora às redes neurais tradicionais na modelagem de sistemas não lineares, classificação, predição e controle é a *Wavelet Neural Network* (WNN). Tal RNA apresenta uma estrutura similar à rede MLP, com a diferença de apenas substituir as funções de ativação da camada escondida por funções *wavelets*. Esta rede teve início com o trabalho de DAUGMAN (1988).

A razão fundamental para este trabalho é buscar a obtenção de um modelo matemático de um sistema dinâmico não linear para que em futuras pesquisas se possa desenvolver um controlador preditivo baseado em modelos para sistemas dinâmicos. Afinal, independentemente da modalidade do controlador, o controle de processos em engenharia se realiza a partir de um modelo. (LANDAU, 2006).

Assim, fundamentado na teoria de funções *Wavelets*, Redes Neurais Artificiais e Identificação de Sistemas, o presente trabalho apresenta a identificação de um sistema dinâmico de tanques de nível acoplados com características não lineares, utilizando como ferramenta a *Wavelet Neural Network* (WNN), com o intuito de verificá-la como uma alternativa para identificação de sistemas dinâmicos. É importante enfatizar que para melhor avaliar as potencialidades não lineares da WNN, optou-se por utilizar um sistema de tanques, que por si só já possui natureza não linear. Além disso, o sistema possui acoplamento de sinais, ruídos de medição e foi construído de forma artesanal. Tais características aumentam ainda mais a não linearidade do sistema e, conseqüentemente, a complexidade para modelagem.

## 1.1 Estado da Arte

No setor industrial a maioria dos processos necessitam de alguma forma de automação, controle ou monitoração de operação. A forte concorrência do mercado faz surgir a necessidade de uma linha de produção veloz, precisa e capaz de otimizar tempo e recursos. Por esse motivo, processos monitorados, controlados e automatizados se fazem tão importantes para indústria de alta produtividade (BHANDARE; KULKARNI, 2015).

Dentre os processos industriais mais comuns, a utilização de tanques de nível são bastantes usuais na fabricação de bebidas, cosméticos e inclusive na indústria do petróleo. O nível é uma variável importante na indústria não somente para a operação do próprio processo, mas também para fins de cálculo de custo e de inventário (BHANDARE; KULKARNI, 2015). Um sistema de nível com dois tanques interligados, desenvolvido por alunos e professores da Universidade Federal do Piauí, constituirá a planta do sistema em estudo e por ser didático é adequado para a pesquisa em questão.

Sistemas semelhantes ao supracitado são utilizados para o estudo e pesquisa de diversos tipos de temas dentro da área de controle, instrumentação, modelagem e inteligência artificial. A exemplo do estudo apresentado por BASTIDA et al. (2013), que implementa um controlador proporcional integral para um sistema de tanques acoplados para controle de nível de líquido. A identificação do modelo é feita por sinal sinusoidal em diferentes frequências. Os resultados da implementação de um PI (Proporcional-Integral) convencional mostram desempenho satisfatório. A aplicação do sinal senoidal não apenas permite modelar um sistema não linear, mas também é usado para projetar o controlador PI (Proporcional-Integral) completo.

Pesquisas mostram, também, que é possível desenvolver técnicas de controle conduzido por dados neste tipo de sistema. A exemplo de RADAC; PRECUP; ROMAN (2018), que propuseram o controle orientado por dados de sistemas de tanques verticais MIMO com VRFT *Virtual Reference Feedback Tuning* e *Q-Learning*. A nova e iterativa estratégia *Q-Learning* usa duas redes neurais para representar a função de valor (crítico) e o controlador, e é chamada de abordagem de *Q-learning* ajustada por *Tuning-Batch*. A aplicação do método é experimentalmente validada para o controle do nível de água de um sistema de dois tanques acoplado não linear multivariável.

Já na área de instrumentação, PAUL; SENGUPTA; RANJAN (2013) empregaram a técnica de remoção de ruído baseada em *wavelets* para remover o ruído de medição da saída do Transmissor de Pressão Diferencial (TPD), responsável por indicar o nível de um tanque de processo. O Sistema de Nível de Líquido (SNL) é aproximado como uma planta de primeira ordem com atraso de tempo. Existem canais suficientes para contaminar a saída do SNL pelo ruído de medição de alta frequência. Foi feita uma tentativa de avaliar o desempenho da técnica de remoção de ruído por Transformada Discreta *Wavelet* (TDW). A resposta disponível usando filtragem baseada em *wavelet* é comparada com a do método convencional de filtragem *Butterworth*, mostrando que a redução de ruído baseada em *wavelets* tem um desempenho melhor do que o método convencional.

Comumente, para processos mais simples, utiliza-se de princípios básicos ou fundamentos matemáticos para obter modelos de sistemas dinâmicos. No entanto, nem sempre se conhece as equações envolvidas no funcionamento de um determinado sistema dinâmico, ou elas são conhecidas mas seriam impraticáveis, por limitações de tempo e recursos,



levantar tais equações e estimar respectivos modelos (AGUIRRE, 2015).

Por esse motivo, o domínio de técnicas de identificação de sistemas é importante, pois a construção de modelos ocorre a partir de dados observados em sistemas dinâmicos reais, de modo que pouco ou nenhum conhecimento prévio do sistema seja necessário. A exemplo de RAHMAT; ROZALI (2008), que determinaram o modelo matemático de um sistema acoplado de tanques usando técnicas de identificação de sistemas e análise da resposta transitória do sistema. Em seguida, projetou controladores que consistem em um controlador PID e Lógica *Fuzzy* para o sistema. Na fase final deste projeto, o uso de ambos os controladores em aplicações industriais é comparado e analisado, apresentando resultados satisfatórios.

Em 2015, LEE et al., propuseram a identificação de um sistema eletro-hidráulico para aplicação do modelo no controle de posição. Para isso foi utilizado o algoritmo recursivo de mínimos quadrados ou, na língua inglesa, *Recursive Least Squares* (RLS) que encontra de forma recorrente os coeficientes da função de transferência de baixa ordem do sistema de malha fechada.

Quando se trata de sistemas não lineares, a utilização de RNAs como ferramenta para identificação de sistemas é usual. Em 2017, GAUTAM propôs a identificação do sistema de pêndulo invertido não linear usando rede neural *Multilayer Perceptron*. A RNA foi avaliada para vários números de neurônios de camada oculta, e observou-se que em condições de malha fechada, o aumento nos números de neurônios ocultos tem menos efeito na precisão do modelo.

Em 2018, NAUNG et al., propuseram a modelagem e controle de um motor CC por meio de técnicas de identificação de sistemas. Para isso foi utilizado a estrutura de identificação NNARX (*Neural Network AutoRegressive with eXogenous input*) e o algoritmo de *Levenberg-Marquardt* para treinamento da rede. Os resultados mostram uma boa aproximação do modelo em relação ao sistema real, em que os resultados dessa modelagem são aplicados no controle do motor CC.

Dentre as RNAs utilizadas como ferramentas de identificação, a MLP é uma das mais usuais. A exemplo de MONESS; DIAA-ELDEEN (2018), que apresentou uma investigação experimental de uma rede *MultiLayer Perceptron* para modelar toda a dinâmica de um Sistema de Manobra Flexível (SMF). Basicamente, trata-se de uma estrutura complexa não linear, semelhante a um helicóptero em menor escala. A abordagem introduzida é comprovada experimentalmente empregando as habilidades de adaptação da RNA para identificar a dinâmica do sistema. Os resultados mostram um desempenho superior das redes neurais em termos de generalização de modelos, predição de resposta e desempenho em domínios de tempo e frequência.

Em (2015), MARTINS; SOUZA; ARAUJO, apresentaram o desenvolvimento e

testes de um *software* para identificação de dinâmicas de um sistema não linear de tanques acoplados, utilizando Redes Neurais Artificiais do tipo *MultiLayer Perceptron*. A estrutura de modelagem utilizada foi do tipo NNARX. O *software* é genérico possibilitando ao usuário fazer vários ensaios com quantidades diferentes de camadas escondidas e número de neurônio. Além disso, é possível alterar as funções de ativação e ainda fazer uma análise da curva de aprendizagem através de sua *interface*. O *software* permite fazer a validação de duas formas diferentes, podendo ser com o sistema em malha aberta ou em malha fechada, possibilitando ainda a realimentação do modelo com dados reais ou dados estimados do modelo.

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* 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*, tem-se apresentado como uma alternativa promissora às redes neurais tradicionais na modelagem de sistemas não lineares, classificação, previsão e controle (JUNIOR, 2014).

Apesar de considerar que as WNNs foram popularizadas pelos trabalhos de SZU et al. (1992), ZHANG; BENVENISTE (1992) e BILLINGS; WEI (2005a) afirmam que a origem das WNNs vem do trabalho de DAUGMAN (1988), no qual *wavelets* são utilizadas na classificação e compressão de imagens. No entanto, atualmente, pesquisadores têm aplicado as redes WNNs com sucesso em diversas outras situações, a exemplo de CHENG et al. (2017), que estudou a viabilidade de utilizar o modelo de identificação baseado na rede neural WNN com a estrutura de identificação NARX por meio de simulação. A pesquisa compara e analisa os impactos de diferentes sinais de acionamento na precisão do modelo de uma Plataforma estabilizada por giroscópio ou, em língua inglesa, *Gyro-stabilized platform* (GSP). O resultado da simulação mostra que o método é viável.

Em 2016, KHODABANDEHLOU; FADALI, apresentou um controlador preditivo baseado num modelo neural *wavelet* com componente *feedforward* para identificação e controle *online* de um sistema não linear de um simulador de veículo autônomo não tripulado. O controlador preditivo usa a WNN para prever as saídas futuras da planta ao longo do horizonte de previsão estendido. Os resultados da simulação mostram que esta metodologia pode compensar o efeito de atraso de comunicação fixo e aleatório da rede de comunicação e perda de pacotes, além de fornecer um desempenho de rastreamento satisfatório. A teoria de *Lyapunov* é usada para provar a estabilidade do controlador preditivo.

Em 2015, OWA; SHARMA; SUTTON, projetaram um controlador preditivo baseado em um modelo neural não linear para um sistema de tanques acoplados de múltiplas

variáveis. A rede utilizada nesse trabalho foi a *Wavelets Neural Network*. O modelo não linear obtido mostrou eficácia no procedimento de identificação do sistema que permite uma ampla gama de capacidades de previsão. Os resultados mostraram ainda que o modelo WNN é bem adequado para executar em todas as regiões de controle operacionais e as capacidades de lidar com a rejeição de perturbações.

## 1.2 Justificativa

Analisando-se de forma geral as pesquisas desenvolvidas nos últimos anos no meio acadêmico, 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, em que as representações lineares são substituídas em algumas aplicações por seus correspondentes não lineares mais acurados, tornando-se possível analisar e reproduzir certos fenômenos e comportamentos dinâmicos mais complexos.

No entanto, 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, 2015).

Ao optar por uma representação não linear, existe ainda o desafio de obter modelos matemáticos partindo-se das equações que descrevem a física do processo, pois os sistemas com os quais se precisa lidar na prática são mais complexos, possuindo características que podem tornar inviável definir as equações básicas do sistema, como o forte acoplamento de sinais e ruídos de medição.

Diante disso, este trabalho tem como justificativa a aplicação das técnicas de identificação de sistemas para solucionar um problema com grande relevância na modelagem de um sistema: A obtenção de um modelo que represente toda dinâmica e faixa de operação de um sistema de nível acoplado, considerando a não linearidade do sistema em estudo, o acoplamento de sinais entre as variáveis de saída e ruídos de medição.

## 1.3 Hipóteses e Objetivos

As pesquisas apresentadas no tópico 1.1 deixam claro a eficiência das aplicações das redes neurais MLP e WNN na estimação de funções, em que por meio de uma série de valores observados, uma rede pode ser treinada para aprender a composição dessas funções e, assim, calcular um valor esperado para uma determinada saída.

Além disso, [LI; MINMING; HAIBO \(2010\)](#) afirmam que as redes WNNs possuem melhor capacidade de aproximação, de tolerância a falhas e de previsão, apresentam convergência de treinamento mais rápida e desempenho melhores do que as tradicionais redes neurais como, por exemplo, a MLP (*Multilayer Perceptron*).

Outra característica importante para o uso das redes WNN é o fato de que estas estruturas são aproximadores universais, capazes de realizar representações de funções com melhor exatidão. A comprovação deste fato tem origem nos trabalhos de [HORNIK; STINCHCOMBE; WHITE \(1989\)](#) e [KREINOVICH; SIRISAENGTAKSIN; CABRERA \(1989\)](#). Diante disso, na presente pesquisa levantou-se as seguintes hipóteses, quanto a aplicação dessas RNAs para modelagem do sistema de tanques de nível:

- a) **Hipótese 1:** A *Wavelet Neural Network* por ser bom aproximador universal, será capaz de realizar representações do sistema de tanques de nível acoplados.
- b) **Hipótese 2:** A *Wavelet Neural Network* apresentará boa exatidão para validação.
- c) **Hipótese 3:** A *Wavelet Neural Network* possui melhor convergência que a *Multilayer Perceptron*.

Assim, para avaliar essas hipóteses, o presente trabalho tem como objetivo geral realizar a identificação multivariável de um sistema com dois tanques de nível acoplados utilizando a *Wavelets Neural Network*, a fim de verificá-la como uma alternativa para identificação de sistemas multivariáveis não lineares. Para avaliar as pontencialidades da WNN, os resultados obtidos serão comparados com a identificação do referido sistema utilizando a rede neural *Multilayer Perceptron*.

Como objetivos específicos tem-se:

- a) Implementar as redes MLP e WNN como ferramentas para identificação da dinâmica do sistema de tanques em questão;
- b) Implementar a estrutura de modelagem do tipo NNARX;
- c) Realizar treinamento das redes neurais;
- d) Realizar a validação dos modelos neurais obtidos;
- e) Comparar os modelos neurais obtidos.

## 1.4 Organização da Dissertação

Esta dissertação é composta de cinco capítulos, incluindo este. No segundo capítulo é realizada a fundamentação teórica, ou seja, descreve-se todas as ferramentas e teorias necessárias para o desenvolvimento e compreensão do trabalho. Tal capítulo inicia com os aspectos das redes neurais artificiais utilizadas, descrevendo-se seus fundamentos e

arquiteturas. Em seguida é abordado a identificação de sistemas dinâmicos e por fim descreve-se as principais características do sinal *Pseudo Random Binary Sequence* (PRBS).

No capítulo seguinte, denominado de “Materiais e Métodos” é apresentado o sistema dinâmico de nível utilizado na pesquisa. Em seguida é introduzido a metodologia de identificação proposta neste trabalho, com as suas principais etapas descritas. São elas: coleta de dados; pré-processamento de dados; escolha da estrutura de identificação; estimação e validação dos modelos neurais obtidos.

No quarto capítulo, os resultados da modelagem utilizando a WNN são apresentados, avaliados e discutidos, comparando-se com o modelo neural MLP por meio de análise gráfica e tabelas.

Por fim, o último capítulo apresenta as conclusões do trabalho desenvolvido e direciona os rumos a serem seguidos em trabalhos futuros.

## 1.5 Trabalhos Publicados e Submetidos

- HERTZ SILVA, HEINRICH; ARAÚJO JÚNIOR, JOSÉ MEDEIROS; SANTANA, HEITOR GONÇALVES. *Identificação multivariável de um sistema de tanques de nível acoplados utilizando RNA*, **INDUSCON** (2018).



## 2 Fundamentação Teórica

Inicialmente este capítulo ilustra a teoria base para o entendimento, fundamentação e implementação de uma rede neural artificial, abordando-se os conceitos, o modelo básico de um neurônio e as funções de ativação (polarização) mais usuais. Em seguida descreve-se as arquiteturas e o processo de aprendizagem e validação das redes neurais *Multilayer Perceptron* e *Wavelet Neural Network*. No tópico 2.4 a atenção volta-se para a identificação de sistemas. São apresentadas algumas estruturas de identificação mais usuais e a representação de sistemas não lineares, ilustrando-se também a estrutura utilizada no presente trabalho.

Como trata-se de uma identificação conduzida por dados, faz-se necessário a fundamentação teórica do sinal de excitação do sistema. Neste trabalho, foi utilizado o sinal *Pseudo Random Binary Sequence* (PRBS), cujas características são abordadas neste capítulo.

### 2.1 Redes Neurais Artificiais

Embora os primeiros trabalhos em Redes Neurais Artificiais (RNA) tenham sido publicados há mais de 76 anos, com [MCCULLOCH; PITTS \(1943\)](#), tal tema começou a ser fortemente pesquisado a partir do final dos anos 1980, com o trabalho [RUMELHART; HINTON; WILLIAMS \(1986\)](#). As aplicações que envolvem a utilização de sistemas considerados inteligentes são as mais variadas possíveis, por exemplo: previsões de ações no mercado financeiro ([CARVALHO, 2018](#)); classificação de padrões usando visão computacional ([YUN; HUYEN; LU, 2018](#)); controle de trajetórias de robôs móveis ([MOHARERI, 2009](#)); classificação e diagnósticos de doenças ([KHAN et al., 2001](#)).

Mais especificamente, na engenharia, essa ferramenta é utilizada na classificação de padrões, filtragem de sinais, análise de imagens, controle e identificação de sistemas dinâmicos, sendo esta última a aplicação alvo desta dissertação ([NORGAARD et al., 2001](#)).

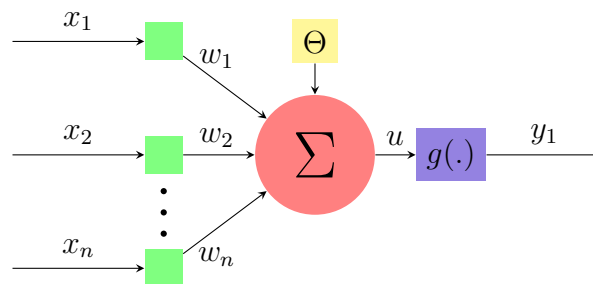
#### 2.1.1 Considerações Gerais

Segundo [SILVA; SPATTI; FLAUZINO \(2010\)](#), Redes Neurais Artificiais são modelos computacionais inspirados no sistema nervoso de seres vivos. Possuem a capacidade de aquisição e manutenção do conhecimento (baseado em informações) e podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um número grande de interconexões (sinapses artificiais).

No entanto, apesar das estruturas das RNAs terem como base de desenvolvimento o complexo sistema nervoso biológico, na prática, os neurônios artificiais são elementos computacionais bem mais simples quando comparado ao neurônio biológico.

O neurônio artificial que representa as características neural e biológica, isto é, paralelismo e alta conectividades, foi proposto por McCulloch & Pitts em 1943, sendo ainda o modelo mais utilizado nas diferentes arquiteturas, cuja estrutura é ilustrada pela Figura 1. (SILVA; SPATTI; FLAUZINO, 2010).

Figura 1 – Neurônio Artificial.



Fonte: Elaborada pelo autor.

Nessa representação, pode-se observar que os diversos sinais de entrada provenientes do meio externo são representados pelo conjunto  $\{x_1, x_2, \dots, x_n\}$ . Após receber os sinais de entrada, as ponderações são exercidas pelo conjunto de pesos sinápticos  $\{w_1, w_2, \dots, w_n\}$  e o combinador linear agrega todos os sinais, realizando a soma ponderada de suas entradas, gerando como saída o potencial de ativação  $\{u\}$ . Por fim, tal potencial de ativação é submetido a uma função de ativação  $g(\cdot)$ . O limiar de ativação  $\{\Theta\}$  é uma variável que especifica qual será o patamar apropriado para que o resultado produzido pelo combinador linear possa gerar um valor de disparo em direção à saída do neurônio.

As funções de ativação são divididas em funções parcialmente diferenciáveis e funções totalmente diferenciáveis. As primeiras possuem pontos cujas derivadas de primeira ordem não existem. As três principais funções deste grupo são: função degrau, função degrau bipolar e função rampa simétrica (SILVA; SPATTI; FLAUZINO, 2010).

Já as funções de ativação totalmente diferenciáveis, possuem derivadas de primeira ordem e são conhecidas em todos os pontos de seu domínio de definição. As quatro principais funções desse grupo são: função logística, tangente hiperbólica, gaussiana e a função linear (HAYKIN, 2003).

#### a) Função logística

O resultado de saída produzido pela aplicação da função logística assumirá sempre valores reais entre zero e um, tendo-se sua expressão matemática dada

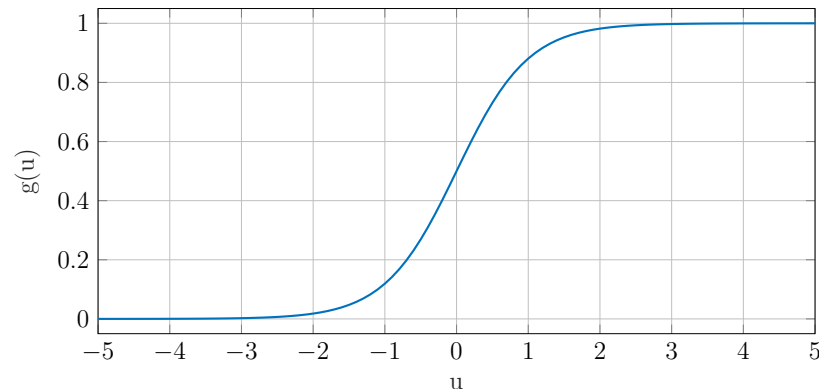


por 2.1.

$$g(u) = \frac{1}{1 + e^{-\beta \cdot u}}, \quad (2.1)$$

onde  $\beta$  é uma constante real relacionada com a inclinação da curva logística. A Figura 2 ilustra a função de ativação logística.

Figura 2 – Função de Ativação Logística.



Fonte: Elaborada pelo autor.

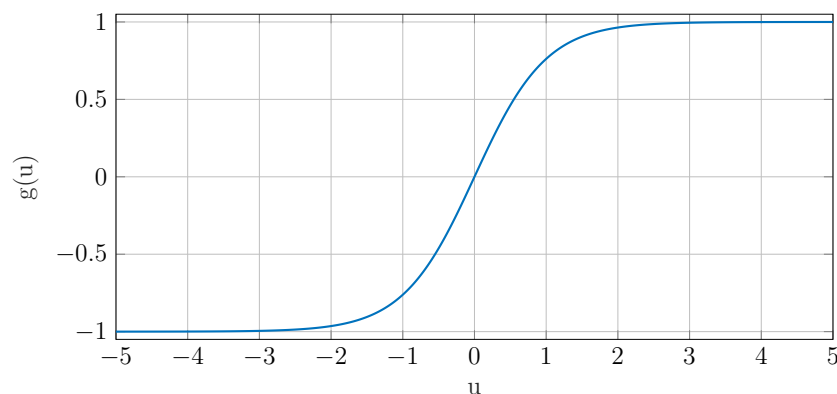
b) Função tangente hiperbólica

Nesta função, diferentemente da função logística, o resultado de saída ficará entre os valores reais -1 e +1, cuja expressão matemática é definida por 2.2.

$$g(u) = \frac{1 - e^{-\beta \cdot u}}{1 + e^{-\beta \cdot u}}, \quad (2.2)$$

onde  $\beta$  é uma constante real relacionada com a inclinação da curva tangente hiperbólica. A Figura 3 ilustra a função descrita.

Figura 3 – Função de Ativação Tangente Hiperbólica.



Fonte: Elaborada pelo autor.

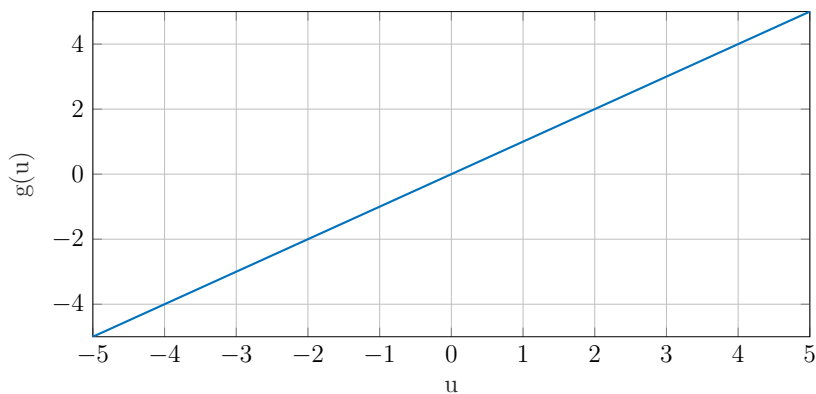
## c) Função Linear (Identidade)

Tal função produz resultados de saída idênticos aos valores do potencial de ativação  $u$ , tendo sua função matemática definida por 2.3.

$$g(u) = u. \quad (2.3)$$

Uma das aplicabilidades dessa função está na utilização de redes neurais artificiais como aproximadoras universais de funções, visando-se mapear o comportamento entre as variáveis de entrada e saída de processos. A representação gráfica dessa função é ilustrada a seguir, Figura 4.

Figura 4 – Função de Ativação Linear (ou Identidade).



Fonte: Elaborada pelo autor.

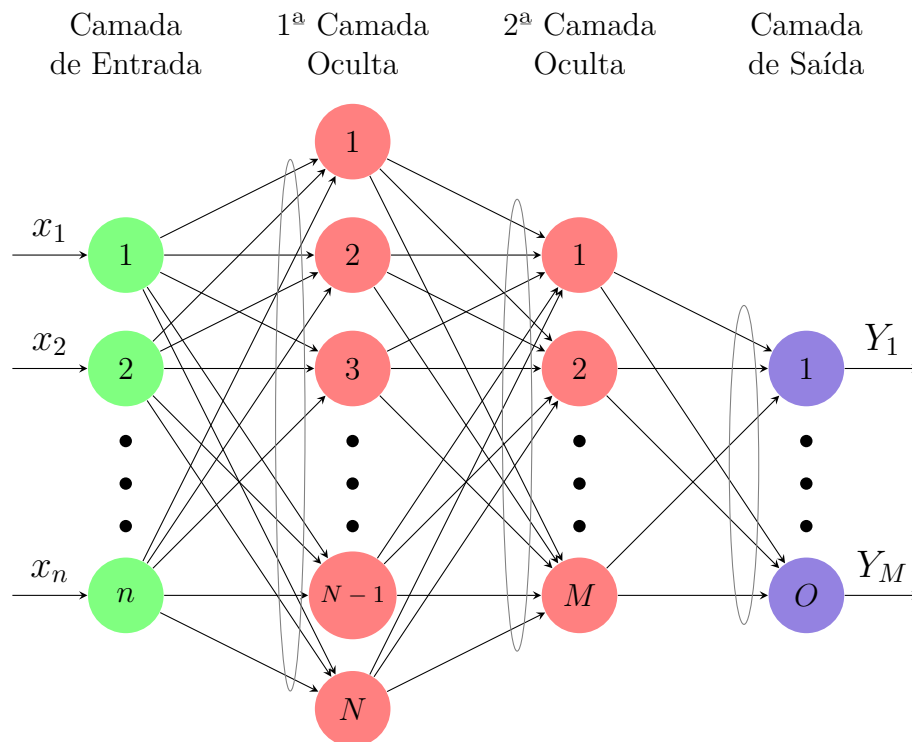
Segundo HAYKIN (2003), uma rede neural artificial pode ser dividida em três partes, denominadas de camadas, as quais são classificadas de acordo com suas respectivas funções em: camada de entrada, camada oculta (ou intermediária) e camada de saída.

A camada de entrada possui a função de receber as informações (dados, sinais ou amostras) advindas do meio externo. Já as camadas escondidas (ocultas), são aquelas compostas de neurônios que tem por função extrair as características associadas ao processo ou sistema a ser inferido. Quase todo o processo interno da rede é realizado nessas camadas. Por fim, a camada de saída é constituída de neurônios com a função de produção e apresentação dos resultados finais da rede, provenientes das camadas ocultas.

A arquitetura de uma RNA define a forma como os seus neurônios estão dispostos, uns em relação aos outros. Essa disposição dos neurônios está relacionada com o direcionamento das conexões sinápticas. Assim, as estruturas podem ser divididas em: redes *feedforward* (alimentação à frente) de camada simples, redes *feedforward* de camadas múltiplas, redes recorrentes, e redes reticuladas (JUNIOR; COSTA; MONTGOMERY, 2007).

O presente trabalho se limitará a discutir a arquitetura *feedforward* de camadas múltiplas, pois foi a estrutura utilizada para o desenvolvimento da pesquisa. De acordo com [SILVA; SPATTI; FLAUZINO \(2010\)](#), tal estrutura é empregada na solução de diversos problemas, tais como aqueles relacionados à aproximação de funções, classificação de padrão, identificação de sistemas, entre outros. A Figura 5 ilustra a arquitetura *feedforward* de camadas múltiplas.

Figura 5 – Exemplo de rede *feedforward* de camadas múltiplas.



Fonte: Elaborada pelo autor.

Como observa-se na Figura 5, a quantidade de neurônios que compõem a camada oculta é normalmente diferente do número de sinais que compõem a camada de entrada. Já na camada de saída, o número de neurônios coincide com o número de saídas do sistema.

Geralmente, o número de neurônios das camadas escondidas só pode ser determinado por meio de testes, dependendo sobretudo do tipo e da complexidade do problema a ser mapeado pela rede, assim como da quantidade e da qualidade dos dados disponíveis ([SILVA; SPATTI; FLAUZINO, 2010](#)).

Como já discutido anteriormente, as RNAs possuem a capacidade de mapear e aprender a partir de dados do comportamento do sistema. Para isso é necessário realizar o treinamento da RNA, pois só após a etapa de treinamento a mesma deverá ser capaz de generalizar soluções para novos dados de entrada ([HAYKIN, 2003](#)). De acordo com [SILVA; SPATTI; FLAUZINO \(2010\)](#), o processo de treinamento é definido da seguinte forma:

O processo de treinamento de uma rede neural consiste da aplicação de passos ordenados que sejam necessários para sintonização dos pesos sinápticos e limiares de seus neurônios, tendo-se como objetivo final a generalização de soluções a serem produzidas pelas suas saídas, cujas respostas são representativas do sistema físico em que estas estão mapeando.

Para que o processo de treinamento tenha resultados satisfatórios é necessário destinar uma margem considerável do conjunto total de amostras para o treinamento da rede neural. Diante disso, costuma-se dividir os dados em subconjunto de teste e subconjunto de treino. O subconjunto de treinamento é composto por cerca de 60 a 90% das amostras do conjunto total. Já o conjunto de teste, geralmente é formado por 10 a 40% do conjunto total de amostras (SILVA; SPATTI; FLAUZINO, 2010).

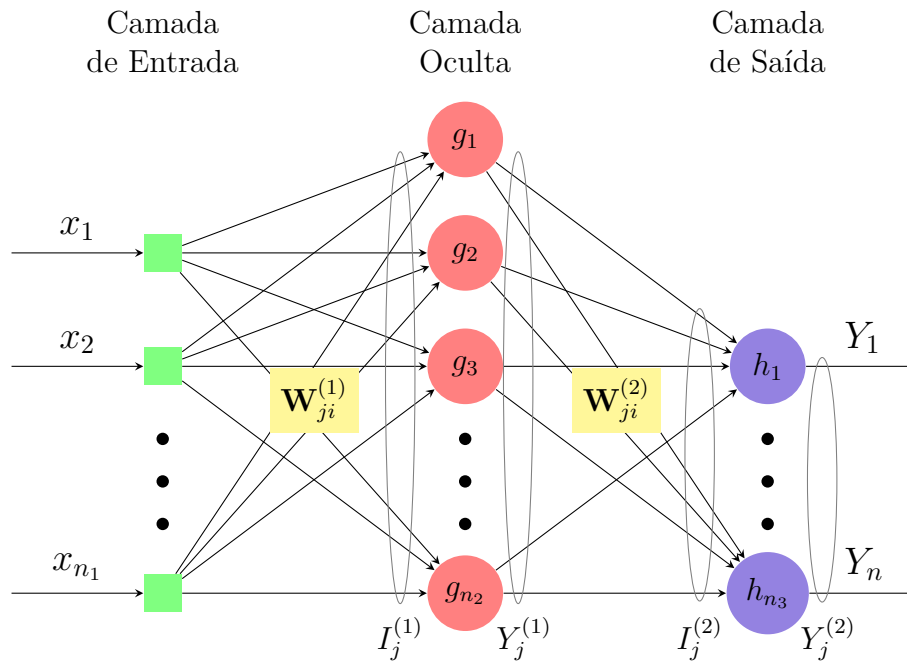
Segundo HAYKIN (2003), existem duas formas de treinar uma rede neural: treinamento supervisionado e o treinamento não-supervisionado. No treinamento supervisionado existe uma resposta desejada que avalia a resposta da rede ao padrão atual de entrada. As alterações dos pesos são calculadas de forma que a resposta da rede se aproxime da resposta desejada. É o tipo de aprendizagem normalmente utilizada para treinar arquiteturas *feedforward*. Já no treinamento não supervisionado não existe saídas desejadas. A rede tem de descobrir sozinha relações, padrões, regularidades ou categorias nos dados que lhes vão sendo apresentados e codifica-os nas saídas. Os pesos sinápticos e limiares dos neurônios desse tipo de treinamento são ajustados pelo algoritmo de aprendizado de forma a refletir esta representação internamente dentro da própria rede. A partir da subseção 2.1.2 será dado ênfase às redes neurais utilizadas na presente pesquisa.

### 2.1.2 *Multilayer Perceptron*

Como afirmado anteriormente, as redes MLP são caracterizadas por sua grande aplicabilidade, nas mais diversas áreas. Entre essas potenciais áreas, têm-se: aproximação universal de funções; reconhecimento de padrões; identificação e controle de processos; previsão de séries temporais, otimização de sistemas, entre outros (SILVA; SPATTI; FLAUZINO, 2010).

Conforme observado na Figura 6, o fluxo de informações na estrutura da rede se inicia na camada de entrada, percorre em seguida as camadas intermediárias, sendo então finalizado na camada neural de saída. Pode-se perceber também, que inexiste qualquer tipo de realimentação de valores produzidos pela camada neural de saída ou pelas próprias camadas intermediárias.

O ajuste dos pesos e do limiar de cada um dos neurônios da rede MLP é efetuado utilizando-se treinamento supervisionado, isto é, para cada amostra dos dados de entrada obtém-se a respectiva saída desejada. Segundo SILVA; SPATTI; FLAUZINO (2010), tal processo de treinamento pode ser realizado utilizando-se o algoritmo *Backpropagation*,

Figura 6 – Ilustração de rede *Multilayer Perceptron*.

Fonte: Elaborada pelo autor.

conhecido também como regra Delta Generalizada, que pode ser dividido em duas etapas: *Forward* e *Backward*. As duas etapas serão descritas a seguir.

a) Etapa de Propagação Adiante (*Forward*)

Nessa etapa, os sinais de entrada (amostras) do conjunto de treinamento são inseridos nas entradas da rede e são propagados camada a camada até a produção das respectivas saídas. Assim, tomando-se como base as variáveis da Figura 6, após a apresentação de um vetor de entrada  $\mathbf{x}$ , na iteração  $t$ , o primeiro passo é calcular as ativações dos neurônios da camada escondida por meio da Equação 2.4.

$$\mathbf{I}_j^{(1)}(t) = \sum_{i=1}^{n_1} \mathbf{W}_{ji}^{(1)}(t) \cdot \mathbf{x}_i(t), \quad (2.4)$$

cujos elementos de  $\mathbf{I}_j^{(1)}$  denotam a entrada ponderada em relação ao  $j$ -ésimo neurônio da camada oculta. Em seguida, as saídas correspondentes são calculadas como:

$$\mathbf{Y}_j^{(1)}(t) = g_j(\mathbf{I}_j^{(1)}(t)) = g_j \left( \sum_{i=1}^{n_1} \mathbf{W}_{ji}^{(1)}(t) \cdot \mathbf{x}_i(t) \right), \quad (2.5)$$

tal que a função de ativação  $g$  adotada, geralmente é a função logística ou a função tangente hiperbólica.

Uma vez calculado o vetor  $\mathbf{Y}_j^{(1)}$ , a próxima etapa é calcular  $\mathbf{I}_j^{(2)}$ , cujos elementos denotam a entrada ponderada em relação ao  $j$ -ésimo neurônio da camada de saída, Equação 2.6.

$$\mathbf{I}_j^{(2)}(t) = \sum_{i=1}^{n_2} \mathbf{W}_{ji}^{(2)}(t) \cdot \mathbf{x}_i(t). \quad (2.6)$$

Logo em seguida as respostas produzidas pelas saídas da rede são definidas, por meio da Equação 2.7. Estas são comparadas com as respectivas respostas desejadas que estejam disponíveis.

$$\mathbf{Y}_j^{(2)}(t) = h_j(\mathbf{I}_j^{(2)}(t)) = h_j \left( \sum_{i=1}^{n_2} \mathbf{W}_{ji}^{(2)}(t) \cdot \mathbf{x}_i(t) \right), \quad (2.7)$$

tal que a função de ativação  $h$  adotada, geralmente é a função linear.

As respostas desejadas, no presente estudo, correspondem às saídas reais de sistemas dinâmicos a serem identificados. Essa comparação gera um erro, o qual será subsequentemente utilizado para ajustar os pesos e limiares de todos os neurônios.

b) Etapa de Propagação Reversa (*Backward*)

Esta etapa de funcionamento da rede MLP envolve o cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada escondida e da camada de saída. Assim, o fluxo de sinais ocorre dos neurônios de saída para os neurônios da camada de entrada, passando pelos neurônios da camada oculta. Diferentemente da etapa anterior, as alterações nos pesos sinápticos e limiares de todos os neurônios da rede são executados no decorrer desta fase.

Assim, após os cálculos das ativações e saídas na etapa *forward*, o primeiro passo da fase *Backward* consiste em calcular os gradientes locais dos neurônios da camada de saída, dado pela Equação 2.8.

$$\delta_j^{(2)}(t) = (\mathbf{d}_j(t) - \mathbf{Y}_j^{(2)}(t)) \cdot g'(\mathbf{I}_j^{(2)}(t)) \quad (2.8)$$

onde  $d_j(t)$  é a saída desejada para o neurônio  $j$ .

A derivada assume diferentes formas dependendo da escolha da função de ativação. Logo, para a função linear, tem-se a Equação 2.9:

$$h'(\mathbf{Y}) = \frac{\partial h(\mathbf{Y})}{\partial \mathbf{I}} = c, \quad (2.9)$$

onde  $c$  é uma constante.

Com isso, o segundo passo da etapa de propagação reversa consiste em calcular os gradientes locais dos neurônios da camada de saída utilizando justamente a Equação 2.9.

O terceiro passo corresponde ao processo de atualização ou ajuste dos parâmetros (pesos sinápticos) que ponderam as entradas dos neurônios camada de saída da rede MLP. Para isso, utiliza-se a seguinte regra de atualização dos pesos,  $W_{ji}$ , dada por:

$$\begin{aligned} \mathbf{W}_{ji}^{(2)}(t+1) &= \mathbf{W}_{ji}^{(2)}(t) + \Delta \mathbf{W}_{ji}^{(2)}(t) \\ &= \mathbf{W}_{ji}^{(2)}(t) + \eta \cdot \delta_i^{(2)}(t) \cdot \mathbf{Y}_j^{(2)}(t), \end{aligned} \quad (2.10)$$

onde  $\eta$  é a taxa de aprendizagem.

Portanto, a Equação 2.10 ajusta os pesos dos neurônios da camada de saída da rede levando-se em consideração a diferença observada entre as respostas produzidas por suas saídas em relação aos respectivos valores desejados.

Diferentemente dos neurônios pertencentes à camada de saída da rede MLP, para os neurônios das camadas intermediárias não se tem acesso de forma direta aos valores desejados para as suas saídas. Nesta situação, os ajustes de seus pesos são efetuados por intermédio de estimativas dos erros de saída produzidos por aqueles neurônios da camada imediatamente posterior, os quais já foram previamente ajustados (SILVA; SPATTI; FLAUZINO, 2010). Logo, calcula-se o gradiente local  $\delta_j^{(1)}$  em relação ao  $j$ -ésimo neurônio da camada intermediária por meio da expressão 2.11.

$$\delta_j^{(1)}(t) = (\mathbf{d}_j(t) - \mathbf{Y}_j^{(1)}(t)) \cdot g'(\mathbf{I}_j^{(1)}(t)) \quad (2.11)$$

Neste caso a derivada assume diferentes formas dependendo da escolha da função de ativação. Logo, para as funções logística e tangente hiperbólica, tem-se as Equações 2.12 e 2.13, respectivamente.

$$g'(\mathbf{Y}) = \frac{\partial g(\mathbf{Y})}{\partial \mathbf{I}} = \mathbf{Y} \cdot [1 - \mathbf{Y}], \quad (2.12)$$

$$g'(\mathbf{Y}) = \frac{\partial g(\mathbf{Y})}{\partial \mathbf{I}} = [1 - \mathbf{Y}^2], \quad (2.13)$$

Por fim, realiza-se a atualização ou ajuste dos parâmetros (pesos sinápticos) que ponderam as entradas dos neurônios camada de intermediária da rede MLP. Para isso, utiliza-se a seguinte regra de atualização dos pesos,  $W_{ji}$ , dada por:

$$\begin{aligned}\mathbf{W}_{ji}^{(1)}(t+1) &= \mathbf{W}_{ji}^{(1)}(t) + \Delta \mathbf{W}_{ji}^{(1)}(t) \\ &= \mathbf{W}_{ji}^{(1)}(t) + \eta \cdot \delta_i^{(1)}(t) \cdot \mathbf{Y}_j^{(1)}(t),\end{aligned}\tag{2.14}$$

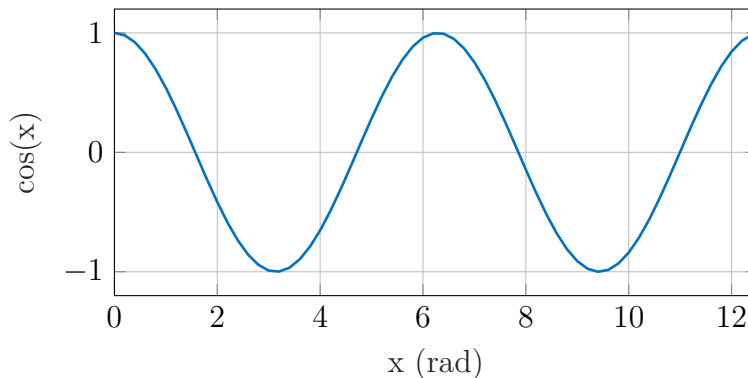
## 2.2 Funções *Wavelets*

Antes de fundamentar as redes neurais *wavelets*, é importante conceituar e contextualizar as funções *wavelets*, que vem despertando interesses em várias áreas. O rápido avanço da teoria dessas funções deve-se, 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. Assim, as *wavelets* estão sendo utilizadas, por exemplo, na mecânica de fluídos, computação numérica, análise de imagens, processamento de sinais, sistemas de controle, fenômenos biológicos, medicina e psicologia (VEITCH, 2005).

Uma *wavelet* é uma forma de onda de duração limitada que tem um valor médio de zero. Essa função cresce e decai em um período de tempo finito apresentando formas irregulares e assimétricas, ao contrário dos sinais sinusóides, que teoricamente se estendem de menos a mais infinito, já as *wavelets* têm um começo e um fim. Por isso, as *wavelets* são melhores em descrever anomalias, pulsos e outros eventos que iniciam e param dentro do sinal. Assim, define-se *wavelet* como sendo uma forma de onda com duração limitada e um valor médio igual a zero (FUGAL, 2009).

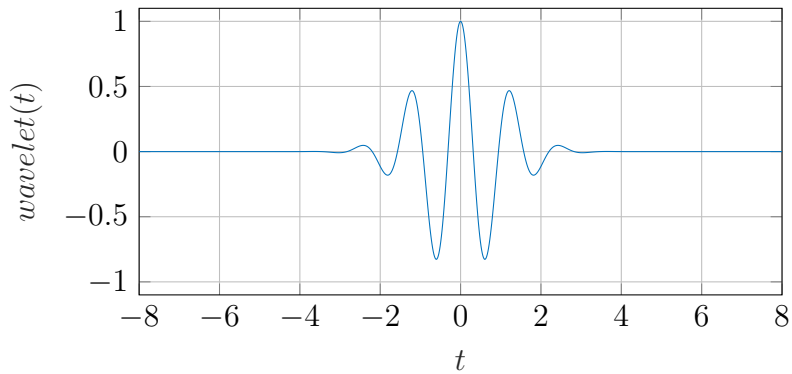
Uma porção de uma função senóide infinitamente longa e uma *wavelet* de comprimento finito, são ilustradas nas Figuras 7 e 8, respectivamente. Observe que a senóide tem uma frequência facilmente discernível, enquanto a *wavelet* tem uma pseudo-frequência que varia sobre o seu comprimento (ROCHA, 2008).

Figura 7 – Função Cosseno.



Fonte: Elaborada pelo autor.



Figura 8 – Função *Wavelet*.

Fonte: Elaborada pelo autor.

Em termos matemáticos, para que uma função seja considerada do tipo *wavelet*, ela tem de satisfazer a certas propriedades matemáticas (VEITCH, 2005). Essas propriedades são citadas a seguir:

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

$$\int_{+\infty}^{-\infty} \psi(t) dt = 0. \quad (2.15)$$

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

$$\int_{+\infty}^{-\infty} \psi^2(t) dt = 1. \quad (2.16)$$

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

$$\int_{+\infty}^{-\infty} |\psi(t)|^2 dt < \infty. \quad (2.17)$$

Considerando a comparação supracitada, entre as funções *wavelets* e funções senoidais, a literatura também faz analogia entre a análise de *wavelets* e a análise de séries de *Fourier*. De acordo com VEITCH (2005), na análise de *Fourier* um sinal original é decomposto em ondas senoidais de várias frequências – via transformada de *Fourier*. Da mesma forma, na análise de *wavelets*, esse mesmo sinal original pode ser decomposto em versões “deslocadas” e “escaloadas” – via transformada de *wavelets*. Assim, um sinal original é representado por versões de uma função mãe comprimida ou alongada através de seu parâmetro de dilatação e/ou com retardo no tempo, por meio de seu parâmetro de translação.

Assim, seja  $\psi(x)$  uma função *wavelet* mãe,  $R$  um espaço de variáveis reais e  $R^n$  seja um espaço  $n$ -dimensional de variáveis reais, a função pode ser deslocada e escalonada  $\psi\left(\frac{x-t}{\lambda}\right)$ , onde  $t$ , chamado de parâmetro de translação, é um vetor real  $n$ -dimensional no

mesmo espaço que  $x$ , e  $\lambda$  é um escalar chamado de parâmetro de dilatação. A transformada *wavelet* de uma função  $f(x)$  é dada por:

$$\tau(\lambda, t) = \int_{R^n} f(x) \lambda^{-n/2} \psi\left(\frac{x-t}{\lambda}\right) dx. \quad (2.18)$$

A transformada *wavelet* transforma a função do domínio original (domínio  $x$ ) em um domínio *wavelet* (domínio  $\lambda, t$ ). Uma função  $f(x)$  possui tanto variações globais suaves quanto variações locais, podendo ser efetivamente representada no domínio *wavelet* por uma função *wavelet* correspondente a  $\tau(\lambda, x)$ . A função original  $f(x)$  pode ser recuperada da função *wavelet* usando uma transformada *wavelet* inversa, definida como:

$$f(\mathbf{x}) = \frac{1}{C_\psi} \int_{0^+}^{\infty} \lambda^{-(n+1)} \int_{R^n} \tau(\lambda, t) \lambda^{-n/2} \psi\left(\frac{x-t}{\lambda}\right) dt d\lambda. \quad (2.19)$$

Em que o parâmetro  $C_\psi$  necessita ser finito e positivo, para que ao operar-se a transformação sobre o sinal, mantém-se informação suficiente para que este possa ser reconstituído. Esta condição permite a existência da transformada inversa de *wavelet* e é chamada de condição de admissibilidade.

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 tempo-frequência. Essa característica é uma vantagem em relação a outros métodos baseados em séries ortogonais, tais como as análises de *Fourier*, em que a representação limita-se às 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 (JUNIOR, 2014).

Já as funções *wavelets*, devido aos parâmetros de dilatação e translação, possuem uma flexibilidade que permite a análise do sinal em frequências diferentes e resoluções diferentes, o que caracteriza uma forma de análise conhecida por Análise Multirresolucional (AMR). Tal propriedade permite que diferentes frequências sejam relevantes para distintos subintervalos do tempo, resultando em uma melhor adaptação temporal, possibilitando que sejam vistos tanto características globais quanto os detalhes locais de um sinal (ROCHA, 2008).

São essas as principais razões que tornam a metodologia *wavelet* extensivamente usada em diversas aplicações, pois a capacidade de discernir tempo e frequência e até mesmo a forma geral de um evento transitório, torna essa metodologia uma ferramenta poderosa que permite explorar completamente qualquer sinal que contenha eventos que iniciem e parem e/ou alterem a frequência (ROCHA, 2008).

Uma vez apresentada a definição e vantagens das funções *wavelets*, agora é possível comentar sobre as principais famílias de *wavelets* e como as propriedades específicas levam a algumas aplicações práticas.

O termo função *wavelet* é usado genericamente para se referir a *wavelets* ortogonais ou não ortogonais. Um conjunto ortogonal de *wavelets* é chamado de *wavelet basis*, e um conjunto de *wavelets* não ortogonais é denominado *wavelet frame*. O uso de uma base ortogonal implica o uso da Transformada Discreta Wavelet (TDW), enquanto as não ortogonais podem ser usados com a transformada discreta ou Transformada Contínua Wavelet (TCW) (ALEXANDRIDIS; ZAPRANIS, 2014). As *Wavelets* Gaussianas, *Morlet* e *Mexican Hat* são funções *Wavelets Frame*.

Funções *Wavelets* Complexas são caracterizadas por possuírem coeficientes complexos na equação matemática explícita, por isso têm capacidade de fornecer informação da amplitude e da fase e é mais bem adaptada para capturar comportamentos oscilatórios comuns, por exemplo, em séries temporais. Além disso, são mais indicadas para análise de sinais complexos. *Wavelets* deste tipo incluem *Shannon*, *Complex Frequency B-Spline*, *Complex Morlet* e *Complex Gaussian* (MALLAT, 2009).

Já as *Wavelets* Ortogonais e Biortogonais são mais adequadas quando precisa-se realizar algum tipo de redução de informação ou quando a ortogonalidade é um fator de importância. Entre funções deste tipo, além das ortogonais já citadas, têm-se: *Haar*, *Biorthogonal Wavelets* e *Reverse Biorthogonal Wavelets*. Geralmente, o TDW é usado para compressão de dados se o sinal já é amostrado, e o TCW é usado para análise de sinal (FUGAL, 2009).

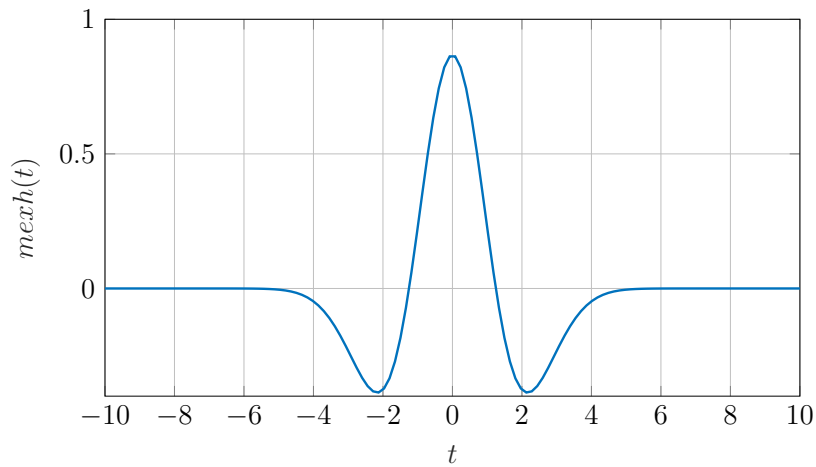
De todas as funções supracitadas, aquelas que são preferencialmente usadas em aplicações com redes neurais são as *Mexican Hat*, *Morlet*, *Gaussian* e *Meyer* por serem mais eficientes nas análises de sinais reais, extração e detecção de características (DOMINGUES et al., 2016). Além disso, são funções continuamente diferenciáveis, característica necessária para a utilização em algoritmos de aprendizagem que envolve o uso de derivadas como, por exemplo, o *backpropagation*. Estas *wavelets* serão agora discutidas.

#### a) *Mexican Hat*

Esta função é um bom exemplo de uma *wavelet* infinitamente regular, conforme discutido anteriormente. A equação matemática explícita para a *Wavelet Mexican Hat* é dada pela Equação 2.20 e a representação gráfica pela Figura 9 (FUGAL, 2009).

$$mexh(t) = \left( \frac{2}{\sqrt{3} \cdot \pi^{0,25}} \right) \cdot \exp\left(\frac{-t^2}{2}\right) \cdot (1 - t^2). \quad (2.20)$$

Pode-se observar que essa *wavelet* é simétrica no tempo e sua atuação efetiva

Figura 9 – Função *Wavelet Mexican Hat*.

Fonte: Elaborada pelo autor.

varia entre  $t = -5$  e  $t = 5$ . Na verdade, tal equação é resultante de uma função que é propriamente a 2ª derivada da Densidade de Probabilidade Gaussiana, por isso a simetria (FUGAL, 2009).

Tendo as características de suavidade (regularidade), simetria e decadência muito rápida, a *wavelet Mexican Hat* é frequentemente usada na análise da visão, porque essas características são semelhantes às do olho humano (FUGAL, 2009).

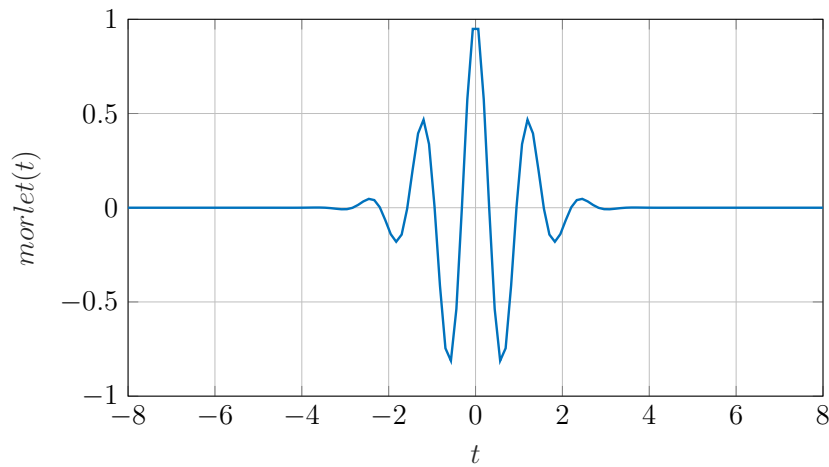
#### b) *Wavelet Morlet*

Nomeada por Jean Morlet, um geofísico e um dos primeiros inventores de *wavelets*, esta foi a primeira *wavelet* (depois do Haar) a ser desenvolvida e é outro exemplo de uma *wavelet* com coeficientes reais. A *Morlet* compartilha a maioria das características da *Mexican Hat*. São definidas a partir de sua equação matemática explícita (Equação 2.21), é simétrica, infinitamente regular e, conforme Figura 10, tem um atuação efetiva de  $t = -4$  a  $t = +4$  (FUGAL, 2009).

$$morlet(t) = \exp\left(\frac{-t^2}{2} \cdot \cos(5t)\right). \quad (2.21)$$

Com suavidade e periodicidade, a função *Morlet* é bom para análise de dados periódicos ou continuamente variáveis. Exemplos do uso dessa *wavelet* incluem a análise de pulsos sinusoidais e de índices atmosféricos (tais como mudanças cíclicas na pressão do ar) (FUGAL, 2009).

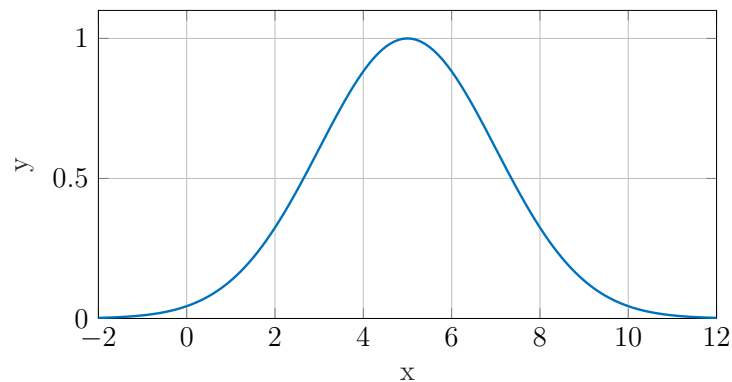
#### c) *Gaussian Wavelets*

Figura 10 – Função *Wavelet Morlet*.

Fonte: Elaborada pelo autor.

As funções *Gaussian Wavelets* têm origem nas derivadas na função densidade de probabilidade gaussiana. No entanto, a função de densidade de probabilidade gaussiana, exemplificada pela Figura 11, não é uma *wavelet*. A bem conhecida equação para a função gaussiana é mostrada na Equação 2.22 (FUGAL, 2009).

Figura 11 – Função Densidade de Probabilidade Gaussiana.

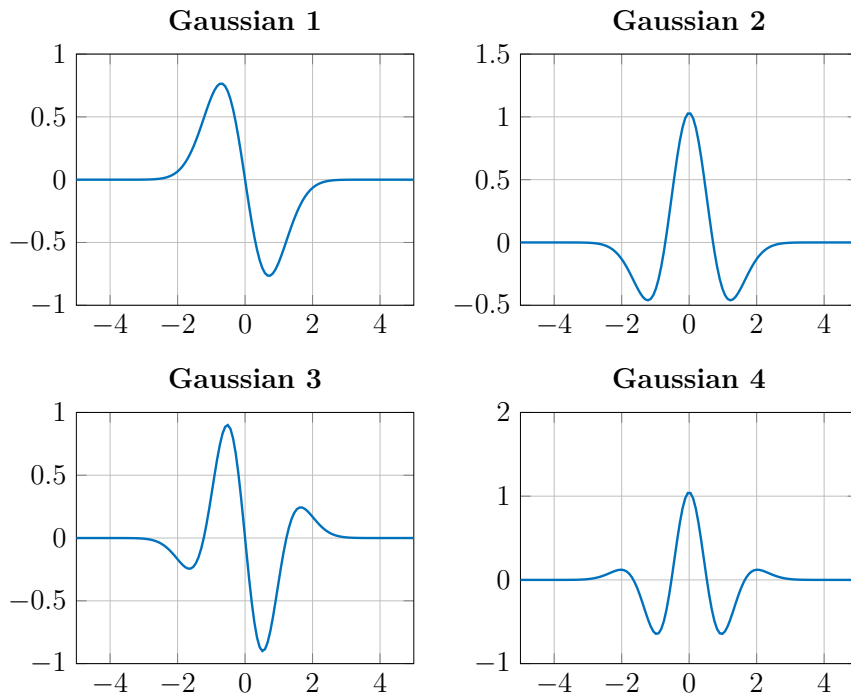


Fonte: Elaborada pelo autor.

$$gaussian(t) = \exp(-t^2). \quad (2.22)$$

Da primeira até a quarta derivada da função gaussiana (normalizada) são apresentadas nos próximos 4 gráficos, Figura 12. O gráfico final, por exemplo, pode ser referido como uma *wavelet Gaussian 4*.

Analisando os gráficos das derivadas, Figura 12, pode-se ver que um padrão se repete para as derivadas apresentadas. As *Gaussian Wavelets* pares (2, 4, 6, etc) são simétricas e as ímpares (1, 3, 5, etc) são assimétricas. Este processo pode

Figura 12 – Funções *Wavelets* Gaussianas.

Fonte: Elaborada pelo autor.

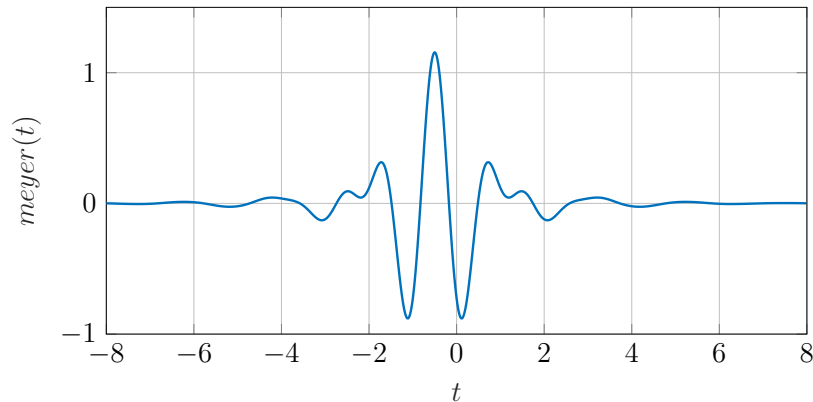
ser continuado com mais derivadas para produzir *wavelets* gaussianas adicionais. Além disso, conforme informado anteriormente, o gráfico da Gaussiana 2 é bem semelhante como uma *wavelet* de *Mexican Hat*.

As *Wavelets* Gaussianas, de forma geral, variam suavemente e têm boa regularidade. Uma *wavelet* gaussiana de baixo número (*Gaussian 1* ou *2*) pode ser representada por um número pequeno de pontos. Uma *wavelet* gaussiana de alto número requer mais pontos, mas fornece melhor resolução de frequência. Essa flexibilidade torna a família gaussiana uma boa escolha para aplicações como música, fala e tomografia (FUGAL, 2009).

#### d) *Wavelet Meyer*

Batizadas com o nome do pioneiro das *wavelets* Yves Meyer, vencedor do Prêmio Nobel 2017, essas *wavelets* são definidas no domínio da frequência e não pelas equações do domínio do tempo que foram apresentadas anteriormente para as outras *wavelets*. Embora teoricamente infinito no domínio do tempo, a atuação efetiva ocorre entre -8 a +8, conforme mostra a Figura 13 (FUGAL, 2009).

As *wavelets* Meyer compartilham as mesmas propriedades das outras *wavelets* reais, simétricas e infinitamente regulares. No entanto, o pico é ligeiramente à esquerda de zero. Além disso, essa função é definida no domínio da frequência em termos de uma função auxiliar  $\nu$  como:

Figura 13 – Funções *Wavelets Meyer*.

Fonte: Elaborada pelo autor.

$$\psi(\omega) := \begin{cases} \frac{1}{\sqrt{2\pi}} \sin\left(\frac{\pi}{2}\nu\left(\frac{3|\omega|}{2\pi} - 1\right) e^{j\omega/2}\right) & \text{se } 2\pi/3 < |\omega| < 4\pi/3, \\ \frac{1}{\sqrt{2\pi}} \cos\left(\frac{\pi}{2}\nu\left(\frac{3|\omega|}{4\pi} - 1\right) e^{j\omega/2}\right) & \text{se } 4\pi/3 < |\omega| < 8\pi/3, \\ 0 & \text{caso contrário,} \end{cases}$$

em que:

$$\nu(x) := \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } 0 < x < 1, \\ 1 & \text{if } x > 1. \end{cases}$$

Há muitas maneiras de definir esta função auxiliar, cada uma delas resultando em uma variante da família de *wavelets* de Meyer.

## 2.3 Wavelet Neural Network

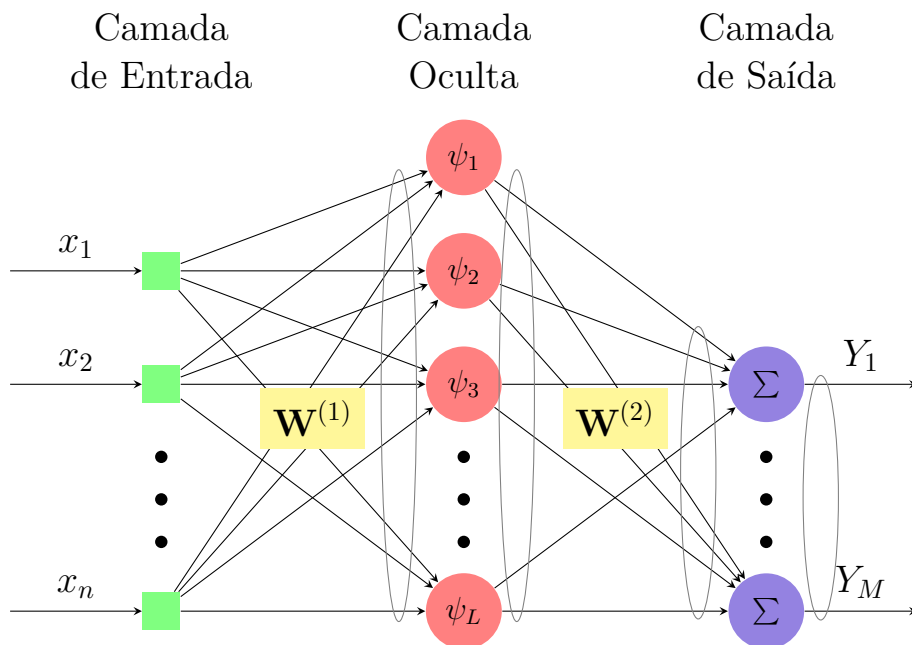
Devido às propriedades das *wavelets* e das redes neurais, a ideia de combinar essas duas técnicas se apresenta quase que naturalmente (MINU; LINEESH; JOHN, 2010). As WNNs herdam as vantagens fornecidas por estas duas estruturas, incluindo as características de análise de sinais em diferentes frequências e resoluções fornecidas pelas *wavelets* e as características de adaptação, robustez 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; SINGH; SRIVASTAVA, 2010).

Na literatura, várias versões de redes neurais *wavelets* foram propostas e geralmente têm a composta por três camadas. O modo como as três camadas estão conectadas e interagem, definem a estrutura da rede. Na camada de entrada as variáveis são inseridas, dependendo da estrutura utilizada essas entradas serão ponderadas ou não por pesos

sinápticos antes de serem propagadas para a camada oculta. A camada oculta consiste em neurônios cujas funções de ativação são extraídas de uma base de família *wavelets*. Esses neurônios *wavelets* são comumente chamados de *wavelons*. Finalmente, todos os *wavelons* são combinados para produzir a saída da rede  $Y$ , na camada de saída, que é uma aproximação do valor desejado (VEITCH, 2005).

A Figura 14 ilustra uma Rede Neural *Wavelet*, e é fácil perceber que a estrutura de uma WNN com múltiplas camadas é muito semelhante à de uma rede neural tradicional. Isto é, uma RNA *feedforward*, recebendo uma ou mais entradas, com uma camada oculta e cuja camada de saída consiste em um ou mais combinadores lineares.

Figura 14 – Estrutura da *Wavelet Neural Network*.



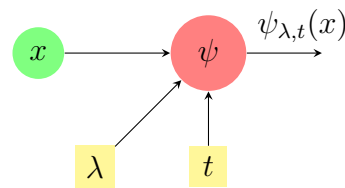
Fonte: Elaborada pelo autor.

Segundo JUNIOR (2014), outra característica importante das WNNs é a forma como os parâmetros internos dos neurônios da camada oculta desse tipo de rede são definidos, ou seja, como são ajustados os parâmetros de dilatação e de translação das funções de ativação *wavelets*. Dependendo como essa parametrização é feita, as redes WNN podem ser classificadas em WNNs de grade fixa e WNNs adaptativas.

As 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. Já as WNNs 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.

Conforme se observa na Figura 15, a função de ativação do neurônio é uma função *wavelet*, cujos parâmetros de entrada incluem os coeficientes de dilatação e translação.



Figura 15 – Neurônio *Wavelet*.

Fonte: Elaborada pelo autor.

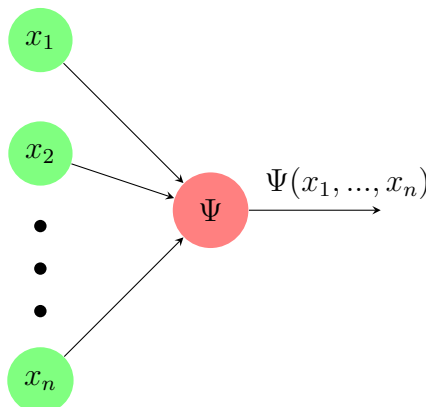
Assim, a saída do *wavelon* é definida conforme Equação 2.23 (VEITCH, 2005):

$$\psi_{\lambda,t}(x) = \psi\left(\frac{x-t}{\lambda}\right), \quad (2.23)$$

onde  $\lambda$  e  $t$  são os parâmetros *wavelet* que representam, respectivamente a dilatação e a translação.

No caso de uma Rede Neural *Wavelet* multidimensional, a entrada é um vetor multidimensional e os *wavelons* consistem em funções de ativação de *wavelets* multidimensionais. A saída de cada *wavelon* é uma ou mais combinações lineares dessas *wavelets* multidimensionais. A Figura 16 mostra a forma de um *wavelon* para a situação supracitada. A saída é definida de acordo com a Equação 2.24.

$$\Psi(x_1, \dots, x_n) = \sum_{i=1}^n \psi_{\lambda_i, t_i}(x_i). \quad (2.24)$$

Figura 16 – *Wavelon* com Função de Ativação *Wavelet* Multidimensional.

Fonte: Elaborada pelo autor.

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

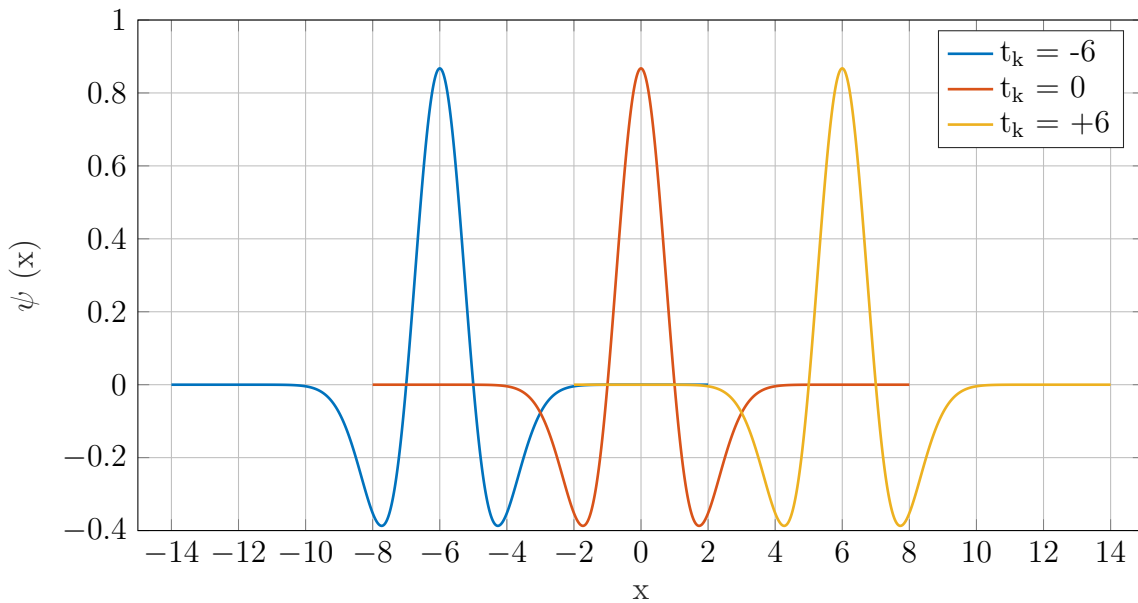
do tipo linear, a saída da rede será calculada de acordo com a Equação 2.25 (ABIYEV; KAYNAK, 2008).

$$Y_M = \sum_{k=1}^L \mathbf{W}_{kj}^{(2)} \Psi(x_1, \dots, x_n), \quad j = 1, \dots, M. \quad (2.25)$$

De acordo com a Figura 14, o índice  $M$  é quantidade de neurônios de saída,  $L$  número de *wavelets* e  $n$  o número de entradas da rede.  $\mathbf{W}_{kj}^{(2)}$  são os pesos sinápticos presentes entre a camada oculta e camada de saída.

A seleção da *wavelet* mãe depende da aplicação e não se limita às *wavelets* apresentadas. A função de ativação pode ser *wavelets basis* (ortogonais) ou *wavelets frame* (não ortogonais). Fundamentado nos trabalhos de BECERIKLI; OYSAL; KONAR (2004), BILLINGS; WEI (2005b) e ZHANG (1994), que apresentaram bons resultados e funcionamento satisfatório em várias aplicações com a *wavelet Mexican Hat*, no presente trabalho foi utilizado essa função como *wavelet* mãe.

Figura 17 – *Wavelets* da Família *Mexican Hat* Transladadas.

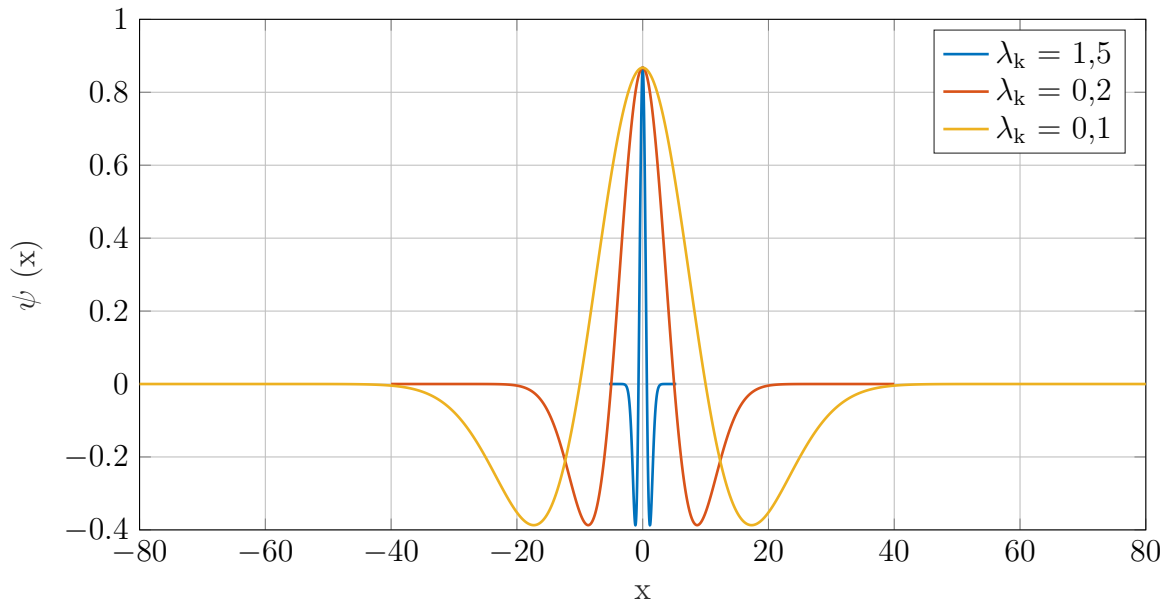


Fonte: Elaborada pelo autor.

É possível observar na Figura 17 que o parâmetro  $t_k$  indica que a função  $\psi(x_i)$  foi transladada de uma distância equivalente, sendo então um parâmetro de translação. Já variando-se o parâmetro  $\lambda_k$  é possível alongar ou comprimir a forma da função. Por isto o parâmetro  $\lambda_k$  é conhecido como parâmetro de dilatação. A Figura 18 ilustra *Mexican Hat* dilatadas.

Conforme afirmado anteriormente, a WNN adaptativa utiliza o algoritmo *back-propagation* para ajustar os pesos sinápticos, parâmetro de dilatação e parâmetro de

Figura 18 – Wavelets da Família Mexican Hat Dilatadas.



Fonte: Elaborada pelo autor.

translação. Na primeira etapa do algoritmo supracitado, para frente (*forward*), 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 (*backward*), para aplicação do algoritmo *backpropagation*, é necessária a utilização da derivada parcial da *wavelet* em relação aos seus parâmetros ajustáveis, denominados de  $\theta$ . Assim, o procedimento de treinamento da WNN envolve a utilização das derivadas do erro,  $E$ , em relação aos seus parâmetros ajustáveis  $\theta\{W_k; \lambda_k; t_k\}$  (JUNIOR, 2014). Segundo VEITCH (2005), essas derivadas são apresentadas nas Equações 2.26 a 2.28.

$$\frac{\partial E}{\partial \lambda_{i,k}} = -e \cdot W_k \cdot \left( \frac{x_i - t_{i,k}}{\lambda_{i,k}^2} \right) \cdot \psi'(z_{i,k}), \quad (2.26)$$

$$\frac{\partial E}{\partial t_{i,k}} = -e \cdot W_k \cdot \frac{1}{\lambda_{i,k}} \cdot \psi'(z_{i,k}), \quad (2.27)$$

$$\frac{\partial E}{\partial W_k} = e \cdot \psi(z_{i,k}). \quad (2.28)$$

Onde,

$$e = Y_\theta(x_i) - D(x_i), \quad (2.29)$$

$$z_{i,k} = \frac{x_i - t_{i,k}}{\lambda_{i,k}}, \quad (2.30)$$

$$\psi'(z) = \frac{d\psi(z)}{dz}. \quad (2.31)$$

Além disso,  $D$  é a saída desejada para a rede,  $Y_\theta$  corresponde a saída da rede,  $x$  é o vetor de variáveis de entradas,  $t_{i,k}$  e  $\lambda_{i,k}$  são os parâmetros de translação e dilatação da  $i$ -ésima *wavelet* presente  $k$ -ésimo neurônio da camada intermediária. Uma vez calculada essas derivadas é possível realizar a atualização dos parâmetros adaptativos da RNA,  $\theta\{\lambda_k; t_k; W_k\}$ , utilizando as Equações 2.32, 2.33 e 2.34, respectivamente.

$$\lambda_{i,k}(n+1) = \lambda_{i,k}(n) - \eta \frac{\partial E}{\partial \lambda_{i,k}} \quad (2.32)$$

$$t_{i,k}(n+1) = t_{i,k}(n) - \eta \frac{\partial E}{\partial t_{i,k}} \quad (2.33)$$

$$W_k(n+1) = W_k(n) - \eta \frac{\partial E}{\partial W_k} \quad (2.34)$$

onde o  $n$  se refere a  $n$ -ésima época de treinamento e  $\eta$  é a taxa de aprendizado constante. A escolha dessa taxa depende da aplicação e das amostras de treinamento. Geralmente, valores entre 0,1 e 0,4 são usados (ALEXANDRIDIS; ZAPRANIS, 2014).

## 2.4 Sistemas Dinâmicos

### 2.4.1 Sistemas dinâmicos e suas classificações

O ponto de partida para o estudo de um sistema dinâmico é a análise preliminar do mesmo por meio da experimentação no próprio sistema, estimulando-o de forma apropriada, observando-se as características da resposta de tal sistema e assim realizar a classificação do mesmo considerando-se diversas propriedades.

A primeira característica a ser analisada nos sistemas, é o seu comportamento quanto à sua evolução no tempo, podendo ser estáticos ou dinâmicos. Como já afirmado anteriormente, o comportamento do sistema dinâmico varia com o tempo, ou seja, a saída atual depende da entrada atual e das passadas. Já no sistema estático suas características independem do tempo, podendo variar espacialmente, logo a saída atual só depende da entrada atual (AGUIRRE, 2015).

Além dessa classificação, várias outras propriedades importantes apresentadas pelos sistemas dinâmicos permitem dividi-los em diversos tipos de subclasses. Tais propriedades possuem uma grande influência no tipo de modelo obtido. Dessa forma, os sistemas dinâmicos podem ser classificados em: lineares ou não lineares; contínuos ou discretos; causais ou não causais; estáveis ou instáveis; invariantes ou variantes no tempo; autônomos ou não

autônomos; de equações homogêneas ou não homogêneas; multivariáveis ou monovariáveis, entre outras (LATHI, 2007).

Dentre essas classificações, duas são de extrema importância a serem observadas para melhor compreensão do presente trabalho: a linearidade e o número de variáveis de entrada e saída.

#### 2.4.1.1 Sistemas lineares e não lineares

Um sistema cuja saída seja proporcional a sua entrada é um exemplo de um sistema linear. No entanto, a linearidade não se resume a isto, ela também implica as propriedades de aditividade, homogeneidade e por consequência das duas anteriores, a propriedade da superposição (LATHI, 2007).

Segundo LATHI (2007), para o sistema ser caracterizado como aditivo, as várias entradas que estão atuando no mesmo, devem ter efeitos independentes uma das outras sobre a saída do sistema. Ou seja, o efeito total no sistema devido a todas as entradas pode ser determinado considerando uma entrada por vez e assumindo todas as outras entradas iguais a zero.

Além disso, um sistema linear deve satisfazer a propriedade de homogeneidade ou escalamento, a qual afirma que para um número real ou imaginário  $k$ , se uma entrada aumentar  $k$  vezes, seu efeito também aumentará  $k$  vezes (LATHI, 2007). Logo, um sistema não linear não satisfaz nenhum desses princípios.

#### 2.4.1.2 Sistema MIMO e acoplamento de sinais

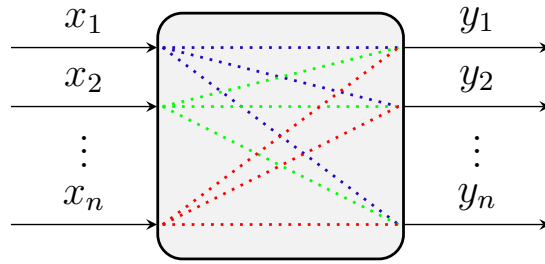
Nem sempre os conceitos utilizados na teoria clássica de identificação de sistemas para a análise de sistemas com uma entrada e uma saída, SISO (*Single Input, Single Output*), podem ser aplicados de forma direta a sistemas com várias entradas e saída, MIMO (*Multiple Input, Multiple Output*), pois geralmente existe um forte acoplamento entre os sinais de entrada e saída. Esta classe de sistemas é bem mais difícil de tratar matematicamente, pois os métodos frequências não são em geral válidos ou possuem uma diferente interpretação quando comparado com sistemas SISO (ROCHA, 2006).

Pode-se dizer que sinais estão acoplados quando um sinal influencia um outro conforme apresentado na Figura 19, na qual as linhas tracejadas representam a interdependência dos sinais (ROCHA, 2006).

O acoplamento de sinais pode ocorrer de três formas: acoplamento de entrada, acoplamento de saída e acoplamento de entrada e saída (ROCHA, 2006).

O acoplamento de entrada ocorre quando mais de um sinal de entrada influencia o sinal de saída resultante. Matematicamente pode-se afirmar que a saída fica em função de

Figura 19 – Ilustração representativa de acoplamento de sinais.



Fonte: Elaborada pelo autor.

todas as entradas, conforme expresso pela Equação 2.35.

$$y = f(x_1(t), x_2(t), \dots, x_n(t)). \quad (2.35)$$

Já no caso do acoplamento de saída a variante da equação fica da seguinte forma:

$$y = f(y_1(t-1), y_2(t-2), \dots, y_n(t-1)). \quad (2.36)$$

Por fim tem-se o acoplamento de entrada e saída, que se trata das situações citadas anteriormente de forma simultânea, obtendo-se:

$$y = f(x_1(t), x_2(t), \dots, x_n(t), y_1(t-1), y_2(t-2), \dots, y_n(t-1)). \quad (2.37)$$

Conforme se observa na equação supracitada, a saída resultante atual passa a ser influenciada por entradas e saídas anteriores.

## 2.4.2 Identificação de sistemas dinâmicos

Segundo LJUNG (1987), identificação de sistemas dinâmicos é permitir a elaboração do modelo matemático de um sistema dinâmico baseado em medidas coletadas e, a partir desses dados, realizar ajuste de parâmetros e/ou do modelo, até que a saída do sistema coincida de forma bem próxima quanto possível com as amostras das saídas medidas no sistema real.

Existem várias formas de classificar as técnicas de modelagem. Levando-se em conta o nível de conhecimento do sistema que se deseja identificar, pode-se realizar três tipos de modelagem: caixa branca, caixa preta e caixa cinza (AGUIRRE, 2015).

Para modelagem caixa branca existe claramente conhecimento de suas entradas, saídas e do processo de funcionamento dos sistemas. Esse tipo de sistema é facilmente identificável e modelável e utiliza a física do processo, ou seja, utiliza princípios básicos para modelagem.

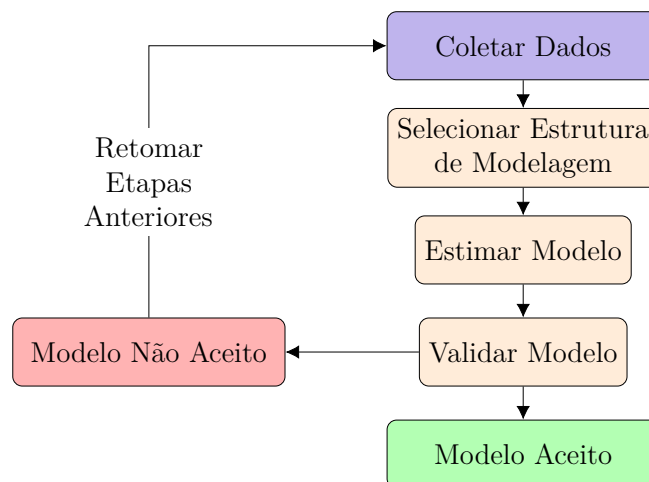
Já para a modelagem caixa cinza, existe o conhecimento das entradas e das saídas, mas o processo de funcionamento é pouco conhecido, ou seja, não se sabe exatamente todos os parâmetros que influenciam o sistema.

Por fim, quando pouco se sabe sobre os sistemas a serem modelados é utilizado o sistema conhecido como caixa preta. Um sistema caixa preta pode servir para representar tanto sistemas lineares como não-lineares, sistemas dinâmicos ou estáticos. Entretanto, é difícil uma análise algébrica das saídas obtidas pelos sistemas caixa preta, ou seja, a modelagem deve ser conduzida por dados.

A identificação de sistemas se enquadra exatamente no tipo de modelagem de sistemas caixa preta, pois uma das características das técnicas de identificação é que pouco ou nenhum conhecimento prévio do sistema é necessário e, conseqüentemente, tais métodos são alternativos para a modelagem de sistemas caixa branca, sendo também referidos como modelagem (ou identificação) caixa preta (AGUIRRE, 2015).

Segundo LANDAU (2006), a operação completa da identificação deve necessariamente incluir os quatro estágios apresentados na Figura 20. Além disso, o autor afirma que os métodos específicos utilizados em cada fase dependem do tipo de modelo desejado (paramétrico ou não paramétrico, de tempo contínuo ou de tempo discreto) e das condições experimentais. Seguindo as etapas apresentadas pela Figura 20, observa-se que a validação é o passo obrigatório para decidir se o modelo identificado é aceitável ou não.

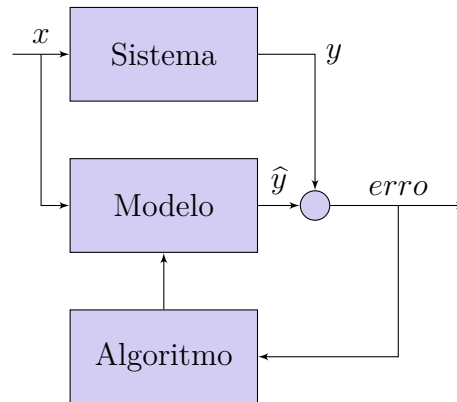
Figura 20 – Procedimento básico de identificação caixa preta.



Fonte: Modificado (LANDAU, 2006).

Essencialmente, a identificação computacional de sistemas ou modelagem empírica envolve coleta de dados das características de entrada-saída do sistema e utilização destes para obtenção de um modelo matemático que aproxima este comportamento observado. O diagrama de blocos para identificação de sistemas, apresentado na Figura 21, ilustra como é este procedimento.

Figura 21 – Diagrama de Blocos - Identificação de processos.



Fonte: Modificado (COELHO, 2000).

Assim, a partir de um mesmo conjunto de sinais observados ou medidos ajustam-se os parâmetros do modelo de forma a minimizar o erro de saída ( $erro = y - \hat{y}$ ), isto é, a diferença entre o sinal real observado e o valor produzido pelo modelo. Este tipo de modelagem é classificada como modelagem direta.

### 2.4.3 Modelos neurais de identificação

Segundo JUNIOR (2007), a principal vantagem de se utilizar estruturas neurais para a identificação de sistemas é a sua capacidade de representar, ou aprender, características não lineares desses sistemas.

Tais estruturas de identificação são caracterizadas pelo seu vetor de regressão, ou seja, pelo vetor que contém os valores passados das variáveis para se estimar a saída do sistema. Essas características regressivas dos modelos neurais de identificação são oriundas de generalizações das estruturas de modelagem linear, podendo-se citar como estruturas lineares: FIR (*Finite Impulse Response*), ARX (*AutoRegressive eXogenous input*), ARMAX (*AutoRegressive, Moving Average, eXogenous input*), OE (*Output Error*) e SSIF (*State Space Innovations Form*) (JUNIOR, 2007).

Conseqüentemente, os modelos neurais não lineares irão variar de acordo com o regressor linear de origem. Por exemplo, se o vetor de regressão for selecionado de forma similar a utilizada pelo modelo ARX, a estrutura de modelagem neural é chamada NNARX (*Neural Network ARX*). Do mesmo modo ocorre para as demais estruturas NNFIR, NNARMAX, NNOE e NNSSIF (JUNIOR, 2007).

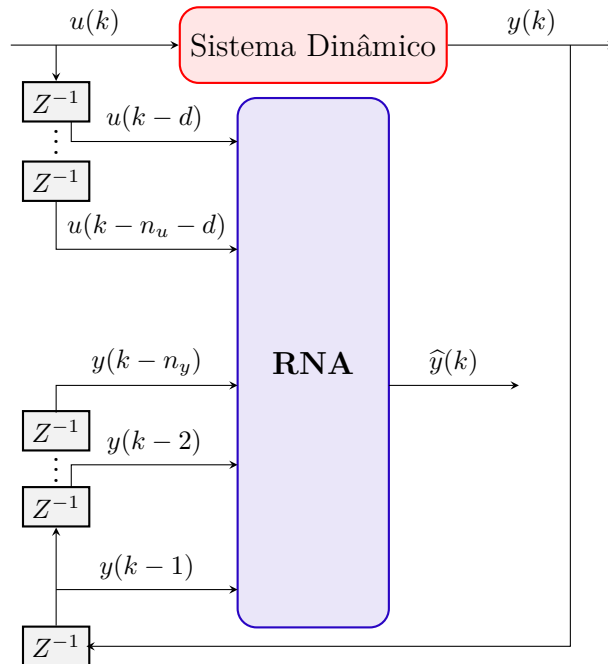
O presente trabalho se limitará a descrever a estrutura NNARX, pois foi a estrutura neural de identificação utilizada para modelagem do sistema de tanques acoplados.

A Figura 22 apresenta o diagrama de uma rede NNARX, onde  $\hat{y}$  é a saída estimada,  $d$  o atraso de transporte,  $n_y$  a ordem da saída,  $n_u$  a ordem da entrada,  $y$  a saída desejada



e  $u$  a entrada do processo.

Figura 22 – Estrutura NNARX para identificação de sistemas.



Fonte: Elaborada pelo autor.

Pode-se observar no diagrama que os regressores fazem com que a saída presente da rede neural esteja relacionada com valores passados da entrada e saída do sistema a ser identificado. Logo, observa-se que os regressores são de extrema importância, uma vez que a estrutura é projetada com o intuito de identificar a dinâmica existente em um sistema físico (LUCAS, 2012). A Equação 2.38 descreve o modelo não linear em questão.

$$\hat{y}(k) = f(y(k-1), \dots, y(k-n_y), u(k-d), \dots, u(k-d-n_u)). \quad (2.38)$$

Outra característica importante nas redes NNARX, é que as saídas geradas pelos modelos são sempre estáveis, uma vez que representa relações puramente algébricas entre estimativa e as medições passadas das entradas e saídas do processo, não existindo a realimentação de saída estimada. Dessa forma, ela é a melhor estrutura para sistemas determinísticos ou com baixo nível de ruído (LUCAS, 2012).

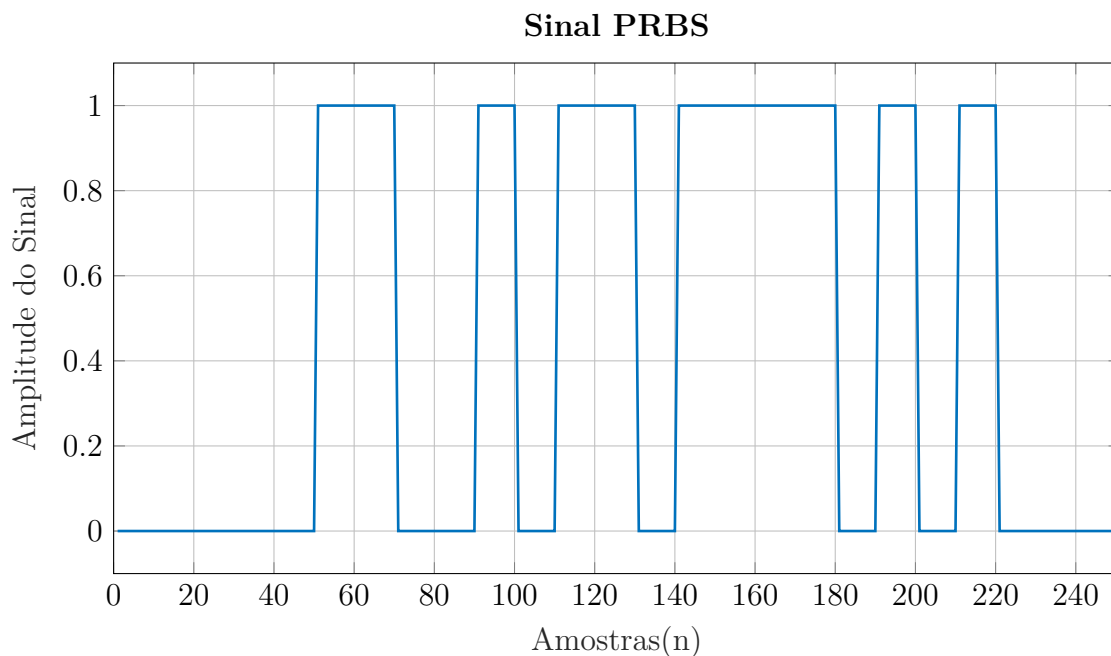
## 2.5 Pseudo Random Binary Signal

Um sinal de perturbação utilizado na identificação de sistemas lineares deve possuir duas características. A primeira delas é ser pequeno suficiente, em amplitude, para minimizar distorções não lineares e ao mesmo tempo ser grande o suficiente para minimizar os efeitos dos ruídos (AGUIRRE, 2015).

Dentre os sinais pseudo-aleatórios, os sinais binários são bastante populares, pois são fáceis de gerar, já que esses sinais possuem apenas dois valores possíveis e comuta entre níveis em instantes discretos. Outro aspecto interessante desses sinais é que a mudança entre valores é definida de forma determinística e, conseqüentemente, exatamente o mesmo sinal pode ser gerado repetidas vezes (AGUIRRE, 2015).

A periodicidade do sinal *Pseudo Random Binary Signal* (PRBS) não deve ser menor do que o tempo de acomodação do sistema que está sendo testado. Caso contrário, o sistema “perceberá” a periodicidade do sinal PRBS reduzindo, assim, o caráter aleatório do teste. A Figura 23 ilustra um exemplo de sinal PRBS.

Figura 23 – Sinal utilizado para excitação do sistema em estudo.



Fonte: Elaborada pelo autor.

Assim, a aplicação de um sinal PRBS deve ter a duração maior possível, porém, um tempo muito prolongado pode gerar problemas, dependendo do ponto de vista. Por exemplo, um sinal muito longo aplicado ao sistema, permite mais dados para o treinamento e conseqüentemente identificação, melhorando a estimativa dos parâmetros. No entanto, pode ocasionar a inclusão de distúrbios no processo de treinamento que terão maior possibilidade de se manifestar com um tempo muito extenso.

Uma vez apresentada a fundamentação teórica necessária para compreensão deste trabalho, o capítulo seguinte apresentará a metodologia utilizada para identificação do sistema de nível com tanques acoplados bem como a caracterização do mesmo.

## 3 Materiais e Métodos

Nesta seção será descrito o sistema de tanques de nível utilizado na pesquisa, bem como todos os passos para coleta e pré-processamento dos dados. Em seguida é descrito o processo de treinamento e validação, das redes utilizadas, por meio da estrutura NNARX.

Uma metodologia exploratória, descritiva e quantitativa mostra-se ideal para investigar identificação de sistemas dinâmicos não lineares utilizando redes neurais artificiais, tendo em vista trabalhar com os diversos conceitos relacionados com inteligência artificial e modelagem de sistemas, que estão enraizadas na regressão linear, mínimos quadrados, método da máxima-verossimilhança, aproximação estocástica, entre outros. Além do mais, medidas de avaliação quantitativa são utilizadas no intuito de permitir uma adequada verificação no comportamento dos algoritmos.

De acordo com [GIL \(2017\)](#), a pesquisa exploratória é bastante flexível, possibilita considerações dos mais variados aspectos relativos ao fato estudado, em que o principal objetivo é aprimorar as ideias ou descobrir intuições, além de proporcionar maior familiaridade com o problema.

### 3.1 Equipamentos e Operação: Sistema de Tanques de Nível

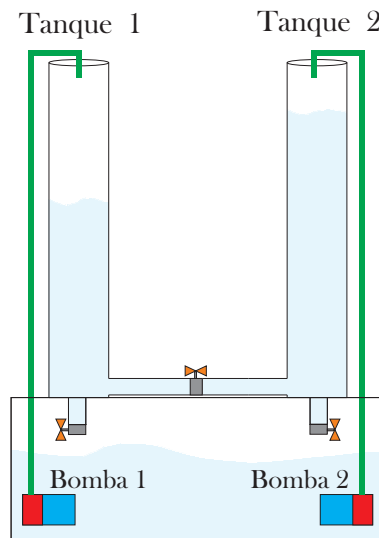
O sistema de tanques de nível utilizado na presente pesquisa foi desenvolvido no curso de engenharia elétrica da Universidade Federal do Piauí. Segundo [SOUSA \(2014\)](#), tal planta tem como principal objetivo disponibilizar um processo robusto e de fácil entendimento que permita aos alunos o domínio dos fundamentos de controle e identificação de processos, que é bastante comum em indústrias.

O processo é composto por duas colunas cilíndricas, construídas em acrílico, cada uma com 70 *cm* de altura e 75 *mm* de diâmetro, interligadas por um tubo cilíndrico de 20*mm* de diâmetro com uma válvula que controla o fluxo de líquido de uma coluna para a outra. Cada coluna possui uma válvula que controla o fluxo que desce para o reservatório de água que fica logo abaixo das colunas. Esse reservatório retangular possui dimensões de 50*cm* x 30*cm* x 30*cm*. As bombas ficam dentro deste reservatório e bombeiam água para as duas colunas de líquido ([SOUSA, 2014](#)).

Um esquemático deste tanque é mostrado na [Figura 24](#) e uma fotografia na [Figura 25](#), onde é ilustrada uma visão frontal dos tanques. Por questão de nomenclatura e referência, denominou-se de Tanque 1 à esquerda e Tanque 2 à direita.

Como observa-se nas figuras supracitadas, as bombas estão localizadas na parte inferior do reservatório e bombeiam água para parte superior dos tanques cilíndricos. Tais

Figura 24 – Esquema do sistema de tanques acoplados.



Fonte: Elaborada pelo autor.

Figura 25 – Sistema de tanques de nível acoplados utilizado na pesquisa.



Fonte: Elaborada pelo autor.

bombas são alimentadas com tensão de 12 V e o controle de vazão é realizado através de transistores de potência acionados por PWM.

A partir da análise do funcionamento do sistema, é possível perceber que o mesmo é caracterizado por possuir atraso de transporte, ou seja, existe um tempo decorrente para que uma variação no sinal de entrada (excitação) seja efetivamente “percebida” pela variável de saída (resposta). Tal característica é resultante do tempo que o líquido

bombeado leva para percorrer a mangueira desde a extremidade conectada na bomba à extremidade localizada na parte superior dos tanques cilíndricos.

A verificação do nível dos tanques é possível através de dois sensores de pressão do tipo diferencial, um em cada tanque, conectados a dois pequenos tubos colocados desde a parte inferior de cada tanque. Esses sensores transformam a pressão da coluna de água em uma tensão linearmente proporcional ao nível da água (SOUSA, 2014).

Geralmente, os sensores do tipo diferencial possuem duas extremidades para entrada de pressão, sendo uma destinada a pressão de referência (atmosférica – conforme a aplicação) e a outra para pressão que se deseja saber. Logo, através da diferença de pressão, é possível se determinar o valor da pressão desejada (LUZETTI, 2009).

Os sensores utilizados na presente pesquisa são produzidos pela Motorola© e pertencem a série MPX5010, Figura 26 (SEMICONDUCTOR, 2012). Esses sensores possuem a capacidade máxima de pressão de 1,5 mca (metros de coluna de água), ou seja, para uma coluna de água de 1,5 m o sensor vai retornar uma tensão de 5 V. A tensão na saída é linearmente proporcional à pressão e varia de 0 a 5 V (SOUSA, 2014).

Figura 26 – Sensor de Pressão Utilizado.



Fonte: Modificado (SEMICONDUCTOR, 2012).

O *hardware* utilizado para gerar o sinal PWM para o controle da vazão das bombas e responsável pela obtenção da tensão gerada pelo sensor é o microcontrolador PIC18F2550 da Microchip® (MICROCHIP, 2006). Para isso, tal dispositivo utiliza um conversor A/D presente em sua arquitetura.

Para a interface entre o microcontrolador e o usuário do sistema de tanques, utiliza-se a placa SanUSB, que trata-se de uma ferramenta composta de *hardware* e *software*, que em tempo real realiza a sincronização entre o tanque e o *software* Matlab/Simulink® (SOUSA, 2014). A placa supracitada é mostrada na Figura 27.

Segundo SOUSA (2014), a placa SanUSB, ilustrada na Figura 27, possui os seguintes atributos:

- a) um gravador de programas no microcontrolador;
- b) conversor TTL - RS-232 para comunicação serial bidirecional, emulado via USB pelo protocolo CDC, que permite também a depuração do programa através da

Figura 27 – Placa SanUSB.



Fonte: Modificado (SOUSA, 2014).

impressão via USB das variáveis do *firmware*;

- c) fonte externa, pois a alimentação do PIC é através da porta USB do PC;
- d) conversor analógico-digital (AD) externo, tendo em vista que ele dispõe internamente de 10 ADs de 10 bits;
- e) *software* de simulação, considerando que a simulação do programa e do *hardware* podem ser feitas de forma rápida e eficaz no próprio circuito de desenvolvimento ou com um *protoboard* auxiliar.

## 3.2 Coleta de Dados

Uma vez que a identificação de sistemas se propõe a obter modelos a partir de dados, é necessário gerar tais dados. Questões importantes relacionadas a esta etapa são a forma de excitação do sistema, a escolha dos sinais de excitação, a execução da validação e a escolha do tempo de amostragem.

Na primeira etapa da presente pesquisa, como já explicado anteriormente, utilizou-se um microcontrolador em conjunto com a placa SanUSB para realizar a amostragem em tempos discretos dos dados de entradas e saídas.

Definido o processo de aquisição, buscou-se estabelecer um sinal de excitação para o sistema. Tendo em vista as vantagens do sinal PRBS, já citadas no presente trabalho, unido a sua facilidade de implementação, optou-se por tal sinal.

A implementação do sinal PRBS ocorreu por meio da função *indinput* presente na *toolbox System Identification* do *software* Matlab/Simulink®, que gera sinais de entrada que são tipicamente usados para a identificação. Essa função possui como parâmetros: a quantidade de amostras ( $N$ ), o tipo de sinal (*type*), os valores de amplitude máximos e mínimos (*levels*) e a frequência máxima e mínima do sinal (*band*). Tal função possui a seguinte sintaxe:

$$u = \text{indinput}(N, \text{type}, \text{band}, \text{levels}), \quad (3.1)$$

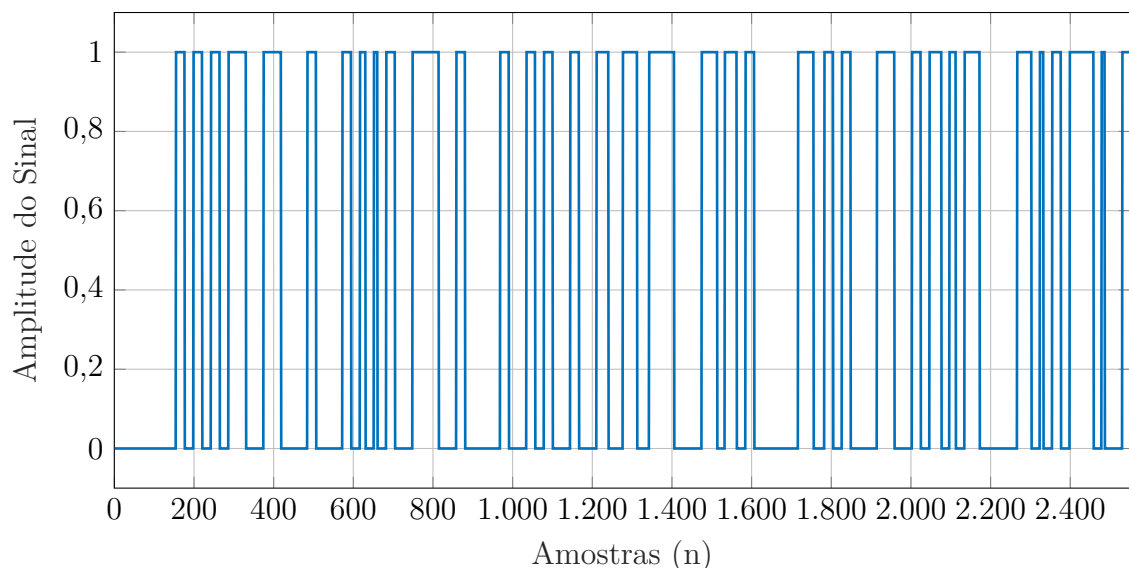
onde  $u$  seria o vetor com os valores do sinal PRBS. Para que não houvesse riscos de transbordamento do tanque, implementou-se um saturador que limita a vazão da bomba, ou seja, ao atingir o valor de 65 *cm* de altura, o saturador atua reduzindo a tensão na bomba a zero por alguns instantes, consequentemente cessando a vazão de líquido nos tanques.

Para gerar o sinal PRBS de excitação do Tanque de Nível 1 e Tanque de Nível 2, foram utilizados os parâmetros apresentados na Tabela 1. As Figuras 28 e 29 mostram os sinais gerados para cada tanque. Os valores do sinal PRBS são utilizados para alterar o *duty cycle* do PWM responsável por acionar as bombas hidráulicas. Observa-se na Tabela 1 que a frequência dos sinais de excitação dos Tanques 1 e 2 são distintas. Tal metodologia foi utilizada para verificar a interdependência entre os tanques de modo que em determinados momentos a bomba de um tanque esteja acionada e a bomba do tanque adjacente desligada.

Tabela 1 – Parâmetros da Função *indinput*: Tanques Acoplados

Parâmetros	Tanque de Nível 1	Tanque de Nível 2
Quantidade de Amostras ( $N$ )	2550	2550
Tipo de Sinal ( <i>type</i> )	prbs	prbs
Frequência ( <i>band</i> )	[0 0,045]	[0 0,1]
Amplitude ( <i>levels</i> )	[0 1]	[0 1]

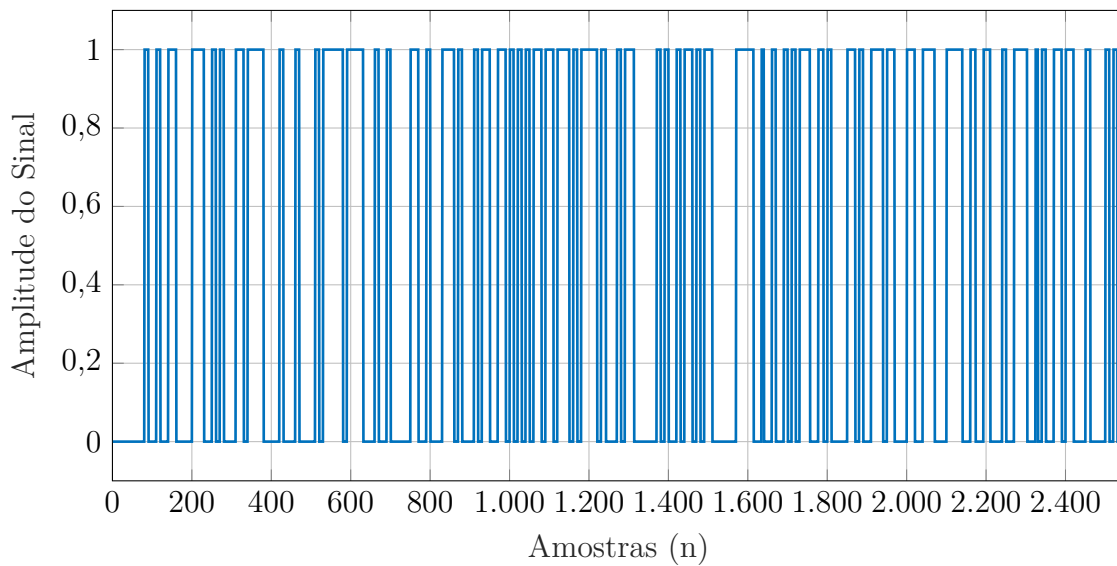
Figura 28 – Sinal PRBS Aplicado à Bomba Hidráulica associada ao Tanque 1.



Fonte: Elaborada pelo autor.

Na prática, segundo [AGUIRRE \(2015\)](#), a frequência de amostragem é normalmente escolhida entre 5 e 10 vezes maior do que a frequência do sinal de interesse, e não apenas 2 vezes maior (chamada de frequência de *Nyquist*). Diante disso, neste trabalho adotou-se

Figura 29 – Sinal PRBS Aplicado à Bomba Hidráulica associada ao Tanque 2.

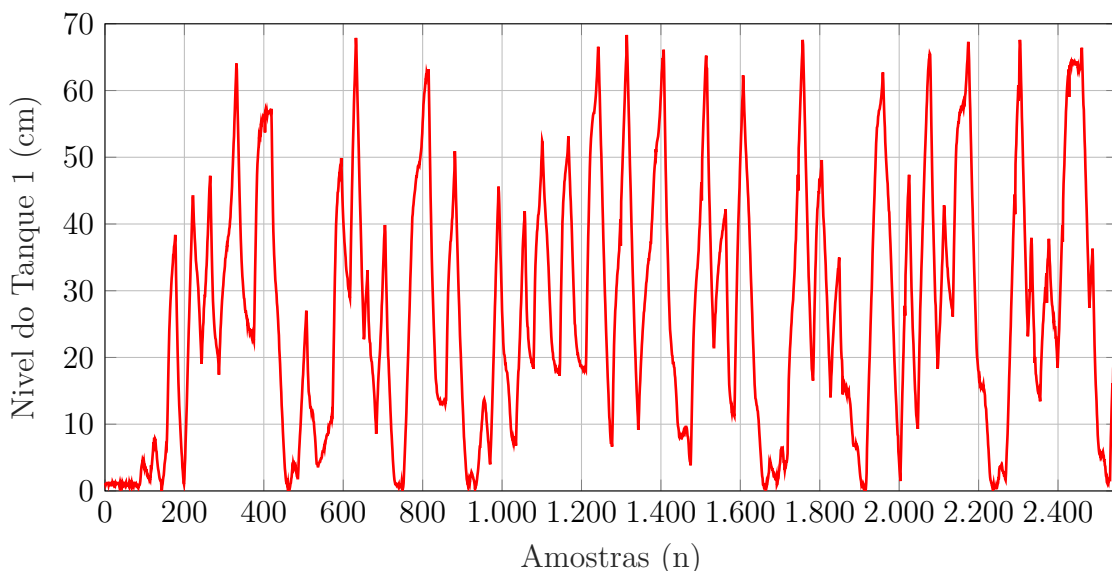


Fonte: Elaborada pelo autor.

o período de amostragem de 0,8 segundos, valor este que se mostrou bastante satisfatório e dentro do intervalo supracitado.

As Figuras 30 e 31 representam as respostas do sistema, correspondendo o nível do tanque 1 e nível do tanque 2, respectivamente.

Figura 30 – Resposta do Sistema: Nível do Tanque 1.

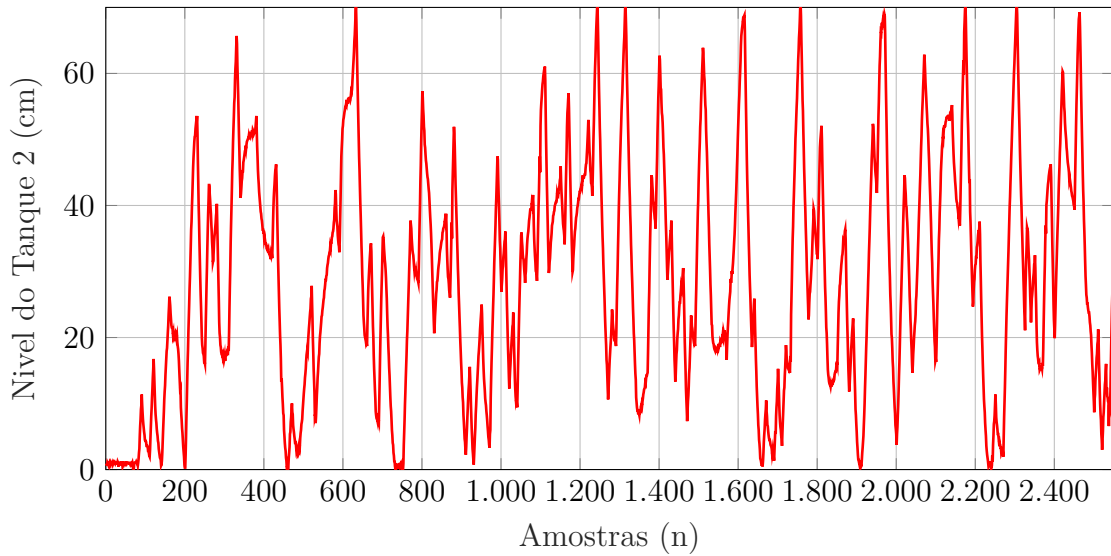


Fonte: Elaborada pelo autor.

Com a finalidade de realizar dois estudos de caso, optou-se por condicionar o sistema de tanques a duas configurações diferentes de vazão de acoplamento. Na primeira configuração manteve-se a válvula responsável pelo acoplamento dos dois tanques total-



Figura 31 – Resposta do Sistema: Nível do Tanque 2.



Fonte: Elaborada pelo autor.

mente fechada, ou seja, os tanques encontraram-se desacoplados para essa configuração. Já na segunda configuração a válvula foi parcialmente aberta (tanques parcialmente acoplados, em torno de 50%). Para o caso desacoplado, foram considerados os parâmetros apresentados na Tabela 2 para gerar o Sinal PRBS. As Figuras 28, 29, 30 e 31 são referentes aos dados coletados para o estudo de caso com acoplamento.

Tabela 2 – Parâmetros da Função *indinput*: Tanques Desacoplados

Parâmetros	Tanques de Nível 1 e 2
Quantidade de Amostras ( $N$ )	2550
Tipo de Sinal ( $type$ )	prbs
Frequência ( $band$ )	[0 0,2]
Amplitude ( $levels$ )	[0 1]

É importante ressaltar que somente na segunda configuração, tanques acoplados, adotou-se sinais de excitação distintos para cada tanque de nível. Para o primeiro estudo de caso, tanques desacoplados, utilizou-se o mesmo sinal de excitação para os dois tanques. A justificativa para tal metodologia, é permitir uma comparação dos tanques quando os dois trabalham de forma isolada.

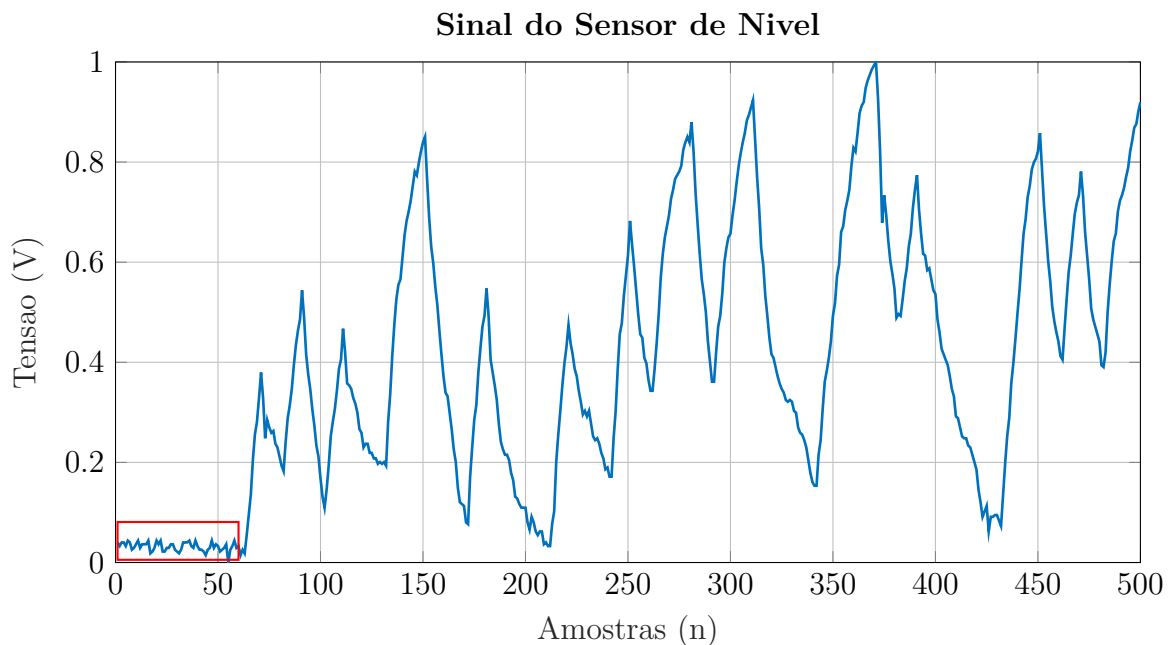
### 3.3 Pré-Processamento de Dados

Nesta etapa é realizada uma análise criteriosa sobre os dados coletados buscando minimizar ambiguidades e possíveis erros associados à etapa de aquisição de dados. Normalmente, os dados coletados são divididos em duas categorias: dados de treinamento

e dados de validação. Neste trabalho coletou-se 2550 amostras que foram divididas em 2040 amostras para treino e 510 amostras para validação.

Definido os conjuntos de dados de treinamento e validação, é importante analisar as características do sensor utilizado. Conforme observa-se o sinal ilustrado pela Figura 32, tal sensor possui um ruído característico de instrumentos de medição, havendo um erro de *offset*, pois existe uma determinada tensão mesmo não havendo líquido no tanque.

Figura 32 – Ruídos de Medição.



Fonte: Elaborada pelo autor.

Antes de inserir os dados na entrada da estrutura utilizada no processo de identificação, é comum mudar a escala original dos dados com a finalidade de adaptar os dados de entrada à faixa dinâmica das funções de ativação da rede neural. No presente trabalho normalizou-se os valores dos dados entre 0 (zero) e 1 (um). Neste caso, deve-se aplicar a seguinte transformação a cada componente do conjunto de dados:

$$x_j^* = \frac{x_j}{x_j^{max}}, \quad (3.2)$$

onde, ao dividir  $x_j$  pelo seu maior valor  $x_j^{max} = \max\{x_j(t)\}$ , tem-se  $x_j^* \in [0, 1]$ . A mesma normalização foi utilizada para o treinamento das redes MLP e WNN.

### 3.4 Seleção da Estrutura de Identificação

Com os dados normalizados, pode-se agora aplicar esses dados nas entradas da estrutura de identificação para realizar o treinamento da RNA e ajuste dos parâmetros do

modelo. No entanto, antes de descrever o treinamento propriamente dito, vale enfatizar novamente que quando se utiliza redes neurais conjuntamente com a estrutura de identificação ARX, a entrada de dados da rede neural passa a ser formada por regressores que fazem com que a saída no instante presente esteja relacionada com valores passados da entrada e saída do sistema a ser identificado. Logo, a matriz de dados de entrada da RNA deve ser formada a partir dos dados do sinal de excitação (PRBS) e resposta (nível) do sistema real (SILVA; JUNIOR, 2018).

Por conta dessas características regressoras do modelo NNARX, as saídas geradas serão sempre estáveis, uma vez que representam relações puramente algébricas entre estimativa e as medições passadas das entradas e saídas do processo. É justamente por isso que optou-se pela escolha dessa estrutura de identificação pois, segundo LUCAS (2012), ela é uma boa estrutura para sistemas determinísticos ou com baixo nível de ruído.

Uma vez definida e justificada a escolha da estrutura de identificação regressiva, deve-se definir a ordem do preditor. Para isso, um estudo experimental inicial foi conduzido para determinar o número ótimo de parâmetros de entrada das RNAs, ou seja, definir o vetor de regressão não linear como a forma de entradas exógenas.

Após análise empírica, a estrutura que apresentou melhores resultados foi aquela formada por 2 (duas) amostras do sinal PRBS (dados de entrada atrasados) e 2 (duas) amostra do sinal de Nível (dados de saída atrasados). O atraso de transporte adotado foi igual a 1 (um). Considerando um caso monovariável (SISO), ou seja, a identificação de um único tanque, teremos a seguinte estrutura de identificação ilustrada na Figura 33. Tal topologia de identificação também foi a mais adequada em outros trabalhos, como por exemplo (OWA; SHARMA; SUTTON, 2015).

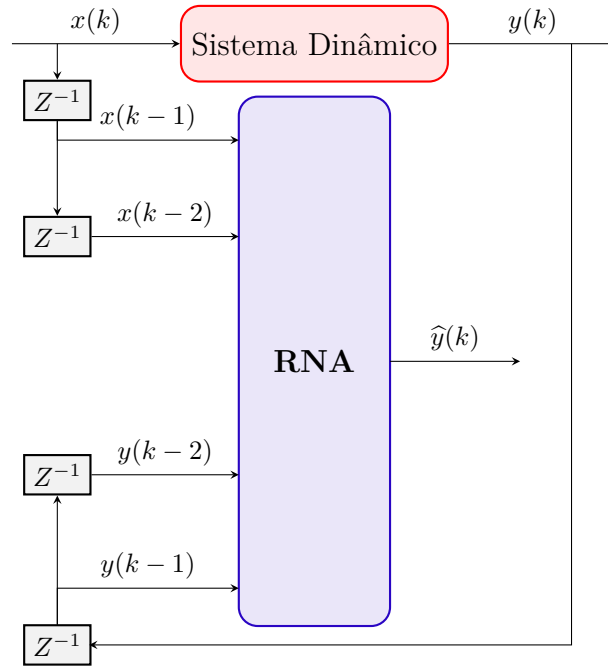
Porém, vale enfatizar novamente que adotou-se duas configurações diferentes para a abertura da válvula de acoplamento dos dois tanques: válvula totalmente fechada (tanques desacoplados) e válvula parcialmente aberta. A configuração acima será aplicada para o caso desacoplado pois, uma vez que a válvula de acoplamento esteja totalmente fechada, o sistema passa a se comportar como dois sistemas monovariáveis SISO independentes, sem acoplamento de sinal. Logo, matematicamente o vetor de regressão para a identificação do Tanque 1 e Tanque 2 é dado pela Equação 3.3 e Equação 3.4, respectivamente.

$$\widehat{y}_1(k) = f(y_1(k-1), y_1(k-2), x_1(k-1), x_1(k-2)). \quad (3.3)$$

$$\widehat{y}_2(k) = f(y_2(k-1), y_2(k-2), x_2(k-1), x_2(k-2)). \quad (3.4)$$

Considerando a válvula de acoplamento parcialmente aberta adotou-se oito entradas, dado que se trata de um sistema multivariável com atraso de transporte igual a 1. Logo, conforme mostra a Figura 34, tem-se como entrada da rede: 2 (duas) amostras de dados

Figura 33 – Estrutura NNARX para identificação do sistema de nível desacoplado.



Fonte: Elaborada pelo autor.

de entrada de cada tanque de nível e 2 (duas) amostras de dados de saída por tanque, todas com atraso. Os índices 1 e 2 em  $x_1$ ,  $y_1$ ,  $x_2$  e  $y_2$  representam amostras dos tanques 1 e 2, respectivamente.

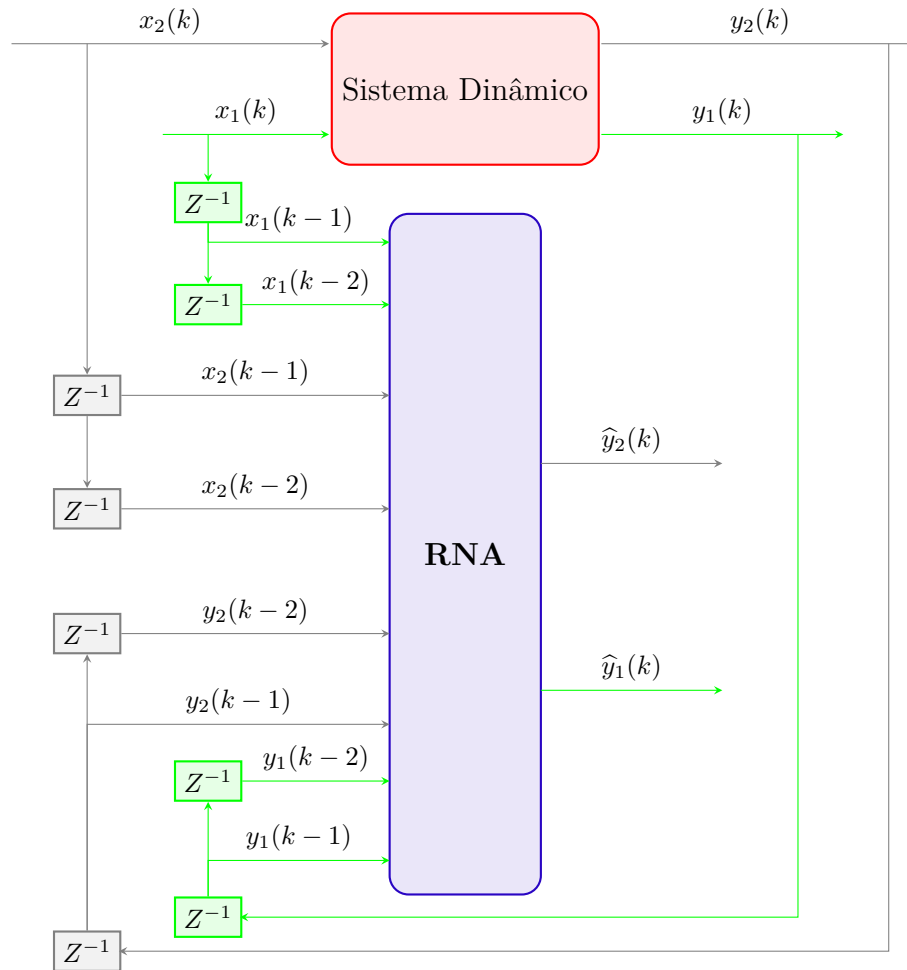
Matematicamente, as duas saídas estimadas dependem dos mesmos dados de entrada e saída do sistema devido ao acoplamento de sinais, logo as equações do modelo são dadas por:

$$\hat{y}_1(k) = f(y_1(k-1), y_1(k-2), x_1(k-1), x_1(k-2), y_2(k-1), y_2(k-2), x_2(k-1), x_2(k-2)) \quad (3.5)$$

$$\hat{y}_2(k) = f(y_1(k-1), y_1(k-2), x_1(k-1), x_1(k-2), y_2(k-1), y_2(k-2), x_2(k-1), x_2(k-2)) \quad (3.6)$$

Conforme afirmado anteriormente, foram utilizadas duas redes neurais artificiais juntamente com a estrutura ARX: *Multilayer Perceptron* e *Wavelet Neural Network*. Segundo [SILVA; SPATTI; FLAUZINO \(2010\)](#), a especificação da topologia dessas redes neurais é usualmente efetuada de forma empírica, pois tal dimensionamento depende, entre outros fatores, da quantidade de neurônios na camada escondida, do algoritmo de aprendizagem utilizado, da forma como as matrizes de pesos foram iniciadas, da complexidade do problema a ser mapeado, da disposição espacial das amostras e, por sua

Figura 34 – Estrutura NNARX para identificação do sistema de nível acoplado.



Fonte: Elaborada pelo autor.

vez, da qualidade do conjunto de treinamento disponível.

Afim de permitir a definição da influência do número de neurônios ocultos e da quantidade de épocas sobre os resultados, coletou-se os erros médios quadráticos variando-se esses dois parâmetros da rede. Inicialmente, manteve-se fixo o número de neurônios e realizou-se 20 (vinte) treinamentos para um dado número de épocas. Por exemplo, para 3 neurônios, realizou-se 20 treinamentos de 1.000 épocas, 20 de 5.000 épocas, 20 de 10.000 épocas, 20 de 15.000 épocas, 20 de 20.000 épocas e 20 de 25.000 épocas. Para cada treinamento registrou-se o valor do erro quadrático médio e ao final dos 20 treinamentos calculou-se a média e a variância dos valores adquiridos para um dado número de neurônios e épocas.

Para o estudo de caso com desacoplamento entre os tanques, esse procedimento foi realizado para o intervalo de 3 a 10 neurônios e para cada tanque de forma isolada (desacoplado). O número de épocas variou-se entre 5.000 e 25.000 épocas para a rede MLP e entre 100 e 300 épocas para a rede WNN. Os parâmetros adotados para o estudo de

caso dos tanques desacoplados são ilustrados na Tabela 3.

Tabela 3 – Dados de Parametrização - Estudo de caso com desacoplamento

<b>Parâmetros para Estimação dos Modelos</b>		
	<b>MLP</b>	<b>WNN</b>
Número de camadas ocultas	1	1
Dimensão da camada de entrada	4	4
Número de neurônios da camada oculta	3 a 10	3 a 10
Número de neurônios da camada de saída	1	1
Atraso de transporte	1	1
Taxa de aprendizagem	0,1 e 0,05	0,1 e 0,05
Número de épocas	60.000 a 500.000	50 e 300

Para considerar os valores supracitados de épocas, realizou-se treinamentos nas duas redes tendo como critério de parada o erro. E verificou-se que a Rede Neural *Wavelet* necessita, em média, 100 vezes menos épocas de treinamento que a Rede Neural Perceptron Múltiplas Camadas, para atingir um dado valor de erro. Justificando, assim, os valores apresentados conforme Tabela 3.

É possível observar, também, que considerou-se dimensão 4 para a camada de entrada dado que se trata de um sistema monovariável, estrutura NNARX de identificação e atraso de transporte igual a 1 (um). Já o número de neurônios da camada de saída varia de acordo com o número de saídas do sistema. Consequentemente, como identificou-se cada tanque de forma isolada, essa camada terá apenas 1 (um) neurônio, pois trata-se de um sistema monovariável, ou seja, uma entrada e uma saída. A taxa de aprendizagem foi definida através de testes, variando-se a mesma entre 0 e 1, definindo-se o valor de 0,1 para as duas redes, WNN e MLP, com a finalidade de gerar os resultados finais e comparação dos desempenhos das RNAs.

A Tabela 4 apresenta mais informações a respeito da parametrização para os estudos de casos com acoplamento entre os tanques. O número de neurônios da camada de saída varia de acordo com o número de saídas do sistema. Consequentemente, como identificou-se os tanques acoplados, essa camada terá dois neurônios. Por sua vez, para a camada oculta, testou-se valores entre 6 e 12 neurônios, utilizando o mesmo procedimento do caso com desacoplamento, afim de se obter os resultados gráficos que serão ilustrados ao longo do capítulo seguinte.

Conforme já explicado anteriormente, considerando que o segundo estudo de caso trata-se de um sistema multivariável e a estrutura de identificação adotada foi NNARX com atraso de transporte, as redes passam a ser formadas por 8 (oito) entradas. Pode-se observar, também, que a diferença de número de épocas para treinamento entre as duas RNAs também foram significativas.

Tabela 4 – Dados de Parametrização - Estudo de caso com acoplamento

Parâmetros para Estimação dos Modelos		
	MLP	WNN
Número de camadas ocultas	1	1
Dimensão da camada de entrada	8	8
Número de neurônios da camada oculta	6 a 12	6 a 12
Número de neurônios da camada de saída	2	2
Atraso de transporte	1	1
Taxa de aprendizagem	0,1 e 0,05	0,1 e 0,05
Número de épocas	100.000 a 600.000	200 e 400

Seguindo os requisitos para a identificação de sistemas, as etapas de obtenção de dados e escolha da estrutura de identificação já tiveram as metodologias descritas e justificadas. Na seção a seguir será apresentado como os algoritmos das redes MLP e WNN realizam os ajustes dos pesos de modo a obter o modelo do sistema de nível.

## 3.5 Estimação dos Modelos do Sistema de Tanques

### 3.5.1 Modelo Neural: *Multilayer Perceptron*

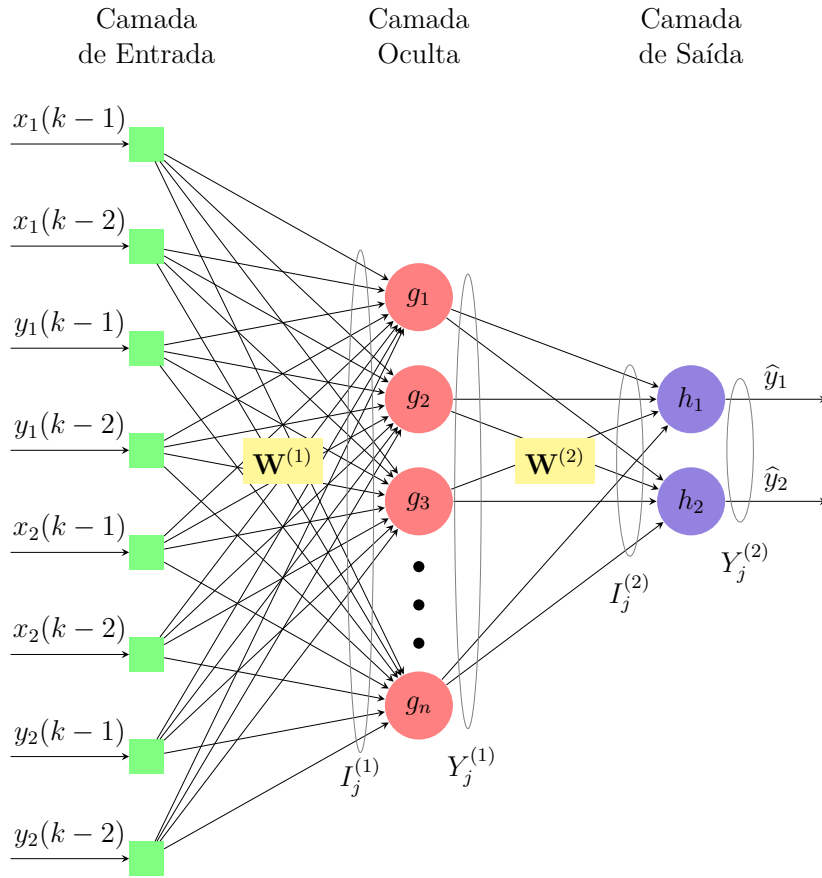
Para uma melhor descrição do procedimento de estimação do modelo, considerou-se como referência a estrutura da rede MLP adotada para o estudo de caso multivariável, ilustrada pela Figura 35.

Conforme se observa, a RNA é formada por uma camada de entrada com uma dimensão igual a 8 (oito), que correspondem a 2 (duas) amostras de dados de saída de cada tanque de nível e 2 (duas) amostras de dados de entrada por tanque. Tem-se uma camada oculta, cuja quantidade de neurônios variou-se durante o experimento. Por fim, uma camada de saída é formada por 2 neurônios em que cada neurônio corresponde ao valor estimado do nível dos tanques.

A partir da figura citada, assume-se então a seguinte terminologia para os parâmetros constituintes da rede:

- $\mathbf{W}_{ji}^{(L)}$  São matrizes de pesos cujos elementos denotam o valor do peso sináptico conectando o  $j$ -ésimo neurônio da camada ( $L$ ) ao  $i$ -ésimo neurônio da camada ( $L - 1$ ). Para a topologia ilustrada na Figura 35, tem-se:  $\mathbf{W}_{ji}^{(2)}$  é o peso sináptico conectando o  $j$ -ésimo neurônio da camada de saída ao  $i$ -ésimo neurônio da camada oculta;  $\mathbf{W}_{ji}^{(1)}$  é o peso sináptico conectando o  $j$ -ésimo neurônio da camada oculta ao  $i$ -ésima entrada da rede.
- $\mathbf{I}_{ji}(L)$  São vetores cujos elementos denotam a entrada ponderada em relação ao

Figura 35 – Representação da MLP para o estudo de caso multivariável.



Fonte: Elaborada pelo autor.

$j$ -ésimo neurônio da camada  $L$ , os quais são definidos por:

$$\mathbf{I}_j^{(1)} = \sum_{i=0}^{n_1} \mathbf{W}_{ji}^{(1)} \cdot x_i \Leftrightarrow I_j^{(1)} = W_{j0}^{(1)} \cdot x_0 + W_{j1}^{(1)} \cdot x_1 + \dots + W_{jn_1}^{(1)} \cdot x_{n_1}, \quad (3.7)$$

$$\mathbf{I}_j^{(2)} = \sum_{i=0}^{n_2} \mathbf{W}_{ji}^{(2)} \cdot Y_i^{(1)} \Leftrightarrow I_j^{(2)} = W_{j0}^{(2)} \cdot Y_0^{(1)} + W_{j1}^{(2)} \cdot Y_1^{(1)} + \dots + W_{jn_2}^{(2)} \cdot Y_{n_2}^{(1)}. \quad (3.8)$$

c)  $\mathbf{Y}_j^{(L)}$  São vetores cujos elementos denotam a saída do  $j$ -ésimo neurônio em relação à camada  $L$ , os quais são definidos por:

$$\mathbf{Y}_j^{(1)} = g \left( I_j^{(1)} \right) \quad (3.9)$$

$$\mathbf{Y}_j^{(2)} = h \left( I_j^{(2)} \right) \quad (3.10)$$

onde  $g$  é a função transferência adotada para cada camada.

Assim, definido os dados de entrada da rede, a sequência de procedimentos computacionais visando o processo de treinamento é explicitada a seguir em



termos do Pseudocódigo 1. No algoritmo a seguir, as entradas da rede neural artificial MLP serão generalizadas pela representação  $x_n$ . Já as saídas por  $Y_j^{(L)}$ .

---

**Pseudocódigo 1** Estimação do Modelo Neural MLP
 

---

```

1: Obter o conjunto de amostras de treinamento  $\{x^{(k)}\}$ 
2: Associar o vetor de saída desejada  $\{d^{(k)}\}$  ▷  $\{x^{(k)}, d^{(k)}\}$ 
3: Iniciar  $\mathbf{W}_{ji}^{(1)}$ ,  $\mathbf{W}_{ji}^{(2)}$ 
4: Especificar taxa de aprendizagem  $\{\alpha\}$ 
5:  $\{\text{época} \leftarrow 0\}$  ▷ Inicia contador de épocas
6: while  $|E_M^{atual} - E_M^{anterior}| \leq \varepsilon$  do
7:    $E_M^{anterior} \leftarrow E_M$  ▷ Armazena o erro da época anterior
8:   for  $k \leftarrow 1$  to  $N$  do ▷  $N$  é quantidade de amostras
9:     Obter  $I_j^{(1)}$  e  $Y_j^{(1)}$  ▷ Equações (2.4) e (2.5)
10:    Obter  $I_j^{(2)}$  e  $Y_j^{(2)}$  ▷ Equações (2.6) e (2.7)
11:    Determinar  $\delta_j^{(2)}$  ▷ Equação (2.8)
12:    Ajustar  $\mathbf{W}_{ji}^{(2)}$  ▷ Equação (2.10)
13:    Determinar  $\delta_j^{(1)}$  ▷ Equação (2.11)
14:    Ajustar  $\mathbf{W}_{ji}^{(1)}$  ▷ Equação (2.14)
15:   end for
16:   Obter  $\mathbf{W}^{(1)}$  e  $\mathbf{W}^{(2)}$  ajustados
17:    $E_M^{atual} \leftarrow E_M$  ▷ Armazena o erro atual
18:    $\text{época} \leftarrow \text{época} + 1$ 
19: end while

```

---

A variável *época* possui o próprio número de época que foram necessárias para a efetivação do treinamento da rede, isto é, quantas vezes foram necessárias apresentar todas as amostras do conjunto de treinamento visando o ajuste das matrizes de pesos.

Como observa-se, a rede MLP é considerada treinada totalmente quando o erro  $\{E_M\}$  entre duas épocas sucessivas for inferior à precisão  $\{\varepsilon\}$  requerida ao problema de modelagem do sistema de tanques. Tal erro é calculado a partir da Equação 3.11.

$$E_M = \sqrt{\frac{\sum (y(t) - \hat{y}(t))^2}{N}}. \quad (3.11)$$

No entanto, na presente pesquisa tanto a variável *época* como o erro de precisão foram utilizados como critério de parada, por considerar que a precisão desejada pode ser inalcançável.

### 3.5.2 Modelo Neural: *Wavelet Neural Network*

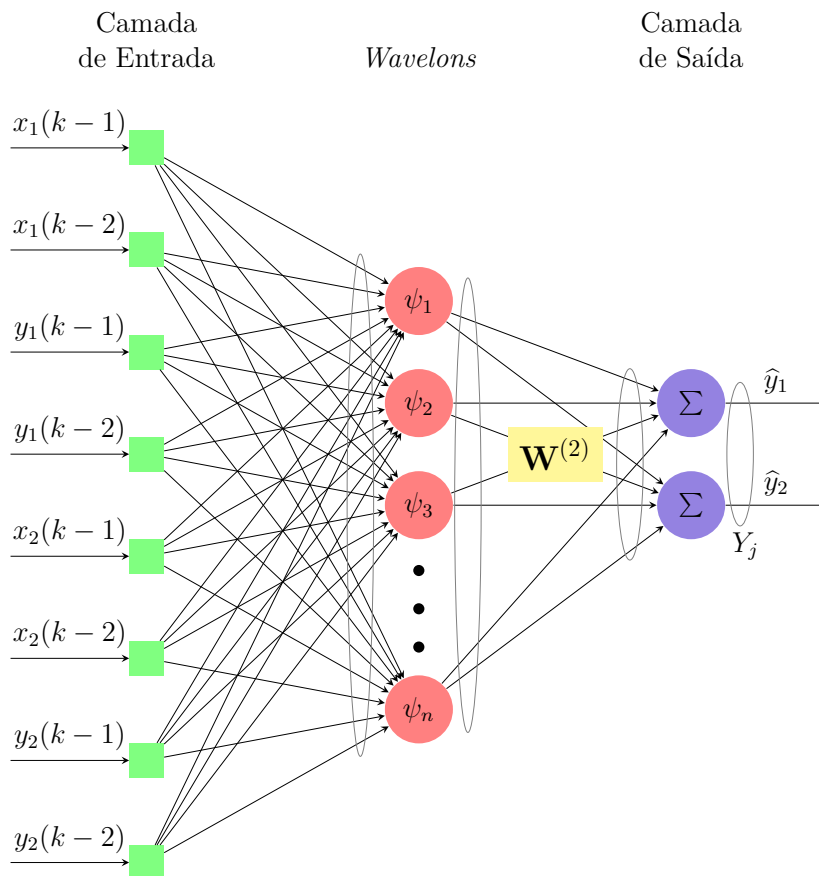
Conforme afirmado anteriormente, as redes neurais WNN e MLP são bem semelhantes em muitos aspectos, como por exemplo, a aprendizagem é realizada a partir de uma amostra aleatória de pares de entrada-saída observados. Além disso, são arquiteturas

do tipo *feedforward* e o algoritmo de aprendizagem utilizado para estimação também é o *backpropagation*, que usa o gradiente reverso da função do erro quadrático médio, sendo a função objetivo a ser minimizada.

A grande diferença entre as duas redes neurais supracitadas é justamente as funções de ativação dos neurônios da camada oculta. Na WNN os parâmetros *Wavelons*  $\{w_k; \lambda_k; t_k\}$  precisam ser inicializados e os neurônios da camada de saída é do tipo linear para ambas as redes neurais. Outra diferença significativa é que as entradas da rede neural *wavelet* são enviadas diretamente para a camada escondida, não havendo ponderação por pesos sinápticos. Geralmente, essa topologia é adotada quando se utiliza WNNs adaptativas, em que os parâmetros de dilatação e translação são ajustados em cada época de treinamento.

Logo, tomando como referência a Figura 36, no algoritmo a seguir é apresentada a sequência de procedimentos computacionais, Pseudocódigo 2, visando a obtenção do modelo neural WNN. As entradas da rede neural artificial WNN serão generalizadas pela representação  $x_n$ . Já as saídas estimadas pela RNA por  $\hat{y}_j$ .

Figura 36 – Representação da WNN para o estudo de caso multivariável.



Fonte: Elaborada pelo autor.

No presente trabalho, o objetivo final da WNN é aproximar uma função  $f(x)$  do domínio  $D = [a, b]$  usando a equação definida em 2.25. Antes de executar a rede neural

*wavelet* para estimar a função desejada, os parâmetros adaptativos  $w_i$ ,  $t_i$  e  $\lambda_i$  precisam ser inicializados. Conforme pode-se observar no pseudocódigo para treinamento da WNN, primeiramente, inicializou-se os pesos sináptico com valores aleatórios entre 0 e 1. Segundo (VEITCH, 2005), para definir os parâmetros  $t_i$  e  $\lambda_i$ , seleciona-se um ponto  $p$  dentro do intervalo  $[a, b]$ , em que a variável  $i$  corresponde aos respectivos *wavelons*, e aplica-se as seguintes equações:

$$t_i = p, \quad (3.12)$$

$$\lambda_i = \xi(b - a), \quad (3.13)$$

onde  $\xi > 0$  é normalmente definido como 0,5. Repete-se agora esta inicialização: tomando os intervalos  $[a, p]$  e  $[p, b]$  e definindo  $t_{i+1}$ ,  $\lambda_{i+1}$ , respectivamente. Isto é recursivamente repetido até que todos *wavelons* sejam inicializados.

---

#### Pseudocódigo 2 Estimação do Modelo Neural WNN

---

```

1: Obter o conjunto de amostras de treinamento  $\{x^{(k)}\}$ 
2: Associar o vetor de saída desejada  $\{d^{(k)}\}$  ▷  $\{x^{(k)}, d^{(k)}\}$ 
3: Iniciar  $w_{ji}^{(1)}$ ,  $t_k$  e  $\lambda_k$ 
4: Especificar taxa de aprendizagem  $\{\eta\}$ 
5: {época ← 0} ▷ Inicia contador de épocas
6: while  $|E_M^{atual} - E_M^{anterior}| \leq \varepsilon$  do
7:    $E_M^{anterior} \leftarrow E_M$  ▷ Armazena o erro da época anterior
8:   for  $k \leftarrow 1$  to  $N$  do ▷  $N$  é quantidade de amostras
9:     Obter Saídas da RNA  $\hat{y}_j$  ▷ Equação (2.25)
10:    Determinar  $\frac{\partial E}{\partial \lambda_{i,k}}$  ▷ Equação (2.26)
11:    Ajustar  $\lambda_{i,k}(n + 1)$  ▷ Equação (2.32)
12:    Determinar  $\frac{\partial E}{\partial t_{i,k}}$  ▷ Equação (2.27)
13:    Ajustar  $t_{i,k}(n + 1)$  ▷ Equação (2.33)
14:    Determinar  $\frac{\partial E}{\partial w_k}$  ▷ Equação (2.28)
15:    Ajustar  $w_k(n + 1)$  ▷ Equação (2.34)
16:   end for
17:   Obter  $w$ ,  $t$  e  $\lambda$  ajustados
18:    $E_M^{atual} \leftarrow E_M$  ▷ Armazena o erro atual
19:   época ← época+1
20: end while

```

---

Da mesma forma que a rede MLP, a WNN será considerada treinada totalmente quando o erro  $\{E_M\}$  entre duas épocas sucessivas for inferior à precisão  $\{\varepsilon\}$  requerida ao problema de modelagem do sistema de tanques (Equação 3.11) ou quando atingir uma determinada quantidade de épocas.

## 3.6 Validação dos Modelos do Sistema de Tanques

Nenhum modelo, por definição, representará o sistema físico em todos os aspectos. Portanto, já que o modelo provavelmente é representativo do sistema em alguns aspectos apenas, ele será considerado válido se incorporar aquelas características do sistema que são fundamentais para a aplicação em questão (AGUIRRE, 2015).

Comparar a simulação do modelo obtido com dados medidos é provavelmente a forma mais usual de se validar um modelo. Nesse caso, deseja-se saber se o modelo reproduz ao longo do tempo os dados observados. Sem dúvida, esse procedimento é importante e, em muitas aplicações, não é trivial obter um modelo cuja simulação se aproxime das observações. Por ser relativamente simples, tal procedimento é muito comum (AGUIRRE, 2015).

No entanto, a simulação do modelo não indica se as falhas do modelo são corrigíveis. Para isso, é necessário realizar análise de resíduos, pois esta análise informa se o procedimento de estimação foi estatisticamente sadio (AGUIRRE, 2015). Diante disso, o presente trabalho adotou as seguintes abordagens para validar os modelos: simulação e análise de resíduos.

### 3.6.1 Simulação dos Modelos: Saída Estimada vs Saída Medida

Após o término do treinamento da rede, pode-se então utilizá-lo para estimar as saídas do sistema que foi mapeado frente às novas amostras que serão apresentadas em suas entradas. Para esta fase de operação, a sequência de passos é apresentada, conforme os pseudocódigos 3 e 4, para as duas redes neurais utilizadas.

---

#### Pseudocódigo 3 Validação do Modelo Neural MLP

---

- 1: Obter o conjunto de amostras de teste  $\{x\}$
  - 2: Assumir  $\mathbf{W}_{ji}^{(1)}$  e  $\mathbf{W}_{ji}^{(2)}$  já ajustadas no treinamento
  - 3: Obter  $I_j^{(1)}$  e  $Y_j^{(1)}$  ▷ Equações (2.4) e (2.5), respectivamente
  - 4: Obter  $I_j^{(2)}$  e  $Y_j^{(2)}$  ▷ Equações (2.6) e (2.7), respectivamente
  - 5: Disponibilizar  $Y_j^{(2)}$  ▷ Saídas da Rede
- 

---

#### Pseudocódigo 4 Validação do Modelo Neural WNN

---

- 1: Obter o conjunto de amostras de teste  $\{x\}$
  - 2: Assumir  $\mathbf{W}_{ji}$ ,  $t_k$  e  $\lambda_k$  já ajustadas no treinamento
  - 3: Obter  $Y_j$  ▷ Equação(2.25)
  - 4: Disponibilizar  $Y_j$  ▷ Saídas da Rede
- 

É importante observar que dos parâmetros adaptativos são realizados somente na fase de treinamento das redes, em que se aplicam os passos *forward* e *backward* a fim de

proceder correções sinápticas necessárias. Já na fase de validação, somente a fase *forward* é processada com o objetivo de gerar as saídas da rede.

### 3.6.2 Análise de Resíduos

Do ponto de vista da validação de modelos os resíduos são a parte dos dados que o modelo não conseguiu identificar. Logo, um modelo ideal consegue estimar tudo o que for informação nos dados. Se isso ocorrer, então os resíduos conterão apenas aquilo que não é estimável e, por tanto, serão dados aleatórios, ruídos brancos (AGUIRRE, 2015). O ponto de partida para essa análise é a definição do vetor de resíduos  $\xi = y - \delta\hat{\theta}$ , em que  $y$  são os dados de identificação usados para estimar  $\hat{\theta}$  e construir  $\delta$ . Para verificar o quanto os resíduos dos modelos obtidos são brancos, utiliza-se o histograma do resíduo (AGUIRRE, 2015).

Um vetor de valores  $\xi$  é linearmente branco se sua função de autocorrelação for nula para todos os valores de atraso maiores ou iguais a um. Ao testar a aleatoriedade de um vetor de resíduos, deve-se ter em mente que tal vetor é o resultado do ajuste de uma estrutura matemática (os regressores do modelo) a um determinado conjunto de dados. Nesse contexto, considera-se que o vetor de resíduos seja não autocorrelacionado, ou seja, suponha que  $\xi$  seja tal que  $r_\xi = 0, \forall \neq 0$ . Assim, o termo “linearmente” branco é usado para indicar situações em que não há autocorrelação (linear) no vetor considerado (AGUIRRE, 2015).

A análise de resíduos também permite verificar o quão particular é um resultado. Nesse caso, utiliza-se a correlação cruzada entre o vetor de resíduos e o sinal de entrada  $u(k)$ , ou seja,  $r_{u\xi}$ . Assim, se  $r_{u\xi}(k) = 0, \forall k$ , tem-se que o vetor de erros cometidos pelo modelo não depende do sinal de entrada usado para obter o modelo (AGUIRRE, 2015). As Equações 3.14 e 3.15 representam as funções de autocorrelação e correlação cruzada (lineares), respectivamente.

$$r_\xi(\tau) = E\{\xi(k - \tau) \cdot \xi(k)\} = \delta(\tau), \quad (3.14)$$

$$r_{u\xi}(\tau) = E\{u(k - \tau) \cdot \xi(k)\} = 0, \forall \tau. \quad (3.15)$$

No caso de sistemas não lineares, é concebível que um modelo linear consiga explicar toda a informação linear dos dados. Se for esse o caso, os resíduos de tal modelo, em princípio, passarão pelos dois testes mencionados: autocorrelação dos resíduos e correlação cruzada. Entretanto, é de se esperar que no vetor de resíduos haja correlações não lineares. A fim de detectar tais correlações, utilizou-se a função não linear representada pela Equação

3.16, que permite calcular a função de correlação cruzada entre o vetor de resíduos e o sinal de entrada  $u(k)$ , ou seja,  $r_{u^2\xi}$  (BILLINGS; VOON, 1986).

$$r_{u^2\xi}(\tau) = E \left\{ (u^2(k - \tau) - \overline{u^2(k)})\xi(k) \right\} = 0, \forall \tau, \quad (3.16)$$

Evidentemente, poderá haver nos resíduos dependências não detectáveis por 3.16 e, nesse caso, funções específicas poderão ser derivadas. De fato, novos conjuntos de funções têm sido propostos para a validação de modelos não lineares (AGUIRRE, 2015).

## 4 Resultados

No desenvolvimento deste trabalho, inicialmente, procurou-se informar a teoria necessária para o desenvolvimento da pesquisa apresentando as principais características da rede WNN, de modo a deixar claro os aspectos dessa ferramenta visando, também, facilitar o processo de implementação da mesma. No capítulo de Materiais e Métodos, foram apresentados os algoritmos e equações para buscar a configuração ótima para a estrutura de identificação, além de apresentar os estudos de casos, em que adotou-se duas configurações diferentes para a abertura da válvula de acoplamento entre os dois tanques: válvula totalmente fechada (tanques desacoplados) e válvula parcialmente aberta.

Considerando esses estudos de casos, o presente capítulo ilustra os resultados da identificação abordando diversos parâmetros e aspectos das estruturas neurais utilizadas, como o número de neurônios, taxa de aprendizagem, quantidade de épocas e o comportamento do erro médio quadrático. Buscando, assim, avaliar a viabilidade da Rede Neural *Wavelet* como ferramenta de identificação de sistemas de tanques multivariável não linear.

Os parâmetros supracitados são considerados para fins de comparação das duas RNAs utilizadas, MLP e WNN, de modo que após apresentação da validação dos modelos estimados com o auxílio dessas redes, será discutido e analisado as hipóteses consideradas neste trabalho a respeito da *Wavelet Neural Network*.

### 4.1 Resultados da Validação: Tanques Desacoplados

Neste estudo de caso, a válvula de acoplamento entre os tanques foi totalmente fechada de modo que não houvesse nenhum fluxo de líquido entre os dois tanques. Conseqüentemente, nessa configuração, não existe nenhum tipo de acoplamento entre os sinais de saída (nível) e muito menos entre os sinais de entrada, ou seja, os tanques de nível funcionam de forma completamente independentes, passando a se comportar como dois sistemas dinâmicos SISO (*Single Input, Single Output*).

Diante disso, buscando-se analisar e comparar a dinâmica do sistema com tanques de nível atuando de forma isolada, aplicou-se os mesmos sinais de excitação *Pseudo Random Binary Signal* nos dois sistema de nível (SISO) formado por bomba hidráulica, sensor e o respectivo tanque de nível associado. Tais sinais são ilustrados pela Figura 37 e Figura 38, correspondendo aos sinais da etapa de treino e etapa de validação, respectivamente.

A resposta de cada sistema de nível para o sinal de excitação aplicado é ilustrado pela Figura 39. É evidente que por mais que os dois sistemas sejam formados por atuadores e sensores com as mesmas características e os tanques possuem as mesmas dimensões,

Figura 37 – Configuração com Tanques de Nível Desacoplados: Sinal PRBS utilizado para excitar as bombas hidráulicas dos Tanques 01 e 02 para obtenção dos dados de treinamento.

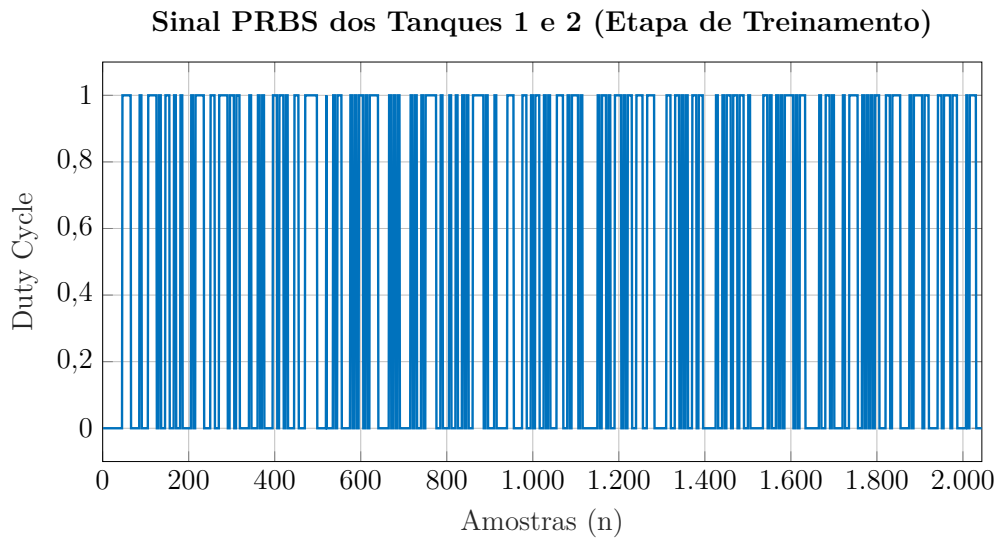
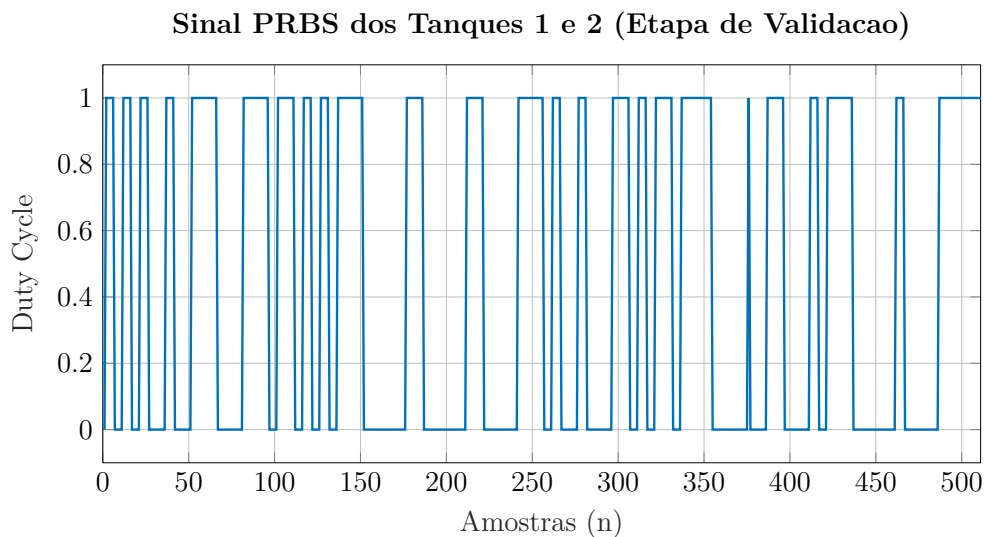


Figura 38 – Configuração com Tanques Desacoplados: Sinal PRBS utilizado para validação dos modelos dos Tanques.

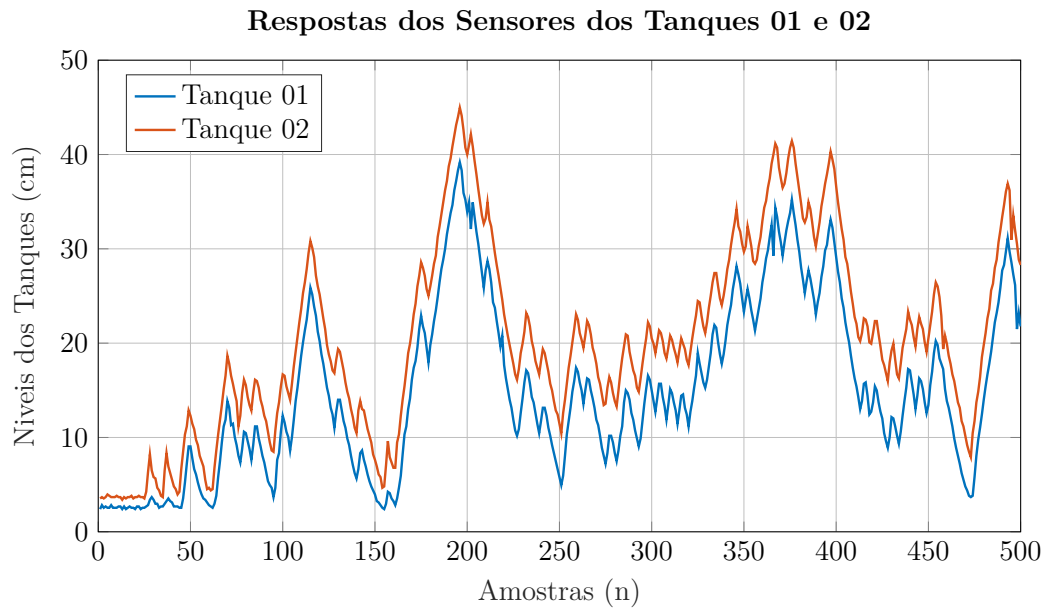


ainda sim existe uma diferença entre as dinâmicas. Isso se deve principalmente à válvula de escoamento de cada tanque, pois são válvulas manuais e certamente terão vazão diferentes. Além disso, o tanque é artesanal, o que faz com que apresente diferenças estruturais, como desníveis.

Resumidamente, para gerar os resultados gráficos de validação a seguir, foi aplicado o mesmo sinal PRBS com 2040 amostras (Figura 37) para estimação do modelo do Tanque



Figura 39 – Configuração com Tanques de Nível Desacoplados: Comparação das dinâmicas entre os tanques de nível 01 e 02.



Fonte: Elaborada pelo autor.

de Nível 1 e Tanque de Nível 2, e conseqüentemente o treinamento das redes neurais utilizadas *Multilayer Perceptron* e *Wavelet*. Da mesma forma, na fase de validação, aplicou-se o mesmo sinal PRBS com 510 amostras (Figura 38) para obter os resultados dos dois modelos neurais de cada tanque.

No entanto, será apresentado a seguir apenas a validação do modelo do Tanque 1 utilizando as redes neurais MLP e WNN, pois os resultados da validação do Tanque de nível 2 foram muito semelhantes aos resultados do Tanque de Nível 1, para o caso desacoplado.

#### 4.1.1 Modelo Neural MLP

Para estimação do modelo neural *Multilayer Perceptron*, basicamente, é necessário calcular os pesos sinápticos de modo aproximar a saída da RNA ao valor de referência desejado. No entanto, para isso é necessário parametrizar a rede, ou seja, definir a quantidade de neurônios ocultos, taxa de aprendizagem e inicialização dos pesos sinápticos.

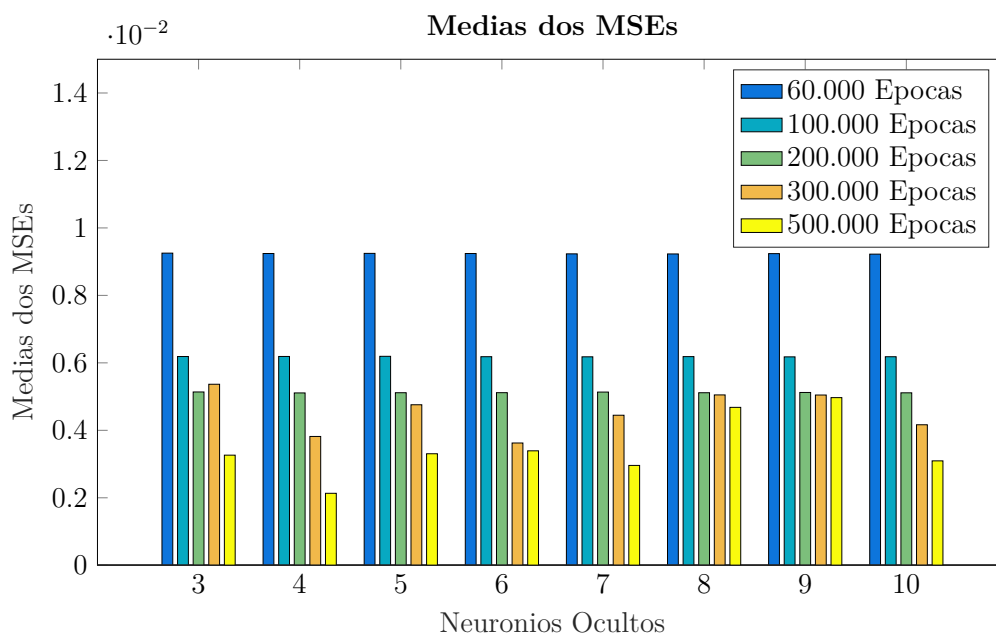
No primeiro momento realizou-se uma análise considerando o erro médio quadrático como critério de parada, adotando-se dois valores para a taxa de aprendizagem 0,1 e 0,05. O objetivo dessa primeira análise é considerar quantas épocas são necessárias para o erro de estimação próximo de  $10^{-2}$  e  $10^{-3}$ . Nessa análise considerou-se a quantidade de neurônios ocultos igual a 4 (quatro). O resultado desse experimento é apresentado na Tabela 5, e a taxa de aprendizagem adotada para as análises seguintes foi de 0,1.

Tabela 5 – Análise da MLP: Erro como Critério de Parada

Erro	Quantidade de Número de Épocas	
	Taxa de Aprendizagem = 0,1	Taxa de Aprendizagem = 0,05
$1 \cdot 10^{-2}$	67.590 épocas	116.588 épocas
$5 \cdot 10^{-3}$	206.760 épocas	570.138 épocas

Em seguida, buscando-se a melhor topologia para a rede neural MLP, são realizados vários treinamentos e validações da rede utilizando os dados dos tanques, de modo que variou-se o número de neurônios ocultos e a quantidade de épocas. Para cada treinamento registra-se o valor do erro médio quadrático e ao final dos 20 treinamentos calcula-se a média e a variância dos valores adquiridos para um dado número de neurônios e épocas. Os parâmetros utilizados para essa análise são apresentados na Tabela 3. Neste segundo momento, o critério de parada utilizado é apenas o número de épocas, variando-se entre 60.000 e 500.000 épocas. Os resultados das médias e variâncias encontradas são ilustrados na Figura 40 e Figura 41, respectivamente.

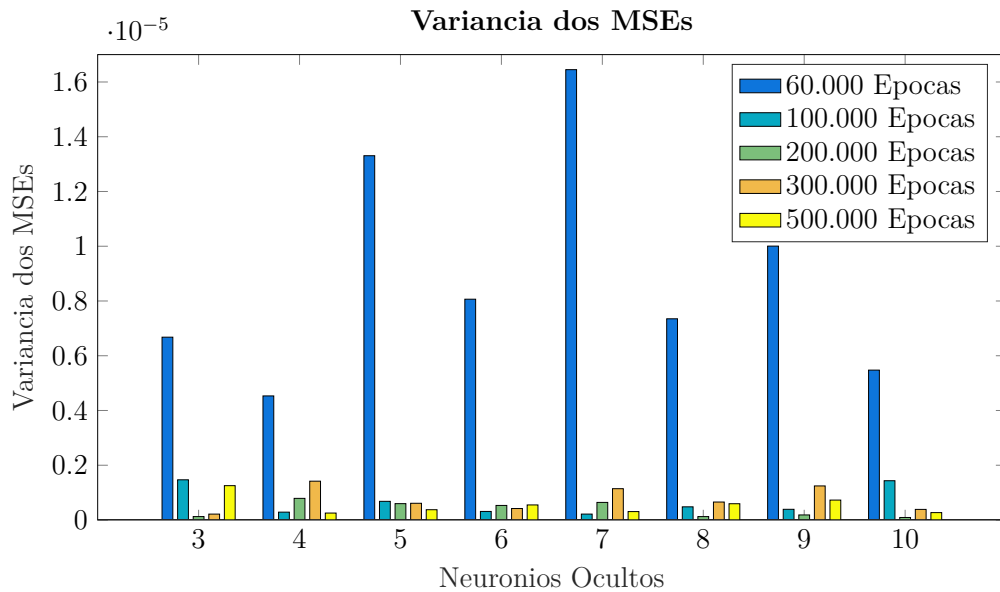
Figura 40 – Configuração com Tanques de Nível Desacoplados (MLP): Médias dos MSEs para validação do Tanque de Nível 01.



Fonte: Elaborada pelo autor.

Esse procedimento é realizado para o intervalo de 3 a 10 neurônios ocultos e para cada tanque de forma isolada (desacoplado). E conforme pode-se observar nos resultados para o Tanque de Nível 1, a quantidade de neurônios ideal para atingir o menor erro depende da quantidade de número de épocas também. No entanto, os valores mínimos são atingidos para quantidade de neurônios igual a 4 (quatro) para camada oculta. Além disso, a variância foi a menor dentre todos os casos, indicando que a maioria das médias

Figura 41 – Configuração com Tanques de Nível Desacoplados (MLP): Variâncias dos MSEs para validação do Tanque de Nível 01.



Fonte: Elaborada pelo autor.

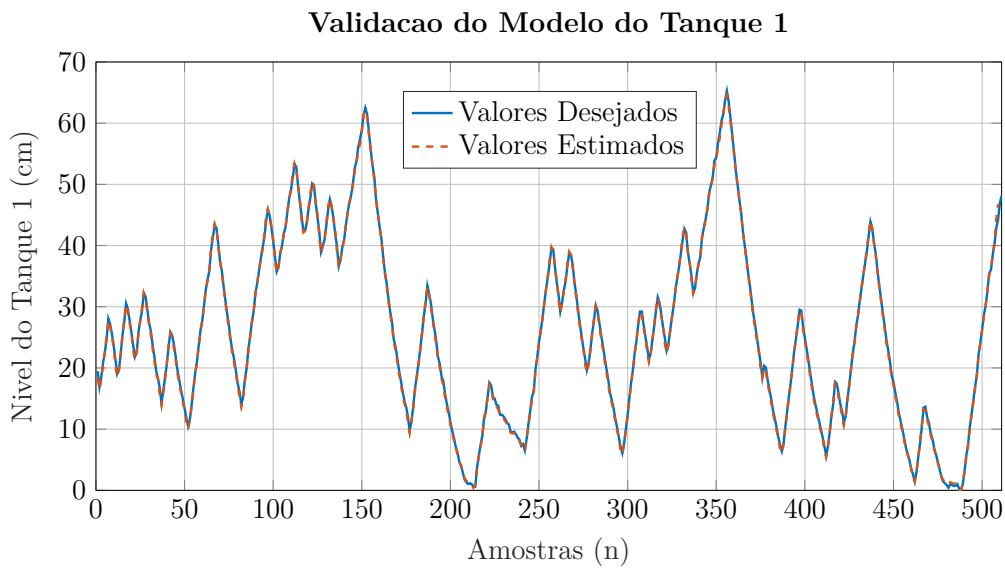
calculadas dentro dos 20 treinamentos foram bem próximas entre si. Logo, adotou-se essa quantidade de neurônios ocultos.

Outro ponto importante a ser observado, analisando a Figura 40, é que o erro diminui com o aumento do número de épocas, havendo uma queda significativa entre 60.000 e 500.000 épocas. Já a variância dos erros são da ordem de  $10^{-5}$  a  $10^{-6}$  quando adota-se mais de 500.000 épocas, indicando que para esses casos o erro médio encontrado em cada topologia estão bem próximos.

Vale lembrar que considerou-se a dimensão igual a 4 para camada de entrada dado que se adotou uma estrutura NNARX com atraso de transporte igual a 1. Já o número de neurônios da camada de saída varia de acordo com o número de saídas do sistema. Conseqüentemente, como identificou-se cada tanque de forma isolada, essa camada teve apenas um neurônio, pois trata-se de um sistema monovariável, ou seja, a RNA só estima apenas uma saída do sistema.

Para apresentação dos resultados de validação a seguir, adotou-se 4 (quatro) neurônios na camada oculta e a rede foi treinada com a topologia 4:4:1 (4 entradas, 4 neurônios oculto e 1 neurônio na camada de saída), tendo como critério de parada de treinamento o erro médio quadrático de 0,005. Para atingir este valor de EMQ, foram necessárias 206.760 épocas. A taxa de aprendizagem também foi definida de forma empírica com o valor de 0,1. Além disso, os pesos sinápticos foram inicializados de forma aleatória. A apresentação do resultado da validação do modelo neural MLP do Tanque de Nível 1 é ilustrado pela Figura 42.

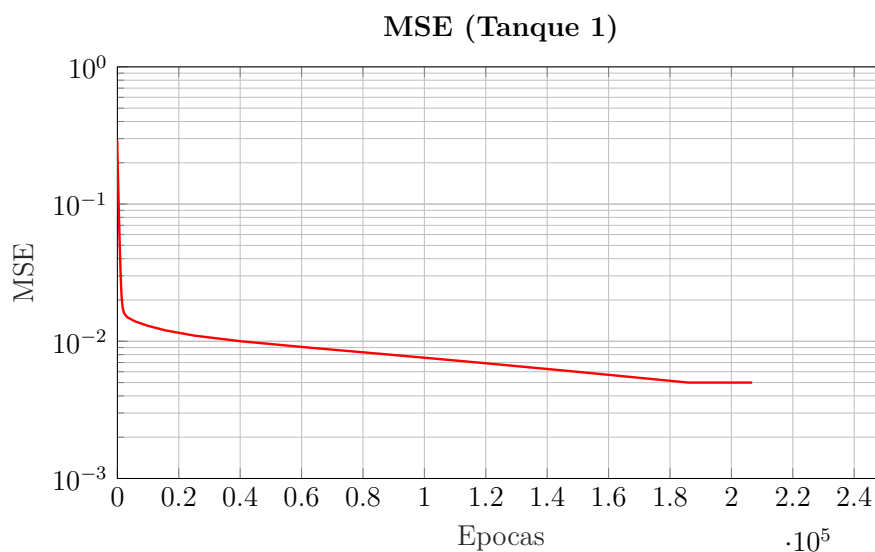
Figura 42 – Configuração com Tanques de Nível Desacoplados (MLP): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 01).



Fonte: Elaborada pelo autor.

É perceptível, diante da Figura 42, que a rede foi capaz de mapear a dinâmica do tanque de forma satisfatória e o modelo apresenta bons resultados de validação. Observa-se que a validação foi feita para toda faixa de operação do sistema de nível e que os valores picos do sinal também foram bem mapeados, por mais que a quantidade de amostras nestes pontos sejam em menor quantidade. O comportamento do erro médio quadrático durante cada época do treinamento é ilustrado pela Figura 43.

Figura 43 – Configuração com Tanques de Nível Desacoplados (MLP): MSE (206.760 Épocas de Treinamento).



Fonte: Elaborada pelo autor.

Conforme se observa na Figura 43, o erro decai de forma significativa nas primeiras 20.000 épocas, atingindo o erro médio quadrático mínimo estipulado após 206.760 épocas.

#### 4.1.2 Modelo Neural WNN

Para estimação do modelo neural *Wavelet Neural Network*, além de calcular de forma recursiva os pesos sinápticos  $w$ , é necessário calcular, também, os parâmetros de translação  $t$  e o parâmetro de dilatação  $\lambda$  descritos em capítulos anteriores. No entanto, para que isso seja possível é necessário parametrizar a rede, ou seja, definir a quantidade de *Wavelons*, taxa de aprendizagem e inicialização dos parâmetros adaptativos.

Assim como é feito para a estimação do modelo neural utilizando MLP, no primeiro momento realiza-se uma análise considerando o erro médio quadrático como critério de parada, adotando-se dois valores para a taxa de aprendizagem 0,1 e 0,05. O objetivo dessa primeira análise é considerar quantas épocas são necessárias para o erro de estimação próximo de  $10^{-2}$  e  $10^{-3}$ . O resultado desse experimento é apresentado na Tabela 6, e a taxa de aprendizagem adotada para as análises seguintes foi de 0,1.

Tabela 6 – Análise da WNN: Erro como Critério de Parada

Erro	Quantidade de Número de Épocas	
	Taxa de Aprendizagem = 0,1	Taxa de Aprendizagem = 0,05
$1 \cdot 10^{-2}$	14 épocas	20 épocas
$5 \cdot 10^{-3}$	163 épocas	276 épocas

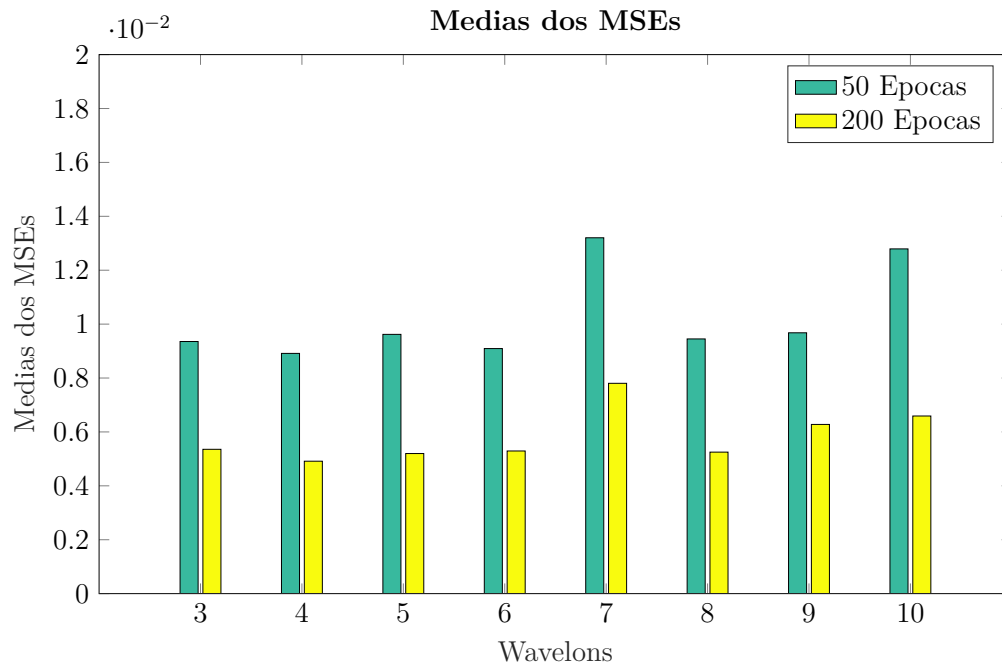
Em seguida, busca-se a melhor topologia para a rede neural WNN realizando vários treinamentos e validações com diferentes números de *wavelons* e a quantidade de épocas. Para cada treinamento registrou-se o valor do erro quadrático médio e ao final dos 20 treinamentos calculou-se a média e a variância dos valores adquiridos para um dado número de neurônios e épocas. Os resultados são ilustrados na Figura 44 e Figura 45.

Conforme se observa nas figuras citadas, considerou-se na análise de topologias a quantidade de 3 a 10 *wavelons*. Diante disso, é perceptível que 4 neurônios na camada oculta já mostra resultados satisfatórios de erros. Logo, adota-se essa quantidade de *wavelons*.

Outro ponto importante a ser observado, analisando a Figura 44, é que o erro diminui com o aumento do número de épocas, havendo uma queda significativa para 200 épocas. Já a variância dos erros são da ordem de  $10^{-4}$ , indicando que para esses casos o erro médio encontrado para cada topologia estão bem próximos.

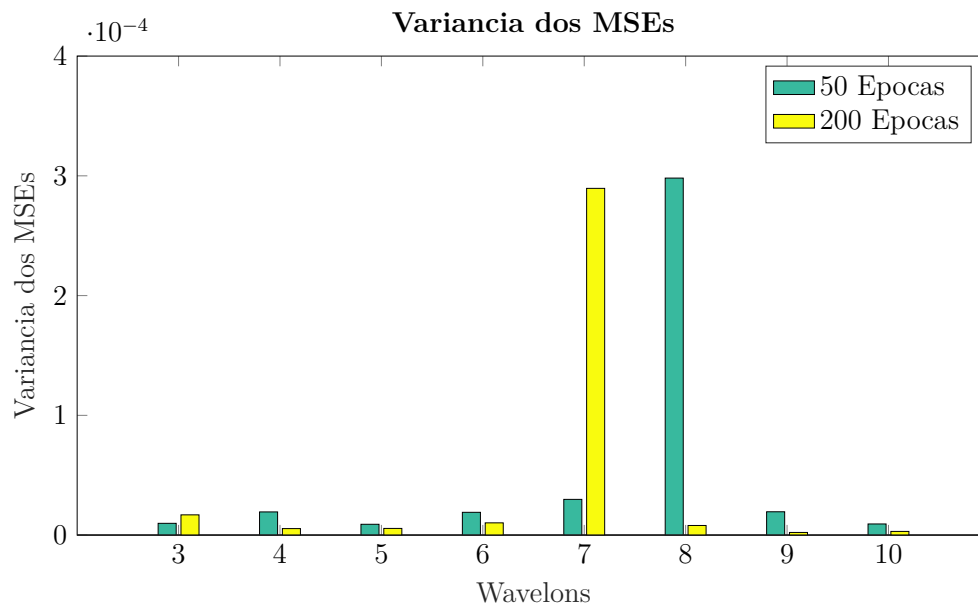
Diante disso, para gerar os resultados apresentados a seguir, adotou-se 4 (quatro) *wavelons* e a rede foi treinada com a topologia 4:4:1, tendo como critério de parada de treinamento o erro médio quadrático de 0,005. Para atingir este valor de EMQ, foram

Figura 44 – Configuração com Tanques de Nível Desacoplados (WNN): Médias dos MSEs para validação do Tanque de Nível 1.



Fonte: Elaborada pelo autor.

Figura 45 – Configuração com Tanques de Nível Desacoplados (WNN): Variâncias dos MSEs para validação do Tanque de Nível 1.



Fonte: Elaborada pelo autor.

necessárias 163 épocas. A taxa de aprendizagem também foi definida considerando o valor de 0,1. Além disso, os pesos sinápticos foram inicializados de forma aleatória e os parâmetros de translação e parâmetros de dilatação foram inicializados por meio das

Equações 3.13 e 3.12.

Os parâmetros ajustados finais para da *Wavelet Mexican Hat* são apresentados nas Tabelas 7 e 8. Como foi adotado a topologia de rede [4:4:1], tem-se 16 (dezesseis) parâmetros de translação ( $t_{ij}$ ), pois para cada  $i$ -ésimo *wavelon* tem-se 4 funções *wavelets* associadas ( $j$ -ésima). Já o parâmetro de dilatação, adotou-se um valor de dilatação para cada *wavelon*.

Tabela 7 – Coeficientes de Translação - *Wavelets*: Tanques Desacoplados

$ij$	$t_{ij}$			
	1	2	3	4
1	0,5014	0,4968	0,5140	1,6185
2	0,5040	0,4928	0,1873	0,4051
3	0,3910	0,3191	-0,0070	-0,0592
4	0,0043	0,1185	0,1003	0,1285

Tabela 8 – Coeficientes de Dilatação - *Wavelets*: Tanques Desacoplados

	$\lambda_i$			
	1	2	3	4
	1,5705	-0,4432	-0,9418	10,6347

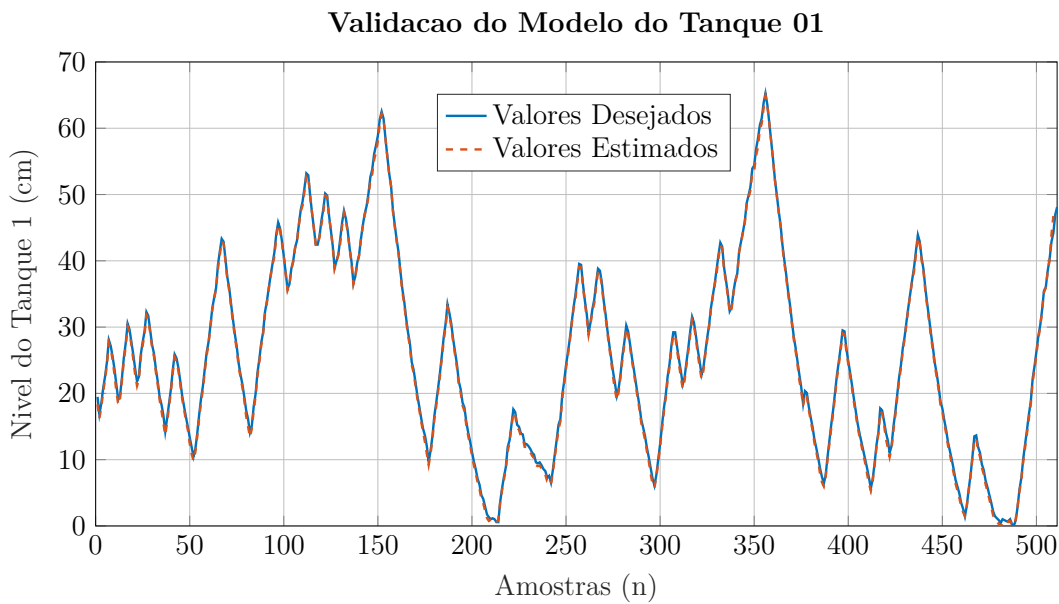
A apresentação do resultado da validação do modelo neural WNN do Tanque de Nível 1 é ilustrado pela Figura 46. Assim como o modelo neural MLP, o modelo WNN foi capaz de mapear a dinâmica do tanque de forma satisfatória e o modelo apresenta bons resultados de validação. Observa-se que a validação foi feita para toda faixa de operação do sistema de nível e que os valores picos do sinal também foram bem mapeados. Vale ressaltar, também, a quantidade consideravelmente menor de épocas, quando comparado com a quantidade de épocas necessárias para o modelo MLP atingir o mesmo erro mínimo. O comportamento do erro médio quadrático durante cada época do treinamento é ilustrado pela Figura 47.

Pode-se observar que o erro decai de forma significativa nas primeiras épocas, passando a ter uma diminuição do erro de forma mais gradual em épocas seguintes, atingindo o erro médio quadrático mínimo estipulado após 163 épocas.

## 4.2 Resultados da Validação: Tanques Acoplados

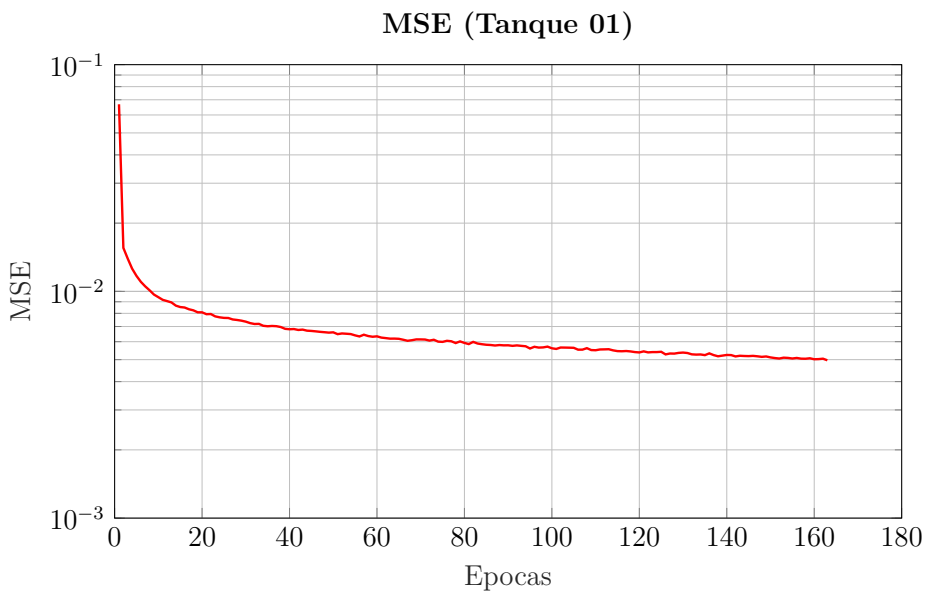
Neste estudo de caso a válvula de acoplamento entre os tanques de nível é parcialmente aberta em aproximadamente 50% da vazão total, ou seja, a partir de agora passa-se a trabalhar com um sistema multivariável com duas entradas e duas saídas, uma vez que a excitação de uma bomba hidráulica, com o sinal PRBS, passa a influenciar no nível dos

Figura 46 – Configuração com Tanques de Nível Desacoplados (WNN): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 1).



Fonte: Elaborada pelo autor.

Figura 47 – Configuração com Tanques de Nível Desacoplados (WNN): MSE (163 Épocas de Treinamento).



Fonte: Elaborada pelo autor.

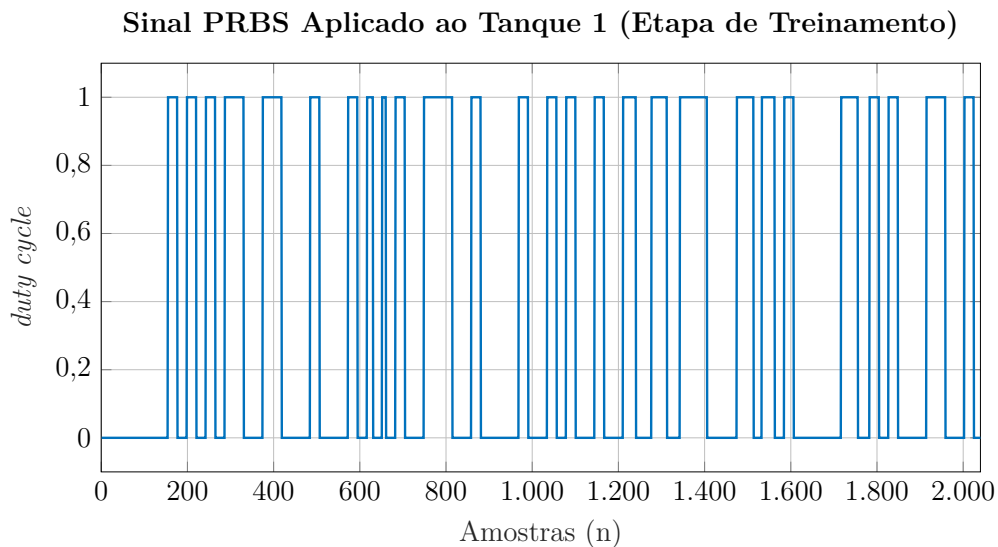
dois tanques. Diante disso, tem-se um acoplamento entre sinais, tornando o processo não linear e mais complexo de se identificar.

Com a finalidade de comparação dos resultados gerados por cada RNA, foram utilizados os mesmos dados para treino e validação dos modelos neurais MLP e WNN.



Além disso, para o caso multivariável (tanques acoplados), adotou-se sinais de excitação diferentes para as bombas hidráulicas com a finalidade de observar o efeito de cada uma das entradas sobre o nível dos tanques. Para o Tanque 1 o sinal aplicado na bomba hidráulica é ilustrado na Figura 48 e o sinal aplicado na bomba do Tanque 2 é ilustrado na Figura 49. É possível observar que os sinais possuem frequências diferentes. Isso foi possível variando-se o parâmetro *band* da função *indinput*, Equação 3.1.

Figura 48 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para excitar a bomba hidráulica do Tanque 1 para obtenção dos dados de treinamento.



Fonte: Elaborada pelo autor.

Os sinais de excitação do sistema para a fase de validação são ilustrados nas Figuras 50 e 51. Conforme informado anteriormente, os dados utilizados para treinamento e validação, foram originados de um único conjunto de amostras que foi dividido 2040 amostras para treino e 510 amostras para validação.

#### 4.2.1 Modelo Neural MLP

Novamente é necessário parametrizar a rede, ou seja, definir a quantidade de neurônios ocultos, taxa de aprendizagem e inicialização dos pesos sinápticos. Só após essa definição que se pode dar início a estimação dos modelos por meio dos ajustes dos pesos sinápticos  $w$ .

Inicialmente, realizou-se uma análise considerando o erro médio quadrático como critério de parada, adotando-se dois valores para a taxa de aprendizagem 0,1 e 0,05. Da mesma forma do estudo de caso com os tanque desacoplados, o objetivo dessa primeira análise é considerar quantas épocas são necessárias para o erro estimação próximo de  $10^{-2}$

Figura 49 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para excitar a bomba hidráulica do Tanque 2 para obtenção dos dados de treinamento.

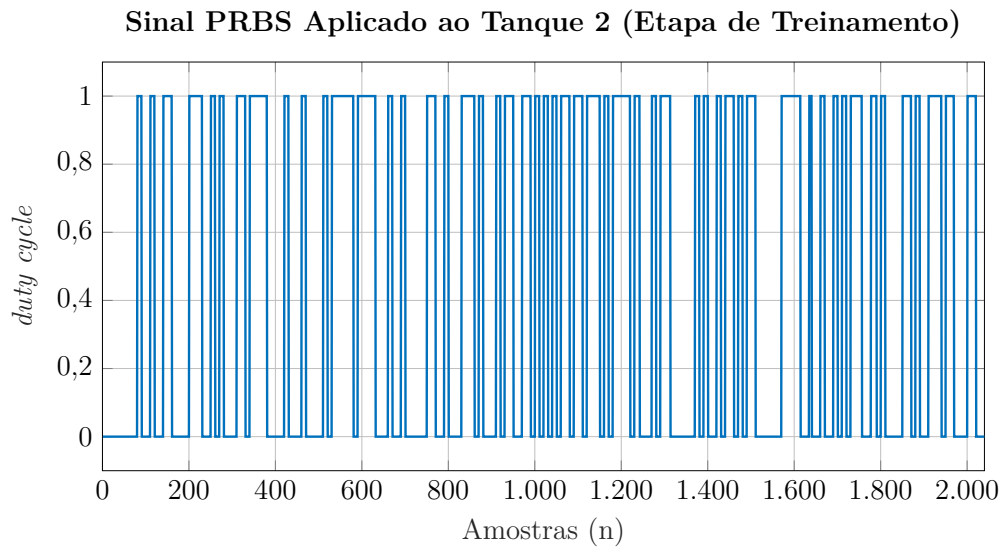
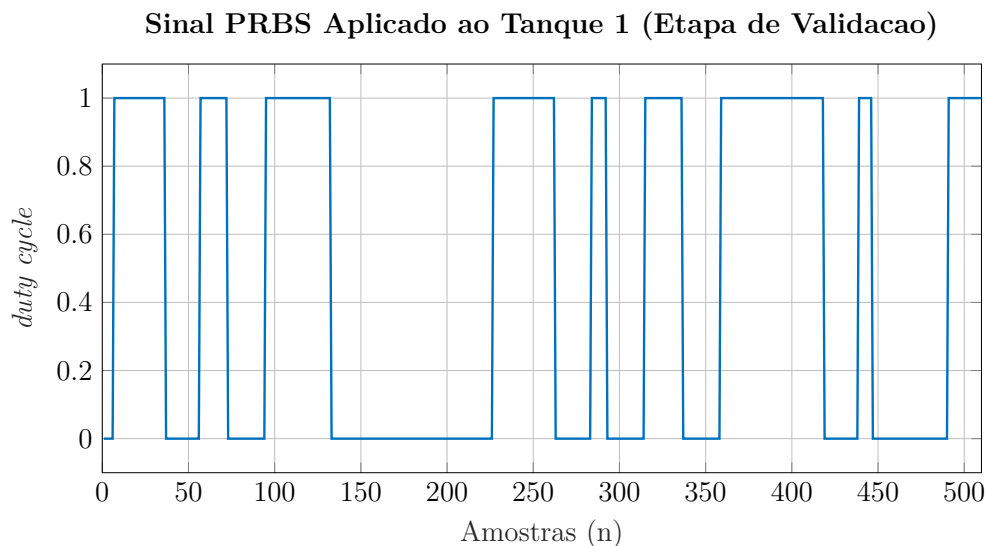


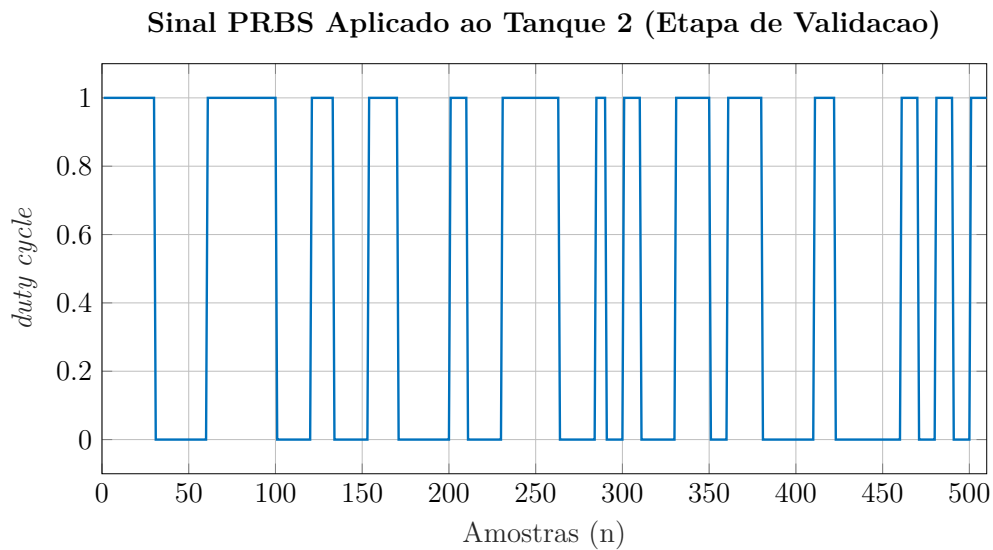
Figura 50 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para Validação do modelo Tanque 1.



e  $10^{-3}$ . O resultado desse experimento é apresentado na Tabela 9, e a taxa de aprendizagem adotada para as análises a seguir foi de 0,1.

Além disso, também foi definido a especificação da topologia dessas redes neurais de forma empírica. Conforme explicado para o estudo de caso anterior, realizou-se vários treinamentos variando-se o número de épocas e neurônios ocultos com a finalidade de encontrar a topologia da rede MLP que ofereça os melhores resultados. Os gráficos a seguir

Figura 51 – Configuração com Tanques de Nível Acoplados: Sinal PRBS utilizado para Validação do modelo Tanque 2.



Fonte: Elaborada pelo autor.

Tabela 9 – Análise da MLP Multivariável: Erro como Critério de Parada

Erro	Quantidade de Número de Épocas	
	Taxa de Aprendizagem = 0,1	Taxa de Aprendizagem = 0,05
$1 \cdot 10^{-2}$	96.413 épocas	175.238 épocas
$5 \cdot 10^{-3}$	354.614 épocas	707.272 épocas

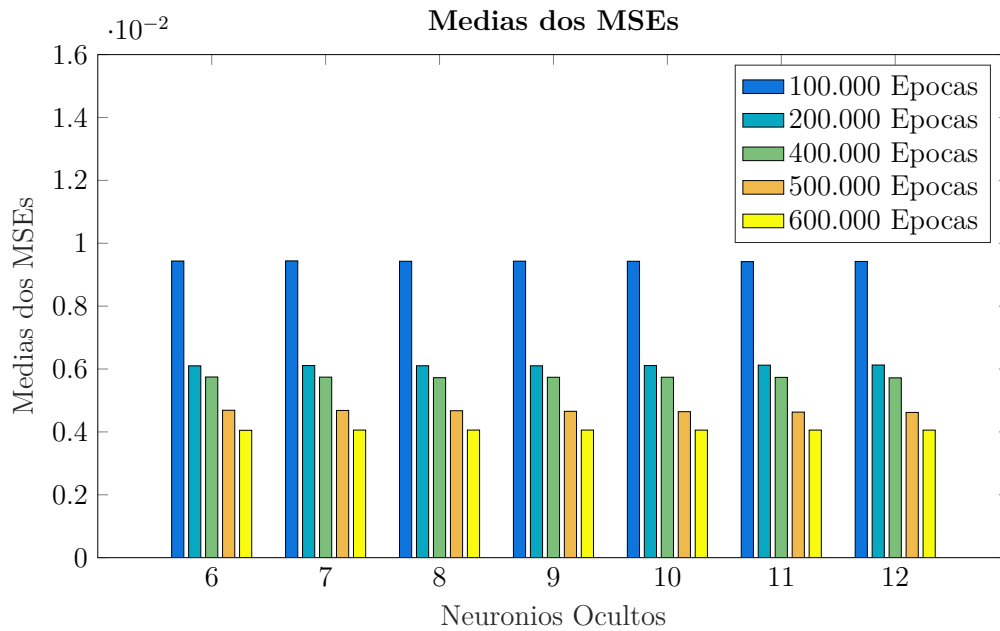
explicitam essa análise.

É possível observar que a quantidade de neurônios ocultos não possui tanta influência na redução do erro médio, uma vez que a média dos EMQs tiveram valores próximos. Percebe-se, então, que para este estudo de caso somente o aumento do número de épocas influencia na minimização do erro, pois a análise do aumento no número de neurônios não leva a nenhuma conclusão quanto a tendência do mesmo.

Diante disso, optou-se por trabalhar com 9 neurônios ocultos, pois para esta situação houve menor variância do erro médio para 600.000 épocas. No entanto, é importante ressaltar que para o mesmo número de épocas, a identificação dos tanques acoplados apresentou valores maiores de erro do que nos tanques desacoplados. Esse resultado era esperado, já que o processo multivariável apresenta maior complexidade para identificação por conta do acoplamento dos sinais de entrada e saída.

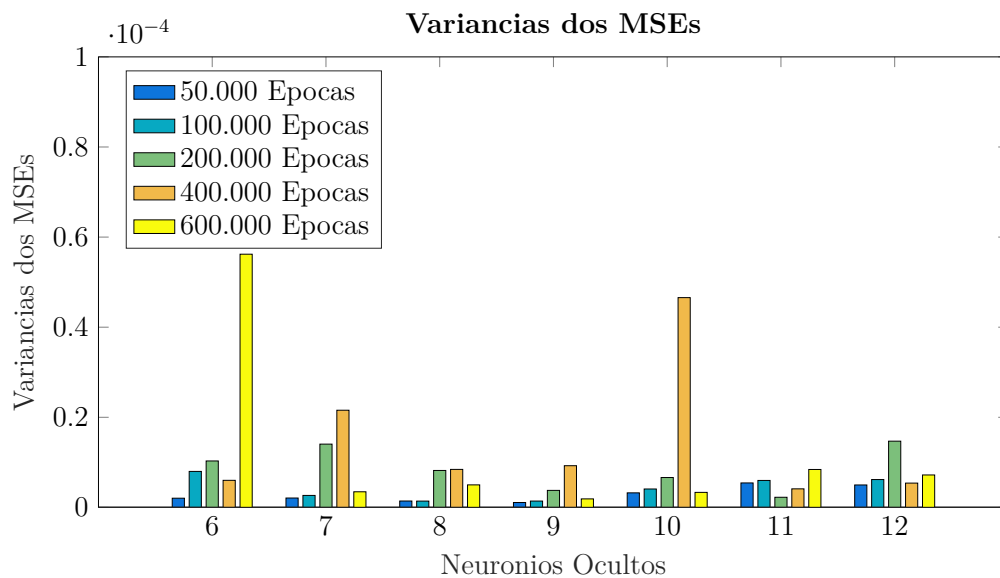
Diante disso, para gerar os resultados apresentados a seguir, adota-se 9 (nove) neurônios ocultos e a rede é treinada com a topologia 8:9:2, tendo como critério de parada de treinamento o erro médio quadrático de 0,005. Vale lembrar que adota-se oito entradas na primeira camada dado que se trata de um sistema multivariável e a estrutura adotada

Figura 52 – Configuração com Tanques de Nível Acoplados (MLP): Médias dos MSEs para Validação do Tanque de Nível 1.



Fonte: Elaborada pelo autor.

Figura 53 – Configuração com Tanques de Nível Acoplados (MLP): Variância dos MSEs para Validação do Tanque de Nível 1.



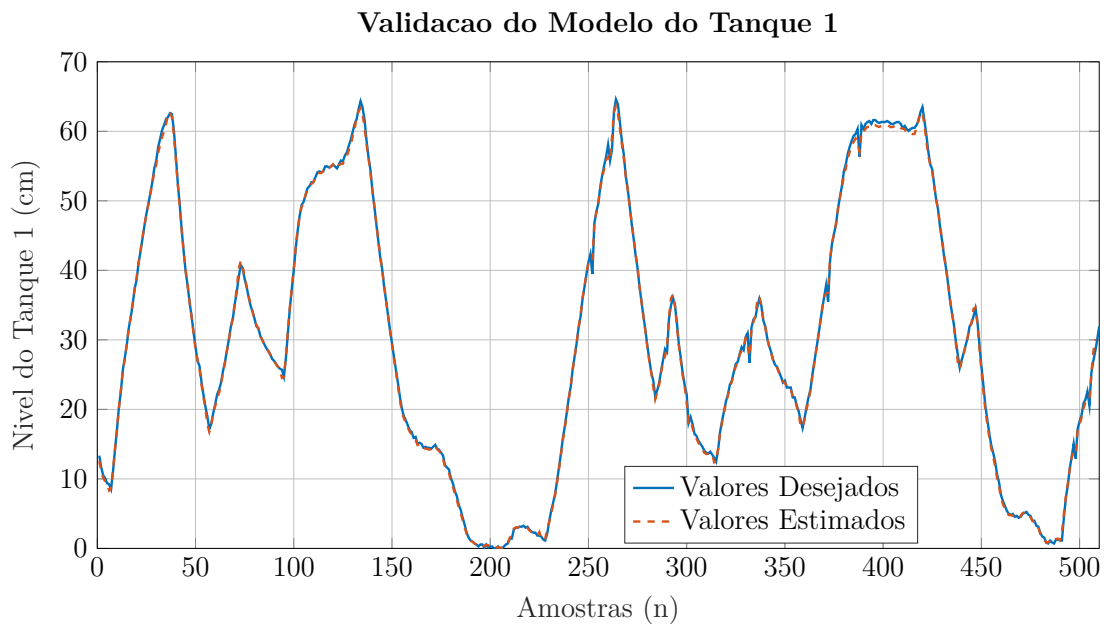
Fonte: Elaborada pelo autor.

foi a NNARX com atraso de transporte igual a 1. Já o número de neurônios da camada de saída varia de acordo com o número de saídas do sistema. Consequentemente, como identificou-se os tanques acoplados, essa camada têm dois neurônios.

Para configuração da rede MLP, supracitada, são necessárias 354.614 épocas de

treinamento para atingir o erro de 0,005. A taxa de aprendizagem também foi definida em 0,1. Além disso, os pesos sinápticos foram inicializados de forma aleatória. A apresentação do resultado da validação do modelo neural MLP do Tanque de Nível 1 e Tanque de Nível 2 é ilustrado pela Figura 54 e 55, respectivamente.

Figura 54 – Configuração com Tanques de Nível Acoplados (MLP): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 01).



É perceptível, diante das Figuras 54 e 55, que a rede foi capaz de mapear a dinâmica dos dois tanques de nível de forma satisfatória, apresentando bons resultados de validação para os dois tanques.

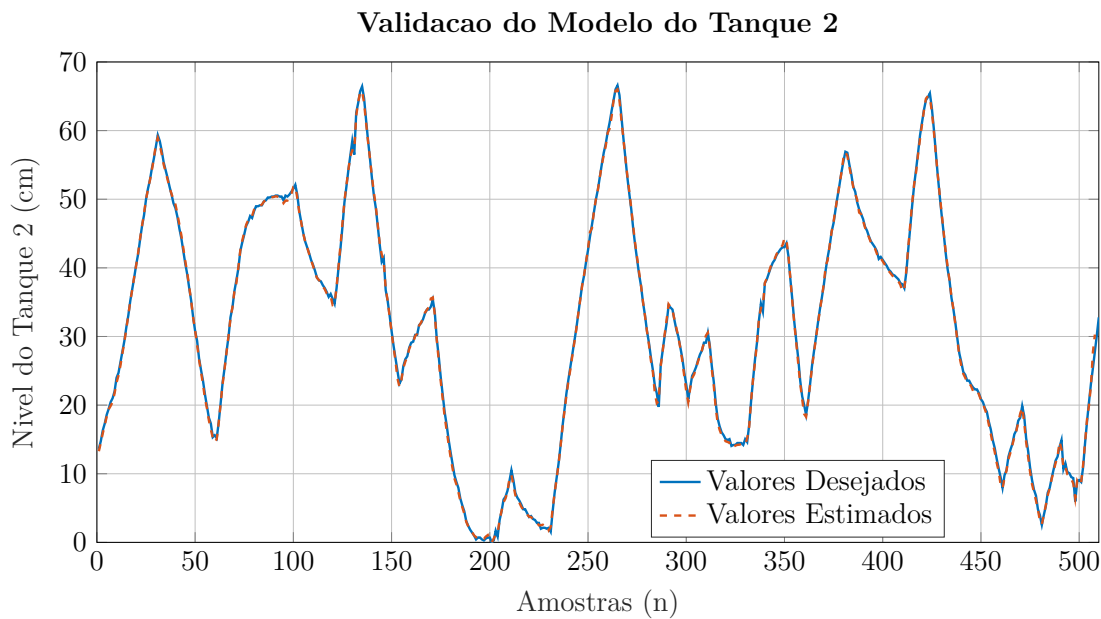
Como trata-se de um sistema multivariável de duas saídas, conseqüentemente tem-se erros de estimação para as duas saídas. No entanto, o erro de estimação final é calculado a partir da média dos EMQs das saídas. Logo, a cada época de treinamento da rede *Multilayer Perceptron*, e conseqüente estimação do modelo, coletou-se o erro médio quadrático para as saídas relacionadas ao tanque e tirou-se a média. O comportamento do erro durante cada época do treinamento é ilustrado pela Figura 56.

Pode-se observar que o erro decai de forma significativa nas primeiras 50.000 épocas, passando a ter uma diminuição do erro de forma mais gradual em épocas seguintes. De forma geral é possível afirmar que a curva MSE comportou-se conforme esperado.

#### 4.2.2 Modelo Neural WNN

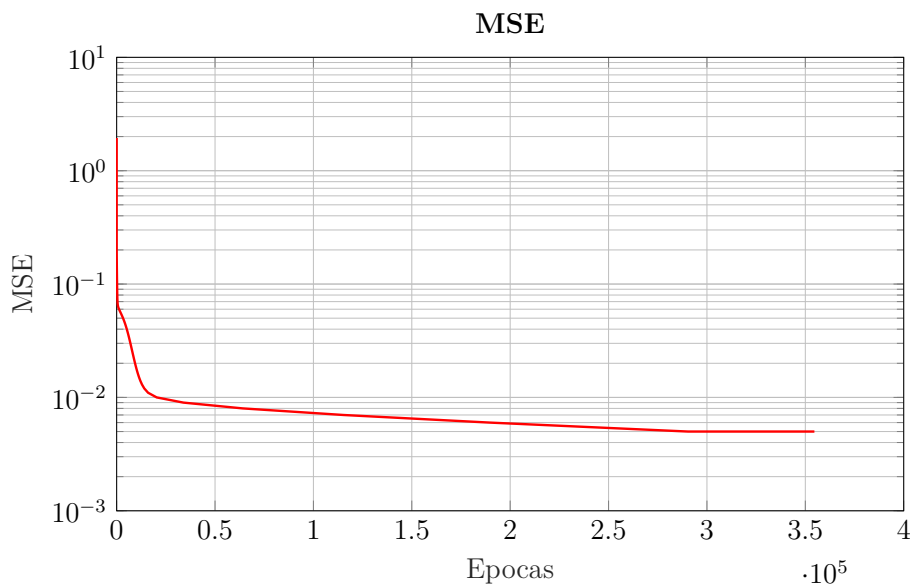
Assim como realizado para a estimação do modelo neural multivariável utilizando MLP, inicialmente, realizou-se uma análise considerando o erro médio quadrático como

Figura 55 – Configuração com Tanques de Nível Acoplados (MLP):Validação do modelo para saída 2 do sistema dinâmico (Tanque de Nível 02).



Fonte: Elaborada pelo autor.

Figura 56 – Configuração com Tanques de Nível Acoplados (MLP): MSE – (354.614 Épocas).



Fonte: Elaborada pelo autor.

critério de parada, adotando-se dois valores para a taxa de aprendizagem 0,1 e 0,05. O resultado desse experimento é apresentado na Tabela 10, e a taxa de aprendizagem adotada para as análises a seguir foi de 0,1.

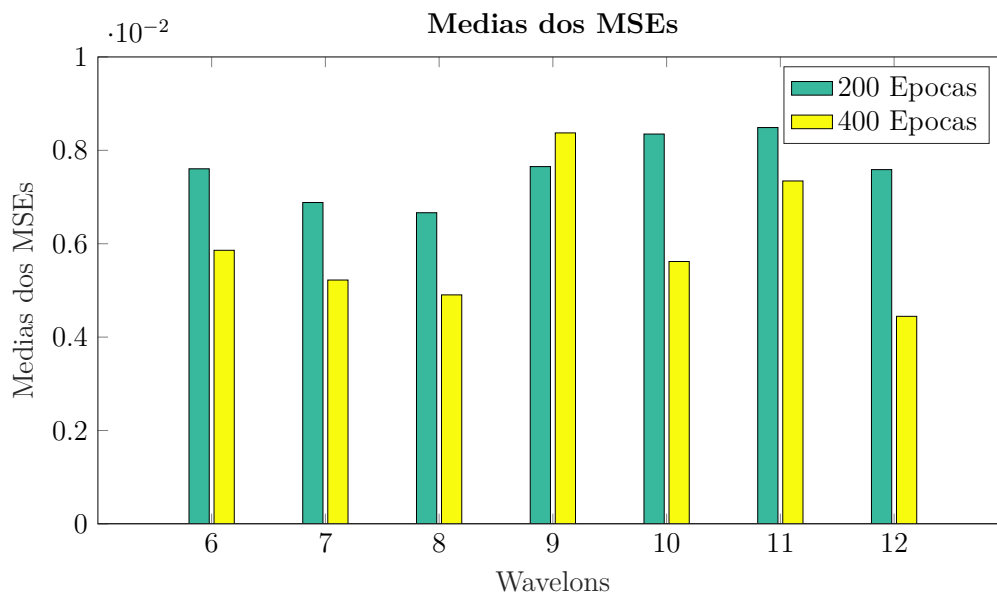
Pode-se observar que para a estimação do modelo utilizando a WNN, necessitou-se

Tabela 10 – Análise da WNN Multivariável: Erro como Critério de Parada

Erro	Quantidade de Número de Épocas	
	Taxa de Aprendizagem = 0,1	Taxa de Aprendizagem = 0,05
$1 \cdot 10^{-2}$	52 épocas	166 épocas
$5 \cdot 10^{-3}$	91 épocas	321 épocas

bem menos épocas de treinamento se comparado com a estimação do modelo MLP. De posse desse conhecimento, buscou-se a melhor topologia para a rede neural WNN realizando vários treinamentos e testes com diferentes números de *wavelons* e a quantidade de épocas. Os resultados para essa análise empírica são ilustrados na Figura 57 e Figura 58.

Figura 57 – Configuração com Tanques de Nível Acoplados (WNN): Médias dos MSEs para Validação do Tanque de Nível 01.



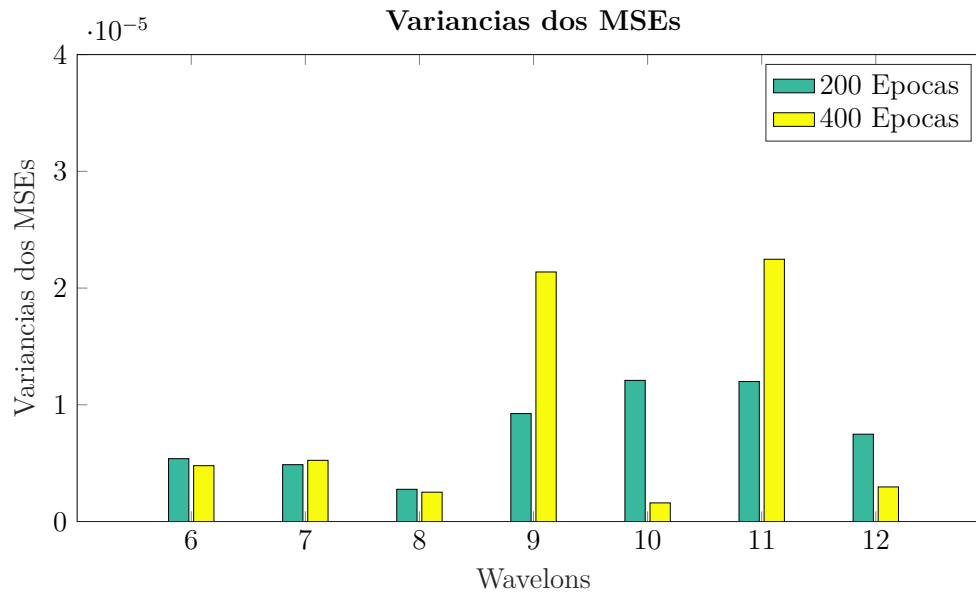
Fonte: Elaborada pelo autor.

É possível observar que a quantidade de *wavelons* possui influência na redução do erro médio, porém não de forma proporcional. Ou seja, o aumento de *wavelons* não levou a redução do erro, pois para alguns casos houve o aumento e em outros diminuição do mesmo. Assim, a análise do aumento no número de neurônios não leva a nenhuma conclusão quanto a tendência do mesmo.

Porém, valores satisfatórios foram atingidos para quantidade de neurônios igual a oito. Percebe-se, também, que para este estudo de caso o aumento do número de épocas não necessariamente influencia na minimização do erro.

Diante disso, optou-se por trabalhar com 8 *wavelons*, pois para esta situação houve menor erro médio e menor variância. No entanto, é importante ressaltar que para o mesmo

Figura 58 – Configuração com Tanques de Nível Acoplados (WNN): Variância dos MSEs para Validação do Tanque de Nível 01.



Fonte: Elaborada pelo autor.

número de épocas, a identificação dos tanques acoplados apresentou valores maiores de erro do que nos tanques desacoplados.

A Rede Neural *Wavelet* foi treinada utilizando a topologia 8:8:2 (8 entradas na primeira camada, 8 *wavelons* e 2 neurônio na camada de saída) até que o erro médio quadrático mínimo de 0,005 fosse alcançado. Para isso, foram necessárias 91 épocas de treinamento. A taxa de aprendizagem também foi definida em 0,1. Além disso, os pesos sinápticos foram inicializados de forma aleatória e os parâmetros de translação e parâmetros de dilatação foram inicializados por meio das Equações 3.13 e 3.12

Os parâmetros ajustados finais da *Wavelet Mexican Hat* são apresentados nas Tabelas 11 e 12. Como foi adotado a topologia de rede [8:8:2], tem-se 64 (sessenta e quatro) parâmetros de translação ( $t_{ij}$ ), pois para cada  $i$ -ésimo *wavelon* tem-se 8 funções *wavelets* associadas ( $j$ -ésima). Já o parâmetro de dilatação, adotou-se um valor de dilatação para cada *wavelon*.

A apresentação dos resultados da validação do modelo neural WNN do tanque de nível 01 e tanque de nível 02 é ilustrado pela Figura 59 e 60, respectivamente.

É perceptível, diante das Figuras 59 e 60, que a rede foi capaz de mapear a dinâmica dos dois tanques de nível de forma satisfatória e o modelo apresenta bons resultados de validação.

O comportamento do erro durante cada época do treinamento é ilustrado pela Figura 61, correspondendo ao erro de estimação.



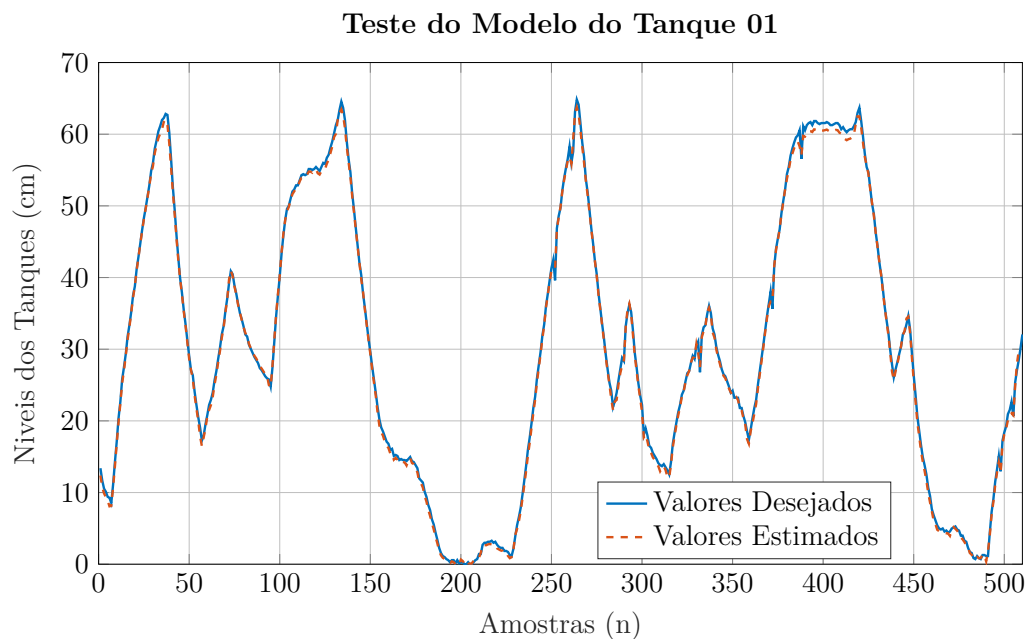
Tabela 11 – Coeficientes de Translação - *Wavelets*: Tanques Acoplados

$ij$	$t_{ij}$							
	1	2	3	4	5	6	7	8
1	0,4726	0,4697	0,6148	0,4868	0,4496	0,4292	1,2021	1,4011
2	0,5003	0,4703	0,0061	0,2677	0,4962	0,4642	0,0070	0,3759
3	0,4801	0,4749	-0,3600	-0,7265	0,5285	0,5851	0,5192	0,7005
4	0,2124	0,1988	0,1542	0,2220	0,2746	0,2644	-0,1414	-0,2616
5	0,5766	0,5554	0,1921	0,2493	0,5836	0,6419	-0,3143	-0,6505
6	0,5062	0,5037	0,0336	-0,3517	0,4615	0,4097	-0,0070	-0,2420
7	0,2116	0,1876	-0,2406	-0,3237	0,3470	0,2999	-0,3701	-0,5414
8	0,3340	0,3743	-0,0658	-0,0856	0,3776	0,3590	-0,1587	-0,1092

Tabela 12 – Coeficientes de Dilatação - *Wavelets*: Tanques Acoplados

$\lambda_i$							
1	2	3	4	5	6	7	8
1,2057	1,3341	-2,3161	-2,9489	-2,6198	-1,9071	-1,7770	-1,1463

Figura 59 – Configuração com Tanques de Nível Acoplados (WNN): Validação do modelo para saída 1 do sistema dinâmico (Tanque de Nível 01).



Fonte: Elaborada pelo autor.

Pode-se observar que o erro decai de forma significativa nas primeiras épocas em ambos os tanques, passando a ter uma diminuição do erro de forma mais gradual em épocas seguintes.

Figura 60 – Configuração com Tanques de Nível Acoplados (WNN): Validação do modelo para saída 2 do sistema dinâmico (Tanque de Nível 02).

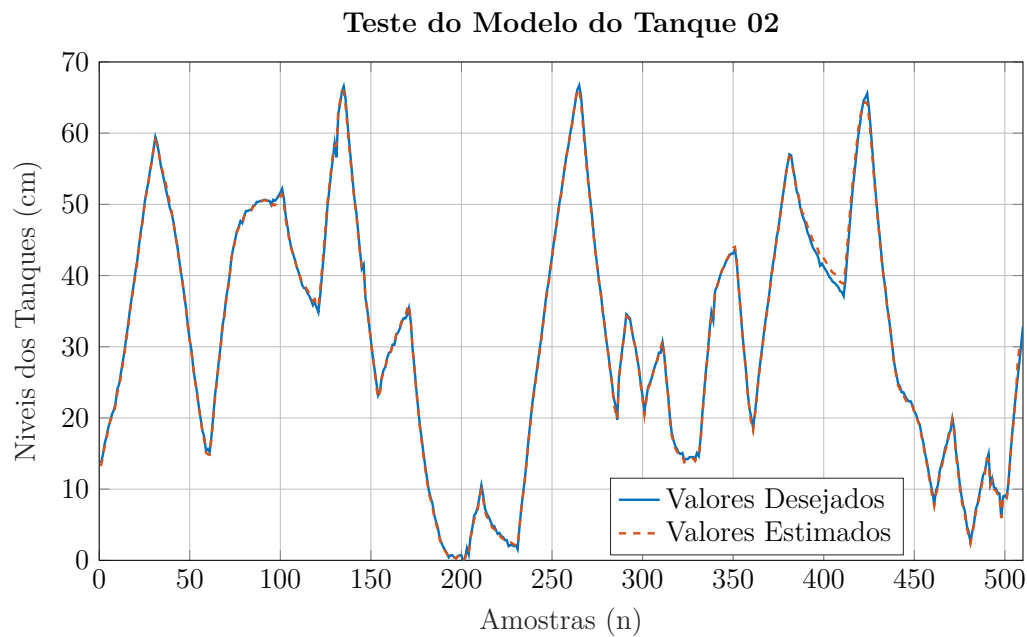
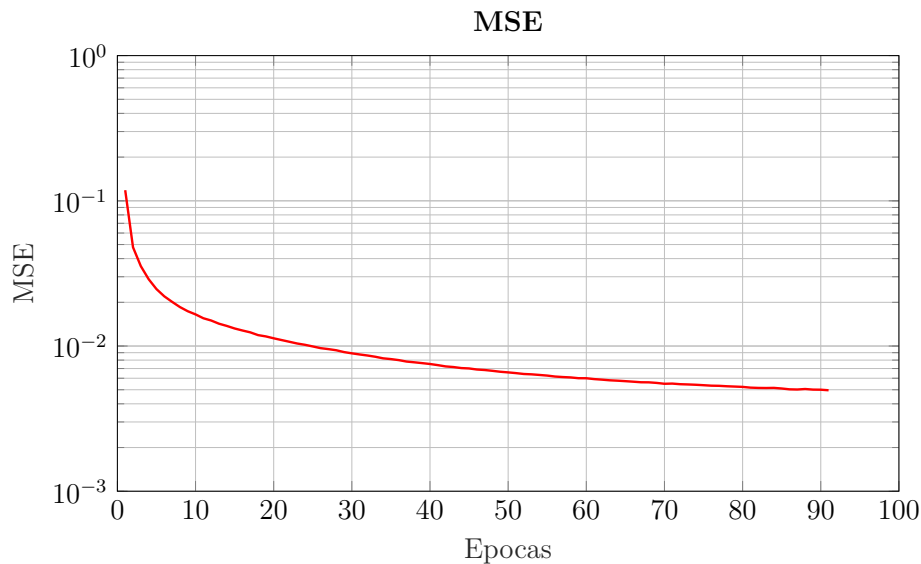


Figura 61 – Configuração com Tanques de Nível Acoplados (WNN): MSE (91 Épocas).



### 4.3 Análise das Hipóteses do Trabalho

Uma vez realizada a identificação do sistema de tanques de nível acoplados, com uma validação bem sucedida utilizando as redes neurais MLP e WNN, o presente tópico apresentará comparações entre as ferramentas utilizadas de modo a avaliar as hipóteses

propostas no início deste trabalho. São elas:

- **Hipótese 1:** A *Wavelet Neural Network* por ser bom aproximador universal, será capaz de realizar representações do sistema de tanques de nível acoplados.
- **Hipótese 2:** A *Wavelet Neural Network* apresentará boa exatidão para validação.
- **Hipótese 3:** A *Wavelet Neural Network* possui melhor convergência que a *Multilayer Perceptron*.

Nas subseções a seguir cada hipótese será discutida e avaliada de modo individual para fins de validação das mesmas e comparação das RNAs.

#### 4.3.1 Hipótese 1: Viabilidade da WNN aplicada na identificação de tanques de nível multivariável

Para validar a hipótese que considera a WNN como uma ferramenta viável para uso em identificação de sistemas, utilizou-se como parâmetro de comparação os resultados da modelagem obtidos por meio da Rede *Multilayer Perceptron*. Assim, considerando-se os resultados gráficos de validação apresentados no Capítulo de Resultados, fica evidente que a Rede Neural *Wavelet* mapeou de forma satisfatória a dinâmica do sistema de tanques de nível, tão quanto a Rede Neural *Multilayer Perceptron*. Além disso, conforme apresentado na Tabela 13, quando as duas Redes Neurais foram submetidas ao mesmo erro mínimo como critério de parada e com a mesma taxa de aprendizagem, observa-se que as duas conseguem alcançar o MSE mínimo estabelecido.

Tabela 13 – Comparação entre os MSEs

EMQ	Quantidade de Épocas			
	Taxa de Aprend. (0,1)		Taxa de Aprend. (0,05)	
	MLP	WNN	MLP	WNN
$1 \cdot 10^{-2}$	96.413	52	175.238	166
$5 \cdot 10^{-3}$	354.614	195	707.272	321

É possível observar que para todos as situações o número de épocas foram menores para o modelo neural *Wavelet Neural Network*. Logo, pode-se deduzir que a *Wavelet Neural Network* por ser melhor aproximador universal, foi capaz de realizar representações do sistema de tanques de nível com boa exatidão.

#### 4.3.2 Hipótese 2: Exatidão e capacidade de generalização da WNN

Para avaliar a hipótese de tolerância da WNN a ruídos de medição deve-se levar em consideração a análise de resíduos, uma vez que, segundo (AGUIRRE, 2015), do ponto

de vista da validação de modelos os resíduos são a parte dos dados que o modelo não conseguiu mapear. Logo, um modelo ideal consegue estimar tudo o que for informação nos dados. Se isso ocorrer, então os resíduos conterão apenas aquilo que não é estimável e, por tanto, serão dados aleatórios, brancos.

Assim, se os resíduos forem brancos, não há informação útil neles, ou seja, o modelo mapeou tudo o que era informação útil. Por outro lado, se os resíduos não forem brancos, haverá informação neles indicando que o modelo não conseguiu estimar toda a dinâmica do sistema real. Obviamente, exigir que o modelo seja ideal e, conseqüentemente, seus resíduos sejam brancos é impraticável. Um vez que para isso a saída estimada  $\hat{y}$  deve ser igual a saída desejada  $y$ . Portanto, para análise é necessário designar o conjunto de dados de validação para verificar se os resíduos são aleatórios ou não (AGUIRRE, 2015).

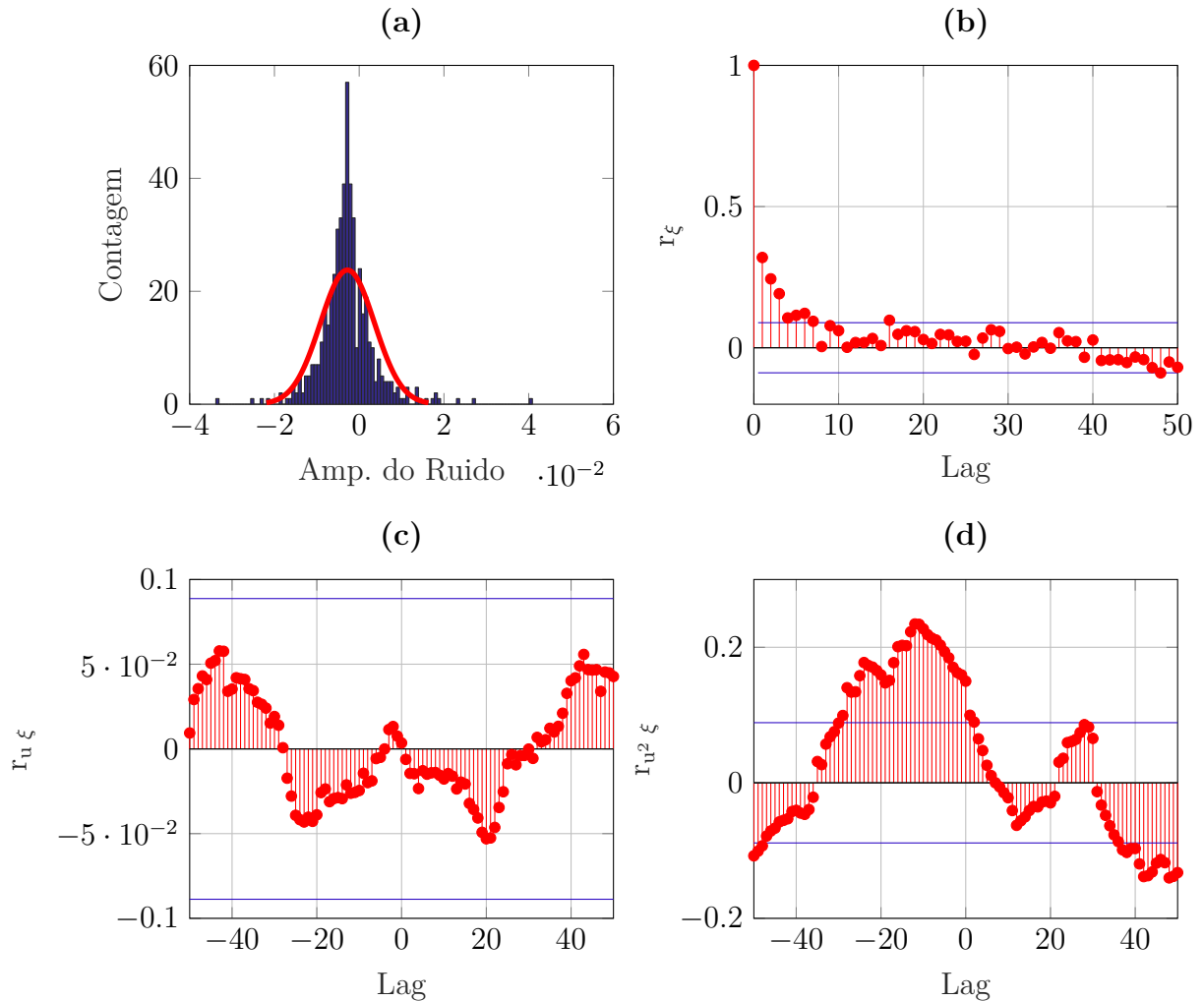
Para verificar o quanto os resíduos dos modelos obtidos são brancos, optou-se por utilizar o histograma do erro, uma vez que este gráfico ilustra como uma determinada amostra de dados está distribuída. Além disso, é utilizado a autocorrelação dos resíduos, correlações cruzadas linear e não linear entre o sinal de resíduo e o sinal de entrada. Considerando o caso multivariável e um erro médio final de treinamento de  $5 \cdot 10^{-3}$ , a Figura 62 e 63 apresentam os resultados para validação dos modelos dos Tanques de nível 1 e 2, respectivamente.

É possível perceber nas Figuras 62(a) e 63(a), que as amostras de dados do resíduo, aproxima-se da forma de uma distribuição gaussiana. No entanto, para ter a certeza o quão próximo está da função de probabilidade normal, gerou-se os gráficos apresentados pela Figura 64, que compara a distribuição dos resíduos dos modelos dos Tanques 1 e 2, respectivamente, com a distribuição normal.

Conforme pode-se observar, os dados com símbolos de marcador  $X$ , nos gráficos mostrados na Figura 64, representam a distribuição dos resíduos, e a reta é a linha de referência que representa a distribuição normal teórica. Se os dados da amostra tivessem uma distribuição normal, os pontos de dados apareceriam ao longo da toda a linha de referência. Como a distribuição dos resíduos não é uma função gaussiana perfeita, pois  $\hat{y}$  não é exatamente igual a  $y$ , conseqüentemente, o gráfico dos resíduos possuem uma curvatura nas extremidades.

Outro forma de testar a aleatoriedade de um vetor de resíduos é por meio da função de autocorrelação do resíduo. Um vetor  $\xi$  é linearmente branco se a autocorrelação for nula para todos os valores de atraso maiores ou iguais a um, ou seja,  $r_{\xi}(k) = 0, \forall k \neq 0$ . Diante disso, analisando as Figuras 62(b) e 63(b), percebe-se que para a maioria dos valores de atrasos a autocorrelação permaneceu dentro do limite de confiança e próximo de zero, evidenciando a aleatoriedade linear do vetor de resíduos. O maior pico ocorre no intervalo zero, quando a variável corresponde exatamente a si mesmo.

Figura 62 – Análise de Resíduos para o Modelo WNN do Tanque de Nível 1: (a) Histograma dos Resíduos; (b) Autocorrelação dos resíduos; (c) Correlação cruzada linear entre os sinais de resíduos e entrada; (d) Correlação não linear entre os sinais de resíduos e entrada.

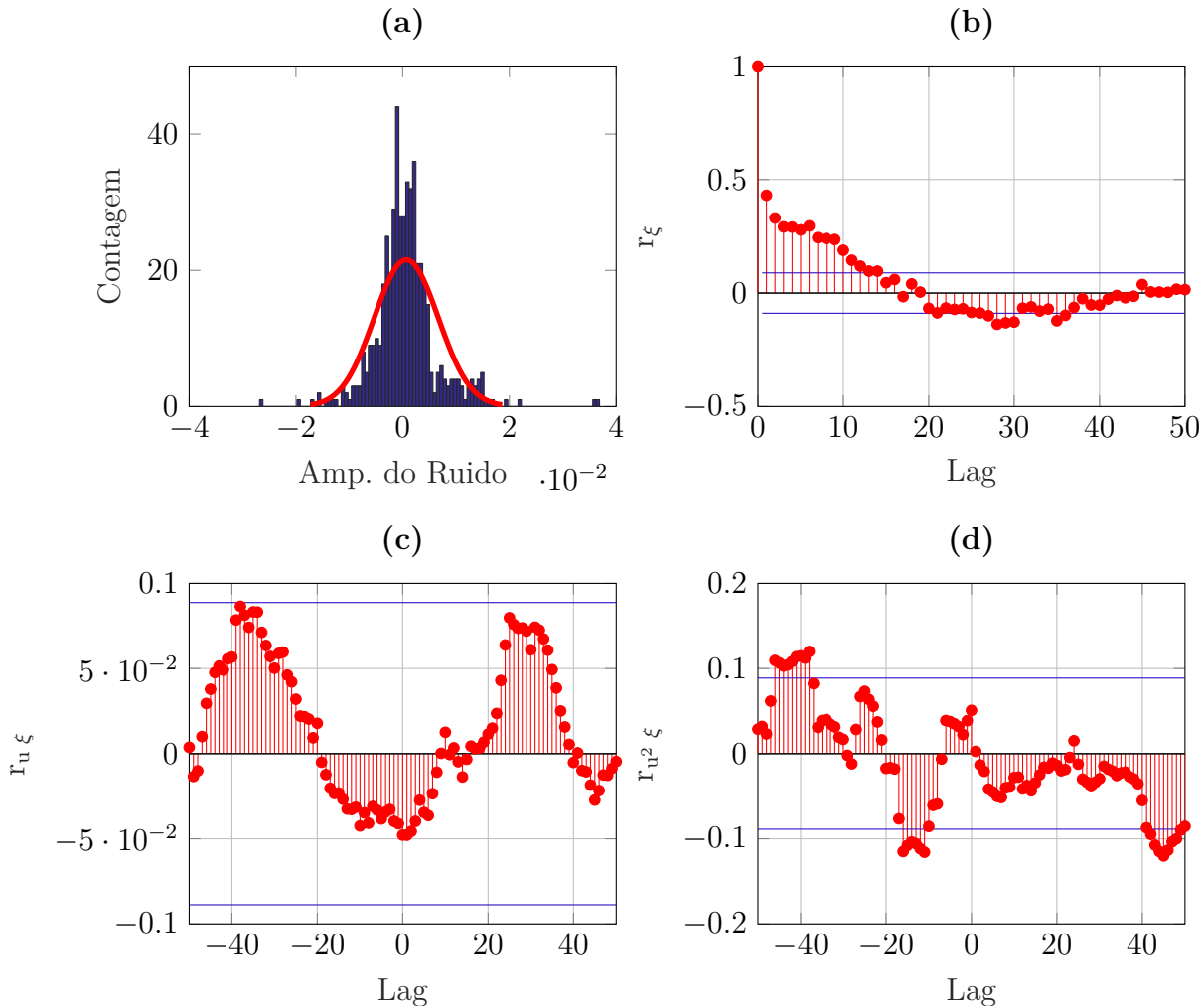


Fonte: Elaborada pelo autor.

Uma forma de verificar quão particular é o resultado de uma validação é calcular a função de correlação cruzada entre o vetor de resíduos e o sinal de entrada. Essa análise é apresentada na Figura 62(c) e 63(c) para os modelos dos tanques de nível 1 e 2, respectivamente. Observa-se que a correlação cruzada permaneceu dentro do limite de confiança e próximo de zero. A implicação desse resultado é que as estimações do modelo terão características semelhantes se calculadas para um outro conjunto de dados.

Os resultados supracitados eram esperados, uma vez que é concebível que um modelo não linear consiga explicar toda a informação linear dos dados (AGUIRRE, 2015). Entretanto, é de esperar que no modelo não linear do sistema de tanques de nível acoplados, haja correlações não lineares. A fim de detectar tais correlações, utilizou-se uma função

Figura 63 – Análise de Resíduos para o Modelo WNN do Tanque de Nível 2: (a) Histograma dos Resíduos; (b) Autocorrelação dos resíduos; (c) Correlação cruzada linear entre os sinais de resíduos e entrada; (d) Correlação não linear entre os sinais de resíduos e entrada.

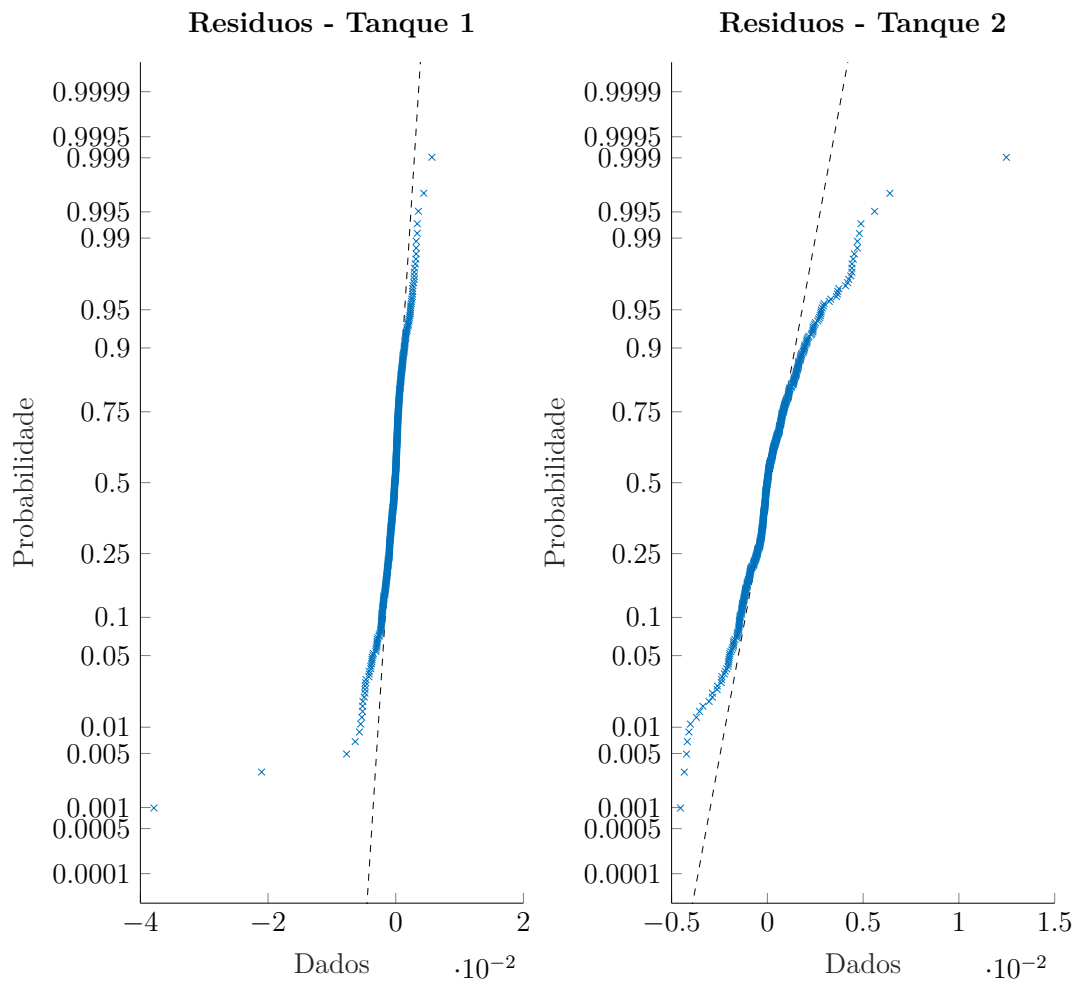


Fonte: Elaborada pelo autor.

não linear (Equação 3.16) e obteve-se os resultados apresentados na Figura 62(d) e 63(d) para os modelos dos tanques de nível 1 e 2, respectivamente. Por meio dessas figuras pode ser constatado que os resíduos possuem correlação não linear com as entradas, uma vez que os resultados estão fora dos limites de confiança (de 95%). AGUIRRE (2015) sugere que este problema pode estar relacionado com a forma com que o ruído foi tratado no modelo e não, necessariamente, que o problema tenha sido uma especificação equivocada da estrutura do modelo.

Diante dos resultados apresentados, é evidente que os resíduos não são brancos, pois existem informações neles indicando que o modelo não conseguiu estimar tudo que é explicável nos dados. No entanto, segundo AGUIRRE (2015), exigir que o modelo explique

Figura 64 – Aproximação das Amostras de Resíduos com a Função Gaussiana para o Modelo do Tanque de Nível 1 - Estudo de Caso Multivariável.



Fonte: Elaborada pelo autor.

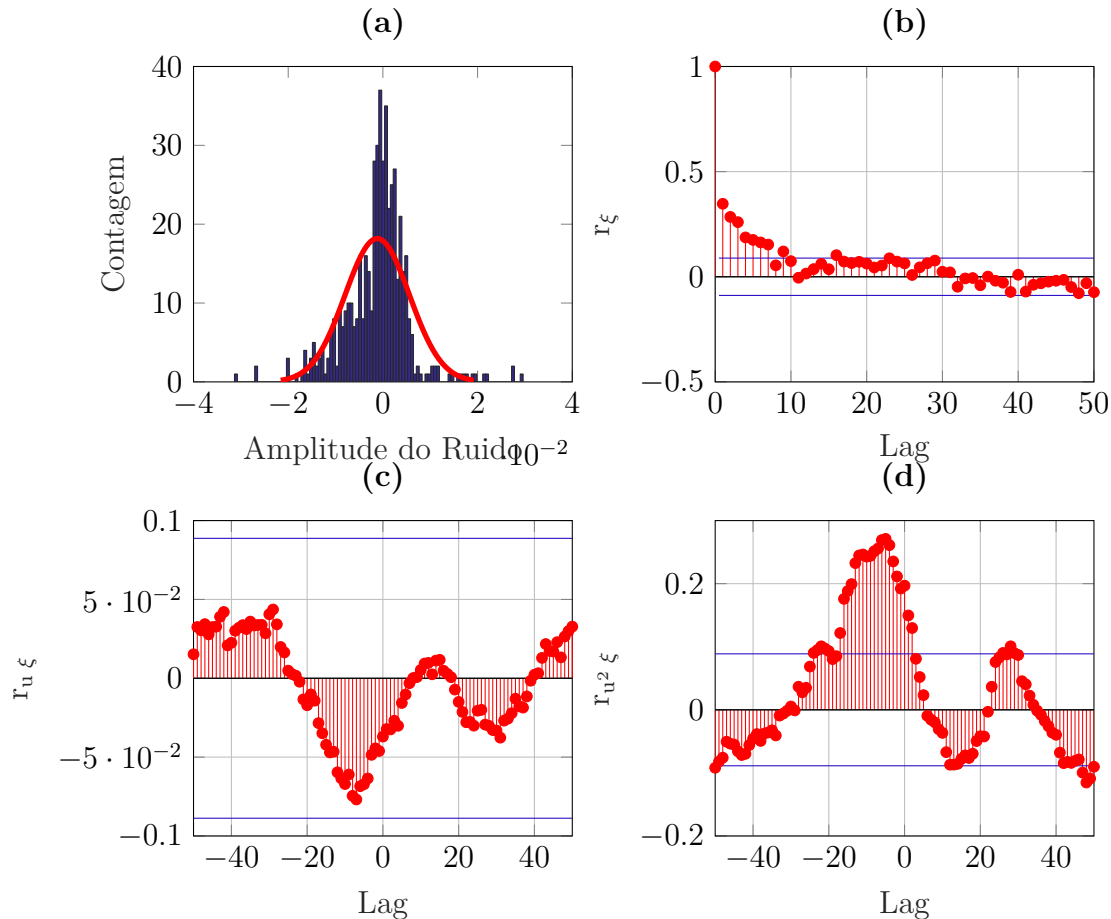
os dados em todos os seu detalhes é algo inviável.

Por fim, buscando um parâmetro de comparação para análise dessa hipótese, gerou-se gráficos para análise dos resíduos do modelo neural MLP. Também foi considerado o caso multivariável e um erro médio final de treinamento de  $5 \cdot 10^{-3}$ . A seguir, na Figura 65 é apresentado a análise de resíduos apenas o modelo do tanque de nível 1, uma vez que para o modelo do tanque de nível 2 obteve-se resultados semelhantes.

Percebe-se que a rede MLP, também, gerou resultados válidos, uma vez que o histograma dos dados dos resíduos apresenta características aleatórias e as análises de autocorrelação e correlação linear apresentam-se dentro do limite de confiança. Da mesma forma da WNN, para a correlação não linear entre resíduos e dados do sinal de entrada, algumas amostras estão correlacionadas com a entrada, indicando que ainda existem informações úteis.

Logo, pode-se afirmar que o modelo neural WNN, para os dois tanques, conseguiram

Figura 65 – Análise de Resíduos para o Modelo MLP do Tanque de Nível 1: (a) Histograma dos Resíduos; (b) Autocorrelação dos resíduos; (c) Correlação cruzada linear entre os sinais de resíduos e entrada; (d) Correlação não linear entre os sinais de resíduos e entrada.



Fonte: Elaborada pelo autor.

estimar e modelar a dinâmica do sistema e explicou parte dos dados de forma satisfatória para a aplicação deste trabalho, possuindo bons desempenhos frente aos ruídos de medição, uma vez que parte dos dados dos resíduos possuem natureza aleatória.

### 4.3.3 Hipótese 3: WNN possui melhor convergência que a MLP

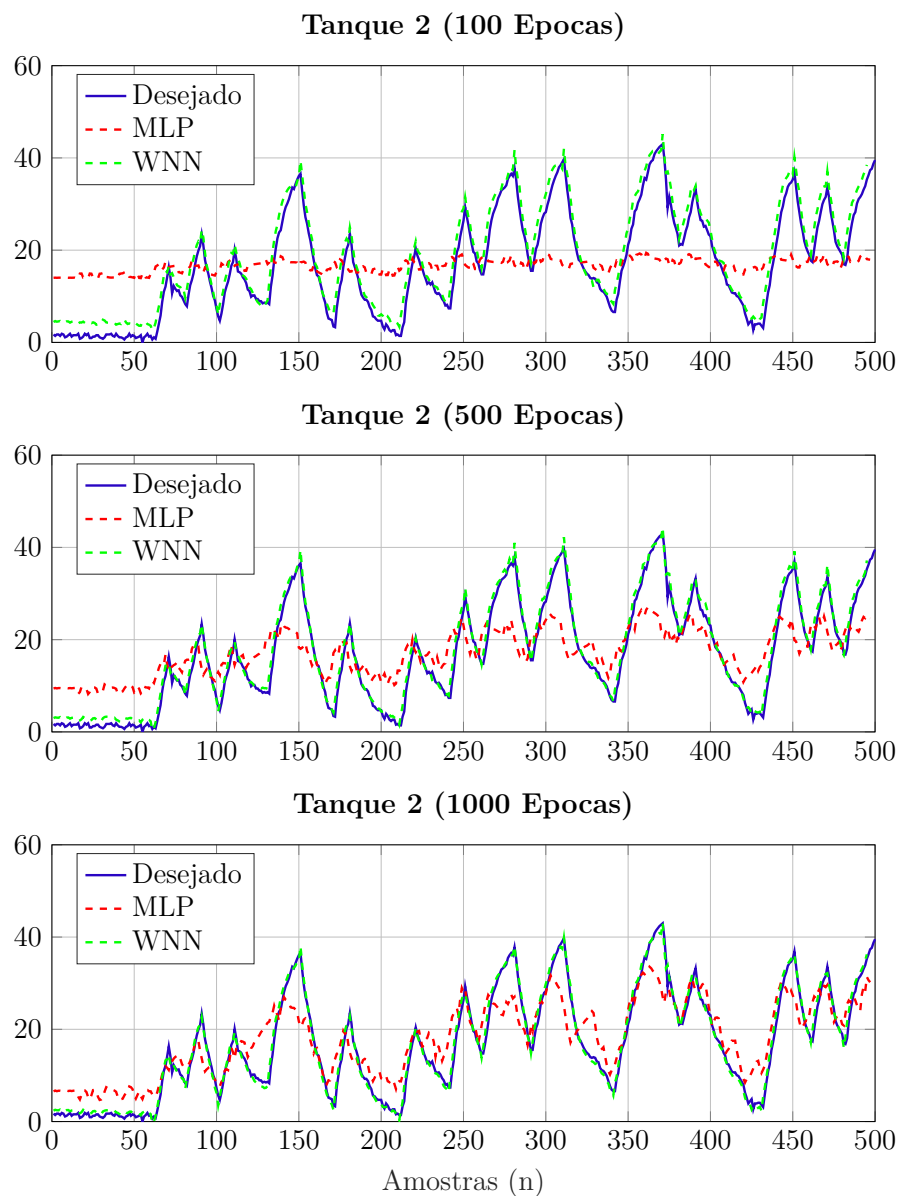
Realizando-se uma comparação entre qualquer dos resultados obtidos para WNN e MLP, seja para o estudo de caso desacoplado ou para o estudo de caso acoplado, em todas as situações o modelo estimado com auxílio da rede *Wavelet Neural Network* necessitou de menores quantidades de épocas para atingir bons resultados.

Isso se deve principalmente às propriedades das *wavelets* combinadas com as propriedades das redes neurais, pois as funções *wavelets* permitem a análise de sinais em diferentes frequências e resoluções, resultando em uma melhor adaptação temporal



e possibilitando que sejam vistos tanto características globais quanto os detalhes locais de um sinal. Aliado a isso, tem-se as características de adaptação, robustez, tolerância a falhas e generalização das redes neurais. Assim, as *Wavelets Neural Network* oferecem taxas de convergência e desempenho melhores do que as tradicionais redes neurais (JAIN; SINGH; SRIVASTAVA, 2010). Para evidenciar a justificativa supracitada, gerou-se um gráfico em que mostra a convergência dos dois modelos para 100, 500 e 1000 épocas.

Figura 66 – Convergência dos Modelos neurais WNN e MLP para 100, 500 e 1000 épocas.



Fonte: Elaborada pelo autor.

Conforme observa-se na Figura 66, a rede WNN converge de forma bem mais rápida e em menos épocas, obtendo-se resultados satisfatórios com apenas 100 épocas, enquanto o modelo neural MLP apresenta erros acentuados para essa mesma quantidade de épocas.

Com isso, pode-se deduzir que a *Wavelet Neural Network* necessitou de menor quantidade de épocas de treinamento para alcançar bons resultados de aproximação de funções do que a rede *MultiLayer Perceptron*, possuindo maior exatidão para o mesmo número de épocas.

## 5 Conclusão

Neste trabalho foi apresentado um estudo para identificação multivariável de um sistema de tanques acoplados utilizando redes neurais artificiais. Mais precisamente, utilizou-se a estrutura neural de identificação regressiva NNARX com a *Wavelet Neural Network*, com a finalidade de verificar esta rede neural como alternativa para identificação de sistemas multivariáveis.

Com base nos resultados apresentados no Capítulo 4, nota-se que a rede neural artificial *wavelet* é eficiente na identificação de sistemas de tanques de nível não lineares com várias entradas e saídas. Além disso, a WNN foi capaz de prever o comportamento do sistema com um passo à frente sem erros significativos. No entanto, estes resultados são dependentes do projeto da WNN, ou seja, número de *wavelons*, método de treinamento usado, taxa de aprendizagem, inicialização dos parâmetros adaptativos *wavelet*, quantidade de épocas e dos dados que são utilizados nas fases de estimação e validação do modelo.

O número de épocas mostrou-se um parâmetro importante para utilização das duas ferramentas para o mapeamento mais preciso da dinâmica dos tanques de nível. Em todos os casos o aumento do número de épocas contribuiu para redução do erro. No entanto, existe um limite para cada situação apresentada, em que o aumento do número de épocas deixa de surtir efeitos na redução do erro. Para as redes *wavelets* alcançou-se erros da ordem  $1 \cdot 10^{-3}$ .

Um estudo comparativo entre os estudos de casos com tanques acoplados e tanques desacoplados, mostrou que, para o mesmo número de épocas independente da RNA utilizada, a identificação dos tanques acoplados apresentou valores maiores de erro do que nos tanques desacoplados. Esse resultado era esperado, já que o processo multivariável apresenta maior complexidade para identificação por conta do acoplamento de sinais que influenciam a dinâmica do sistema.

Quanto a comparação do desempenho dos modelos neurais estimados MLP e WNN, as hipóteses validadas mostram que de fato as redes *wavelets* apresentam bom desempenho na análise de sinais não estacionários e na modelagem de funções não lineares, além de oferecem taxas de convergência e desempenho melhores do que a tradicional rede neural MLP.

Diante de todos os resultados e conclusões supracitados, pode-se afirmar que os mesmos validam a aplicabilidade das Redes Neurais *Wavelets* em sistemas nos quais ocorrem acoplamentos de sinais, com isso atingindo o objetivo deste estudo.

Para futuras pesquisas, planeja-se fazer a implementação de controladores preditivos

do mesmo sistema de tanques utilizando os modelos estimados no presente trabalho. Acredita-se que a exatidão do modelo neural WNN, com erros da estimação da ordem de  $10^{-3}$ , seja suficiente para realizar essa modalidade de controle que utiliza a saída explícita do modelo estimado para calcular a ação de controle. Além disso, pretende-se propor alguma modalidade de controle que trabalhe com esses modelos de forma *online* como, por exemplo, o controle preditivo baseado em modelos.

## Referências

ABIYEV, R. H.; KAYNAK, O. Identification and control of dynamic plants using fuzzy wavelet neural networks. *IEEE Int. Symp. Intell. Control (ISIC)*, v. 4, p. 1295—1301, 2008. Citado na página 56.

AGUIRRE, L. A. *Introdução à Identificação de Sistemas: técnicas lineares e não lineares aplicadas a sistemas reais*. 4. ed. Minas Gerais: Editora UFMG, 2015. Citado 17 vezes nas páginas 27, 28, 31, 33, 58, 60, 61, 63, 64, 69, 82, 83, 84, 105, 106, 107 e 108.

ALEXANDRIDIS, A. K.; ZAPRANIS, A. D. *Wavelet Neural Networks: With Applications in Financial Engineering, Chaos, and Classification*. New Jersey: WILEY, 2014. Citado 2 vezes nas páginas 49 e 58.

BASTIDA, H. et al. Model and Control for Coupled Tanks using Labview. *International Conference on Mechatronics, Electronics and Automotive Engineering Model*, n. 1, p. 127–133, 2013. Citado na página 30.

BECERIKLI, Y.; OYSAL, Y.; KONAR, A. F. On three intelligent systems: dynamic neural, fuzzy and wavelet networks for training trajectory. *Neural Computation and Applications*, p. 339–351, 2004. Citado na página 56.

BHANDARE, D. S.; KULKARNI, N. Performances evaluation and comparison of pid controller and fuzzy logic controller for process liquid level control. In: IEEE. *15th International Conference on Control, Automation and Systems (ICCAS)*. [S.l.], 2015. p. 1347–1352. Citado 2 vezes nas páginas 29 e 30.

BILLINGS, S. A.; VOON, W. S. F. Correlation based model validity tests for nonlinear models. *Int. J. Control*, p. 235–244, 1986. Citado na página 84.

BILLINGS, S. A.; WEI, H. A new class of wavelet networks for nonlinear system identification. *IEEE Trans. Neural Networks*, v. 16, n. 4, p. 862–874, 2005. Citado na página 32.

BILLINGS, S. A.; WEI, H. L. On a dynamic wavelet network and its modeling application. *IEEE Transactions on Neural Networks*, p. 862–874, 2005. Citado na página 56.

CARVALHO, V. P. de. *Previsão de séries temporais no mercado financeiro de ações com o uso de rede neural artificial*. Dissertação (Mestrado) — Universidade Presbiteriana Mackenzie, 2018. Citado na página 37.

CHENG, Z. et al. Modeling of gyro-stabilized platform based on narx neural network. In: IEEE. *10th International Symposium on Computational Intelligence and Design Modeling*. [S.l.], 2017. p. 284–288. Citado na página 32.

COELHO, A. A. R.; COELHO, L. dos S. *Identificação de Sistemas Dinâmicos Lineares*. 2. ed. Florianópolis: UFSC, 2016. Citado na página 28.

COELHO, L. dos S. *Identificação e controle de processos multivariáveis via metodologias*. Florianópolis: UFSC, 2000. Citado na página 62.

- DAUGMAN, J. G. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression, *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 36, n. 7, p. 1169–1179, 1988. Citado 2 vezes nas páginas 29 e 32.
- DOMINGUES, M. O. et al. Explorando a transformada wavelet continua. v. 38, 2016. Citado na página 49.
- FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. *Sistemas de Controle para Engenharia*. 6<sup>a</sup>. ed. Porto Alegre: Bookman, 2013. Citado na página 27.
- FUGAL, D. L. *Conceptual Wavelets: In Digital Signal Processing*. 1<sup>a</sup>. ed. Utah: Elsevier, 2009. Citado 5 vezes nas páginas 46, 49, 50, 51 e 52.
- GAUTAM, P. System identification of nonlinear Inverted Pendulum using artificial neural network. *2016 International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2016*, IEEE, p. 1–5, 2017. ISSN 5090-2806. Citado na página 31.
- GIL, A. C. *Como Elaborar Projetos de Pesquisa*. 6. ed. Porto Alegre: Atlas, 2017. Citado na página 65.
- HAYKIN, S. *Redes Neurais: Princípios e Prática*. 2. ed. Porto Alegre: Bookman, 2003. Citado 4 vezes nas páginas 38, 40, 41 e 42.
- HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, n. 5, p. 359–366, 1989. Citado 2 vezes nas páginas 29 e 34.
- JAIN, T.; SINGH, S. N.; SRIVASTAVA, S. C. Adaptive wavelet neural network-based fast dynamic available transfer capability determination. *ET Generation, Transmission & Distribution*, v. 4, n. 4, p. 519–529, 2010. Citado 2 vezes nas páginas 53 e 111.
- JUNIOR, J. M. de A. *Técnicas de Inteligência Artificial para o Controle Inferencial de Colunas de Destilação*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte (UFRN), 2007. Citado na página 62.
- JUNIOR, J. M. de A. *Identificação Não Linear Usando uma Rede Fuzzy Wavelet Neural Network Modificada*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Natal, Março 2014. Citado 5 vezes nas páginas 28, 32, 48, 54 e 57.
- JUNIOR, L.; COSTA, O. E.; MONTGOMERY, E. E. *Redes Neurais: Fundamentos e Aplicações com Programas em C*. Rio de Janeiro: Ciência Moderna, 2007. Citado na página 40.
- KEESMAN, K. J. *System Identification: An Introduction*. 2<sup>a</sup>. ed. London: Springer, 2011. Citado na página 27.
- KHAN, J. et al. Classification and diagnostic prediction of cancers using gene expression proliferating and artificial neural networks. *Nature Medicine*, v. 7, p. 673–679, 2001. Citado na página 37.
- KHODABANDEHLOU, H.; FADALI, M. S. Networked Model Predictive Control Using a Wavelet Neural Network. p. 1–8, 2016. Citado na página 32.

- KREINOVICH, V.; SIRISAENGTAKSIN, O.; CABRERA, S. Wavelet neural networks are asymptotically optimal approximators for functions of one variable. *IEEE International Conference on Neural Networks*, v. 1, p. 299–304, 1994. Citado na página 34.
- LANDAU, I. D. *Digital Control Systems: Design, Identification and Implementation*. London: Springer, 2006. Citado 2 vezes nas páginas 29 e 61.
- LATHI, B. P. *Sinais e Sistemas Lineares*. Porto Alegre: Bookman, 2007. Citado na página 59.
- LEE, K. H. et al. System identification of hydraulic system using RLS algorithm for feedforward position control. *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2015*, IEEE, v. 1, n. Urai, p. 63–65, 2015. ISSN 0029-8115, 0029-8115. Citado na página 31.
- LI, C.; MINMING, T.; HAIBO, D. Predicting water irruption quantity from coal floor based on wavelet neural network. *International Forum on Information Technology and Applications*, v. 3, p. 154–157, 2010. Citado na página 34.
- LJUNG, L. *System Identification: Theory for the user*. New Jersey: Prentice Hall, 1987. Citado na página 60.
- LUCAS, M. *Sensor inteligente para estimação de composição em colunas de destilação*. São Carlos: [s.n.], 2012. Citado 2 vezes nas páginas 63 e 73.
- LUZETTI, D. G. *Sistema de Controle para Nível de Reservatório*. Itatiba: [s.n.], 2009. Citado na página 67.
- MALLAT, S. *A Wavelet Tour of Signal Processing: The Sparse Way*. 1ª. ed. Burlington: Space and Signals Technologies LLC, 2009. Citado na página 49.
- MARTINS, J. K. E. da C.; SOUZA, F. R. de L.; ARAUJO, F. M. U. de. Identificação de um sistema de tanques acoplados não linear com atraso utilizando rna. In: INSTITUTO BRASILEIRO DE PETROLEO, GAS E BIOCMBUSTIVEIS (IBP). *Congresso Rio Automação 2015*. [S.l.], 2015. Citado na página 31.
- MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943. Citado na página 37.
- MICROCHIP. *PIC18F2455/2550/4455/4550 Data Sheet, 28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology*. [S.l.], 2006. Disponível em: <<https://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>>. Citado na página 67.
- MINU, K. K.; LINEESH, M. C.; JOHN, C. J. Wavelet neural networks for nonlinear time series analysis. *Applied Mathematical Sciences*, v. 4, n. 50, p. 2485–2495, 2010. Citado na página 53.
- MOHARERI, O. *Mobile Robot Trajectory Tracking Using Neural Networks*. Dissertação (Mestrado) — American University of Sharjah, 2009. Citado na página 37.

MONESS, M.; DIAA-ELDEEN, T. Experimental nonlinear identification of a lab-scale helicopter system using MLP neural network. *ICENCO 2017 - 13th International Computer Engineering Conference: Boundless Smart Societies*, v. 2018-Janua, p. 166–171, 2018. Citado na página 31.

NAUNG, Y. et al. Implementation of data driven control system of DC motor by using system identification process. *Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2018*, IEEE, v. 2018-Janua, p. 1801–1804, 2018. Citado na página 31.

NORGAARD, M. et al. *Neural Networks for Modeling and Control of Dynamic Systems*. Londres: Springer-Verlag London Limited, 2001. Citado na página 37.

OWA, K.; SHARMA, S.; SUTTON, R. A wavelet neural network based non-linear model predictive controller for a multi-variable coupled tank system. *International Journal of Automation and Computing*, v. 12, n. 2, p. 156–170, 2015. ISSN 17518520. Citado 2 vezes nas páginas 32 e 73.

PAUL, R.; SENGUPTA, A.; RANJAN, R. Measurement Wavelet based denoising technique for liquid level system. *Measurement*, Elsevier Ltd, v. 46, n. 6, p. 1979–1994, 2013. ISSN 0263-2241. Disponível em: <<http://dx.doi.org/10.1016/j.measurement.2013.02.004>>. Citado na página 30.

RADAC, M.-b.; PRECUP, R.-e.; ROMAN, R.-c. Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-Learning. *ISA Transactions*, Elsevier Ltd, v. 73, p. 227–238, 2018. ISSN 0019-0578. Disponível em: <<https://doi.org/10.1016/j.isatra.2018.01.014>>. Citado na página 30.

RAHMAT, M. F.; ROZALI, S. M. Modeling and controller design for a coupled- tank liquid level system: analysis & comparison. v. 48, n. D, p. 113–141, 2008. Citado na página 31.

ROCHA, F. L. *Identificação de sistemas não-lineares multivariáveis usando redes neurais perceptron multicamadas e função de base radial*. Curitiba: [s.n.], 2006. Citado na página 59.

ROCHA, V. B. da. *Um abordagem de Wavelets Aplicada à Combinação de Previsões: Uma análise teórica e experimental*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2008. Citado 2 vezes nas páginas 46 e 48.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. In: RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. (Ed.). Cambridge, MA, USA: MIT Press, 1986. cap. Learning Internal Representations by Error Propagation, p. 318–362. ISBN 0-262-68053-X. Disponível em: <<http://dl.acm.org/citation.cfm?id=104279.104293>>. Citado na página 37.

SEMICONDUCTOR, F. *Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated*. [S.l.], 2012. Disponível em: <<https://www.nxp.com/docs/en/data-sheet/MPX5010.pdf>>. Citado na página 67.



- SILVA, H. H.; JUNIOR, J. M. de A. Identificação multivariável de um sistema de tanques de nível acoplados utilizando redes neurais artificiais. In: INDUSCON. *Universidade de São Paulo*. [S.l.], 2018. Citado na página 73.
- SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. A. *Redes Neurais Artificiais para engenharia e ciências aplicadas*. São Paulo: Artliber, 2010. Citado 7 vezes nas páginas 29, 37, 38, 41, 42, 45 e 74.
- SOUSA, D. L. *Controle PID Multivariável de um Tanque Duplo com Acoplamento entre as Malhas*. Teresina: [s.n.], 2014. Monografia (Bacharelado em Engenharia Elétrica), Universidade Federal do Piauí, Teresina. Citado 3 vezes nas páginas 65, 67 e 68.
- SZU, H. H. et al. Neural network adaptive wavelets for signal representation and classification. *Optical Engineering*, v. 31, p. 1907–1916, 1992. Citado na página 32.
- VEITCH, D. *Wavelet Neural Networks and Their Application in the Study of Dynamical Systems*. Dissertação (Mestrado) — University of York, Agosto 2005. Citado 6 vezes nas páginas 46, 47, 54, 55, 57 e 81.
- YUN, K.; HUYEN, A.; LU, T. Deep neural networks for pattern recognition. *CoRR*, abs/1809.09645, 2018. Disponível em: <<http://arxiv.org/abs/1809.09645>>. Citado na página 37.
- ZHANG, Q. *Using Wavelet Network in Nonparametric Estimation*. [S.l.]: INRIA, 1994. Citado na página 56.
- ZHANG, Q.; BENVENISTE, A. Wavelet networks. *IEEE Trans. Neural Networks*, v. 3, n. 6, p. 889–898, 1992. Citado na página 32.



# Apêndices



# APÊNDICE A – Script para Coleta de Dados

Código A.1 – Geração dos Sinais de Excitação e Coleta de Dados

```
1 % Sinal PRBS
2 % Tanque de Nivel (Heinrich Hertz Silva)
3
4 % Inicializacao para Coleta de Dados
5
6 clc;
7 pause(5);
8
9 % set_pwm_duty(Tanque 1 ou 2, duty cycle, frequencia do PWM)
10
11 set_pwm_duty(1,1,3000); % aciona a bomba hidraulica 1 de
    acordo com o PWM
12 set_pwm_duty(2,1,3000); % aciona a bomba hidraulica 2 de
    acordo com o PWM
13
14 pause(5);
15 recebe(1); % Realiza leitura do sinal de nivel do tanque 1
16 recebe(2); % Realiza leitura do sinal de nivel do tanque 2
17
18 PRBS_T1 = idinput(2555, 'prbs', [0 0.2], [0 1]); %Gera Sinal
    PRBS do Tanque 1
19 PRBS_T1=PRBS_T1';
20 N=length(PRBS_T1);
21
22 PRBS_T2 = idinput(2555, 'prbs', [0 0.2], [0 1]); %Gera Sinal
    PRBS do Tanque 2
23 PRBS_T2=PRBS_T2';
24 N=length(PRBS_T2);
25
26 %Coleta de Dados
27
28 for j=1:N
```

```
29     set_pwm_duty(1,(1-PRBS_T1(j)),3000);
30     nivel_T1(j)= recebe(1);
31
32     % Saturador para evitar Transbordamento do Tanque 1
33     if nivel_T1(j)>=2.0
34         if (N-j)<18
35             PRBS_T1(j:N-j)=0;
36         else
37             PRBS_T1(j:j+18)=0;
38         end
39     end
40     set_pwm_duty(2,(1-PRBS_T2(j)),3000);
41     nivel_T2(j)= recebe(2);
42     pause(1);
43
44     % Saturador para evitar Transbordamento do Tanque 2
45     if nivel_T2(j)>=2.1
46         if (N-j)<18
47             PRBS_T2(j:N-j)=0;
48         else
49             PRBS_T2(j:j+18)=0;
50         end
51     end
52 end
53
54 % Desliga Bombas
55 set_pwm_duty(2,1,3000);
56 set_pwm_duty(1,1,3000);
57
58 % Graficos dos Dados Coletados
59
60 figure(1);
61 plot(PRBS_T1,'b','LineWidth',1);
62 xlabel('Amostras (n)','FontSize',13);
63 ylabel('Amplitude do Sinal','FontSize',13);
64 title('Sinal PRBS aplicado ao Tanque 01');
65 axis([0 2550 -0.05 1.05]);
66 grid on;
67
```

```
68 figure(2);
69 plot(PRBS_T2,'b','LineWidth',1);
70 xlabel('Amostras (n)','FontSize',13);
71 ylabel('Amplitude do Sinal','FontSize',13);
72 title('Sinal PRBS aplicado ao Tanque 02');
73 axis([0 2550 -0.05 1.05]);
74 grid on;
75
76 figure(3);
77 plot(V2HT1(nivel_T1),'b','LineWidth',1);
78 xlabel('Amostras (n)','FontSize',13);
79 ylabel('Nivel (cm)','FontSize',13);
80 title('Nivel do Tanque 01');
81 axis([0 2550 0 70]);
82 grid on;
83 hold on;
84 plot(30*ones(N),'k');
85
86 figure(4);
87 plot(V2HT2(nivel_T2),'r','LineWidth',1);
88 title('Nivel do Tanque 02');
89 xlabel('Amostras (n)','FontSize',13);
90 ylabel('Nivel (cm)','FontSize',13);
91 axis([0 2550 0 70]);
92 grid on;
93 hold on;
94 plot(30*ones(N),'k');
```





# APÊNDICE B – Script Principal

## Código B.1 – Obtenção dos Modelos Neurais

```

1 % Universidade Federal do Piaui - UFPI
2 % Programa de Pos Graduacao em Eng Eletrica - PPGEE
3 % Script Produto da Dissertacao: Identificacao Multivariavel
4 % de Um Sistema de Tanques de Nivel Acoplados Utilizando WNN
5 % Mestrando: Heinrich Hertz Silva
6
7 % Apaga Variaveis e Carrega novos Dados
8 clear all;close all;clc;
9 load('nivel_T1');
10 Qt_Amostras = length(nivel_T1);
11
12 % Divisao das Amostras Treino e Validacao
13
14 % N: NUMERO TOTAL DE AMOSTRAS DE SINAL
15 N_distribution=[4/5 1/5]; % Divisao das Amostras em Treino e
    Validacao
16 N_train=floor(N_distribution(1)*Qt_Amostras); % Qt de
    amostras p/ treino
17 N_test=floor(N_distribution(2)*Qt_Amostras); % Qt de amostras
    p/ teste
18
19
20 % Definicao das Variaveis para o Tanque 01
21
22 % Dados de Treino para o Tanque de Nivel 1
23 PRBS_T1(1+length(PRBS_T1)-(length(PRBS_T1)-length(PRBS_T2
    )):length(PRBS_T1))=[];
24 X1 = PRBS_T1(1:N_train); % Sinal PRBS - Tanque de Nivel 1
25 D1 = nivel_T1(1:N_train); % Sinal de Nivel - Tanque de
    Nivel 1
26
27 % Dados de Teste para o Tanque de Nivel 1
28 P1 = PRBS_T1(1:N_test); % Sinal PRBS para Teste - Tanque
    de Nivel 1

```

```
29     L1 = nivel_T1(1:N_test); % Sinal de Nivel para Teste -
        Tanque de Nivel 1
30
31 % Definicao das Variaveis para o Tanque 02
32
33     % Dados de Treino para o Tanque de Nivel 2
34     X2 = PRBS_T2(1:N_train); % Sinal PRBS para Treino -
        Tanque de Nivel 2
35     D2 = nivel_T2(1:N_train); % Sinal PRBS para Treino -
        Tanque de Nivel 2
36
37     % Dados de Teste para o Tanque de Nivel 2
38     P2 = PRBS_T2(1:N_test); % Sinal PRBS para Teste - Tanque
        de Nivel 2
39     L2 = nivel_T2(1:N_test); % Sinal PRBS para Teste - Tanque
        de Nivel 2
40
41 % Pre-Processamento dos Dados
42
43     % Normalizacao dos Dados
44     % Dados de Treino
45     D1=mapminmax(D1,0,1);
46     D2=mapminmax(D2,0,1);
47     % Dados de Teste
48     L1=mapminmax(L1,0,1);
49     L2=mapminmax(L2,0,1);
50
51 % Ordem NNARX
52     % Nx -> Ordem do preditor
53     % Ny -> Ordem de contexto
54     % AT -> Atraso de Transporte
55
56     Nx = 2;
57     Ny = 2;
58     AT = 1;
59
60 % Gerar Matriz de Entrada para Treino por meio das Funcoes
    Abaixo
61     % Tanque 1
```

```
62     Dados_train_T1 = [X1' D1'];
63     [X_train_T1] = gera_padroes_Ordens_modificado(
        Dados_train_T1,Nx,Ny,AT);
64     X_train_T1=X_train_T1';
65
66     % Tanque 2
67     Dados_train_T2 = [X2' D2'];
68     [X_train_T2] = gera_padroes_Ordens_modificado(
        Dados_train_T2,Nx,Ny,AT);
69     X_train_T2=X_train_T2';
70
71     % Matriz de dados para Treino
72     X_train=[X_train_T1;X_train_T2];
73
74
75 % Gerar Matriz de Entrada para Teste
76     % Tanque 1
77     Dados_test_T1 = [P1' L1'];
78     [X_test_T1] = gera_padroes_Ordens_modificado(
        Dados_test_T1,Nx,Ny,AT);
79     X_test_T1=X_test_T1';
80
81     % Tanque 2
82     Dados_test_T2 = [P2' L2'];
83     [X_test_T2] = gera_padroes_Ordens_modificado(
        Dados_test_T2,Nx,Ny,AT);
84     X_test_T2=X_test_T2';
85
86     % Matrix de dados para Teste
87     X_test=[X_test_T1;X_test_T2];
88
89 % Gerar Matriz de dados desejados
90     D=[D1;D2];
91     L=[L1;L2];
92 % Neuronios por Camada
93     p = 2*(Nx+Ny); % Quantidade de Neuronios da Entrada da
        Rede
94     H = p; % Quantidade de Neuronios da Camada Oculta (
        Wavelons)
```

```
95     m = 2; % Quantidade de Neuronios da Camada de Saida
96
97 % Parametros da Rede Neural
98     mu = 0.05; % taxa de aprendizagem
99     epoch = 5000; % Quantidade de Epocas
100    MSEmin = 0.0009; % Erro Minimo (Critério de Parada)
101    bias = rand(1,m); % Vetor Bias
102
103 % Treino
104    [Wy,MSE,t,lambda,BIAS,E,W,Tt,Llambda,Bbias]=
        treino_WNN_MSE(p,H,m,mu,X_train,D,epoch,MSEmin,bias);
        % Funcao para treino usando WNN
105    [Wx,Wy,MSE_PLOT,E]=trainNNARX_MSE(p,H,m,mu,alpha,X,D,
        epochMax,MSETarget); % Funcao para treino usando MLP
106
107    % Observacao: Comentar uma das linhas das funcoes de
        treino com "%".
108
109    %Grafico do MSE
110    figure(1);
111    semilogy(MSE,'r-','LineWidth',1);
112    xlabel('Epocas','FontSize',13);
113    ylabel('EQM','FontSize',13);
114    title('EQMs (Tanques 01 e 02)');
115    grid on;
116
117 % Validacao
118
119    Y = teste_WNN(X_test,Wy,t,lambda,H,m,BIAS); % Funcao para
        Validacao para WNN
120
121    Y = runNNARX(X_test,Wx,Wy); % Funcao para Validacao para
        MLP
122
123    % Observacao: Comentar uma das linhas das funcoes de
        teste com "%".
124
125 % Resultados
126    % Plotar Validacao - Tanque de Nivel 1
```

```
127     figure(2);
128     plot(V2HT1(L1max*L(1,:)), '-','LineWidth',1); % Funcao
        V2HT1 converte sinal de tensao em nivel(cm)
129     hold on;
130     plot(V2HT1(L1max*Y(1,:)), '--','LineWidth',1);
131     xlabel('Amostras (n)','FontSize',13);
132     ylabel('Niveis dos Tanques (cm)','FontSize',13);
133     title('Teste do Modelo do Tanque 01');
134     legend('Valores Desejados','Valores Estimados','Location'
        , 'southeast');
135     grid on;
136
137     % Plotar Validacao - Tanque de Nivel 2
138     figure(3);
139     plot(V2HT2(L2max*L(2,:)), '-','LineWidth',1); % Funcao
        V2HT2 converte sinal de tensao em nivel(cm)
140     hold on;
141     plot(V2HT2(L2max*Y(2,:)), '--','LineWidth',1);
142     xlabel('Amostras (n)','FontSize',13);
143     ylabel('Niveis dos Tanques (cm)','FontSize',13);
144     title('Teste do Modelo do Tanque 02');
145     legend('Valores Desejados','Valores Estimados','Location'
        , 'northwest');
146     grid on;
```



# APÊNDICE C – Funções Utilizadas no Script

Código C.1 – Função para Gerar Vetores Regressores da Estrutura NNARX

```

1 function [D]=gera_padroes_Ordens_modificado(dados,Nx,Ny,AT)
2     % D: Matriz de dados
3     % Cada linha eh um vetor de estados de dimensao Ny
4
5     [LIN COL]=size(dados);
6     NN = abs(Nx-Ny);
7     XX=[]; YY=[];
8     for i=Nx:LIN-NN-AT,
9         X = [dados(AT+i-Nx+1:i+AT,1)];
10        XX = [XX; fliplr(X')];
11    end
12
13    for i=Ny:LIN,
14        Y = [dados(i-Ny+1:i,2)];
15        YY = [YY; fliplr(Y')];
16    end
17
18    [a b]=size(XX);
19    [c d]=size(YY);
20    S=abs(a-c);
21
22    for k=1:S
23        YY(a-k,:)=[];
24    end
25
26    D=[XX YY];

```

Código C.2 – Função para Treinamento da Rede Neural MLP

```

1 function [Wx,Wy,MSE_PLOT,E]=trainNNARX_MSE(p,H,m,mu,alpha,X,D
2     ,epochMax,MSETarget)
3 [p1 N] = size(X);

```

```
4 bias = -1;
5 MSE_PLOT=[];
6 X = [bias*ones(1,N) ; X];
7
8 % Inicializa Variaveis
9 Wx = rand(H,p+1);
10 WxAnt = zeros(H,p+1);
11 Tx = zeros(H,p+1);
12 Wy = rand(m,H+1);
13 Ty = zeros(m,H+1);
14 WyAnt = zeros(m,H+1);
15 DWy = zeros(m,H+1);
16 DWx = zeros(H,p+1);
17 MSETemp = zeros(2,epochMax);
18 Wxx=[];
19 Wyy=[];
20 mse_epoc=inf;
21 a=1;
22 p=1;
23
24 % Etapa de Propagacao
25 while (mse_epoc > MSETarget)&&(a~=epochMax)
26
27 k = randperm(N);
28 X = X(:,k);
29 D = D(:,k);
30
31 V = Wx*X;
32 Z = 1./(1+exp(-V));
33
34 S = [bias*ones(1,N);Z];
35 G = Wy*S;
36
37 Y = p*G;
38
39 E = D - Y;
40
41 mse_S1 = sqrt(mean(E(1,:).^2));
42 mse_S2 = sqrt(mean(E(2,:).^2));
```



```

43 MSETemp(1,a) = mse_S1;
44 MSETemp(2,a) = mse_S2;
45 mse_epoc=mean([mse_S1 mse_S2]);
46 MSE_PLOT=[MSE_PLOT mse_epoc];
47
48 % Etapa de Retropropagacao
49
50 df = p*(Y./Y);
51 dGy = df .* E;
52
53 DWy = mu/N * dGy*S';
54 Ty = Wy;
55 Wy = Wy + DWy + alpha*WyAnt;
56 WyAnt = Ty;
57
58 df= S.*(1-S);
59
60 dGx = df .* (Wy' * dGy);
61 dGx = dGx(2:end,:);
62 DWx = mu/N* dGx*X';
63 Tx = Wx;
64 Wx = Wx + DWx + alpha*WxAnt;
65 WxAnt = Tx;
66
67 a=a+1;
68 end
69
70 MSE = MSETemp;
71 end

```

### Código C.3 – Função para Treinamento da Rede Neural WNN

```

1 function [Wy,MSE_PLOT,t,lambda,bias,E,W,Tt,Lambda,Bbias]=
   treino_WNN_MSE(p,H,m,mu,X,D,epochMax,MSETarget,bias)
2 [p1 N] = size(X);
3 Wy = rand(m,H); %Vetor de Pesos entre a camada de oculta e
   saida
4 MSETemp = zeros(2,epochMax);
5 MSE_PLOT=[];
6 % lambda=0.2*ones(H,1); % dilacao(escala)

```

```

7 % t=0.5*ones(H,p); %vetor de translacao
8 [t,lambda]=inicia_param(0,1,p,H);
9 GAMA=zeros(H,N);
10 E=zeros(m,N);
11 W=[];
12 Tt=[];
13 Llambda=[];
14 Bbias=[];
15 mse_epoc=inf;
16 a=1;
17
18 while (mse_epoc > MSETarget)
19 %while (mse_epoc > MSETarget)&&(a~=epochMax)
20 % Etapa de Propagacao
21     % Permutacao da ordem dos dados de entrada
22     k = randperm(N);
23     X = X(:,k);
24     D = D(:,k);
25     % Calculo de Gama
26     for b=1:1:N
27         Y_est=teste_WNN(X(:,b),Wy,t,lambda,H,m,bias);
28         desj=D(:,b);
29         %Inicializando Parametros Adaptativos
30         Wy_new=zeros(m,H);
31         t_new=zeros(H,p);
32         lambda_new=zeros(H,1);
33         c=1;
34
35 % Etapa de Retropropagacao do Erro
36     for i=1:1:H
37         Wy_new(:,i)=Wy_new(:,i)*0;
38         t_new(i,:)=t_new(i,:)*0;
39
40         gama=0;
41         for j=1:1:p1
42             gama=gama+((X(j,b)-t(i,j))/lambda(i,1))^2;
43         end
44         gama=gama^(1/2);
45

```

```
46     back=0;
47     for k=1:1:m
48         E(k,b)=Y_est(k)-desj(k);
49
50         if c==1
51             if isnan(Y_est(k))
52                 return;
53             end
54             bias(1,k)=bias(1,k)-mu*E(k,b);
55             %MSETemp(a)=MSETemp(a)+E(k,b)^2;
56         end
57         back=back+E(k,b)*Wy(k,i);
58         %Ajuste do Parametro Adaptativo Wy
59         Wy_new(k,i)=Wy(k,i)- mu*E(k,b)*MexWav(gama);
60     end
61     c=0;
62
63     %Ajuste do Parametro Adaptativo lambda (dilatacao
64     ou escala)
65     lambda_new(i,1)=lambda(i,1)+mu*back*dMexWav(gama)
66     /(lambda(i,1)^3*gama);
67
68     %Ajuste do Parametro Adaptativo t (translacao)
69     for j=1:1:p
70         t_new(i,j)=t(i,j)+mu*back*dMexWav(gama)*(X(j,
71         b)-t(i,j))/(gama*(lambda(i,1)^2));
72     end
73
74     end
75
76     for ii=1:1:H
77         for kk=1:1:m
78             Wy(kk,ii)=Wy_new(kk,ii);
79         end
80
81         for jj=1:1:p
82             t(ii,jj)=t_new(ii,jj);
83         end
84     end
```

```

82         if abs(lambda_new(ii,1))<0.0002;
83
84             if lambda_new(ii,1) >= 0
85                 lambda(ii,1)=0.0002;
86             else
87                 lambda(ii,1)=-0.002;
88             end
89         else
90             lambda(ii,1)=lambda_new(ii,1);
91         end
92     end
93
94     end
95     mse_S1 = sqrt(mean(E(1,:).^2));
96     mse_S2 = sqrt(mean(E(2,:).^2));
97     mse_epoc=mean([mse_S1 mse_S2]);
98     MSE_PLOT=[MSE_PLOT mse_epoc];
99     MSETemp(1,a) = mse_S1;
100    MSETemp(2,a) = mse_S2;
101    for v=100:100:epochMax
102        if v==a
103            W=[W Wy'];
104            Tt=[Tt t];
105            Llambda=[Llambda lambda];
106            Bbias=[Bbias bias'];
107        end
108    end
109    a=a+1;
110    end
111
112    end

```

Código C.4 – Função para Validação da Rede Neural MLP

```

1 function [Y,df]=runNNARX(X,Wx,Wy)
2
3 [p1 N] = size (X);
4
5 bias = -1;
6

```

```

7 X = [bias*ones(1,N) ; X];
8
9 V = Wx*X;
10 Z = 1./(1+exp(-V));
11
12 S = [bias*ones(1,N);Z];
13 df= [S.*(1-S)];
14 G = Wy*S;
15
16 Y = G;

```

#### Código C.5 – Função para Validação da Rede Neural WNN

```

1 function [Y]=teste_WNN(X,Wy,t,lambda,H,m,bias)
2
3 [p1 N] = size (X);
4 Y=zeros(m,N);
5
6 % Calculo de Gama
7 for a=1:1:N
8     for ii=1:1:m
9         for b=1:1:H
10            gama=0;
11            for j=1:1:p1
12                gama=gama+((X(j,a)-t(b,j))/lambda(b,1))^2;
13            end
14            gama=gama^(1/2);
15            Y(ii,a)=Y(ii,a)+Wy(ii,b)*MexWav(gama);
16        end
17        Y(ii,a)=Y(ii,a)+bias(1,ii);
18    end
19 end
20
21 end

```

#### Código C.6 – Função para Converter um Sinal de Tensão em Nível (cm)

```

1 function [H]=V2HT1(V)
2     Vmin=min(V);
3     Voff=V-Vmin;
4     H=(65*Voff)/(2.08-Vmin);

```

```
5 end
```

Código C.7 – Função *Wavelet Mexican Hat*

```
1 function PSI = MexWav(gama)
2 c = 2/(sqrt(3)*pi^(1/4));
3 PSI = c*exp(-gama^2/2) * (1-gama^2);
4 end
```

Código C.8 – Derivada da Função *Wavelet Mexican Hat*

```
1 function dPSI = dMexWav(gama)
2 c = 2/(sqrt(3)*pi^(1/4));
3 dPSI=c*gama*exp(-gama^2/2)*(gama^2 - 1) - 2*c*gama*exp(-gama
   ^2/2);
4
5 end
```