

Desenvolvimento de Sensores Virtuais para Monitoramento de Processos Não Lineares Multivariáveis Utilizando Redes Neurais

Nathália Arthur Brunet Monteiro

Tese de Doutorado apresentada à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Instrumentação e Controle

Professor José Sérgio da Rocha Neto, D. Sc.

Orientador

Professor Jaidilson Jó da Silva, D. Sc.

Orientador

Campina Grande, Paraíba, Brasil

©Nathália Arthur Brunet Monteiro, Agosto de 2018

**"DESENVOLVIMENTO DE SENSORES VIRTUAIS PARA MONITORAMENTO DE
PROCESSOS NÃO LINEARES MULTIVARIÁVEIS UTILIZANDO REDES NEURAIIS"**

NATHÁLIA ARTHUR BRUNET MONTEIRO

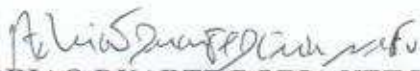
TESE APROVADA EM 13/08/2018



JOSÉ SÉRGIO DA ROCHA NETO, D.Sc., UFCG
Orientador(a)



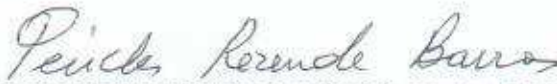
JAIDILSON JÔ DA SILVA, D.Sc., UFCG
Orientador(a)



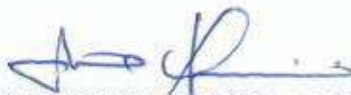
ADRIAIO DUARTE DORIA NETO, Dr., UFRN
Examinador(a)



GIOVANNI CORDEIRO BARROSO, D.Sc., UFC
Examinador(a)



PÉRICLES REZENDE BARROS, Ph.D., UFCG
Examinador(a)



ANGELO PERKUSICH, D.Sc., UFCG
Examinador(a)

CAMPINA GRANDE - PB

Resumo

Em geral, os processos industriais são complexos, não lineares, com múltiplas entradas e múltiplas saídas. Representar esse tipo de sistema por modelos lineares (apesar da sua simplicidade e facilidade de implementação), é, muitas vezes, inadequado quando se faz necessária uma aproximação mais realística e detalhada do processo em estudo. Atualmente, sensores virtuais têm sido utilizados nas indústrias para fazer com que o sistema físico atenda às especificações de desempenho previamente estabelecidas com sucesso, auxiliando no monitoramento e otimização de processos em geral. Para simplificar essa complexidade de se identificar e modelar sistemas não lineares, pode-se utilizar as redes neurais artificiais, que representa uma parte importante do conhecimento na área de sistemas inteligentes, com resultados eficientes na identificação de sistemas complexos e não lineares. Com foco nessas questões, nesta tese é proposto o desenvolvimento de sensores virtuais para monitoramento de variáveis de processos complexos não lineares e com múltiplas entradas e múltiplas saídas, utilizando redes neurais nas estimativas das variáveis. Para a validação dos experimentos realizados, são implementados sensores virtuais seguindo a metodologia proposta para a realização do monitoramento de uma plataforma de teste experimental (sistema de transporte fluídico). As variáveis de interesse no monitoramento da plataforma são valores de pressão e vazão. Com o monitoramento utilizando sensor virtual, é possível obter processos com melhores desempenhos e com menor dificuldade para detectar e solucionar possíveis falhas.

Palavras-chave: Sensor Virtual, Redes Neurais, Sistemas Não Lineares, Monitoramento, Identificação de Sistemas.

Abstract

In general, industrial processes are complex, nonlinear, with multiple inputs and multiple outputs. Representing this type of system by linear models (despite its simplicity and ease of implementation) is often inadequate when a more realistic and detailed approximation of the process under study is necessary. Currently, soft sensors have been used in industries to make physical systems meet the specifications of performance previously established successfully, assisting in monitoring and optimization of general process. To simplify this complexity of identifying and modeling nonlinear systems, artificial neural networks can be used, which represents an important part of the knowledge in the area of intelligent systems, with efficient results in the identification of complex and nonlinear systems. Focusing on these issues, in this thesis the development of soft sensors for the monitoring of variables of complex nonlinear processes with multiple inputs and multiple outputs using neural networks in the estimation of variables is proposed. For the validation of the realized experiments, soft sensors are implemented following the methodology proposed for the monitoring of an experimental test platform (fluid transport system). The variables of interest in platform monitoring are pressure and flow values. By monitoring using soft sensor, it is possible to obtain processes with better performances and with less difficulty to detect and to solve possible failures

Keywords: Soft Sensor, Neural Networks, Nonlinear Systems, Monitoring, System Identification.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivo Geral	3
1.3	Objetivos Específicos	4
1.4	Contribuição	4
1.5	Organização do Texto	5
2	Revisão Bibliográfica	6
3	Fundamentação Teórica	12
3.1	Modelagem e Identificação de Sistemas Não Lineares	12
3.2	Redes Neurais Artificiais	16
3.2.1	Tipos de Redes Neurais	21
3.2.2	Algoritmos de Treinamento	25
3.3	Sensores Virtuais	27
3.4	Monitoramento de Processos	30
4	Descrição da Plataforma de Teste (Sistema de Transporte Fluídico)	32
5	Sensores Virtuais Desenvolvidos e Resultados	43
5.1	Sensores Virtuais Desenvolvidos	43
5.2	Interface Gráfica	49
5.3	Resultados das Simulações	50
5.3.1	Medições para Obtenção da Base de Dados	50
5.3.2	Simulações para Análise de Desempenho das Redes Neurais Proje- tadas	51
5.3.3	Resultados dos Experimentos com Aplicação Prática dos Sensores Virtuais no Monitoramento da Plataforma	64
6	Conclusões e Sugestões para Trabalhos Futuros	69
6.1	Conclusões Gerais	69

6.2 Sugestões para Trabalhos Futuros	70
Referências Bibliográficas	72
A Erros Médios Obtidos nas Estimativas dos Valores de Pressão (Rede 1)	77
B Erros Médios Obtidos nas Estimativas dos Valores de Vazão (Rede 2)	84
C Medições para Obtenção do Conjunto de Dados (Redes 3 e 4)	91
D Erros Médios Obtidos nas Estimativas dos Valores de Pressão (Rede 3)	99
E Erros Médios Obtidos nas Estimativas dos Valores de Vazão (Rede 4)	106

Lista de Tabelas

3.1	Algoritmos de Treinamento e Seus Métodos de Otimização.	26
5.1	Valores de Pressão Reais Medidos e Valores Estimados com as Redes <i>Feedforward</i> e <i>Feedforward</i> em Cascata (Rede 3).	59
5.2	Valores de Vazão Reais Medidos e Valores Estimados com as Redes <i>Feedforward</i> e <i>Feedforward</i> em Cascata (Rede 4).	60
5.3	Erros Obtidos Entre Valores Reais e Estimados de Pressão e Vazão com as Redes 3 e 4.	61
5.4	Tabela Resumo	63
5.5	Erros Obtidos nas Estimativas dos Valores de Pressão e Vazão Aplicados na Plataforma Utilizando as Redes 1 e 2.	66
5.6	Erros Obtidos nas Estimativas dos Valores de Pressão e Vazão Aplicados na Plataforma Utilizando as Redes 3 e 4.	68
A.1	Erros obtidos com Rede <i>Feedforward</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 1).	78
A.2	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 1).	79
A.3	Erros obtidos com Rede <i>Feedforward</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 1).	80
A.4	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 1).	81
A.5	Erros obtidos com Rede <i>Feedforward</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 1).	82
A.6	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 1).	83
B.1	Erros obtidos com Rede <i>Feedforward</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 2).	85

B.2	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 2).	86
B.3	Erros obtidos com Rede <i>Feedforward</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 2).	87
B.4	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 2).	88
B.5	Erros obtidos com Rede <i>Feedforward</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 2).	89
B.6	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 2).	90
C.1	Valores Obtidos nas Medições Para a Coleta do Conjunto de Dados.	91
C.2	Média e Desvio Padrão dos Valores Obtidos com as Medições.	93
C.3	Valores de Pressão Reais Medidos e Valores Estimados com as Redes <i>Feedforward</i> e <i>Feedforward em Cascata</i> (Rede 3).	94
C.4	Valores de Vazão Reais Medidos e Valores Estimados com as Redes <i>Feedforward</i> e <i>Feedforward em Cascata</i> (Rede 4).	96
C.5	Erros Obtidos Entre Valores Reais e Estimados de Pressão e Vazão com as Redes 3 e 4.	97
D.1	Erros obtidos com Rede <i>Feedforward</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 3).	100
D.2	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 3).	101
D.3	Erros obtidos com Rede <i>Feedforward</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 3).	102
D.4	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 3).	103
D.5	Erros obtidos com Rede <i>Feedforward</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 3).	104
D.6	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 3).	105
E.1	Erros obtidos com Rede <i>Feedforward</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 4).	107
E.2	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 4).	108
E.3	Erros obtidos com Rede <i>Feedforward</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 4).	109

E.4	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 4).	110
E.5	Erros obtidos com Rede <i>Feedforward</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 4).	111
E.6	Erros obtidos com Rede <i>Feedforward em Cascata</i> com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 4).	112

Lista de Figuras

3.1	Representação do Processo de Modelagem Simples.	13
3.2	Procedimento para Identificação de Processos.	14
3.3	Representação em Diagrama de Bloco do Sistema Nervoso Humano Visto como Sistema de Três Estágios.	18
3.4	Modelo de Neurônio Base para Projetos de RNA.	19
3.5	Gráfico Função Sigmóide.	21
3.6	Redes Alimentadas Adiante com uma Camada Oculta e uma de Saída. . .	23
3.7	Rede Neural em Cascata	24
3.8	Diagrama do Processo de Funcionamento do Sensor Virtual.	29
3.9	Sensor Virtual Conectado a uma Planta.	29
4.1	Fotografia da Plataforma de Detecção de Incrustação.	33
4.2	Curva Característica Estimada por Ajuste Polinomial (Traço Superior) So- breposta pela Curva característica Experimental Vazão <i>versus</i> Frequência (Traço Sobreposto).	38
4.3	Curva experimental Vazão <i>versus</i> Tempo (linha tracejada) e Frequência <i>versus</i> Tempo (linha tracejada e pontilhada).	39
4.4	Gráfico Logarítmico $h_{1t}X\bar{V}$, com Destaque as Velocidades Inicial (\bar{V}_i) e Final (\bar{V}_f) da Zona de Transição.	40
4.5	Gráfico Logarítmico $h_{1t}X\bar{V}$ (Comportamento Hidrodinâmico da Plataforma de Testes).	41
4.6	Ampliação da Figura 4.5 para Verificar a Zona de Transição de Regimes. .	41
5.1	Fluxograma das Etapas Seguidas para o Desenvolvimento dos Sensores Virtuais.	44
5.2	Fluxograma da Divisão das Amostras do Processo.	45
5.3	Esquema com Entradas e Saídas para Treinamento das Redes.	47
5.4	Interface para Monitoramento.	50
5.5	Conjunto Total de Amostras Utilizado na Criação das Redes Neurais. . . .	51

5.6	Redes Neurais como um Diagrama Gráfico (Rede 1). <i>Feedforward</i> (a) e <i>Feedforward</i> em Cascata (b).	53
5.7	Comparação do Desempenho das Redes do Tipo <i>Feedforward</i> e <i>Feedforward</i> em Cascata nas Estimativas dos Valores de Pressão (Rede 1).	54
5.8	Redes Neurais como um Diagrama Gráfico (Rede 2). <i>Feedforward</i> (a) e <i>Feedforward</i> em Cascata (b).	55
5.9	Comparação do Desempenho das Redes do Tipo <i>Feedforward</i> e <i>Feedforward</i> em Cascata nas Estimativas dos Valores de Vazão (Rede 2).	56
5.10	Redes Neurais como um Diagrama Gráfico (Rede 3). <i>Feedforward</i> (a) e <i>Feedforward</i> em Cascata (b).	57
5.11	Redes Neurais como um Diagrama Gráfico (Rede 4). <i>Feedforward</i> (a) e <i>Feedforward</i> em Cascata (b).	58
5.12	Monitoramento dos Valores de Pressão (a) e Vazão (b) Utilizando as Redes 3 e 4.	62
5.13	Monitoramento dos Valores de Pressão (a) e Vazão (b) Durante o Mesmo Processo (Redes 1 e 2).	64
5.14	Monitoramento dos Valores de Pressão (a) e Vazão (b) Durante o Mesmo Processo (Redes 3 e 4).	67

Capítulo 1

Introdução

1.1 Motivação

Para processos mais seguros, eficientes e mais rentáveis em plantas industriais, há uma crescente necessidade de monitoramento e controle [41]. Devido à crescente complexidade e grau de integração nesses sistemas, soluções adequadas para um bom monitoramento exigem um profundo conhecimento quantitativo do processo e dos parâmetros relevantes deste processo. O surgimento de aplicações industriais e de controle de processos por *hardware*, tecnologia de interface gráfica do usuário, pacotes de computação de alto nível e melhoria contínua de *hardware* e *software* levou ao surgimento do paradigma de instrumentação virtual [35, 46]. Essa complexa inter-relação entre as variáveis que compõem o processo, tem levado ao desenvolvimento de técnicas que possibilitam a estimação de determinadas variáveis através de informações adquiridas por meio de medições de outras variáveis disponíveis no processo [34].

O uso de técnicas como redes neurais artificiais (RNA) na modelagem de sistemas dinâmicos não lineares tem sido difundido na literatura nos últimos anos, motivando diversos trabalhos e uma ampla gama de aplicações na literatura. O uso da RNA pode auxiliar na implementação do sensor virtual para o processo de monitoramento, em busca de processos com melhores desempenhos [7, 52].

Sistemas mais complexos, como plantas de processamento industrial, são em geral fortemente instrumentados com um grande número de sensores. Com o uso dos sensores é possível adquirir dados que podem ser usados no processo de monitoramento e controle das plantas [22]. O conceito de utilização de sensores virtuais visa à modelagem matemática de processos com enfoque na predição de propriedades, a partir de medições disponíveis de outras variáveis da planta [21]. No contexto do processo industrial, esses modelos preditivos são chamados sensores virtuais (*soft sensors*). Esse termo é uma combinação das palavras "*software*" e "*sensors*", pois os modelos são geralmente rotinas de *software*,

e estão representando grandezas semelhantes às dos sensores reais [22].

Existem várias aplicações para a indústria em que os sensores reais requerem aquisição, instalação, operação e altos custos de manutenção. Os sensores virtuais são úteis para essas aplicações. Todas essas necessidades levaram a um crescente interesse no uso de sensores virtuais para fornecer monitoramento eficiente, robusto e em tempo real [23,36]. Assim, os sensores virtuais são uma boa solução para reduzir a complexidade dos processos, podem atuar substituindo ou trabalhando em conjunto com sensores reais auxiliando no monitoramento e controle de falhas e manutenção preventiva do processo, além da possibilidade de serem úteis em casos onde a medição direta do sinal não é possível ou quando o sensor disponível pode não garantir as características de detecção necessárias [3].

Reda *et al.* [44], para monitorar as temperaturas e as localizações de *hot spots* (pontos aquecidos) em processadores reais, propôs uma técnica computacional de sensor virtual que utiliza as medições dos sensores reais para calcular de forma otimizada as temperaturas onde não há sensores embutidos. A técnica com sensor virtual pode melhorar a resolução de monitoramento térmico e contornar restrições de projeto sobre a colocação do sensor real.

et al. [43] apresenta um sensor virtual adaptativo baseado no processo Gaussiano de regressão. Para tornar o treinamento do modelo mais eficiente, um algoritmo para seleção de variáveis baseado em informações mútuas é proposto. A adaptação do modelo é derivada usando a abordagem de janela móvel. Deste modo, é alcançado um modelo não linear probabilístico adaptativo que é adequado para modelar processos não lineares. As capacidades de previsão do método proposto são analisadas em dados industriais reais obtidos em uma coluna de destilação de petróleo.

Ge *et al.* [56] apresentaram um modelo de sensor virtual probabilístico baseado em regressão por componentes principais e, em seguida, o modelo é estendido para processos com múltiplos modos de operação. O novo modelo de regressão por componentes principais probabilístico é baseado na estimativa de máxima verossimilhança e no algoritmo de maximização de expectativa. Em comparação com os sensores virtuais baseados em modelos locais e globais, o novo modelo é mais eficiente para a previsão de qualidade em processos multimodo. Além disso, ao localizar as informações de modo da nova amostra de dados, o sensor virtual proposto tem seu potencial para melhorar a compreensão do processo e a compreensão da qualidade do produto para o engenheiro de operação. Dois estudos de caso mostraram que resultados mais satisfatórios foram obtidos com o novo sensor virtual.

Lotufo [34] apresenta uma metodologia para desenvolvimento de sensores virtuais capazes de inferir variáveis de processos altamente não lineares e variantes no tempo. A

metodologia proposta emprega modelagem nebulosa de sistemas dinâmicos complexos, em que a parte antecedente das regras emprega a técnica de agrupamento (clusterização) de *Gustafson-Kessel* (*product space clustering*), e os parâmetros da parte conseqüente das regras são estimados utilizando-se o algoritmo dos mínimos quadrados recursivos, com fator de esquecimento variável. O algoritmo proposto foi avaliado por meio de um modelo virtual implementado no *software* MATLAB para o processo de neutralização de pH. O algoritmo de identificação nebulosa proposto e implementado no trabalho, utilizado como sensor virtual de pH, forneceu resultados coerentes, quando comparado com outras técnicas de modelagem da literatura, no tocante ao tempo de resposta, erro de predição, capacidade de adaptação e número de amostras necessárias à fase de treinamento.

Shardt and Yang [13] examina o projeto do termo de atualização de polarização para sistemas de sensores virtuais em condições de malha aberta e fechada para diferentes tipos de atrasos de tempo aleatórios. Foi demonstrado que, se o sensor virtual estiver executando em malha aberta, é melhor usar somente o valor disponível mais recente. Por outro lado, para sensores virtuais em malha fechada, se vários valores estiverem disponíveis em um dado ponto, então é melhor para alguns uma média ponderada da informação disponível. A ponderação exata dependerá da natureza do atraso de tempo (estocástico ou determinístico) e do tipo de sistema que está sendo considerado. A taxa de amostragem não teve nenhum impacto no desempenho geral do sistema.

Pode-se observar a importância e utilidade do conceito de sensor virtual no monitoramento de processos em geral e as dificuldades encontradas para sua realização, assim como de se estudar a modelagem e identificação de sistemas não lineares aliado às redes neurais artificiais, verificando que são áreas de estudo promissoras e de grande importância para diversas aplicações, buscando entender e explicar fenômenos que ocorrem, por exemplo, na natureza, em sistemas sociais, biomédicos; sistemas de controle e monitoração; estimação; simulação e treinamento. Nesta tese, é proposto o desenvolvimento de sensores virtuais utilizando redes neurais, visto as dificuldades encontradas, como inexistência de sensores para algumas aplicações específicas, preços elevados, atrasos ou imprecisão das medidas.

1.2 Objetivo Geral

O objetivo principal neste trabalho é o desenvolvimento de sensores virtuais para monitoramento de variáveis de processos complexos não lineares e com múltiplas entradas e múltiplas saídas, utilizando redes neurais nas estimativas das variáveis. O intuito é monitorar o sistema utilizando o conceito de sensor virtual aliado às redes neurais nas estimativas, o que ainda é pouco difundido na literatura técnico científica, sendo uma área

promissora de pesquisa.

1.3 Objetivos Específicos

Implementar sensores virtuais com o intuito de realizar o monitoramento dos valores de pressão e vazão em uma plataforma de teste experimental (sistema de transporte fluídico) seguindo algumas etapas:

- aquisição de amostras para compor base de dados de valores das variáveis de entradas e saídas que se relacionam no processo;
- tratamento dessa base de dados, separação de amostras e normalização;
- desenvolvimento de redes neurais com diferentes algoritmos de treinamento;
- aplicação do sensor virtual desenvolvido para monitoramento da plataforma de teste utilizada.

1.4 Contribuição

A contribuição neste trabalho é o desenvolvimento de sensores virtuais para monitoramento de processos não lineares multivariáveis. O fato de não precisar do modelo matemático, das equações que regem a planta, é um ponto positivo, porque evita trabalhar com a complexidade inerente a esse tipo de processo. Com dados de entradas e saídas que se relacionam no processo é possível desenvolver sensores virtuais para a aplicação de interesse.

Existem algumas áreas de atuações dos sensores virtuais que são importantes e podem ser abordadas no estudo para monitoramento de processos em geral. Pode-se citar casos em que se deseja obter medidas de sensores em ambientes que são inacessíveis fisicamente, ou de complicado acesso. Outro caso seria o de confiabilidade, realizar a verificação do estado do sensor real, usar o sensor virtual monitorando o sensor real para ser possível identificar se existe algum problema. Um outro caso, que foi o de interesse e abordado nesta tese, é o cenário em que se deseja que o sensor virtual acompanhe o mesmo comportamento do sensor real, que funcione idealmente igual ao sensor real de interesse. Com isso, pode-se ter cenários com plantas altamente complexas que precisam de uma instrumentação ampla, em que se tem gastos elevados pela necessidade de uma grande quantidade de sensores, verificando-se a necessidade de se ter processos monitorados com um menor número de sensores, consequentemente reduzindo custos.

Nesse contexto, a ideia proposta pode ser ampliada para os demais casos citados, de confiabilidade e de predição de medidas inacessíveis, por exemplo.

1.5 Organização do Texto

O texto deste trabalho está organizado da seguinte forma:

Neste Capítulo foi realizada uma introdução do trabalho, sendo apresentados a motivação, os objetivos e as contribuições.

No Capítulo 2, é apresentada uma revisão bibliográfica, apontando uma amostragem de abordagens e aplicações de modelagem e identificação de sistemas não lineares aliado às redes neurais artificiais e do uso de sensores virtuais no monitoramento de processos.

No Capítulo 3 é apresentada a fundamentação teórica, sobre modelagem e identificação de sistemas, redes neurais artificiais, sensor virtual e monitoramento de processos.

No Capítulo 4 é apresentada a descrição da plataforma (sistema não linear) utilizada para os experimentos realizados.

No Capítulo 5 é apresentado o desenvolvimento dos sensores virtuais com base em redes neurais, com a explicação sobre as etapas para o desenvolvimento, como a descrição das entradas e saídas do processo, a escolha dos algoritmos de treinamento e as estimativas obtidas. Neste capítulo são apresentadas também as medições realizadas, a interface gráfica desenvolvida para uma melhor interação com o usuário e, por fim, as simulações e os resultados obtidos.

As conclusões gerais em relação ao trabalho realizado e as sugestões para trabalhos futuros são apresentadas no Capítulo 6.

Capítulo 2

Revisão Bibliográfica

As medições em tempo real são bastante desejadas para monitoramento do processo, mas há questões que dificultam isso, como a ausência de métodos de detecção apropriados e o alto custo dos métodos conhecidos. Isto faz com que essas variáveis de interesse sejam monitoradas através de análises *offline* em laboratório, ocorrendo perda de informação, demora na obtenção e análise dos resultados e normalmente requerem aumento do esforço humano, resultando em uma redução de rendimento [19].

Assim, visualizar os dados de um processo é um requisito importante para sistemas de teste, medição ou controle, surgindo a necessidade do emprego de elementos de interface com o mundo real, ou mundo físico [22].

Modelagem matemática é a área do conhecimento que estuda métodos para desenvolver e implementar modelos matemáticos de sistemas reais [2]. Entende-se o modelo matemático de um sistema como sendo um conjunto de equações que descrevem seu comportamento dinâmico e estático. É importante que a obtenção de um modelo matemático estabeleça um compromisso entre a simplicidade do modelo e a precisão dos resultados da análise [41].

Métodos computacionais para modelagem e simulação são significativos na indústria, onde a qualidade desempenha um papel fundamental. O interesse industrial e acadêmico pelo papel dos computadores na tecnologia de fabricação está aumentando, uma vez que a modelagem e a simulação são capazes de levar à otimização de processos e, ao mesmo tempo, reduzir o trabalho experimental caro e demorado [38].

A utilização de técnicas para modelagem de sistemas baseadas em modelos lineares é, em parte, devido à simplicidade dos modelos empregados para representar o comportamento do processo, no entanto, isto também constitui um possível problema, porque tais modelos lineares são, muitas vezes, inadequados quando se faz necessária uma melhor aproximação de um processo. Por outro lado, os esquemas para sistemas não lineares, os quais empregam modelos mais realistas e, portanto, mais complexos, eliminam a simpli-

cidade associada às técnicas lineares para conseguir a descrição de processos não lineares. Modelos não lineares possibilitam um retrato mais fiel do processo quando este se faz necessário. Apesar de apresentar uma complexidade maior, apenas a representação através de um modelo não linear permite a análise de algumas características do sistema, como oscilações [12]. Atualmente, uma solução interessante para permitir qualidade, eficiência e segurança no monitoramento de um processo é utilizar redes neurais artificiais.

O interesse pelas redes neurais é motivado principalmente pelas características destas técnicas que, permitem a construção de modelos que são aproximadores universais de funções. De fato, dependendo da técnica utilizada, é possível aproximar com precisão arbitrária uma função não linear contínua definida em uma região compacta (limitada e fechada) tomando-se como base tanto informações quantitativas quanto qualitativas [7, 52].

As RNAs podem ser aplicadas em diversos tipos de tarefas, por exemplo, no reconhecimento de padrões, aproximação de funções e predição. Nesta tese, a tarefa adotada é a estimativa de variáveis para monitoramento de processos.

Visto a necessidade e importância do monitoramento de processos em aplicações industriais, e as dificuldades encontradas para sua realização, surge o interesse no desenvolvimento do sensor virtual. A utilização do conceito de sensores virtuais já é estudada e está ligada à modelagem matemática de processos com abordagem na predição de propriedades a partir de dados da planta [21]. Há muitas variáveis medidas "*online*" e que são amostradas de forma relativamente frequente, e estas variáveis podem estar indiretamente relacionadas às variáveis difíceis de serem medidas [34].

Assim, pode-se dizer que um sensor virtual é um mecanismo utilizado para substituir a indisponibilidade momentânea ou permanente de um sensor real em uma planta, que pode acontecer devido a uma falha no sensor real ou por sua remoção para manutenção, por exemplo [34].

Aplicações de sensores virtuais estão aumentando com o crescente avanço tecnológico, principalmente por seu custo relativamente baixo em comparação com os sensores reais, quando se trata de grandes processos industriais, onde as plantas de processamento são em geral fortemente instrumentadas, com um grande número de sensores [35].

Muitas vezes é impraticável ter processos completamente monitorizados, para isso é necessário a construção de plantas fortemente instrumentadas por um custo elevado. Por esta razão, o uso de sensores virtuais é uma alternativa valiosa a esta solução. O monitoramento de processos possibilita o diagnóstico de problemas, isto é particularmente útil em processos descontínuos (processos divididos em etapas). Em um processo industrial, por exemplo, o diagnóstico de problemas reduz o número de produtos rejeitados, permite a detecção de falhas antes que o produto em questão esteja concluído, e ajuda a prevenir

falhas nos produtos subsequentes [1].

Angelos *et al.* [38] desenvolveram um sistema para predição de rugosidade da superfície em processo de moagem, que são processos de remoção de materiais que estão envolvidos em vários produtos finais do setor manufatureiro e que exigem alta qualidade, geralmente quantificada através da rugosidade superficial do produto final. A rugosidade da superfície influencia vários atributos de uma peça, como o comportamento de fadiga, desgaste, corrosão, lubrificação e atrito superficial. Alguns tipos de redes neurais artificiais são examinados e comparados para a previsão da rugosidade superficial na tecnologia de fabricação, como por exemplo rede neural de base radial (RBF) e rede neural *feedforward* com diferentes algoritmos de treinamento. Quatro variáveis independentes foram usadas como entradas, velocidade do fuso, taxa de alimentação, profundidade de corte e vibrações e a saída das redes foi a rugosidade da superfície. Com o trabalho, pode-se concluir que as redes neurais artificiais podem prever satisfatoriamente a rugosidade superficial em processo de moagem.

Singh *et al.* [50] apresentaram um sistema de diagnóstico de câncer de mama baseado em redes neurais. O câncer de mama é uma das doenças fatais em mulheres, que tem mais possibilidade de cura se detectado em um estágio inicial. Sistema de rede neural tem a capacidade de ser treinado por grandes volumes de dados e informações ocultas ou recursos nas amostras. Os sistemas de diagnóstico com base em redes neurais artificiais (RNA) para o câncer de mama foi desenvolvido utilizando dois estágios. Em primeiro lugar, a base de dados WDBC (*Wisconsin Breast Cancer Database*) é usada para desenvolver o sistema. É realizado o treinamento supervisionado da rede neural usando o algoritmo BP (*Backpropagation*). Duas variantes do algoritmo são investigados, uma rede neural com três e outra rede neural com quatro camadas. Com a rede neural desenvolvida de forma eficiente, o método de teste é aplicado para imagens de biópsia de lâminas de tecidos mamários. O pré-processamento de imagens é feito para extrair características-chave necessárias que utilizam algoritmos de processamento de imagens usando a ferramenta *Simulink* do MATLAB. Assim, um conjunto de dados é criado usando os estudos de caso de câncer de mama. No trabalho foi demonstrado que a rede neural pode ser uma ferramenta eficiente na tomada de decisões no campo de diagnóstico de câncer através de lâminas de biópsia.

Liu *et al.* [30] propuseram uma nova abordagem de sensor virtual, que envolve um modelo de resposta ao impulso finito não linear com parâmetros lineares adaptáveis para a predição em tempo real de viscosidade de fusão com base nas entradas do processo; a saída do modelo é então usada como uma entrada de um modelo com uma estrutura fixa simples para prever a pressão que pode ser medida *online*. Finalmente, a pressão predita é comparada com o valor medido e o erro correspondente é usado com um sinal

de retorno para corrigir a viscosidade estimada. Essa nova estrutura de retorno permite a adaptabilidade *online* do modelo de viscosidade em resposta a erros de modelagem e perturbações, portanto, produzindo uma estimativa de viscosidade confiável. Os resultados experimentais confirmam a eficácia do método de sensor virtual proposto para o monitoramento e controle de processos de extrusão de polímeros em tempo real.

Tang *et al.* [51] propuseram um método que se diferencia do modelo de aproximação linear na captura do comportamento do sistema não linear sobre uma região maior do ponto de operação. No trabalho são considerados a classe de sistemas não lineares em cascata ou com combinação de *feedback* de elementos estáticos não lineares e um sistema dinâmico linear. É mostrado que a similaridade arbitrária entre a saída do sistema linear e do sistema não linear pode ser conseguida. O esquema proposto é verificado em sistemas não lineares simulados, e testados em um sistema físico (*servomotor*), e encontra aplicação na identificação de sistemas para projetos de controle, diagnósticos de falhas, e análise do comportamento de sistemas não lineares.

Serpas *et al.* [47] desenvolveram uma abordagem para detecção de falhas que envolve sensores virtuais para o monitoramento de processos. Ao contrário de abordagens existentes, as quais comparam medições atuais, ou combinações lineares das mesmas, com valores dessas medições representando operações normais, a metodologia apresentada lida diretamente com as estimativas de estado que precisam ser monitoradas. A estratégia do processo de monitoramento compara o estado estimado a limites que representam valores de estado derivados das operações normais do processo. É importante falar que esses limites são diferentes dos limites que seriam derivados se os estados fossem medidos em vez de estimados. A razão para isso é que o projeto do sensor virtual tem impacto direto na variância desses resultados estimados. Conformemente, os limites precisam ser ajustados baseados no projeto do sensor virtual. A vantagem de tal abordagem é que o efeito das condições anormais do processo nas variáveis de estado pode ser observado diretamente e que é possível incluir relações não lineares entre medições e estados. Ao mesmo tempo, esse tipo de abordagem tem o inconveniente que as variâncias dos estados não medidos não são iguais às variâncias das variáveis do processo atuais devido ao uso do sensor virtual. Entretanto, para muitas técnicas de sensores virtuais, como filtros de Kalman e abordagens relacionadas, é possível computar variâncias dos estados preditivos que correspondem a condições de operações normais. A abordagem apresentada, de realizar verificação de limites em estimativas de estados fisicamente significativas de variáveis não medidas, em vez de simplesmente em medições disponíveis, foi mostrada para oferecer melhor desempenho.

Santos [12] apresenta um estudo de modelagem, identificação e controle preditivo não linear baseado em modelos de *Hammerstein*. As pesquisas associadas às estratégias de con-

trole preditivo não linear têm apresentado grande crescimento ultimamente registrando, também, um número considerável de aplicações na indústria. A representação de um processo complexo através de um modelo não linear, com o objetivo de melhorar seu desempenho dinâmico, tende a sacrificar a simplicidade de projeto do controlador preditivo. No trabalho, são apresentadas algumas técnicas de seleção de estrutura e identificação do modelo de *Hammerstein* e algumas inovações são propostas. A implementação prática realizada numa planta solar de climatização mostrou-se desafiadora pela complexidade de operação e características peculiares: grandes perturbações no sistema; fortes restrições de amplitude e velocidade nas variáveis envolvidas; atraso de transporte. Foram identificados e validados modelos baseados na estrutura de *Hammerstein* e a estratégia pelo controlador preditivo foi implementada visando controlar uma das malhas da planta solar. Apesar das dificuldades encontradas o desempenho do sistema de controle foi satisfatório e os modelos obtidos mostraram-se representativos.

Li *et al.* [29] aplicaram uma rede neural artificial com algoritmo BP (*Backpropagation*) para determinar fontes de influxo de água de mina com base no *software* MATLAB. Por causa do ambiente de desenvolvimento, das condições de ocorrência, da aleatoriedade, incerteza e imprecisão do influxo de água de mina, se torna um sistema não linear muito complexo para a determinação das fontes do influxo dessa água. Nos experimentos para criação, treinamento e teste foram usadas 78 amostras de água do poço como amostras de treinamento e outras 31 amostras de água do poço como amostras de teste. O modelo ganha uma melhor aplicação em projetos práticos e a precisão da avaliação é acima de 90%. Os seus resultados de avaliação não são facilmente afetados pelos fatores humanos. A rede neural BP projetada é eficaz e é viável sua popularização e aplicação para determinar fontes de influxo de água de mina.

Arsie *et al.* [3] descreveram o desenvolvimento de sensores virtuais baseados em redes neurais usados para estimar as emissões de NOx (óxidos de nitrogênio) na exaustão de motores Diesel automotivos. Metodologias de identificação adequadas e testes experimentais foram desenvolvidos com o objetivo de atender às necessidades conflitantes de uma implementação a bordo viável e precisão satisfatória na predição. Além disso, como a predição de modelos orientados ao controle é tipicamente afetada pelo envelhecimento do motor e espalhamento da produção, os recursos da técnica de mínimos quadrados foram explorados para superar esses problemas, adaptando a saída do sensor virtual. O sensor virtual adaptativo de NOx foi testado através da comparação com dados experimentais reais, que foram tratados para permitir uma avaliação completa das capacidades propostas de adaptação do algoritmo baseado em mínimos quadrados.

Os processos, na sua maioria, são complexos, apresentando características fortemente não lineares, difíceis de serem representadas, necessitando do bom conhecimento do pro-

cesso, além de exigir procedimentos computacionais normalmente complexos principalmente no que se refere a sistemas multivariáveis, ou seja, compostos por múltiplas entradas e múltiplas saídas que configuram sistemas do tipo (*MIMO -Multi Input Multi Output*), com elevada interação entre as variáveis do processo e com fortes restrições.

Com a necessidade de modelos mais detalhados e completos, aliado ao aumento do poder computacional, foi possível estabelecer especificações de desempenho mais rígidas nos projetos de engenharia, tornando viável suprir a necessidade de modelos mais detalhados nos processos em estudo.

Neste capítulo teve-se o objetivo de apresentar uma amostragem de abordagens e aplicações do uso do conceito de sensor virtual no monitoramento e controle de processos em geral, e da modelagem e identificação de sistemas não lineares aliado às redes neurais, proporcionando uma visão da evolução e do cenário atual desses assuntos, mostrando que são áreas de estudo promissoras e de grande importância para diversas aplicações.

Capítulo 3

Fundamentação Teórica

Neste capítulo é apresentada uma fundamentação teórica, onde são abordados os conceitos de modelagem e identificação de sistemas, monitoramento de processos, redes neurais artificiais e sensor virtual.

3.1 Modelagem e Identificação de Sistemas Não Lineares

Um modelo é um análogo matemático que representa algumas das características observadas em tal sistema. Evidentemente, há outros tipos de modelos além dos modelos matemáticos. Por exemplo, a maquete de um edifício também é um modelo, ainda que não seja matemático. É importante perceber que uma maquete possui algumas das características da construção real, mas não todas. A mesma observação é válida para os modelos matemáticos [2].

A modelagem de um processo dinâmico é uma ferramenta para responder a perguntas sobre o sistema sem ter que fazer um experimento, é uma forma de representar adequadamente os aspectos essenciais do processo em estudo [32].

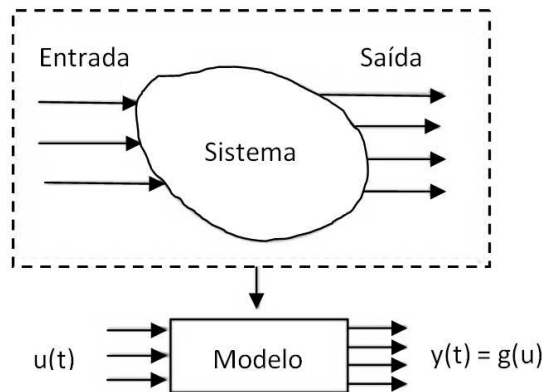
Suponha que, por diferentes razões (econômicos ou de segurança, por exemplo), experimentos em um processo real apresentem restrições ou não possam ser executados, tornando-se útil a realização de estudos de simulação a partir de um modelo do processo, nesses casos o modelo pode ser utilizado para calcular ou decidir como o sistema teria reagido [32].

Esses modelos têm sido utilizados para os mais diversos fins, como, tentar entender e explicar fenômenos que ocorrem, por exemplo, na natureza, em sistemas sociais, biomédicos; sistemas de controle e monitoramento ; estimação; simulação e treinamento [2].

Na Figura 3.1 está ilustrado um sistema de modelagem simples. Estritamente falando,

um sistema é algo real (como um amplificador, um carro ou um corpo humano), enquanto que o modelo é uma abstração (um conjunto de equações matemáticas). Pode-se observar que o modelo do processo não é uma cópia real do sistema, e sim uma aproximação do seu comportamento, simulando o sistema como um todo [5].

Figura 3.1: Representação do Processo de Modelagem Simples.



Fonte: Cassandras *et al.* [5].

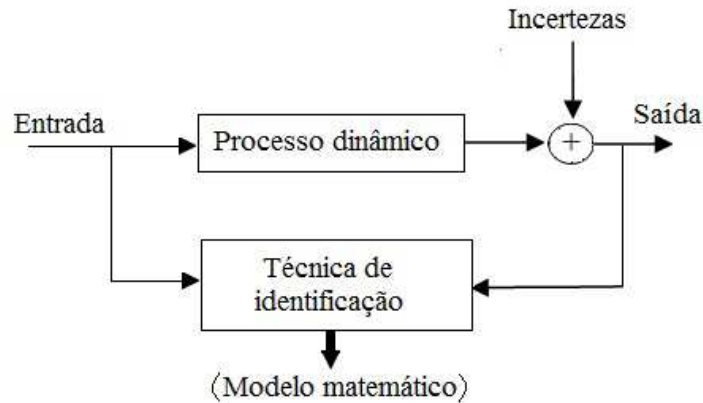
A utilização do modelo para simulação do sistema constitui-se em um procedimento de baixo custo e seguro para experimentar o sistema. Entretanto, a validade (adequação) dos resultados de simulação depende completamente da qualidade do modelo matemático do sistema [6]. O modelo de um sistema pode ser obtido de duas formas: a partir das equações básicas do sistema - *Modelagem Fenomenológica* ou a partir da medição de dados de entrada e saída do sistema - *Identificação de Sistemas*. A dificuldade na obtenção de um modelo fenomenológico adequado, devido à complexidade dos sistemas reais, aliada à grande evolução dos computadores e ao desenvolvimento de estratégias de identificação eficientes, fizeram a identificação de sistemas tornar-se o principal procedimento para a obtenção de modelos matemáticos sendo, atualmente, objeto de estudo de inúmeros pesquisadores das mais diversas áreas de atuação [6,32]. Nesta tese o foco é a identificação e modelagem de sistemas voltada ao campo de pesquisas na área de monitoramento de processos.

A ideia da identificação de sistemas é permitir a elaboração do modelo matemático de um sistema dinâmico, baseado em medidas coletadas pelo ajuste de parâmetros e/ou do modelo, até que a saída do sistema coincida, tão bem quanto possível, com as amostras das saídas reais medidas [31]

Para os processos industriais, o modelo pode ser obtido a partir do tratamento das medidas (procedimento estatístico, filtragem de dados) coletadas a partir de uma realização experimental. O modelo matemático final é uma forma do conhecimento da relação existente entre os sinais de entrada e saída, caracterizada no processo físico pela função de transferência [6]. Na Figura 3.2 está ilustrada a composição básica em blocos de uma

tarefa de identificação.

Figura 3.2: Procedimento para Identificação de Processos.



Fonte: Coelho (Modificada) [6].

Através da técnica de identificação, a partir dos dados de entrada e saída do sistema, obtém-se o modelo matemático que represente os aspectos essenciais do sistema.

Outras informações importantes sobre identificação são:

1. o tipo de modelo do processo (linear/não linear, variante/invariante no tempo, paramétrico/não paramétrico, contínuo/discreto, monovariável/multivariável, característica do ruído, atraso de transporte);
2. a precisão requerida do modelo (baixa, média, alta);
3. o método de estimação do processo (*offline/online*, malha aberta/malha fechada, resposta ao degrau, resposta em frequência, análise espectral e de Fourier, métodos de estimação de parâmetros, etc).

A noção de linearidade é fundamental na ciência e na engenharia e está intimamente associada ao princípio da "superposição", descrito pela seguinte propriedade: se um estímulo S_1 produz uma resposta R_1 , e um estímulo S_2 produz uma resposta R_2 , então a superposição dos dois estímulos, $(S_1 + S_2)$, irá produzir a superposição das duas respostas, $(R_1 + R_2)$. Na sua forma mais simples, isto é, $S_1 = S_2$, a superposição segue a proporcionalidade. Por exemplo, duplicar a entrada para um sistema resulta na duplicação da saída [5].

A classe de sistemas lineares é um pequeno subconjunto dos possíveis sistemas. Abrange muitos casos de interesse ou fornece aproximações adequadas que pode-se usar para fins práticos. Grande parte dos estudos relacionados ao monitoramento e controle de processos é de fato baseada na análise de sistemas lineares, e levou a muitos casos de êxito, desde

a concepção de estruturas eletromecânicas complexas até a descrição do comportamento de economias e crescimento populacional [5].

Então, é bem verdade que em alguns casos, aproximações lineares são sim suficientes para aplicações práticas. O apelo pelo uso desses modelos lineares acontece, muitas vezes, devido à sua simplicidade comparada com modelos não lineares. Mas, em muitas situações práticas, algumas de suas suposições, como a normalidade e a linearidade nos parâmetros, não são satisfeitas, e representações não lineares devem ser usadas. A escolha de modelos não lineares, entretanto, traz consigo um inevitável aumento na complexidade dos algoritmos a serem estudados, comprometem a simplicidade associada às técnicas lineares a fim de alcançar um desempenho elevado [2].

Quando se parte de sistemas lineares para sistemas não lineares, depara-se com uma situação mais difícil. O princípio da superposição não se sustenta mais e as ferramentas de análise envolvem matemática mais avançada. A dinâmica de um sistema não linear é muito mais rica que a dinâmica de um sistema linear. Existem "fenômenos essencialmente não lineares" que podem ocorrer apenas na presença de não linearidade e, portanto, não podem ser descritos por modelos lineares [26]. São exemplos de fenômenos não lineares:

- **Tempo de fuga finito**

O estado de um sistema não linear pode ir ao infinito em tempo finito, o que não acontece para sistemas lineares.

- **Equilíbrios isolados múltiplos**

Um sistema linear pode ter apenas um ponto de equilíbrio e, portanto, apenas um ponto de operação em estado estacionário que atrai o estado do sistema, independentemente do estado inicial. Um sistema não linear pode ter mais de um ponto de equilíbrio isolado. O estado pode convergir para um dos vários pontos operacionais do estado estacionário, dependendo do estado inicial do sistema.

- **Ciclo limite**

Um sistema linear pode ter uma oscilação estável se tiver um par de autovalores no eixo imaginário. A amplitude da oscilação dependerá das condições iniciais. Existem sistemas não lineares que podem entrar em uma oscilação de amplitude e frequência fixas, independentemente das condições iniciais. Este tipo de oscilação é conhecido como um ciclo limite.

- **Caos**

Um sistema não linear pode ter um comportamento estacionário mais complicado que não seja equilíbrio ou oscilação periódica. Tal comportamento é geralmente

referido como caos. Alguns desses movimentos caóticos exibem aleatoriedade, apesar da natureza determinista do sistema.

- **Múltiplos modos de comportamento**

Um sistema não linear pode exibir formas muito diferentes de comportamento, dependendo dos valores dos parâmetros externos. Ou pode até saltar de uma forma de comportamento para outra de forma autônoma. Esses comportamentos não podem ser observados em sistemas lineares.

Esta listagem reúne alguns dos principais fenômenos atribuídos às não linearidades, embora tenham inúmeras outras possibilidades.

A escolha de modelos não lineares traz consigo um inevitável aumento na complexidade dos algoritmos a serem utilizados. Existem razões fortes para, em uma dada aplicação, optar por modelos não lineares como, por exemplo, o fato de os modelos não lineares produzirem certos regimes dinâmicos que os modelos lineares não conseguem representar [2].

Deste modo, a área de identificação tem tido considerável interesse nos últimos anos para fins de previsão, supervisão, diagnóstico e controle. Observa-se a aplicação em diversos campos da engenharia, tais como processos químicos, sistemas biomédicos, sistemas socioeconômicos, sistemas elétricos e outros. Contudo, o enfoque neste trabalho é a identificação de sistemas voltada ao campo de pesquisas na área de monitoramento de processos.

Procurando minimizar a complexidade que envolve esse tipo de sistema, surge a motivação de se buscar formas mais simples e eficazes na identificação e modelagem voltada para o monitoramento desse tipo de processos. Então, observa-se estudos promissores utilizando ferramentas como as redes neurais artificiais, que representam uma parte importante do conhecimento na área de sistemas inteligentes, pela sua ampla aplicabilidade em diversos sistemas, com resultados eficientes na identificação de sistemas complexos e não lineares.

3.2 Redes Neurais Artificiais

Um grande impulso no desenvolvimento de algoritmos é o projeto de modelos de algoritmos para resolver problemas cada vez mais complexos. O homem aprendeu muito com estudos de sistemas naturais, bons resultados foram alcançados através da modelagem da inteligência biológica e natural, chamados assim de sistemas inteligentes [15].

As tentativas de encontrar definições do que seria inteligência ainda provoca grande debate. Os dicionários definem a inteligência como a capacidade de compreender, en-

tender e lucrar com experiência, para interpretar a inteligência, tendo a capacidade de pensamento e da razão. Outras palavras-chave que descrevem aspectos da inteligência incluem criatividade, habilidade, consciência, emoção e intuição [15].

Os computadores podem ser inteligentes? Alan Turing deu muita atenção a esta questão. Ele acreditava que as máquinas poderiam ser criações que imitam os processos do cérebro humano e que não havia nada que o cérebro poderia fazer que um computador bem projetado não pudesse também. Enquanto bons resultados têm sido alcançados na modelagem de pequenas partes de sistemas neurais biológicos, ainda não há soluções para o complexo problema de modelar intuição, consciência e emoção [15].

As redes neurais são um campo em evolução, com origens na neurobiologia. As redes neurais artificiais são uma maneira de criar modelos matemáticos que tentam imitar alguns dos métodos básicos de processamento de informações encontrados no cérebro. Como nossos cérebros realizam tarefas complexas, as redes neurais modeladas também são consideradas úteis na solução de problemas complexos [45].

Uma definição mais ampla de uma rede neural é que ela é uma coleção de neurônios interconectados que aprendem incrementalmente a partir de seu ambiente (dados) para capturar tendências lineares e não lineares essenciais em dados complexos, para fornecer previsões confiáveis para novas situações que contenham até mesmo ruído e informação parcial [45].

Os trabalhos nessa área têm sido motivados desde o começo, pelo reconhecimento de que o cérebro humano processa informações de uma forma diferente do computador digital convencional. O cérebro é um sistema de processamento de informação altamente complexo, não linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar certos processamentos (reconhecimento de padrões, percepção, e controle motor) [18].

Segundo Haykin [18], as principais características do que seja uma RNA são:

Uma rede neural é um processador paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

1. *o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem;*
2. *forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.*

O procedimento usado para realizar o processo de aprendizagem é chamado de algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos da rede de uma maneira ordenada para atingir o objetivo do projeto desejado [18].

A modificação de pesos sinápticos é o método tradicional para o projeto de redes neurais. Este tipo de abordagem é bastante próximo da teoria dos filtros adaptativos lineares, que já está bem estabelecido e com aplicações em diversas áreas [18, 54]. Entretanto, é possível também para uma rede neural modificar sua própria topologia, o que é motivado pelo fato de os neurônios no cérebro humano poderem morrer e que novas conexões sinápticas possam crescer [18].

O sistema nervoso humano pode ser visto como um sistema de três estágios, como apresentado no diagrama em blocos da Figura 3.3. O centro do sistema é o cérebro, representado pela rede neural (nervosa), que recebe continuamente informação, percebe-a e toma decisões apropriadas. Dois conjuntos de setas são mostrados na Figura 3.3. As setas que apontam da esquerda para a direita indicam a transmissão para frente do sinal portador de informação, através do sistema. As setas apontando da direita para a esquerda indicam a presença de realimentação do sistema. Os receptores convertem estímulos do corpo humano ou do ambiente externo em impulsos elétricos que transmitem informação para a rede neural (cérebro). Os atuadores convertem impulsos elétricos gerados pela rede neural em respostas discerníveis como saídas do sistema [18].

Figura 3.3: Representação em Diagrama de Bloco do Sistema Nervoso Humano Visto como Sistema de Três Estágios.



Fonte: Haykin [18].

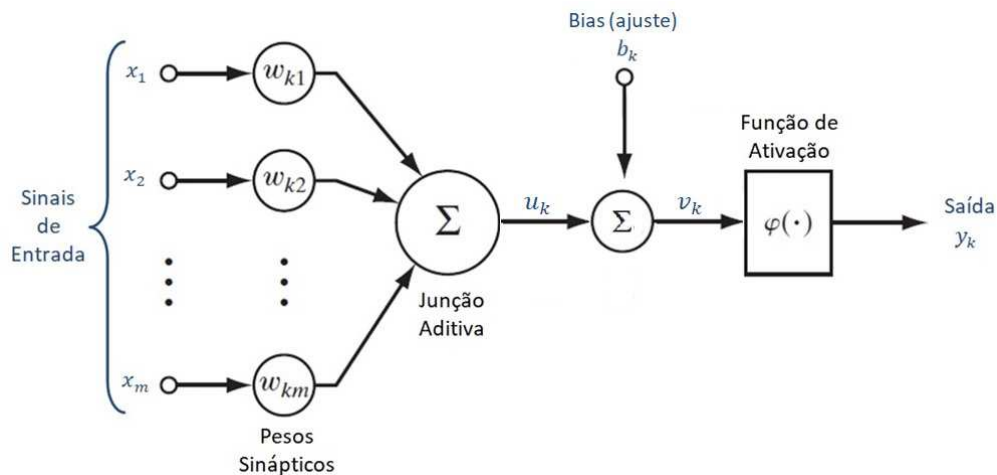
Como foram inspiradas na estrutura e na função de neurônios biológicos, as redes neurais aprendem com a interação de padrões de exemplo, sem necessitarem de um conhecimento das relações entre as variáveis que estão sob investigação. A rede neural artificial é um conjunto de unidades processadoras que simulam neurônios biológicos e são interconectados por um conjunto de pesos (semelhante às conexões sinápticas no sistema nervoso) [18].

As sinapses são unidades estruturais e funcionais elementares que medeiam as interações entre os neurônios. O tipo mais comum de sinapse é a sinapse química, que opera da seguinte forma: um processo pré-sináptico libera uma substância transmissora que se difunde através da junção sináptica entre neurônios e então age sobre um processo pós-sináptico. Assim, uma sinapse converte um sinal elétrico pré-sináptico em um sinal químico e então de volta em um sinal elétrico pós-sináptico. Nas descrições tradicionais da organização neural, assume-se que uma sinapse é uma conexão simples que pode impor ao neurônio receptivo excitação ou inibição, mas não ambas [18].

É importante saber que esses níveis estruturais de organização são uma característica única do cérebro. Eles não são encontrados em lugar algum em um computador digital convencional, e não se está próximo de recriá-los com redes neurais artificiais. Os neurônios artificiais utilizados para construir as redes neurais são realmente primitivos em comparação com aqueles encontrados no cérebro. Apesar disso, com a analogia neurobiológica como fonte de inspiração e com a riqueza das ferramentas teóricas e tecnológicas que se tem acumulado, se tem avançado gradualmente na compreensão das redes neurais artificiais e o seu uso tem gerado resultados satisfatórios [18].

Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural. No diagrama em blocos da Figura 3.4 pode ser observado o modelo de um neurônio, que forma a base para o projeto de redes neurais (artificiais).

Figura 3.4: Modelo de Neurônio Base para Projetos de RNA.



Fonte: Haykin (Modificada) [18].

Aqui se identifica três elementos básicos do modelo neuronal:

1. conjunto de sinapses ou elos de conexão, cada uma se caracteriza por seu peso ou força própria. Um sinal x_j na entrada da sinapse j conectado a um neurônio k é multiplicado pelo peso sináptico w_{kj} , onde o índice k refere-se ao neurônio em questão e o índice j refere-se ao terminal de entrada da sinapse. Ao contrário de uma sinapse do cérebro, o peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos bem como positivos;
2. um somador para somar os sinais de entrada, ponderados pelas respectivas sinapses do neurônio. Estas funções constituem um combinador linear;
3. uma função de ativação para restringir a amplitude da saída de um neurônio. A função de ativação é também referida como função restritiva já que restringe (filtra) o intervalo aceitável de amplitude do sinal de saída a um valor finito. Tipicamente,

o intervalo normalizado da amplitude de saída de um neurônio é escrito como o intervalo unitário fechado $[0,1]$ ou $[-1,1]$.

Em resumo, as entradas são apresentadas e ponderadas pelos pesos correspondentes, somados e transformados através de uma função de ativação para produzir uma saída [45].

Neste modelo, pode ser notado o acréscimo de um *bias* (ajuste), que tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente, identificado por b_k .

Matematicamente, um neurônio k é descrito com as seguintes equações:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (3.1)$$

$$v_k = u_k + b_k \quad (3.2)$$

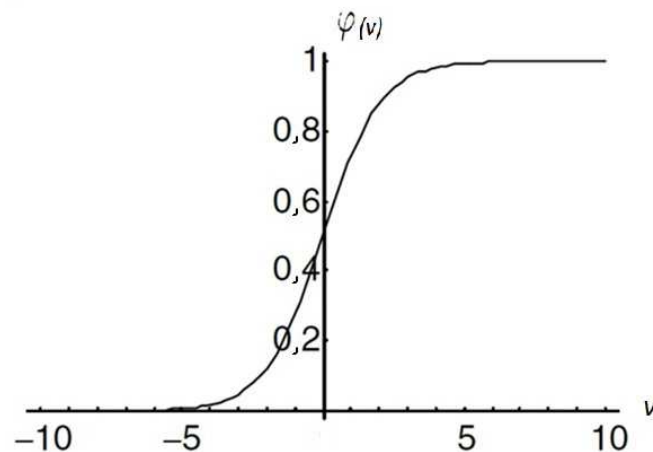
$$y_k = \varphi v_k \quad (3.3)$$

em que $x_1, x_2, x_3, \dots, x_m$, são sinais de entrada; $w_{k1}, w_{k2}, w_{k3}, \dots, w_{km}$, são os pesos sinápticos do neurônio k ; u_k é a saída do combinador linear devido aos sinais de entrada; b_k é o *bias*; v_k é u_k com a aplicação do *bias*; $\varphi(\cdot)$ é a função de ativação; e y_k é o sinal de saída do neurônio. O uso do *bias* b_k é feito para aumentar os graus de liberdade, permitindo uma melhor adaptação, por parte da rede neural, ao conhecimento a ela fornecido. Tem o efeito de aplicar uma transformação na saída u_k do combinador linear, como demonstrado na Equação (3.2).

A função de ativação, definida pelo termo $\varphi(\cdot)$, é responsável pela saída do neurônio, conforme já citado anteriormente na Equação (3.3). Existem alguns tipos básicos de funções de ativação, por exemplo, função de limiar, função linear por partes (*piecewise*) e função sigmóide. A função sigmóide é um dos modelos mais utilizados em projetos de redes neurais [18].

O gráfico desta função (sigmóide), observado na Figura 3.5, forma um "S", é uma função contínua não linear que permanece dentro de alguns limites superiores e inferiores. Não linear significa que a saída da função varia não linearmente com a entrada; esse aspecto permite que redes neurais façam mapeamento não linear entre entradas e saídas [45].

Figura 3.5: Gráfico Função Sigmóide.



Fonte: Sandhya [45].

Algumas regras foram desenvolvidas para os diferentes tipos de aprendizado. Um tipo de aprendizado é o supervisionado, que requer um conjunto de treinamento que consiste em vetores de entrada e um vetor de destino (saída desejada) associado a cada vetor de entrada. O objetivo com o treinamento supervisionado é ajustar os valores de peso de forma que o erro entre a saída real e a saída desejada seja minimizado [15].

Existem diferentes tipos de redes neurais que aprendem sob supervisão, pode-se citar, por exemplo, redes neurais alimentadas adiante (*feedforward*) de múltiplas camadas, redes *feedforward* em cascata e recorrentes.

3.2.1 Tipos de Redes Neurais

Os tipos de redes neurais com aprendizado supervisionado podem ser diferenciados uns dos outros por ter ou não laços de realimentação.

As redes que não possuem laços de realimentação são chamadas de redes neurais alimentadas adiante (*feedforward*). Esse tipo de rede recebe sinais externos e simplesmente propaga esses sinais por todas as camadas para obter o resultado (saída) da rede neural. Não há conexões de realimentação para as conexões anteriores, as conexões são feitas somente para neurônios da camada seguinte [15].

Já as que possuem laços de realimentação são chamadas de recorrentes. As redes neurais recorrentes, por outro lado, têm conexões sinápticas realimentadas (ou laços de realimentação) permitindo o fluxo de sinais de ativação e saída neurais entre neurônios de camadas distintas para modelar as características temporais do problema que está sendo aprendido [15].

As redes *feedforward* têm sido aplicadas com êxito para resolver diversos problemas considerados difíceis, complexos [18]. Visto isto, nesta tese são testadas redes neurais com

aprendizado supervisionado sem realimentação.

Redes Neurais Alimentadas Adiante (*Feedforward*)

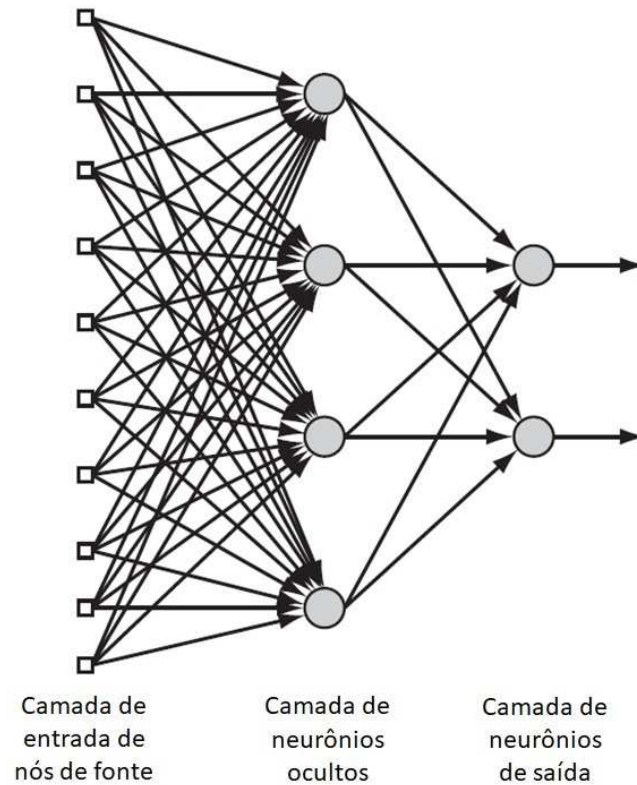
- **Redes Neurais *Feedforward* de Múltiplas Camadas**

As redes alimentadas adiante de múltiplas camadas (*multilayer feedforward networks*) comumente chamadas de perceptron multicamada (*multilayer perceptron - MLP*) são redes bastante difundidas na literatura. Neste tipo de redes há a presença de uma camada de entrada que recebe os sinais, uma camada de saída e uma ou mais camadas ocultas (intermediárias). O termo "oculto" é devido ao fato de que essa parte da rede neural não é vista diretamente da entrada ou saída da rede [18].

Os nós de origem na camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós de computação) na segunda camada (ou seja, a primeira camada oculta). Os sinais da segunda camada são usados como entradas para a terceira camada, e assim por diante, para o resto da rede. Tipicamente, os neurônios em cada camada da rede têm como entradas apenas os sinais de saída da camada precedente. Os sinais de saída dos neurônios na camada de saída (final) da rede constituem a resposta geral da rede ao padrão de ativação fornecido pelos nós de origem na (primeira) camada de entrada.

Na Figura 3.6 é apresentada uma rede neural *feedforward* multicamada para o caso de uma única camada oculta. A rede da Figura 3.6 é referida como uma rede 10-4-2, porque possui 10 nós de origem, 4 neurônios ocultos e 2 neurônios de saída. Por exemplo, uma rede *feedforward* com m nós de origem, h_1 neurônios na primeira camada oculta, h_2 neurônios na segunda camada oculta e q neurônios na camada de saída é referida como uma rede $m-h_1-h_2-q$ [18].

Figura 3.6: Redes Alimentadas Adiante com uma Camada Oculta e uma de Saída.



Fonte: Haykin [18].

A saída de uma rede neural alimentada adiante para qualquer padrão de entrada z_p é calculado como uma única passagem direta pela rede. Para cada unidade de saída o_k , tem-se (assumindo que não há conexões diretas entre as camadas de entrada e saída) [15]:

$$\begin{aligned}
 o_{k,p} &= f_{ok}(net_{o_k,p}) \\
 &= f_{ok}\left(\sum_{j=1}^{J+1} w_{kj} f_{yj}(net_{y_j,p})\right) \\
 &= f_{ok}\left(\sum_{j=1}^{J+1} w_{kj} f_{yj}\left(\sum_{i=1}^{I+1} v_{ji} z_{i,p}\right)\right)
 \end{aligned} \tag{3.4}$$

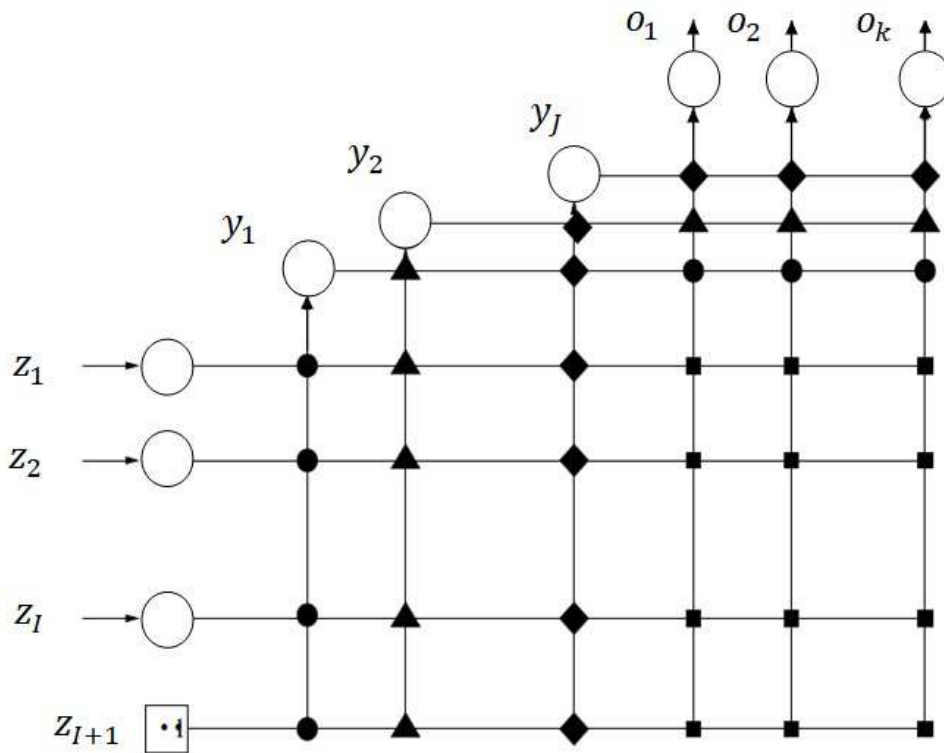
em que f_{ok} e f_{y1} são, respectivamente, a função de ativação para a unidade de saída o_k e a unidade oculta y_j ; w_{kj} é o peso entre a unidade de saída o_k e a unidade oculta y_j ; $z_{i,p}$ é o valor da unidade de entrada z_i do padrão de entrada z_p ; a $(I + 1)^{th}$ unidade de entrada e a $(J + 1)^{th}$ unidade oculta são unidades de polarização que representam os valores de limiar de neurônios na próxima camada.

Uma rede é dita totalmente conectada quando cada um dos nós da rede está conectado a todos os nós da camada adjacente seguinte. Entretanto, se alguns dos elos de comunicação (conexões sinápticas) estiverem faltando na rede, pode-se dizer que ela está parcialmente conectada [18].

• **Redes Neurais Alimentadas Adiante (*Feedforward*) em Cascata**

A rede neural alimentada adiante em cascata é uma variação da rede *Feedforward*, é uma rede multicamada em que todas as unidades de entrada têm conexões diretas com todas as unidades ocultas e com todas as unidades de saída. Além disso, as unidades ocultas estão em cascata. Ou seja, a saída de cada unidade oculta serve como uma entrada para todas as unidades ocultas subsequentes e todas as unidades de saída [15]. Na Figura 3.7 é apresentada uma rede neural em cascata.

Figura 3.7: Rede Neural em Cascata



Fonte: Andries [15].

A saída de uma rede neural em cascata pode ser obtida pela Equação 4.1 [15]:

$$o_{k,p} = f_{ok} \left(\sum_{i=1}^{I+1} u_{ki} z_i + \sum_{j=1}^J w_{kj} f_{yj} \left(\sum_{i=1}^{I+1} v_{ji} z_i + \sum_{l=1}^{j-1} s_{jl} y_l \right) \right) \quad (3.5)$$

em que u_{ki} representa o peso entre a unidade de saída k e a unidade de entrada i , s_{jl} é um peso entre unidades ocultas j e l , e y_l é a ativação da unidade oculta l .

Neste ponto, é importante notar que o treinamento da rede neural em cascata consiste em encontrar valores de peso e o tamanho da rede neural. O treinamento começa com a arquitetura mais simples contendo apenas $(I + 1)K$ pesos diretos entre as unidades de entrada e saída (indicado por quadrado sólido na Figura 3.7) Se a precisão da rede neural em cascata é inaceitável, uma unidade oculta é adicionada, o que adiciona outros $(I + 1)J + (J - 1) + JK$ pesos para a rede. Se $J = 1$, a rede inclui os pesos indicados pelos quadrados e círculos preenchidos na Figura 3.7. Quando $J = 2$, os pesos marcados por triângulos preenchidos são adicionados [15].

Até este ponto, comentou-se como as redes neurais podem ser usadas para calcular um valor de saída dada uma entrada. Alguns tipos de algoritmos podem ser utilizados para o treinamento das redes neurais.

3.2.2 Algoritmos de Treinamento

Existem vários algoritmos para realizar o treinamento de uma rede *Feedforward*. Pode-se citar alguns métodos, por exemplo, os métodos utilizando gradiente descendente com e sem *momentum*, gradiente conjugado e de *quasi-newton*, no qual se pode incluir o método de *Levenberg-Marquardt*. Esses métodos possuem em comum o emprego da retropropagação para o cálculo das derivadas parciais das variáveis de saída da rede em relação aos parâmetros de ajuste ou pesos sinápticos [53].

O treinamento com retropropagação (*backpropagation*) é um algoritmo de aprendizagem supervisionada que utiliza pares (entrada, saída desejada) para, por meio de um mecanismo de correção de erros, ajustar os pesos sinápticos da rede, minimizando o erro quadrático médio (EQM) entre o valor desejado e o valor da saída atual da rede [30].

Neste tipo de treinamento, um sinal de erro é propagado de volta através da rede, começando com os neurônios de saída e se movendo em direção aos neurônios de entrada. As sinapses são então modificadas com base na atividade neural e no sinal de erro retropropagado [24].

Então, existem duas fases para o treinamento, cada uma em um sentido da rede. A fase de propagação (*forward*), em que é realizado o cálculo das saídas e seus respectivos erros e a fase de retropropagação (*backward*), em que é realizada a atualização dos pesos sinápticos de suas conexões [30].

Uma representação completa do conjunto de treinamento é denominada *época*. O processo de aprendizagem é repetido de época em época até o erro sobre todo o conjunto de treinamento convergir para um valor mínimo, ou até que o número de iterações, determinado anteriormente, tenha sido atingido [18].

O treinamento com retropropagação pode ser realizado por algoritmos diferentes. Na

Tabela 3.1 estão apresentados alguns algoritmos de treinamento, sendo diferenciados pela estratégia de otimização utilizada para realizar os ajustes nos pesos e *bias* da rede neural, como também a sigla que representa cada um. As funções de treinamento apresentadas precisam do algoritmo com retropropagação (*backpropagation*). Como nesta tese utilizou-se o *software* MATLAB nas simulações para o projeto das redes neurais, também estão apresentadas na Tabela 3.1 as funções do MATLAB referentes aos métodos citados.

Tabela 3.1: Algoritmos de Treinamento e Seus Métodos de Otimização.

Método	Algoritmo de Treinamento	Sigla	Função MATLAB
Gradiente Conjugado (2° ordem)	Gradiente conjugado escalonado (Moller, 1993)	SCG	<i>trainscg</i>
quasi-Newton (2° ordem)	Levenberg-Marquardt (Hagan and Menhaj, 1994)	LM	<i>trainlm</i>
Gradiente Conjugado e <i>quasi-Newton</i> (2° ordem)	Método da secante de um passo (Battiti, 1992)	OSS	<i>trainoss</i>

Fonte: Werle (Modificada) [53].

O *backpropagation* padrão emprega o gradiente descendente como método de aproximação do mínimo da função erro, o algoritmo de *Levenberg-Marquardt* (LM) utiliza uma aproximação pelo método de Newton, obtida a partir da modificação do método de Gauss-Newton [25].

Métodos do tipo *quasi-newton* ou secante é uma classe de algoritmos que é baseada nos métodos de Newton. O método *Levenberg-Marquardt* é considerado do tipo *quasi-newton*. Foi incorporado ao algoritmo de retropropagação do erro para resolver problemas de otimização que aparecem no treinamento de redes multicamadas [53].

O método de Gradiente Conjugado é um método alternativo aos métodos de *quasi-newton* e gradiente descendente. É projetado para exigir menos cálculos que o método de Newton e apresentar taxas de convergência maiores que as do método do gradiente. Como exemplo, pode-se citar o algoritmo Gradiente Conjugado em Escala (SCG) [53].

Como os algoritmos de *quasi-newton* requerem mais armazenamento e recurso computacional em cada iteração que os algoritmos de gradiente conjugado, existe a necessidade de uma aproximação secante com menor armazenamento e computação. O método de secante em um passo (*One Step Secant - OSS*) é uma tentativa intermediária entre os algoritmos de gradiente conjugado e os algoritmos *quasi-newton*. Neste método, o termo

secante provém do fato de que as derivadas são aproximadas por secantes avaliadas em dois pontos da função, onde neste caso a função é o gradiente [53].

São realizadas comparações do comportamento de diferentes algoritmos de treinamento, com o objetivo de contribuir para as pesquisas na área, além de encontrar o algoritmo que mais se adapte ao conjunto de dados para a implementação dos sensores virtuais para o monitoramento da plataforma em estudo seguindo a metodologia proposta.

Nos últimos anos, aumentou a quantidade de trabalhos que tratam do desenvolvimento de sensores virtuais que utilizam redes neurais artificiais. Isto acontece devido aos bons resultados apresentados quando essa tecnologia é aplicada a sistemas complexos e não lineares, como é o caso do transporte fluídico em tubulações. As redes neurais podem ser aplicadas em diversas áreas, assim como, para o que é de interesse desta tese, no monitoramento de processos, utilizando a linha preditiva.

3.3 Sensores Virtuais

O sensor é um dispositivo que responde a estímulos, fazendo com que a variável de interesse seja expressada de outra forma, em uma variável correspondente mensurável, como deslocamento, pressão ou tensão elétrica, por exemplo [41]. As medições de um sensor real podem ficar indisponíveis devido a falhas, remoção para manutenção ou reparos. O uso de sensores virtuais pode ser uma boa alternativa para solucionar esse tipo de problema, com a possibilidade de inferir valores dessas variáveis que estejam indisponíveis [17].

Os sensores virtuais são modelos matemáticos que permitem inferir variáveis relevantes de interesse (não medidas) com base em sua dependência de um conjunto de outras variáveis (disponíveis medidas no processo) que se relacionam com as variáveis de interesse que por algum motivo não estão sendo medidas diretamente [35].

A utilização do conceito de sensores virtuais já é estudada e está ligada à modelagem matemática de processos com abordagem na predição de propriedades a partir de dados da planta. Os primeiros sensores virtuais surgiram a partir de sistemas de controle indiretos, nos quais a variável a ser controlada (primária) era controlada através do comportamento de outras variáveis (secundárias). Um dos primeiros sensores virtuais desenvolvidos foi o estimador inferencial de *Brosilow*, apresentado no final da década de 70 [21].

A ideia do uso do sensor virtual é a existência de situações em que haja algum impedimento do uso de sensores reais, podendo acontecer por falha ou remoção para manutenção, por exemplo. Nestes casos, o sensor virtual poderia agir substituindo a indisponibilidade (momentânea ou permanente) do sensor real na planta [34].

O sensor virtual pode ser considerado como o resultado da associação das técnicas de modelagem e identificação de sistemas e da tecnologia de instrumentos inteligentes, que

aliados a sistemas digitais como microprocessadores ou microcontroladores, modificam seu comportamento, operando computacionalmente as informações medidas para melhor se adaptar à coleta e manipulação dos dados de um processo transmitindo-os da melhor forma possível [34].

Essa associação entre sensor e modelo é uma ideia fundamental do sensor virtual. Podemos separar essa ideia em duas partes distintas: a parte do sensor real ou conjunto desses sensores, que adquirem variáveis relacionadas com a variável desejada e a parte do sensor virtual, o modelo desta relação, geralmente implementado em *software*, capaz de fornecer, por meio de simulação, uma estimativa da variável desejada [34].

Os sensores virtuais possuem um grande número de propriedades atrativas, dentre elas podem ser destacadas as seguintes [35]:

- podem trabalhar de forma paralela com sensores reais, dando informações úteis para a detecção de falhas, permitindo assim uma operação mais confiável do processo;
- podem ser implementados no *hardware* existente, e podem retornar os parâmetros do sistema quando se faz necessária alguma modificação;
- são capazes de fazer estimativas de dados em tempo real, superando os atrasos introduzidos por meio de sensores reais lentos, melhorando assim o desempenho das estratégias de controle.

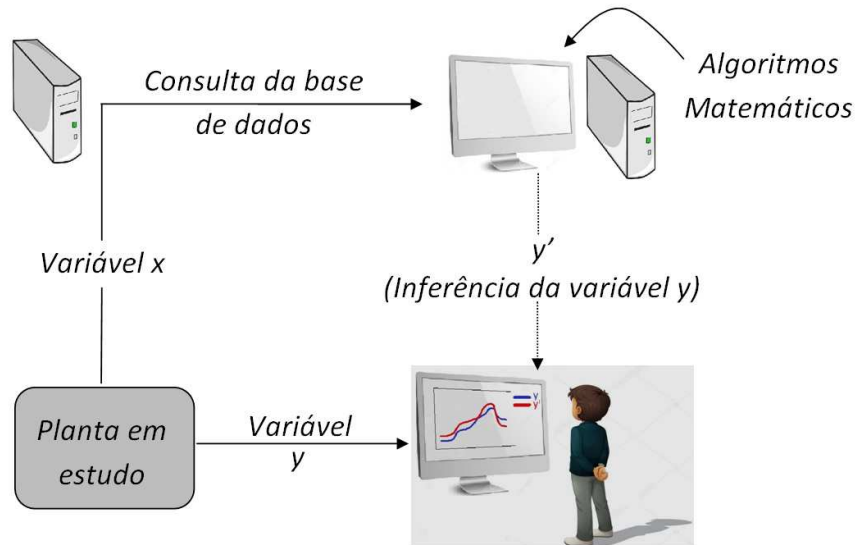
Pode-se dividir a construção de um sensor virtual em cinco etapas, como sendo [34]:

1. aquisição e armazenamento de dados do processo;
2. pré-processamento dos dados, com o objetivo de eliminar em grande parte a presença de ruído e *outliers* (dados fora do padrão);
3. escolha das variáveis a serem usadas e construção do modelo;
4. validação do modelo por meio de um conjunto de dados independente;
5. aplicação do modelo na planta e monitoramento *online*.

Na Figura 3.8 é apresentado um diagrama do funcionamento de um sensor virtual de forma simplificada. A variável y é a saída real do processo. Se acontecer alguma falha que faça com que esta variável y não possa ser medida diretamente, é possível utilizar a variável x (outra variável medida do processo que se relaciona com a variável de interesse) para realizar a inferência do valor de interesse que está indisponível. Consultando um banco de dados histórico do processo e, utilizando algoritmos matemáticos de inferência,

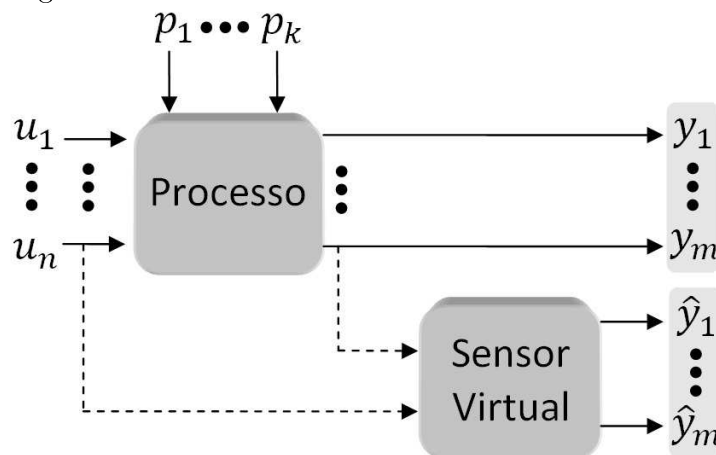
é possível realizar cálculos para gerar uma estimativa da variável y . Este valor inferido é chamado de y' na Figura 3.8.

Figura 3.8: Diagrama do Processo de Funcionamento do Sensor Virtual.



O objetivo nesta tese é utilizar os sensores virtuais funcionando em paralelo com os sensores reais, de forma redundante, podendo substituir os sensores reais caso venham a falhar. Na Figura 3.9 pode ser observado o diagrama de um processo com sensor virtual conectado paralelamente. Na Figura, as variáveis $u_1 \dots u_n$ são as entradas, $p_1 \dots p_k$ são perturbações, $y_1 \dots y_m$ os sinais de saídas dos sensores reais e $\hat{y}_1 \dots \hat{y}_m$ os sinais de saídas estimados (sensores virtuais).

Figura 3.9: Sensor Virtual Conectado a uma Planta.



As setas pontilhadas são para indicar que as entradas e saídas do processo são variáveis utilizadas no desenvolvimento dos sensores virtuais. Com as entradas $u_1 \dots u_n$, as saídas $y_1 \dots y_m$ e a base de dados histórica armazenada do processo (relação de entradas e saídas que se relacionam no processo), o sensor virtual infere os valores de $\hat{y}_1 \dots \hat{y}_m$. Nesta tese,

para o desenvolvimento dos sensores virtuais, é utilizado um processo com duas entradas e duas saídas. Mas este número não necessariamente precisa ser o mesmo, pode-se ter processos com diferentes números de entradas, saídas e perturbações. Além das entradas e saídas do processos, válvulas manuais existentes na plataforma de teste (apresentada no Capítulo 4) são utilizadas para gerar perturbações no sistema e avaliar o desempenho dos sensores virtuais desenvolvidos.

Técnicas de modelagem específicas para sensores virtuais usando redes neurais constituem um interessante desenvolvimento a ser pesquisado. Em geral, a área de sensores virtuais é um importante campo para pesquisa, que pode conduzir a aplicações industriais, fornecendo monitoramento confiável e aumentando a robustez das malhas de controle [34].

3.4 Monitoramento de Processos

O monitoramento de processos é importante em operações de sistemas automatizados. É responsável pela observação e identificação de problemas que ocorram prejudicando o desempenho do processo. O monitoramento de processo industrial permite uma ação focada e eficaz. Então, torna-se uma importante solução, aumentando significativamente a produção e reduzindo problemas que afetam o processo [48].

A tendência nos sistemas de engenharia é aumentar sua complexidade, principalmente em virtude da necessidade de realizar tarefas complexas e de alta precisão. À medida que esses sistemas, com muitas entradas e saídas, se tornam mais complexos, a descrição do sistema requer um grande número de equações [41]. Essa crescente complexidade e grau de integração nesses sistemas aumentou a possibilidade da ocorrência de falhas acontecendo localmente em algum ponto do sistema e que possam ter seus efeitos propagados para outras partes, tornando maiores as consequências [48].

A motivação para o projeto de um sistema de monitoramento de processos bem sucedido é clara; se não forem verificadas, falhas podem causar problemas que variam de impacto, desde incômodos até desastres [48].

É necessário ser feita a identificação de sistemas para realizar o monitoramento. Os métodos por inferência são amplamente difundidos nas indústrias, mas uma quantidade limitada de técnicas são utilizadas para seu desenvolvimento [53]. Neste trabalho, utiliza-se as redes neurais artificiais no processo de identificação de sistemas para realizar inferências para o monitoramento de processos.

Esses métodos utilizando redes neurais, voltados para a inferência de valores (sistemas de inferência) podem atuar auxiliando no monitoramento e otimização de plantas industriais. Fornecem medições mais rápidas e confiáveis a um custo reduzido, quando

comparado a métodos tradicionais de análises [53]. Os métodos de caracterização compostos por sistemas inferenciais, são ferramentas valiosas para serem empregadas em diversos setores da indústria, tais como refinarias, plantas químicas e processamento de alimentos, dentre outras.

Com os temas abordados, é possível observar a complexidade de se identificar e modelar sistemas não lineares complexos; o avanço do uso dos sensores virtuais e das redes neurais artificiais com o objetivo de estimar valores para o monitoramento de processos, motivando assim, a construção desta tese, que tem como objetivo principal apresentar uma metodologia para desenvolvimento de sensores virtuais para monitoramento de variáveis de processos complexos não lineares e com múltiplas entradas e múltiplas saídas, utilizando redes neurais nas estimativas das variáveis.

No capítulo 4 é apresentada a descrição da plataforma de testes (sistema de transporte fluídico) utilizada no trabalho.

Capítulo 4

Descrição da Plataforma de Teste (Sistema de Transporte Fluídico)

O processo de incrustação é o acúmulo e depósito de materiais orgânicos e/ou inorgânicos em superfícies. Este processo ocorre comumente no interior das tubulações utilizadas, principalmente, nas indústrias química, petrolífera, alimentícia e farmacêutica. Tal fato torna-se um grande problema, pois a incrustação reduz o diâmetro interno do tubo e consequentemente aumenta a pressão interna, podendo ocasionar o entupimento ou até mesmo o rompimento da tubulação. Além disso, aumenta o consumo de energia e o custo com a manutenção [8]. Fatores inerentes a este processo são as variações de grandezas físicas (temperatura, pressão e vazão) e químicas (natureza do material incrustante).

Logo, o monitoramento de um sistema de transporte fluídico baseado nas variáveis de processo de incrustação é uma das soluções consideradas para a redução e a prevenção deste fenômeno, de forma a entender e quantificar o papel das várias condições operacionais no sistema com incrustação [37].

Embora quantificar a quantidade de incrustação no interior de tubulações em processos industriais seja um fator importante, não é o objetivo nesta tese. Nesta tese é realizado o monitoramento das variáveis de interesse (valores de pressão e vazão) no processo de transporte fluídico. Essas variáveis podem sofrer alterações caso exista um aumento de incrustação, e o monitoramento é realizado justamente para o acompanhamento dos valores dessas variáveis, para se garantir que o processo esteja funcionando dentro do esperado, mas a conexão com o valor da quantidade de incrustação não é realizada.

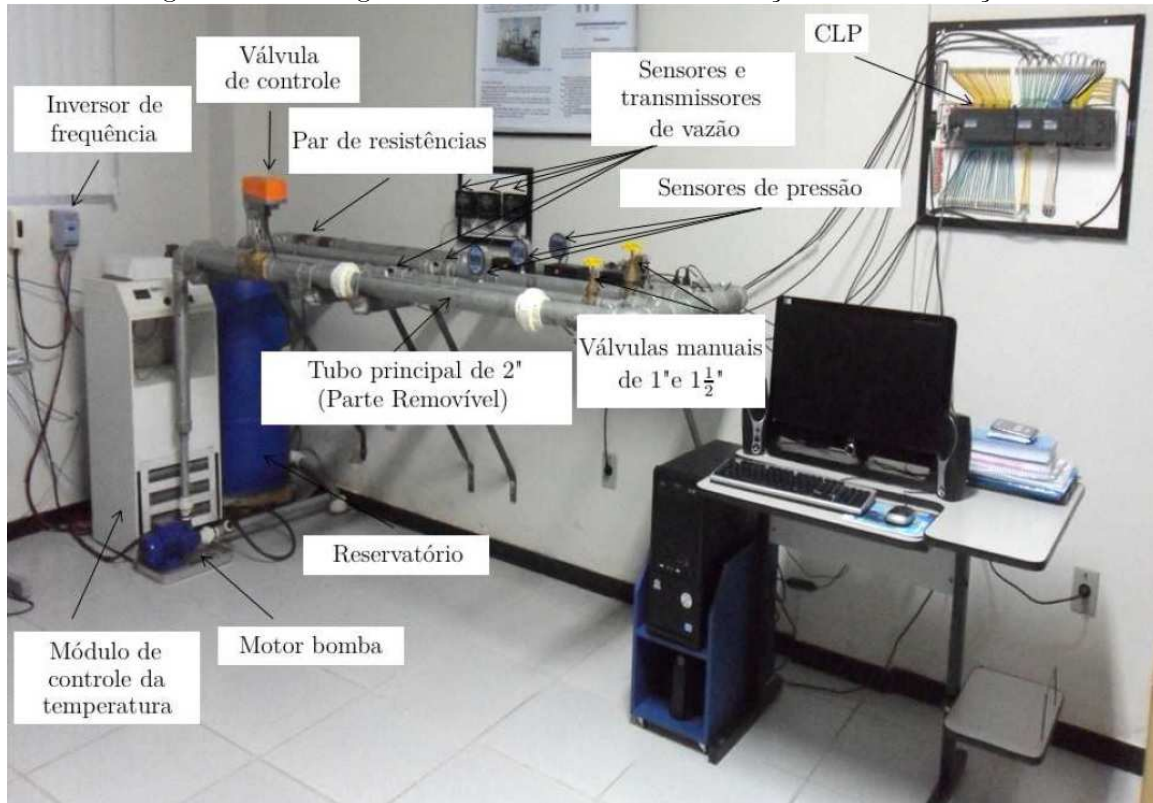
A plataforma utilizada nos experimentos é uma plataforma experimental de transporte fluídico em uma tubulação, que está localizada no Laboratório de Instrumentação Eletrônica e Controle (LIEC), na Universidade Federal de Campina Grande (UFCG).

A plataforma é composta de sensores, para adquirir os dados de pressão e vazão, atuadores, para receber um sinal proveniente do controlador e agir sobre o sistema e o

Controlador Lógico Programável (CLP), que processa os dados adquiridos por meio de uma linguagem própria (*Ladder*) e que são visualizados por meio da interface homem-máquina (IHM).

Uma fotografia da plataforma de testes é apresentada na Figura 4.1.

Figura 4.1: Fotografia da Plataforma de Detecção de Incrustação.



Fonte: Melo *et al.* [10].

A plataforma possui três tipos de sensores. São eles o sensor de vazão, o sensor de pressão e o sensor de temperatura. Comentaremos apenas sobre o sensor de pressão e o de vazão, que são as variáveis de interesse para este trabalho.

A plataforma é caracterizada como um sistema de monitoramento com tubos de ferro galvanizado de diferentes diâmetros, 1", 1 $\frac{1}{2}$ " e 2", cada um com seus sensores de vazão e de pressão, respectivamente (neste trabalho, foi observado o tubo de 2" de diâmetro). Utilizou-se sensores de pressão e de vazão, além de atuadores, no intuito de realizar o monitoramento da plataforma de transporte fluídico nos tubos de 2" de diâmetro.

Quanto à comunicação, o CLP se comunica com os outros dispositivos da planta por uma malha de corrente com sinal na faixa de 4 a 20 mA [40].

Cada tubo possui um sensor de vazão do tipo *Turbine flowmeters* e um de pressão do tipo *StrainGauge*. Os de vazão medem a velocidade de escoamento do fluido de acordo com a frequência de pulsos produzidos devido à rotação da hélice instalada na direção da vazão. A eles conectados existem transmissores da medida com painel de visualização fornecendo

leituras na escala de vazão de 3 a 38 LPM (litros por minuto) [39, 49]. Já o sensor de pressão do tipo *Strain Gauge* baseia-se no princípio de variação da resistência de um fio, em que através da interligação das quatro tiras extensiométricas constituintes em um circuito tipo ponte de *Wheatstone* ajustada e balanceada para condição inicial, consegue medir a pressão por meio do desbalanceamento proporcional à variação da resistência de cada fita [4, 55].

No tubo principal há uma válvula de controle que pode ser acionada remotamente através do CLP, ela é considerada um dos atuadores do sistema. Além da válvula de controle, há um inversor de frequência que controla a velocidade de rotação do rotor da bomba centrífuga.

O fluido utilizado no processo (água) é armazenado em um reservatório de 100 litros. No sistema de transporte fluídico em estudo, tem-se que os dados adquiridos pelos sensores são enviados às entradas do módulo EM235 (formado por conversores A/D, que convertem um sinal analógico de entrada de tensão (0 a 10 V), ou corrente (4 a 20 mA), em um valor digital). Tais dados são processados pela CPU do CLP em um programa escrito em Ladder e visualizados no microcomputador através de uma interface homem-máquina (IHM), implementada no *software LabVIEW* e funcionando via padrão OPC (*OLE for Process Control*). Este padrão, através da relação servidor-cliente, permite que o *LabVIEW* tenha acesso às variáveis de interesse do código em *Ladder*, desde que o servidor OPC seja configurado corretamente com as especificações do CLP.

Para realizar a atuação no sistema, a frequência de operação da bomba pode ser definida pelo usuário via IHM, enviando-se ao inversor de frequência um sinal com valores na faixa de 0 a 10 V pela saída do módulo EM232 do CLP (para cada frequência definida, tem-se um valor de tensão referente), correspondente ao valor de frequência especificado pelo usuário na interface.

Da mesma forma, o estado da válvula de controle pode ser alterado via IHM, enviando-se para o atuador um sinal na faixa de 4-20 mA pela saída do módulo EM235 do CLP (Para cada posição de abertura da válvula, tem-se uma corrente referente). Esta alteração também pode ser acompanhada na interface, já que o sinal de *feedback* da válvula de controle, presente na escala de 1-5 V, é uma das entradas do módulo EM235.

Assim, nesta tese considera-se quatro variáveis, duas entradas e duas saídas. A posição da válvula de controle e a frequência de operação da bomba são utilizadas como entradas do processo, atuando no sistema. E a vazão e pressão medidas na tubulação, são as saídas do sistema, sendo as variáveis de interesse para o monitoramento e controle do processo.

Para um melhor entendimento físico sobre o sistema de transporte fluídico montado, é apresentado um estudo na área de mecânica dos fluidos para análise do comportamento hidrodinâmico da plataforma, realizado anteriormente por Melo [10].

Comportamento Hidrodinâmico na Plataforma Alvo

Quando um fluido viscoso escoar no interior de um conduto, seja ele forçado ou não, há o surgimento de forças cisalhantes exercidas pela face interna da parede do conduto na camada limite do fluido, que reduzem a capacidade de fluidez ao longo do escoamento. O fluido, ao escoar, dissipa parte de sua energia, principalmente sob a forma de calor. Essa perda de energia hidráulica é denominada de perda de carga, que pode ser dividida em:

- Perda localizada (devido à presença de acessórios em pontos particulares do conduto, tais como ampliações, reduções, curvas, válvulas com área transversal não constante);
- Perda distribuída (devido ao atrito do fluido com as paredes do conduto, ao longo de toda a sua extensão, com área transversal constante) [14].

Sendo assim, a perda de carga total do sistema fluídico h_{1T} resulta da soma das perdas distribuída (h_1) e localizada (h_{1m}) na tubulação, podendo ser expressa pela Equação (4.1) [16].

$$h_{1T} = h_1 + h_{1m} = \left(f \cdot \frac{L}{D} + \sum_{i=0}^u k_i \right) \frac{\bar{V}^2}{2g} \quad (4.1)$$

em que: h_{1T} é a perda de carga total [*m.c.a = metros de coluna d'água*]; f é o coeficiente de atrito de Darcy-Weisbach [*adimensional*]; L é o comprimento da tubulação [*m*]; D é o diâmetro da tubulação [*m*]; k_i é o coeficiente de perda de carga cinética característico do acessório [*adimensional*]; u é o número de coeficientes de perda de carga cinética característico (cada tipo de acessório tem um); g é a aceleração da gravidade [*m/s²*] e \bar{V} é a velocidade média de escoamento [*m/s*] obtida pela Equação (4.2).

$$\bar{V} = \frac{Q}{A} = \frac{Q}{\pi R^2} \quad (4.2)$$

sendo Q a vazão medida [*m³/s*] e R é o raio do tubo [*m*].

O escoamento de um fluido pode ocorrer em regime laminar ou turbulento, existindo diferenças qualitativas e quantitativas entre eles. Quantitativamente, o parâmetro que determina o tipo de regime de escoamento é o número de Reynolds, calculado com a Equação (4.3).

$$Re = \frac{\rho D \bar{V}}{\mu} \quad (4.3)$$

em que: Re é o número de Reynolds [*adimensional*]; ρ é a massa específica do fluido [*kg/m³*]; e μ é a viscosidade dinâmica do fluido [*Pa . s*].

Dependendo de qual seja o regime de escoamento, o coeficiente de atrito é obtido pelas Equação (4.4) ou Equação (4.5):

- regime laminar [16]:

$$f_{laminar} = \frac{64}{Re} \quad (4.4)$$

- regime turbulento: Descrita pela equação de Colebrook-White na forma explícita, proposta por Haaland em 1983 e com desvios relativos inferiores a $\pm 1,5\%$ [11]

$$\frac{1}{\sqrt{f_{turbulento}}} = -1,8 \cdot \log \left[\frac{6,9}{Re} + \left(\frac{\varepsilon}{3,7D} \right)^{1,11} \right] \quad (4.5)$$

em que: $\frac{\varepsilon}{D}$ = rugosidade relativa do fluido.

Entre as regiões laminar e turbulenta existe uma zona de transição, na qual o escoamento tem características alternadamente laminares e turbulentas. A gama de valores de número de Reynolds correspondentes a esta região de transição pode ser muito extensa, variando entre 2300 (valor usualmente aceito para projetos com condutos de superfície rugosa) a 4000. Além disso, o coeficiente f é impreciso nessa faixa, não podendo ser predito, devendo ser usada a solução gráfica. Para fluidos Newtonianos, emprega-se o Diagrama de Moody para estimar tal valor [33, 42].

Utilizando-se as Equações (4.1) a (4.5) e os valores pré-definidos dos coeficientes de perda de carga cinética de cada acessório presente na plataforma, foi implementada uma rotina no MATLAB que permitiu a caracterização gráfica em escala logarítmica do comportamento hidrodinâmico do sistema. Na plataforma de testes, foram utilizados os seguintes valores como parâmetros [10]:

Tubulação principal (Tubo de 2")

- diâmetro do tubo: $D = 0,0508 \text{ m}$;
- rugosidade absoluta do tubo: $\varepsilon = 1,5 \cdot 10^{-4} \text{ m}$ (valor referente a tubos de ferro galvanizado);
- comprimento total da tubulação: $L = 5,12 \text{ m}$;

Fluido (Água)

- massa específica da água: $\rho_{\acute{a}gua} = 10^3 \text{ kg/m}^3$;
- viscosidade dinâmica da água: $\mu_{\acute{a}gua} = 1,01 \cdot 10^{-3} \text{ Pa} \cdot \text{s}$

Acessórios (Tubulação principal) [28]

- joelho de 90° padrão (4 unid.): $k_1 = 0,90$;
- união (2 Unid.): $k_2 = 0,40$;
- tê com derivação fechada (2 unid.): $k_3 = 0,60$;
- válvula globo totalmente aberta (1 unid.): $k_4 = 10,0$;
- ampliação gradual (2 unid.): $k_5 = 0,30$;
- redução gradual (1 unid.): $k_6 = 0,15$.

O valor adequado do *set-point* está associado à determinação experimental da zona de transição do regime laminar para o regime turbulento durante o escoamento do fluido na plataforma de testes. Dessa forma, realiza-se um ensaio, iniciando o sistema com um escoamento de baixa velocidade média e aumentando-a gradualmente por meio da variação da frequência do inversor de frequência. Ao atingir a frequência máxima de operação da bomba (no caso, 60 Hz), retorna-se gradualmente ao modo de escoamento inicial de maneira análoga [10].

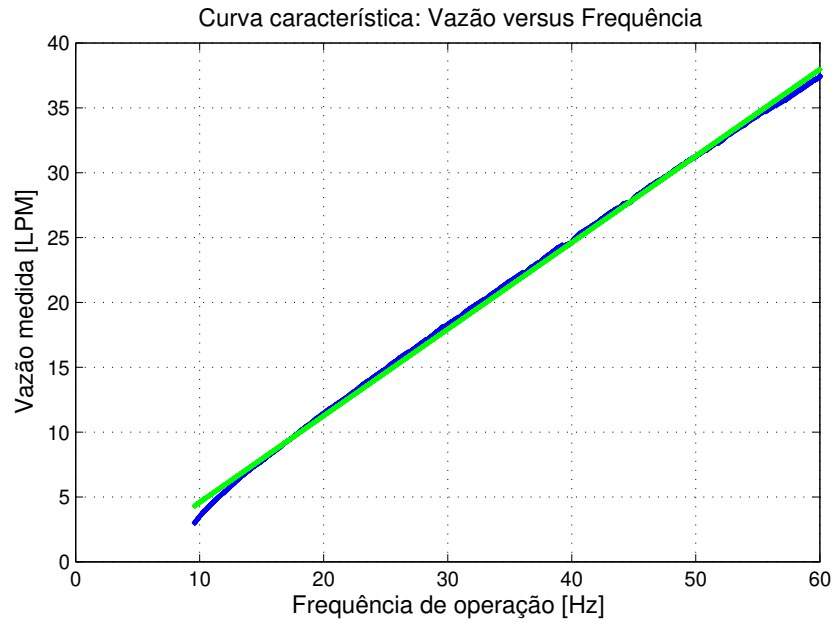
Foi realizado um ensaio para o levantamento da curva característica, e consequentemente, da lei de formação, que expressasse a relação entre a vazão medida no tubo principal e a frequência de operação da bomba utilizada. Para cada incremento de 0,02 Hz, obteve-se uma respectiva medição de vazão, de forma a contemplar aproximadamente os fundos de escala do sensor de vazão e do inversor de frequência. Dessa maneira, fez-se a aquisição de 2608 medidas de vazão e armazenou-se em um arquivo *.txt* gerado pelo bloco *Write LabVIEW Measurement File*, presente no diagrama de blocos da Interface Homem Máquina (IHM) implementada via LabVIEW, que aplicadas na função *polyfit* e *plot*, permitiram obter os coeficientes do polinômio estimado, representado na Equação (4.6), e traçar a curva característica experimental, respectivamente [10].

$$vaz = 0,6597 * freq - 2,0131 \quad (4.6)$$

em que: *vaz* é a vazão medida [$LPM = litros por min$], tendo um desvio de aproximadamente $\pm 0,5$ com relação ao valor real medido; *freq* é a frequência de operação [Hz].

Por meio do polinômio estimado aplicado na função *polyval*, pode-se confirmar a linearidade existente entre os parâmetros, como pode ser observado na Figura 4.2.

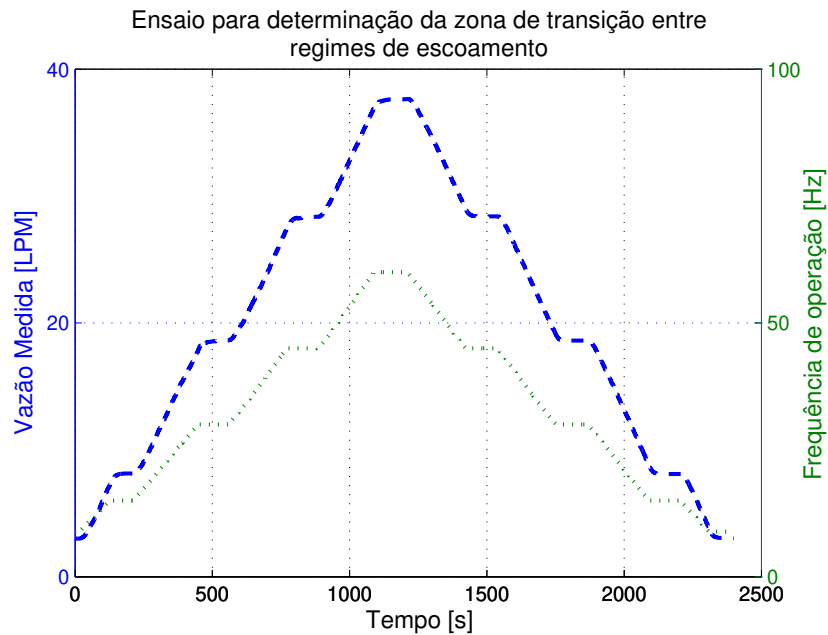
Figura 4.2: Curva Característica Estimada por Ajuste Polinomial (Traço Superior) Sobreposta pela Curva característica Experimental Vazão *versus* Frequência (Traço Sobreposto).



Fonte: Melo *et al.* [10].

Em relação ao ensaio de determinação da zona de transição entre os regimes, iniciou-se com uma frequência de 15 Hz, aumentou-se a frequência com um passo de mesmo valor (0,02 Hz) até o valor máximo e em seguida diminuiu-se no mesmo passo até o valor inicial, armazenando-se todos os valores de vazão obtidos. Em um intervalo de 2400 segundos, o ensaio foi realizado completamente. A curva experimental obtida é apresentada na Figura 4.3 [10].

Figura 4.3: Curva experimental Vazão *versus* Tempo (linha tracejada) e Frequência *versus* Tempo (linha tracejada e pontilhada).

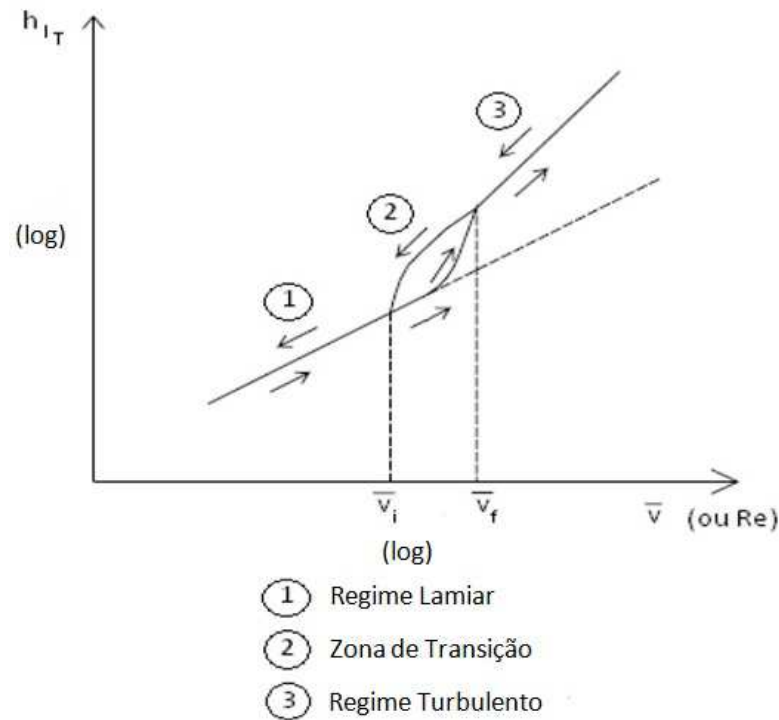


Fonte: Melo *et al.* [10].

De posse dos valores de vazão (ou de frequência, utilizando a Equação (4.6)), calculou-se as velocidades médias de escoamento e os números de Reynolds através da Equação (4.2) e Equação (4.3), respectivamente. Posteriormente, foram calculados os coeficientes de atrito relacionados a cada região de escoamento e conseqüentemente, as perdas de carga do sistema. No caso da zona de transição, como o coeficiente de atrito não pode ser predito, utilizou-se o Diagrama de Moody, no qual o valor estimado foi igual a 0,046, que está associado graficamente ao valor da rugosidade relativa para tubos de ferro galvanizado [10].

A análise do comportamento hidrodinâmico dá-se pelo gráfico logarítmico $h_{lt}X\bar{V}$, em que segundo Lopes *et al.* [33], ocorrerá um fenômeno de histerese devido a perda de carga nesta zona também depender do sentido de variação da velocidade de escoamento, como observado na Figura 4.4.

Figura 4.4: Gráfico Logarítmico $h_{1T}X\bar{V}$, com Destaque as Velocidades Inicial (\bar{V}_i) e Final (\bar{V}_f) da Zona de Transição.

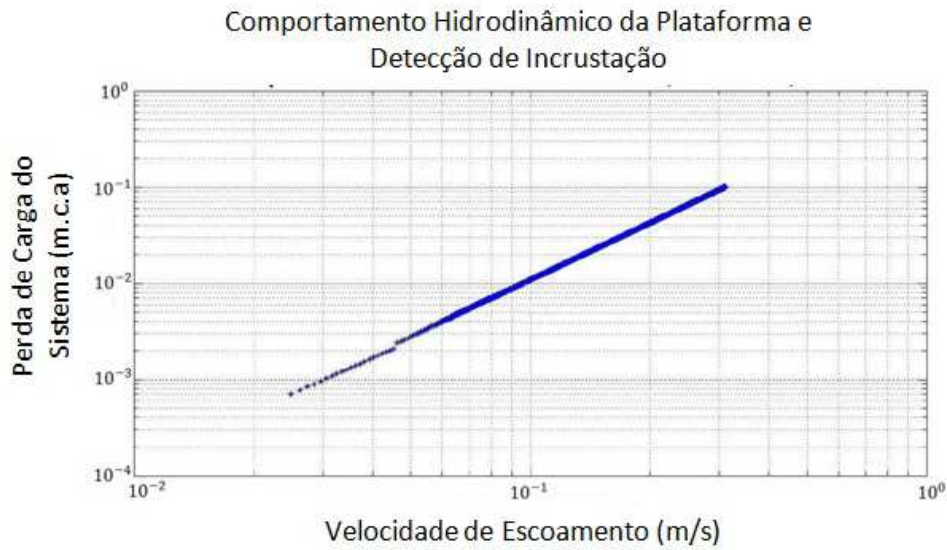


Fonte: Melo *et al.* [10].

Porém, tal gráfico foi gerado por variações na abertura de uma válvula de regulação acoplada à bomba centrífuga. Por isso, esse método de determinação da zona de transição foi usado com modificações, ou seja, com variações na frequência do inversor para se obter um resultado próximo do esperado [10].

O gráfico logarítmico da plataforma de testes pode ser observado na Figura 4.5. Como não é utilizada uma válvula de regulação no ensaio, a perda de carga, por sua vez, não terá adicionado um coeficiente correspondente a cada posição de abertura da válvula, que favorece a formação acentuada da histerese. Por isso, a curva de histerese torna-se uma curva praticamente sobreposta quando usada a frequência, já que esta não altera a quantidade de coeficientes na medida em que varia [10].

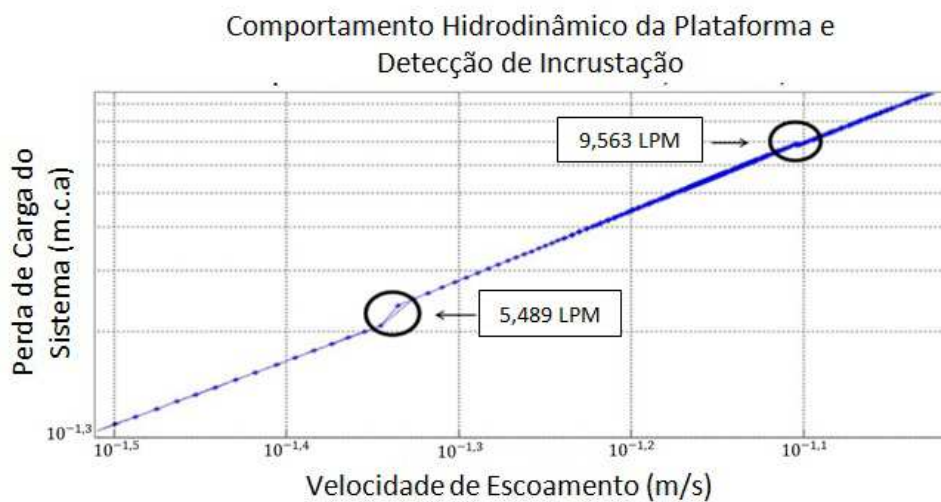
Figura 4.5: Gráfico Logarítmico $h_{lt}X\bar{V}$ (Comportamento Hidrodinâmico da Plataforma de Testes).



Fonte: Melo *et al.* [10].

Em relação a Figura 4.5, pode-se observar que na medida em que o escoamento do fluido acontece, surge uma pequena região de descontinuidade que representa o início da zona de transição entre o escoamento laminar e o turbulento. Fazendo uma ampliação no gráfico da Figura 4.5, nota-se também o fim da zona de transição, menos evidente que o início (Figura 4.6). As velocidades médias de escoamento críticas associadas à zona são iguais a $0,0451 \text{ m/s}$ (velocidade crítica inicial) e a $0,0786 \text{ m/s}$ (velocidade crítica final), fornecendo, respectivamente, valores de vazão iguais $5,489$ e $9,563 \text{ LPM}$ [10].

Figura 4.6: Ampliação da Figura 4.5 para Verificar a Zona de Transição de Regimes.



Fonte: Melo *et al.* [10].

A partir da análise realizada e dos resultados observados, pode-se concluir o quanto

é importante conhecer o funcionamento de um sistema a ser monitorado e controlado, pois sabendo previamente suas particularidades, é possível realizar uma implementação do monitoramento e controle que melhor atenda às especificações do projeto.

No capítulo seguinte são apresentadas a metodologia proposta para o desenvolvimento dos sensores virtuais, a interface gráfica desenvolvida, as simulações e os testes experimentais realizados.

Capítulo 5

Sensores Virtuais Desenvolvidos e Resultados

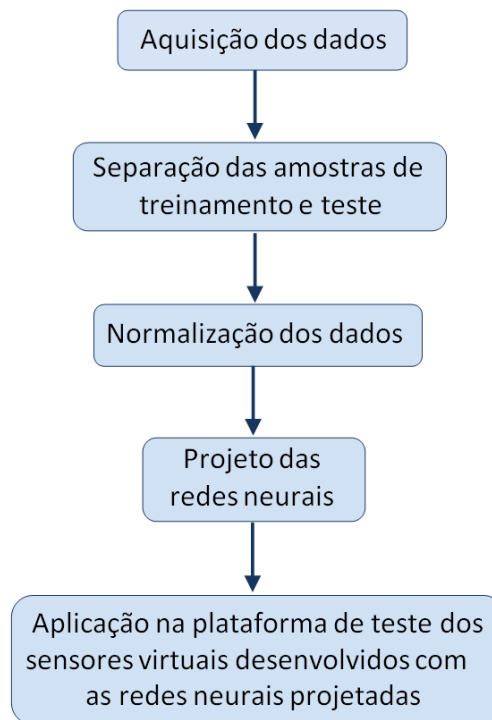
Neste capítulo são apresentadas as simulações e os testes experimentais práticos realizados com os sensores virtuais desenvolvidos utilizando redes neurais nas estimativas de valores para atuar no monitoramento da plataforma de transporte fluídico. É apresentado o desenvolvimento dos sensores virtuais, a interface gráfica desenvolvida para uma melhor interação com o usuário e, por fim, as simulações e os resultados dos testes experimentais obtidos.

O objetivo é fazer a relação das variáveis (posição da válvula (corrente), frequência (tensão) de operação da bomba, pressão e vazão) na plataforma utilizando redes neurais para realizar a modelagem do sistema, com o intuito de fazer a estimativa das variáveis de interesse na plataforma para realizar o monitoramento do processo.

5.1 Sensores Virtuais Desenvolvidos

Nesse contexto, deu-se início ao desenvolvimento dos sensores virtuais. São seguidas algumas etapas para este desenvolvimento. Na Figura 5.1 está ilustrado o fluxograma em que são apresentadas as etapas seguidas para o desenvolvimento e aplicação prática dos sensores virtuais para o monitoramento de processos. Para validação, são desenvolvidos sensores virtuais para monitoramento dos valores de vazão e pressão na plataforma de teste utilizada (tubulação de transporte fluídico).

Figura 5.1: Fluxograma das Etapas Seguidas para o Desenvolvimento dos Sensores Virtuais.



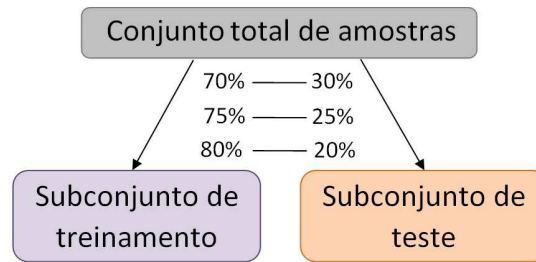
- **Aquisição dos dados**

Nesta etapa é realizada a aquisição dos valores das variáveis que compõem o processo e que se relacionam entre si. A frequência (tensão), a posição da válvula de controle (corrente) e as válvulas manuais foram variadas para obter valores de pressão e vazão que sofrem alterações com essas mudanças na entrada do processo. Obtendo então um conjunto de variáveis (frequência, posição de válvulas, pressão e vazão) que se relacionam e que podem ser usadas como entradas e saídas para o treinamento das redes neurais.

- **Separação das amostras de treinamento e teste**

Com a aquisição dos dados realizada, parte-se para a separação das amostras. Para o projeto das redes neurais, é necessário um conjunto de dados para o treinamento da rede e outro para teste. Um conjunto de pares de entrada e saída, com cada par consistindo de um sinal de entrada e resposta desejada correspondente, é referido como um conjunto de dados de treinamento ou amostra de treinamento [18]. Na Figura 5.2 é apresentado um fluxograma que representa a divisão das amostras de dados disponíveis sobre o comportamento do processo, em que o subconjunto de treinamento é usado para o processo de aprendizado da rede e o conjunto de teste (não participa do treinamento da rede) para verificar se as soluções por parte da rede estão em patamares aceitáveis.

Figura 5.2: Fluxograma da Divisão das Amostras do Processo.



O tamanho dos conjuntos de treinamento e teste variam para cada caso, por exemplo, em Adriano [9] foram separadas cerca de 80% das amostras para o conjunto de treinamento e 20% para teste e em Li *et al.* [29] 70% para o treinamento e 30% para teste. Então, com a base de dados obtida com as medições realizadas, para a definição do tamanho do conjunto de treinamento e do conjunto de teste que gerasse melhor resultado no projeto das redes foram testadas três disposições, 80% e 20%, 75% e 25% e 70% e 30%. Nesta tese, o resultado com melhor desempenho foi obtido com 70% das amostras para treino e 30% para teste.

- **Normalização dos dados**

Normalmente os dados de entrada estão em intervalos de variação distintos, isto faz com que seu treinamento e aprendizagem fiquem prejudicados, pois a rede neural pode interpretar valores mais altos como de maior importância e valores menores como menos importantes [45]. Para contornar este problema aplica-se a técnica de normalização dos dados, isto faz com que todos os atributos fiquem com valores no intervalo da função de ativação dos neurônios. Neste tese as normalizações foram feitas por meio da função *mapminmax* do MATLAB, conforme Equação 5.1:

$$x_N = \frac{(x_{Nmax} - x_{Nmin}) * (x - x_{min})}{x_{max} - x_{min}} + x_{Nmin} \quad (5.1)$$

em que x_{Nmin} e x_{Nmax} são iguais aos extremos do intervalo da normalização, 0 e 1 no caso desta tese, respectivamente, e x_N é o valor normalizado.

- **Projeto das redes neurais**

A forma pela qual os neurônios de uma RNA estão estruturados (interconectados) está intimamente relacionada ao algoritmo de aprendizagem a ser utilizado para treiná-la. Assim, com os amostras separadas para treinamento e teste, inicia-se o processo para projeto das redes neurais para a estimativa dos valores das variáveis de interesse (valores de pressão e vazão). Utilizou-se o *software* MATLAB (funções *trainscg*, *trainlm* e *trainoss*) para projetar as redes e realizar a simulação para avaliar o desempenho das mesmas.

São consideradas duas possíveis situações no monitoramento do processo. Na primeira situação apenas um sinal de sensor real é perdido, mas ainda se tem disponível outro sinal de sensor real de alguma variável de saída que tenha relação com a variável de interesse para monitoramento. A segunda situação acontece quando há a perda de todos os sinais de sensores reais e o processo permanece por um tempo sem a disponibilidade de valores de sensores reais para o monitoramento, dependendo somente dos sensores virtuais até os sinais de sensores reais serem recuperados.

Assim, em cada situação (com a perda de um sensor por vez e com a perda simultânea de sensores) é necessário uma rede neural para cada variável de interesse. Se o interesse é de monitorar um número n de variáveis, é necessário um número $2 * n$ de redes neurais. Por exemplo, para monitorar 4 variáveis, o cenário seria de 5 possibilidades, 4 considerando a perda de um sensor por vez (com 4 redes neurais, uma para cada variável), e 1 possibilidade com a perda dos 4 sensores simultaneamente (também com 4 redes neurais, uma para cada variável), totalizando a necessidade de 8 redes neurais. O número de redes neurais necessárias dado o número de sensores reais que se quer monitorar segue a relação apresentada na Equação 5.2:

$$NR = 2 * NS \quad (5.2)$$

em que NS é o número de sensores reais para monitoramento e NR é o número de redes neurais necessárias.

Visto esta análise, para o desenvolvimento dos sensores virtuais para aplicação na plataforma de teste utilizada, com o interesse de se monitorar duas variáveis (pressão e vazão), são projetadas quatro redes neurais. Duas para as estimativas dos valores de pressão, uma atuando apenas quando não há sinal algum de sensores reais (sinais do sensor de pressão e de vazão perdidos simultaneamente) e uma atuando durante o restante do processo (há pelo menos um sinal de sensor real). E duas para as estimativas dos valores de vazão, também uma atuando quando ocorre a perda dos dois sinais de sensores simultaneamente e a outra durante o restante do processo (há pelo menos um sinal de sensor real).

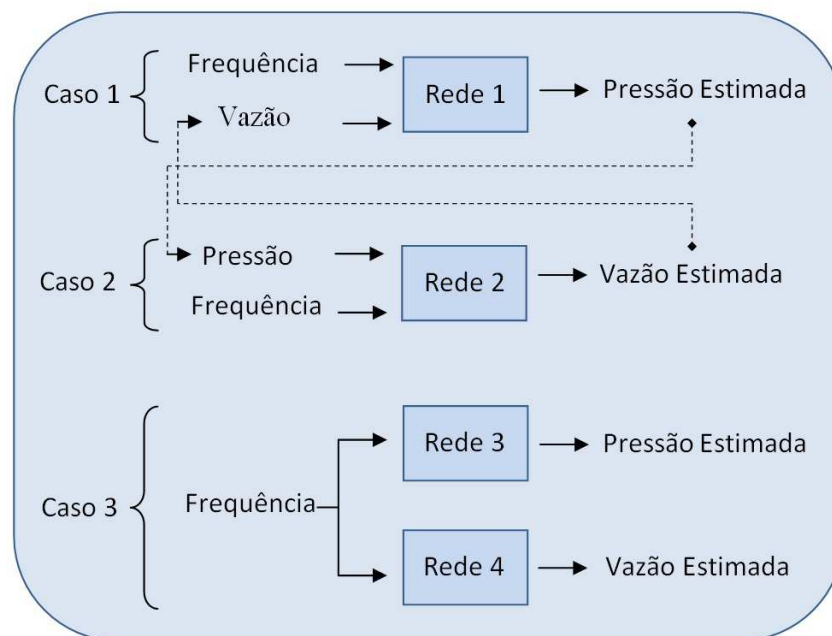
É importante diferenciar as entradas do processo das entradas utilizadas para o treinamento das redes. As entradas que podem interferir no funcionamento do processo são frequência, posição da válvula remota e posição de válvulas manuais (usadas para perturbações no sistemas). Já as entradas utilizadas para o treinamento das redes são valores de frequência, pressão e vazão, dependendo da rede.

No treinamento de rede para estimativa dos valores de pressão, que é executada quando há pelo menos um sinal de sensor real, os valores de frequência e vazão

são usados como entrada. Já no treinamento de rede para estimativa dos valores de vazão, que também é executada quando há pelo menos um sinal de sensor real, os valores de frequência e pressão são usados como entrada. E no treinamento das redes para as estimativas que ocorrem quando os dois sinais dos sensores reais são perdidos simultaneamente (pressão e vazão), apenas o valor de frequência é usado como entrada. Os valores da posição da válvula não são utilizados para treinamento e teste das redes, mas como entradas no processos, funcionando como perturbação no sistema.

Na Figura 5.3 pode-se observar o esquema com entradas e saídas para treinamento das redes. Assim, usando as quatro redes neurais, três possíveis casos são considerados durante o monitoramento do processo. O primeiro caso (rede 1) acontece quando os valores de pressão estão sendo monitorados e o sinal do sensor de vazão é perdido (sinal usado na rede neural que estima os valores de pressão). O segundo caso (rede 2) ocorre quando os valores de vazão estão sendo monitorados e o sinal do sensor de pressão é perdido (sinal usado na rede neural que estima os valores de vazão). E o terceiro caso ocorre quando, durante o processo, os sinais dos dois sensores são perdidos simultaneamente (redes 3 e 4).

Figura 5.3: Esquema com Entradas e Saídas para Treinamento das Redes.



As setas pontilhadas (redes 1 e 2) são para identificar os casos em que um dos sinais é perdido, necessitando o uso de valores estimados para continuar o monitoramento do processo. Por exemplo, quando o sinal de vazão real é perdido, sua estimativa continua funcionando porque está usando os valores de frequência e pressão como

entradas da rede que estimam seu valor, mas as estimativas de valores de pressão ficam comprometidas porque, para estimar o valor de pressão, é usado o valor de vazão (que foi perdido).

Para solucionar essa questão, as estimativas dos valores de vazão são usadas como entradas da rede (em vez do sinal de vazão que foi perdido) para continuar estimando o valor de pressão. Assim, mesmo perdendo um dos sinais dos sensores reais, é possível realizar o monitoramento do processo.

As entradas das redes são valores atuais do processo. Mas como o treinamento é realizado de forma *offline*, ou seja, dados são coletados e armazenados anteriormente para serem utilizados no treinamento e só assim aplicados no processo. Então esses valores atuais são como amostras de testes, em que funcionam como entradas das redes já treinadas com dados históricos adquiridos e armazenados.

As redes para a estimativa dos valores de pressão e vazão quando os dois sinais são perdidos simultaneamente (redes 3 e 4) são utilizadas somente quando os sinais são perdidos, sendo utilizada apenas a frequência aplicada para bombear a água no tubo como entrada, diferentemente das outras duas redes, em que são utilizadas durante o restante do processo, quando há pelo menos um sinal de sensor real (redes 1 e 2).

- **Aplicação na plataforma de teste do sensor virtual desenvolvido com as redes neurais projetadas**

Após o projeto das redes neurais, a etapa seguinte é a aplicação prática dos sensores virtuais desenvolvidos no monitoramento do processo, no caso testado, da plataforma de testes experimental utilizada (sistema de transporte fluídico).

Para testar o monitoramento, um valor de frequência foi aplicado para bombear o líquido dentro da tubulação e os valores de pressão e vazão que estavam sendo medidos (sensores reais) e estimados (sensores virtuais) foram observados. Em um determinado momento, com o processo em operação, os sensores reais foram removidos, inicialmente um de cada vez (utilizando as redes 1 e 2) e em seguida os dois simultaneamente (utilizando as redes 3 e 4), perdendo os sinais dos sensores reais que estavam sendo medidos e usando as redes neurais para estimar os valores de interesse, sendo possível realizar o monitoramento do processo utilizando os sensores virtuais implementados. A posição da válvula de controle também foi alterada. E por fim, a abertura da válvula manual foi realizada para gerar perturbação no processo.

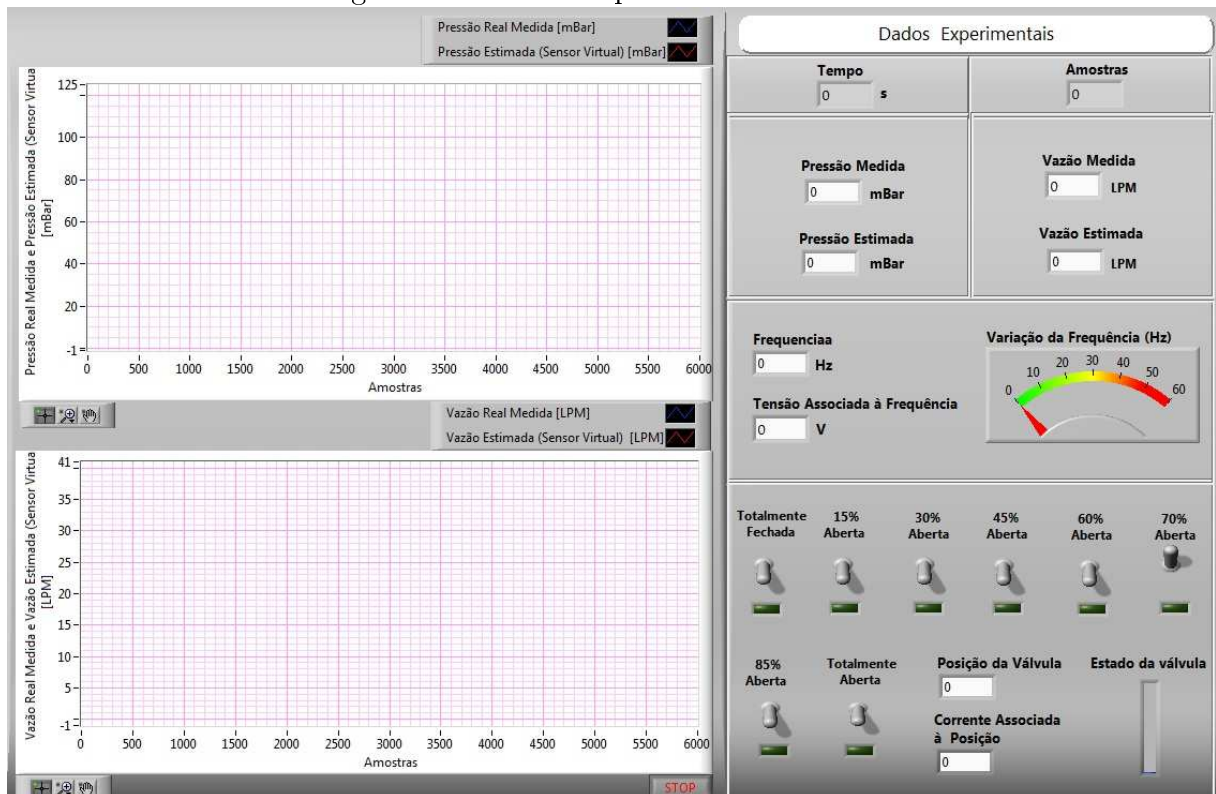
A ideia geral é ter o processo completamente monitorado, assim, se por algum motivo a

medição que está sendo realizada falhar, usar sensores virtuais e ser possível estimar valores das variáveis de interesse utilizando redes neurais nas estimativas para evitar que, por exemplo, um processo industrial seja paralisado, tornando o processo de monitoramento menos invasivo, o que diminui gastos, por exemplo, com interrupções para solucionar problemas que poderiam ser resolvidos sem a necessidade de parada do processo. Isto torna os processos mais confiáveis, com melhores desempenhos e com menores dificuldades de detectar e solucionar possíveis falhas.

5.2 Interface Gráfica

Visualizar os dados adquiridos é um requisito fundamental para sistemas de teste, medição ou controle. Foi desenvolvida uma interface de monitoramento para facilitar a comunicação entre o usuário e as rotinas de monitoramento, apresentando dados e recebendo comandos. Para a criação da interface utilizou-se o *software* LABVIEW, uma ferramenta que contém uma abrangente coleção de controladores e indicadores que pode ser usada para criar uma interface para a sua aplicação específica. Com uma interface de usuário flexível, projetada especificamente para a sua aplicação, é possível alterar parâmetros, visualizar os dados e ter os seus resultados analisados em menos tempo do que nos ambientes de programação tradicionais. Na Figura 5.4 pode ser observada a interface implementada.

Figura 5.4: Interface para Monitoramento.



Fonte: Elaborada pela Autora.

Nesta interface se ativa a abertura da válvula, altera-se os valores de frequência, e se observa os valores de pressão e vazão reais medidos e estimados (sensores virtuais). Também é possível observar a tensão associada à frequência aplicada, o tempo de duração do processo e o número de amostras.

5.3 Resultados das Simulações

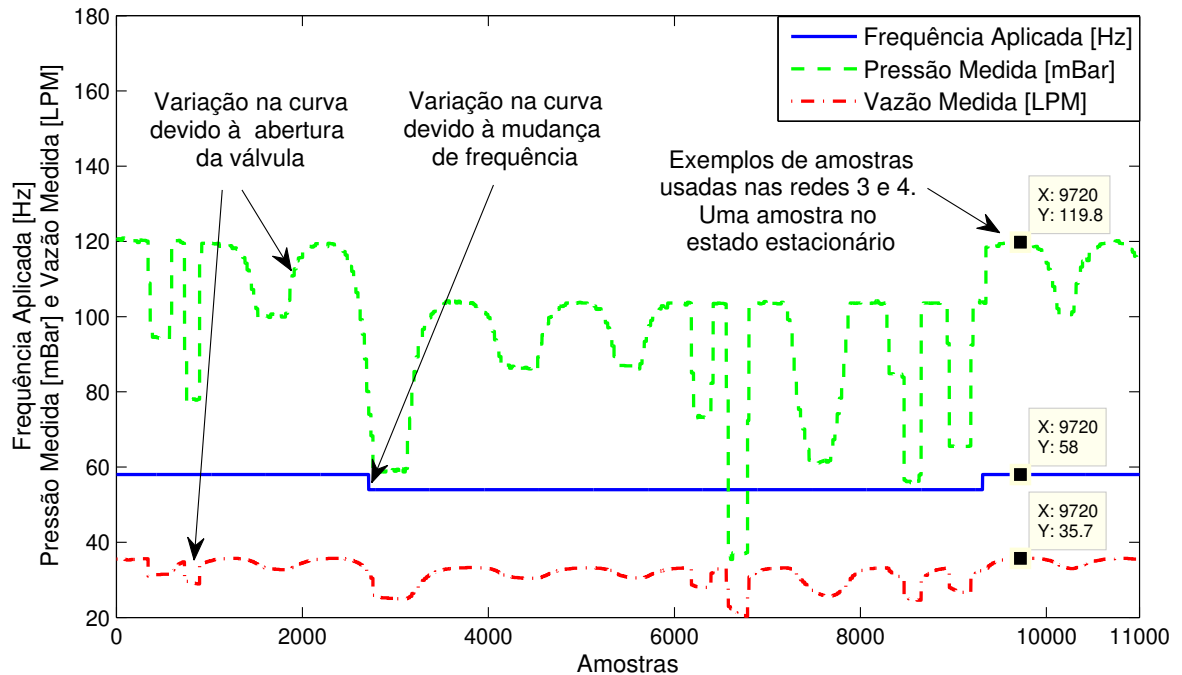
Foram realizadas simulações para avaliar o desempenho das redes para serem aplicadas na plataforma. Para a realização das simulações, inicialmente foram feitas medições para obter uma base de dados que possa ser utilizada no treinamento das redes neurais.

5.3.1 Medições para Obtenção da Base de Dados

Foram feitas as medições com o intuito de se obter uma base de dados em que seja possível observar a relação dos valores de pressão e vazão que estão relacionadas com os valores de frequência (tensão) e posição da válvula (corrente). Para isso, a frequência, posição da válvula e válvulas manuais foram variadas, obtendo um conjunto de amostras (frequência, posição da válvula, pressão e vazão) que se relacionam e que são utilizadas como entradas e saídas no treinamento e testes das redes neurais.

Com as medições, obteve-se um conjunto total com 11000 amostras, que é subdividido entre o subconjunto de treinamento e o de teste. Na Figura 5.5 pode ser observado o gráfico com os valores que compõem o conjunto total de amostras utilizado.

Figura 5.5: Conjunto Total de Amostras Utilizado na Criação das Redes Neurais.



Fonte: Elaborada pela Autora.

Para as redes 1 e 2 foi utilizado o conjunto total com todas as amostras (valores das amostras dispostos sequencialmente no processo de treinamento e teste), para cada frequência aplicada e mudança de posição da válvula, foi utilizado o conjunto de valores que formam a curva. E para as redes 3 e 4 adotou-se um cenário pré definido, com a posição da válvula 100% aberta e utilizando-se os valores quando o processo está em estado estacionário (valores dispostos aleatoriamente no processo de treinamento e teste), isto é, as variáveis de interesse não variam com o tempo (variação zero ou quase zero em um sistema real) por exemplo, na amostra 9720 apresentada na Figura 5.5, para a frequência de 58 Hz, utiliza-se o valor de pressão igual a 119,8 mBar e valor de vazão igual a 35,7 LPM como parâmetros para o treinamento dessas redes (3 e 4).

5.3.2 Simulações para Análise de Desempenho das Redes Neurais Projetadas

É necessário configurar alguns parâmetros no projeto das redes neurais. Para encontrar a configuração de rede que gerasse o melhor desempenho, foram simuladas redes do tipo *feedforward* e *feedforward* em cascata. Parâmetros que são alterados para o projeto das

redes neurais com diferentes configurações:

- função de treinamento - *trainlm*, *trainscg* e *trainoss*
- número de *epochs* (épocas de treinamento) - 30, 50 e 100 *epochs*
- número de neurônios na camada oculta - 1 a 10.

O uso de uma camada oculta é considerada suficiente quando se usa funções de transferências regulares (Ex: sigmoidal), embora o número de neurônios na camada oculta necessários possa ser alto [20]. Nesta tese simulou-se redes com uma camada oculta, variando-se o número de neurônios de 1 a 10 [27].

Em resumo, foram simuladas redes do tipo *feedforward* e *feedforward* em cascata com uma camada oculta, variando-se o número de neurônios na camada oculta de 1 a 10, com três tipos diferentes de funções de treinamento (*trainlm*, *trainscg*, *trainoss*), para 30, 50 e 100 *epochs*. Esse processo foi repetido três vezes, para as três subdivisões do conjunto total de amostras, 70% para treinamento e 30% para teste, 75% e 25%, e 80% e 20%.

Para avaliar o desempenho das redes simuladas, é realizado o cálculo do erro entre os valores reais e estimados. A equação utilizada para o cálculo destes erros é apresentada na Equação 5.3:

$$Erro = \frac{|Valor Real - Valor Simulado|}{Valor Real} * 100\% \quad (5.3)$$

Para definir o método de melhor desempenho, calculou-se a média dos erros. As tabelas completas com os erros médios calculados para as redes 1 e 2, que atuam durante o processo enquanto há pelo menos um sinal de sensor real, são apresentadas nos Apêndices A e B, e para as redes 3 e 4, quando não há sinal de sensores reais, são apresentadas nos Apêndices D e E.

Rede para Estimativa de Pressão (Rede 1)

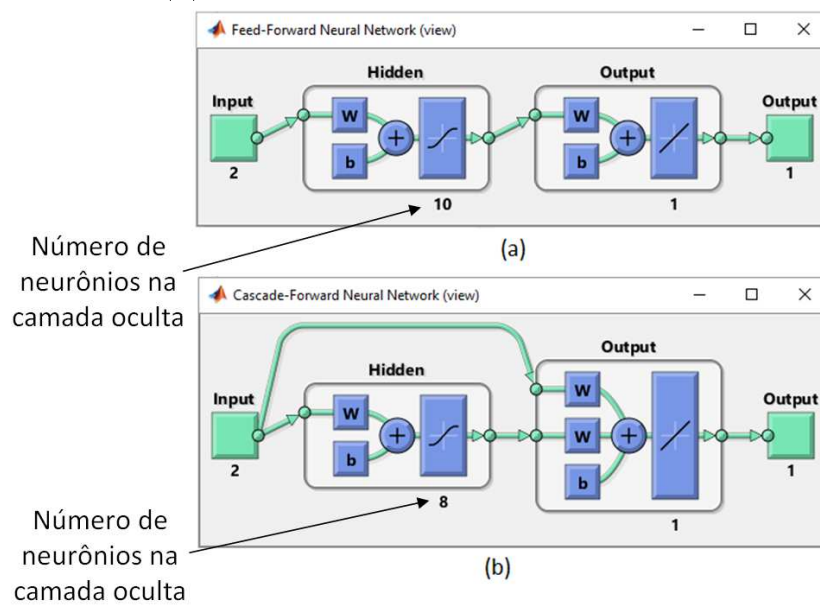
Neste caso, o monitoramento é realizado durante todo o processo enquanto há pelo menos um sinal de sensor real, então, para cada entrada, obtém-se um conjunto de valores de pressão e de vazão, resultando em um número maior de amostras do que o utilizado nas redes 3 e 4. Assim, utilizou-se um conjunto com 11000 amostras, subdividido em conjunto de treinamento e conjunto de teste. Neste caso, a frequência e o valor da vazão real medida foram usados como entradas para o treinamento da rede. Em caso de perda do sinal do sensor real de vazão, a frequência e o valor de vazão estimado são usados como entradas.

As configurações de redes que geraram menores erros para cada tipo de rede foram:

- Rede *Feedforward*: 80% treino - 20% teste, 50 *epochs*, função de treinamento *trainlm*, 10 neurônios na camada oculta. Erro médio de 2,62%.
- Rede *Feedforward* em Cascata: 80% treino - 20% teste, 30 *epochs*, função de treinamento *trainlm*, 8 neurônios na camada oculta. Erro médio de 2,64 %.

Na Figura 5.6 são apresentadas as redes neurais como um diagrama gráfico gerado pelo *software* MATLAB.

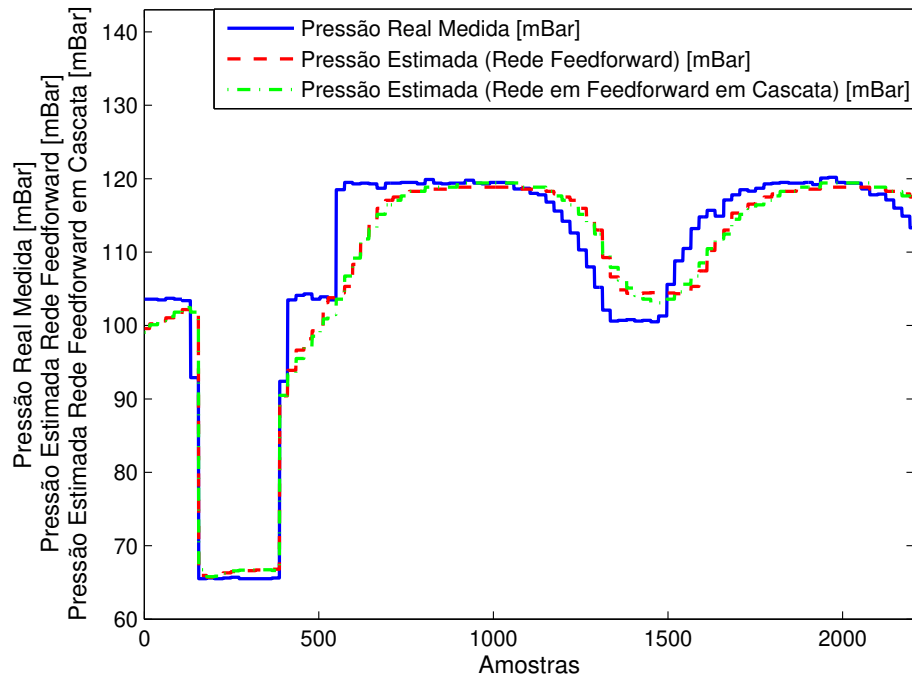
Figura 5.6: Redes Neurais como um Diagrama Gráfico (Rede 1). *Feedforward* (a) e *Feedforward* em Cascata (b).



Fonte: Elaborada pela Autora.

No gráfico da Figura 5.7 é apresentada a comparação desses dois tipos de redes com as configurações citadas, apresentando os valores reais medidos e os resultados das estimativas dos valores de pressão (sensor virtual).

Figura 5.7: Comparação do Desempenho das Redes do Tipo *Feedforward* e *Feedforward* em Cascata nas Estimativas dos Valores de Pressão (Rede 1).



Fonte: Elaborada pela Autora.

Assim, a rede para a estimativa dos valores de pressão utilizada para aplicação prática na plataforma de teste em estudo, foi do tipo *Feedforward* configurada com uma camada oculta (com 10 neurônios), 50 *epochs* e função de treinamento *trainlm*, que gerou melhor desempenho, com erro médio de 2,62%.

Rede para Estimativa de Vazão (Rede 2)

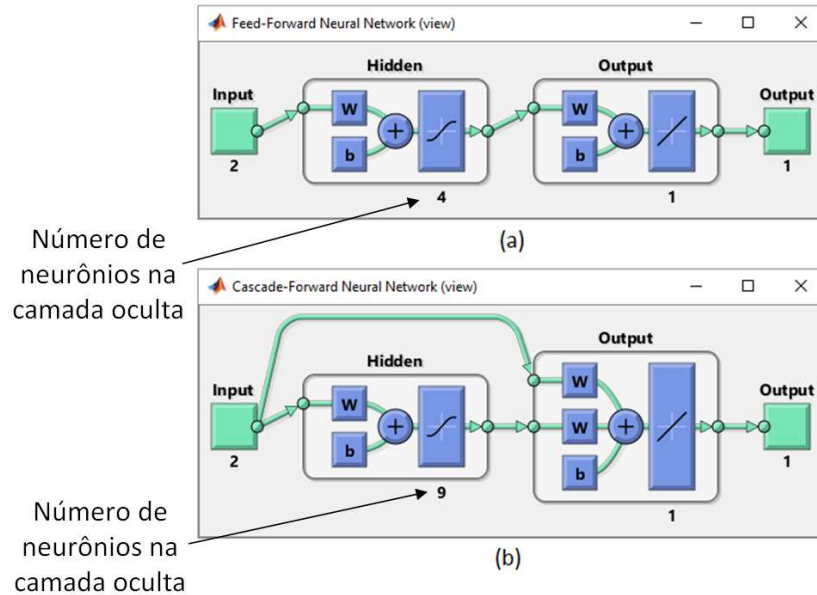
Neste caso, o monitoramento também é realizado durante todo o processo enquanto há pelo menos um sinal de sensor real. A frequência e o valor da pressão real medido foram utilizados como entradas para o treinamento da rede. Em caso de perda do sinal do sensor real de pressão, a frequência e o valor de pressão estimado são usados como entradas. Foi utilizado o mesmo conjunto com 11000 amostras, subdividido em conjunto de treinamento e de teste.

As configurações de redes que geraram menores erros para cada tipo de rede foram:

- Rede *Feedforward*: 80% treino - 20% teste, 100 *epochs*, função de treinamento *trainlm*, 4 neurônios na camada oculta. Erro médio de 1,35%.
- Rede *Feedforward* em Cascata: 80% treino - 20% teste, 50 *epochs*, função de treinamento *trainscg*, 9 neurônios na camada oculta. Erro médio de 1,36%.

Na Figura 5.8 são apresentadas as redes neurais utilizadas nas estimativas dos valores de vazão como um diagrama gráfico gerado pelo *software* MATLAB.

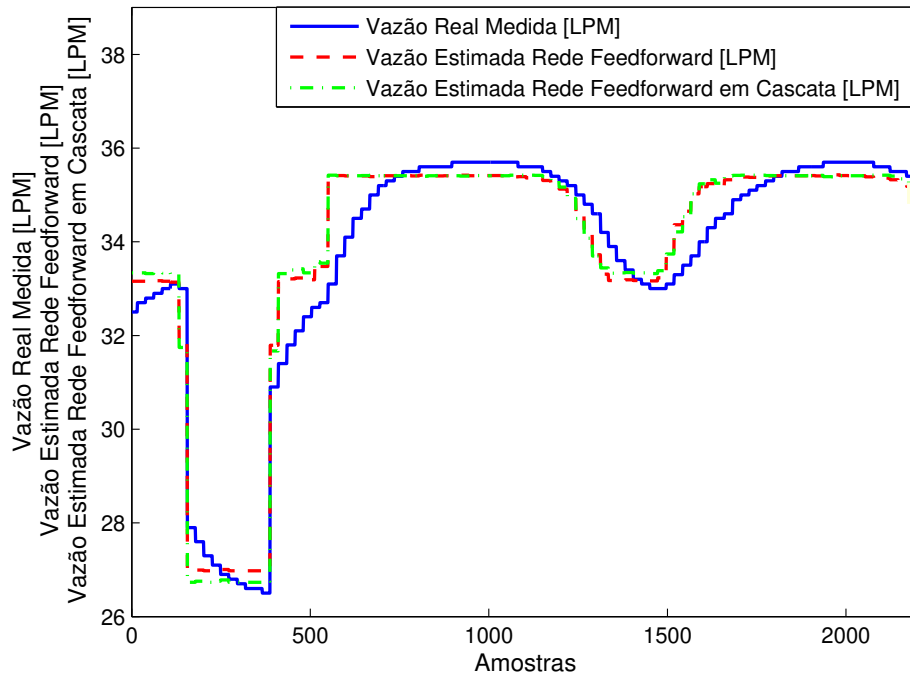
Figura 5.8: Redes Neurais como um Diagrama Gráfico (Rede 2). *Feedforward* (a) e *Feedforward* em Cascata (b).



Fonte: Elaborada pela Autora.

No gráfico da Figura 5.9 é apresentada a comparação desses dois tipos de redes com as configurações citadas, apresentando os valores reais medidos e os resultados das estimativas dos valores de vazão (sensor virtual).

Figura 5.9: Comparação do Desempenho das Redes do Tipo *Feedforward* e *Feedforward* em Cascata nas Estimativas dos Valores de Vazão (Rede 2).



Fonte: Elaborada pela Autora.

Assim, a rede para a estimativa dos valores de vazão utilizada para aplicação prática na plataforma de teste em estudo, foi do tipo *Feedforward* configurada com uma camada oculta (com 4 neurônios), 100 *epochs* e função de treinamento *trainlm*, que gerou melhor desempenho, com erro médio de 1,35 %.

Redes para Estimativa de Pressão e Vazão Quando os Sinais dos Dois Sensores Reais São Perdidos Simultaneamente (Rede 3 e 4)

Neste caso, o monitoramento é realizado quando os dois sinais dos sensores reais são perdidos simultaneamente no estágio estacionário. Para obtenção do conjunto de dados utilizado para treinamento e teste das redes, foram feitas as medições com o intuito de se obter a relação dos valores pressão e vazão que estão relacionadas com os valores de frequência (tensão). Com a válvula de controle totalmente aberta (fixa), variou-se os valores de frequência, de forma aleatória, dentro da faixa de valores de 20 Hz a 60 Hz, resultando em um conjunto com 30 amostras, também subdividido no conjunto de treinamento e conjunto de teste. Os testes foram repetidos 3 (três) vezes, número de vezes suficiente para o critério de parada estabelecido para o trabalho, que foi o valor do desvio padrão menor que 1 (um). Os valores obtidos nas medições são apresentados no Apêndice C.

Após as medições, foi realizado o tratamento estatístico com os valores obtidos para

se ter uma faixa de confiança, assim, foi feito o cálculo da média e do desvio padrão dos valores (também apresentados no Apêndice C). A Equação 5.4 utilizada para o cálculo do desvio padrão é apresentada a seguir:

$$DesvioPadrao = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{n - 1}} = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n - 1} \quad (5.4)$$

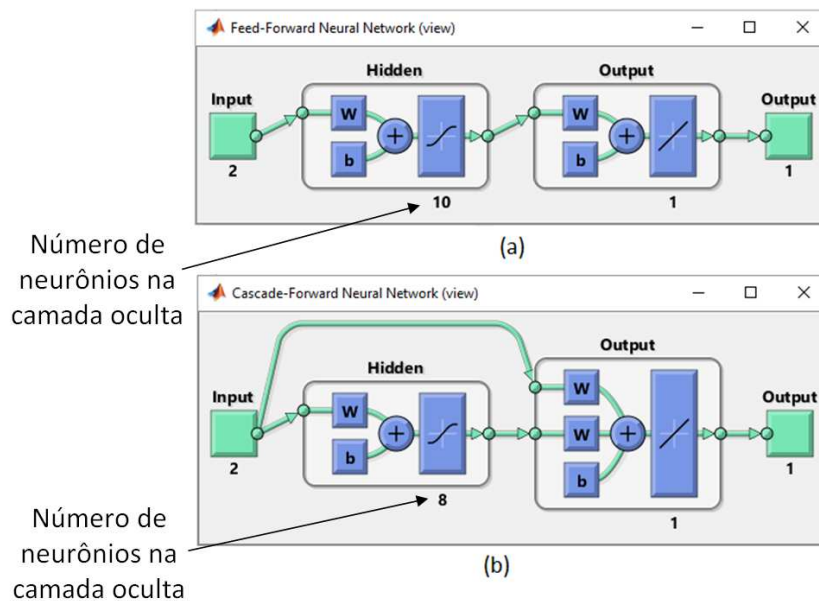
Onde \bar{x} a média, x_i é o valor individual e n é o número de valores.

Com os dados obtidos após o tratamento estatístico, partiu-se para o projeto das redes neurais, buscando realizar as estimativas dos valores de pressão e vazão. Para a estimativa dos valores de pressão quando os dois sinais dos sensores reais são perdidos simultaneamente (rede 3), as configurações de redes que geraram menores erros para cada tipo de rede foram:

- Rede *Feedforward*: 75% treino - 25% teste, 50 *epochs*, função de treinamento *trainlm*, 10 neurônios na camada oculta. Erro médio de 0,62%.
- Rede *Feedforward* em Cascata: 75% treino - 25% teste, 50 *epochs*, função de treinamento *trainlm*, 8 neurônios na camada oculta. Erro médio de 0,58%.

Na Figura 5.10 são apresentadas as redes neurais utilizadas nas estimativas dos valores de pressão (rede 3) como um diagrama gráfico gerado utilizando o *software* MATLAB.

Figura 5.10: Redes Neurais como um Diagrama Gráfico (Rede 3). *Feedforward* (a) e *Feedforward* em Cascata (b).



Fonte: Elaborada pela Autora.

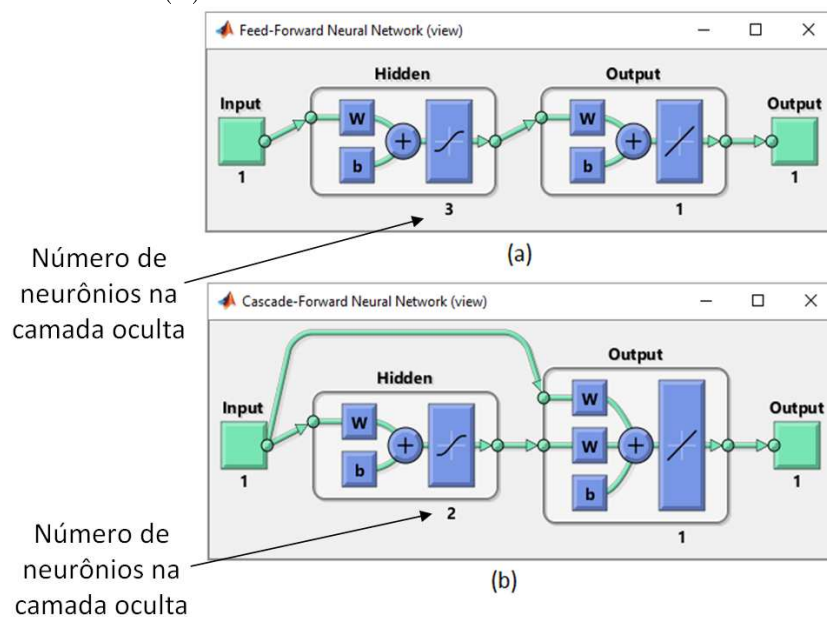
É possível observar que as redes para comparação deste caso têm o mesmo número de neurônios na camada escondida que as redes para comparação do caso 1 (Diagrama Gráfico da Figura 5.6), por isso o diagrama gráfico é o mesmo, mas se diferenciam pelo número de *epochs* e pelo tamanho dos conjuntos de treinamento e teste utilizados.

Para a estimativa dos valores de vazão quando os dois sinais dos sensores reais são perdidos simultaneamente (rede 4), as configurações que geraram menores erros para cada tipo de rede foram:

- Rede *Feedforward*: 75% treino - 25% teste, 30 *epochs*, função de treinamento *trainlm*, 3 neurônios na camada oculta. Erro médio de 0,13%.
- Rede *Feedforward* em Cascata: 75% treino - 25% teste, 50 *epochs*, função de treinamento *trainlm*, 2 neurônios na camada oculta. Erro médio de 0,15%.

Na Figura 5.11 são apresentadas as redes neurais utilizadas nas estimativas dos valores de vazão (rede 4) como um diagrama gráfico gerado utilizando o *software* MATLAB.

Figura 5.11: Redes Neurais como um Diagrama Gráfico (Rede 4). *Feedforward* (a) e *Feedforward* em Cascata (b).



Fonte: Elaborada pela Autora.

Os valores de pressão reais medidos e estimados com as redes *feedforward* e *feedforward* em cascata com as configurações citadas são apresentados (resumidamente) na Tabela 5.1. A Tabela completa pode ser observadas no Apêndice C.

Tabela 5.1: Valores de Pressão Reais Medidos e Valores Estimados com as Redes *Feedforward* e *Feedforward* em Cascata (Rede 3).

Frequência (Hz)	Valores Reais Medidos de Pressão (mBar)	Valores de Pressão Estimados Rede Feedforward (mBar)	Valores de Pressão Estimados Rede Feedforward em Cascata (mBar)
20	6,30	6,28	6,36
25	14,51	14,36	14,50
35	41,20	40,98	41,21
38	49,00	49,12	48,94
45	71,50	70,95	72,00
47	77,27	78,67	78,39
56	111,00	110,70	109,73
58	117,43	117,45	117,67

E os valores de vazão reais medidos e estimados com as redes *feedforward* e *feedforward* em cascata com as configurações citadas são apresentados (resumidamente) na Tabela 5.2. A Tabela completa também pode ser observadas no Apêndice C.

Tabela 5.2: Valores de Vazão Reais Medidos e Valores Estimados com as Redes *Feedforward* e *Feedforward* em Cascata (Rede 4).

Frequência (Hz)	Valores Reais Medidos de Vazão (LPM)	Valores de Vazão Estimados Rede Feedforward (LPM)	Valores de Vazão Estimados Rede Feedforward em Cascata (LPM)
20	11,09	11,11	11,07
25	14,51	14,54	14,48
35	21,10	21,09	21,14
38	23,10	23,12	23,13
45	27,70	27,74	27,71
47	28,94	28,98	28,99
56	34,80	34,79	34,78
58	36,06	35,98	35,98

São calculados os erros (utilizando a Equação 5.3) entre as medições reais e suas estimativas (sensores virtuais) para os valores de pressão e vazão apresentados nas Tabelas 5.1 e 5.2. Os erros calculados podem ser observados na Tabela 5.3.

Tabela 5.3: Erros Obtidos Entre Valores Reais e Estimados de Pressão e Vazão com as Redes 3 e 4.

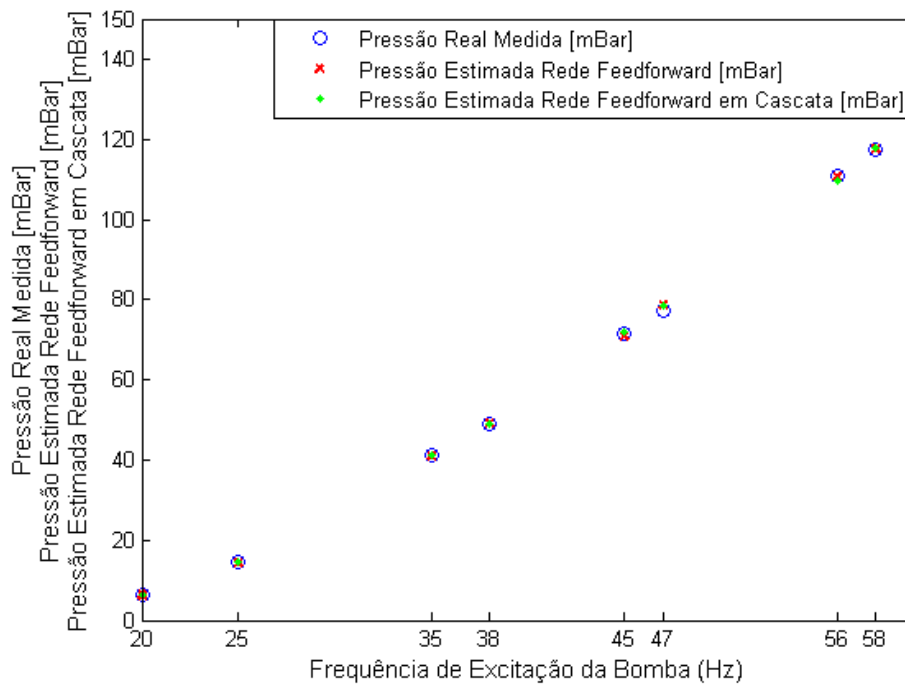
	Erro para a Rede 3		Erro para a Rede 4	
	Pressão (%)		Vazão (%)	
Frequência (Amostras de Teste) (Hz)	<i>Feedforward</i>	<i>Feedforward em Cascata</i>	<i>Feedforward</i>	<i>Feedforward em Cascata</i>
20	0,32	0,95	0,18	0,18
25	1,03	0,07	0,21	0,21
35	0,53	0,02	0,05	0,19
38	0,24	0,12	0,09	0,13
45	0,77	0,70	0,14	0,04
47	1,81	1,45	0,14	0,17
56	0,27	1,14	0,03	0,06
58	0,02	0,20	0,22	0,22
Média Aritmética	0,62	0,58	0,13	0,15

Para definir o método de melhor desempenho, calculou-se a média aritmética dos erros apresentados na Tabela 5.3, chegando a conclusão que os melhores desempenhos para estes casos foram obtidos com a rede *feedforward* em cascata (no caso da pressão), com erro médio de 0,58%, e *feedforward* (no caso da vazão), com erro médio de 0,13%.

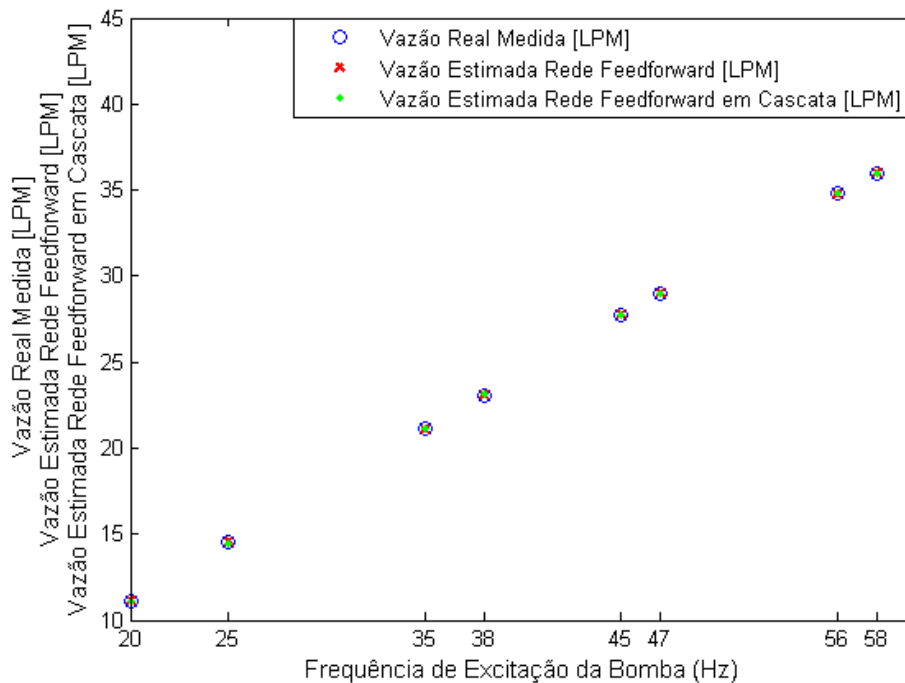
Pode ser observado que o maior erro obtido para os dois casos escolhidos, foi na estimativa do valor de pressão quando 50 Hz foi aplicado, com o valor igual a 1,85%, considerado aceitável para o que está sendo proposto, sem comprometer o monitoramento do processo.

No gráfico da Figura 5.12 é apresentada a comparação desses dois tipos de redes com as configurações citadas, apresentando os valores reais medidos e os resultados das estimativas (sensores virtuais) dos valores de pressão (a) e vazão (b) obtidos nas simulações utilizando os 25% das amostras para teste (8 amostras).

Figura 5.12: Monitoramento dos Valores de Pressão (a) e Vazão (b) Utilizando as Redes 3 e 4.



(a)



(b)

Assim, a rede para a estimativa dos valores de pressão utilizada para aplicação prática na plataforma de teste em estudo, quando os dois sinais de sensores são perdidos simultaneamente, foi do tipo *Feedforward* em cascata configurada com uma camada oculta (com

8 neurônios), 50 *epochs* e função de treinamento *trainlm*, que gerou melhor desempenho, com erro médio de 0,58%. E a rede para a estimativa dos valores de vazão, também quando os dois sinais de sensores são perdidos simultaneamente, foi do tipo *Feedforward* configurada com uma camada oculta (com 3 neurônios), 30 *epochs* e função de treinamento *trainlm*, que gerou melhor desempenho, com erro médio de 0,13%.

Em resumo, são projetadas quatro redes neurais. Caso 1 com rede 1 para estimativa de pressão quando apenas um sinal de sensor real é perdido. Caso 2 com rede 2 para estimativa de vazão quando apenas um sinal de sensor real é perdido. Caso 3 com rede 3 para estimativa de pressão quando os dois sinais de sensores são perdidos simultaneamente (não há sensores reais funcionando no processo). E caso 3 com rede 4 para estimativa de vazão quando os dois sinais de sensores são perdidos simultaneamente (não há sensores reais funcionando no processo).

Na Tabela 5.4 é apresentado o resumo das configurações das redes com melhores desempenho para cada caso e que foram utilizadas para aplicação prática na plataforma de teste em estudo.

Tabela 5.4: Tabela Resumo

	Caso 1	Caso 2	Caso 3 (sinais dos sensores reais perdidos simultaneamente)	
			Rede 1 (Pressão)	Rede 2 (Vazão)
Tipo de Rede	<i>Feedforward</i>	<i>Feedforward</i>	<i>Feedforward em cascata</i>	<i>Feedforward</i>
Número de Neurônios na Camada Oculta	10	4	8	3
Função de Treinamento	<i>trainlm</i>	<i>trainlm</i>	<i>trainlm</i>	<i>trainlm</i>
Divisão Subconjunto de Treinamento e Teste	80(%) e 20(%)	80(%) e 20(%)	75(%) e 25(%)	75(%) e 25(%)
Número de Epochs	50	100	50	30
Erro Médio	2,62 (%)	1,35 (%)	0,58 (%)	0,13 (%)

5.3.3 Resultados dos Experimentos com Aplicação Prática dos Sensores Virtuais no Monitoramento da Plataforma

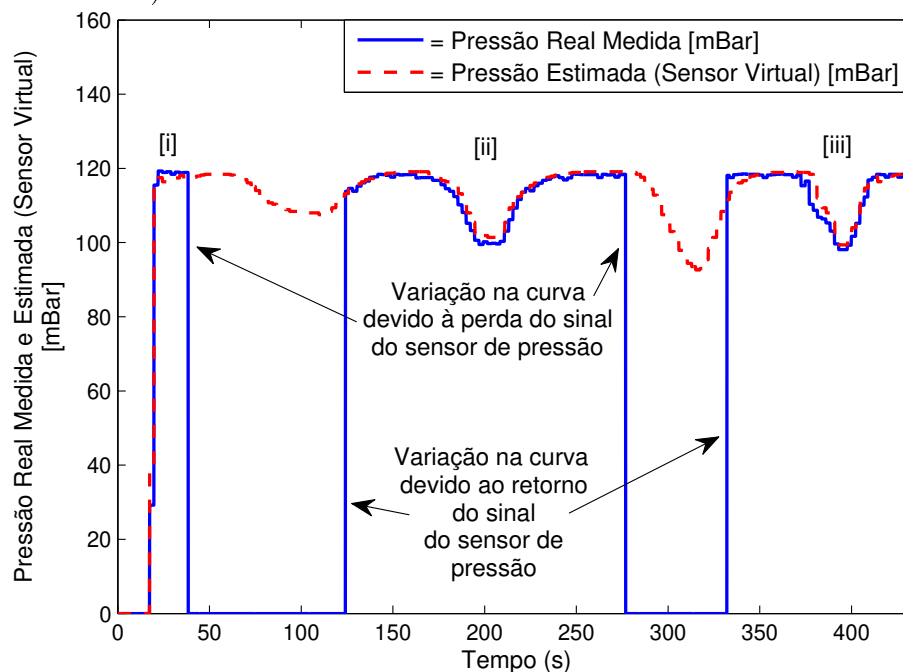
Com as simulações realizadas e a definição da rede com melhor desempenho para cada caso, o próximo passo foi a aplicação prática dos sensores virtuais implementados seguindo a metodologia proposta no monitoramento da plataforma de testes em estudo.

Resultados com a Retirada de um Sensor por vez e Alterando as Posições das Válvulas para Causar Distúrbio (Caso 1 e 2, com Redes 1 e 2)

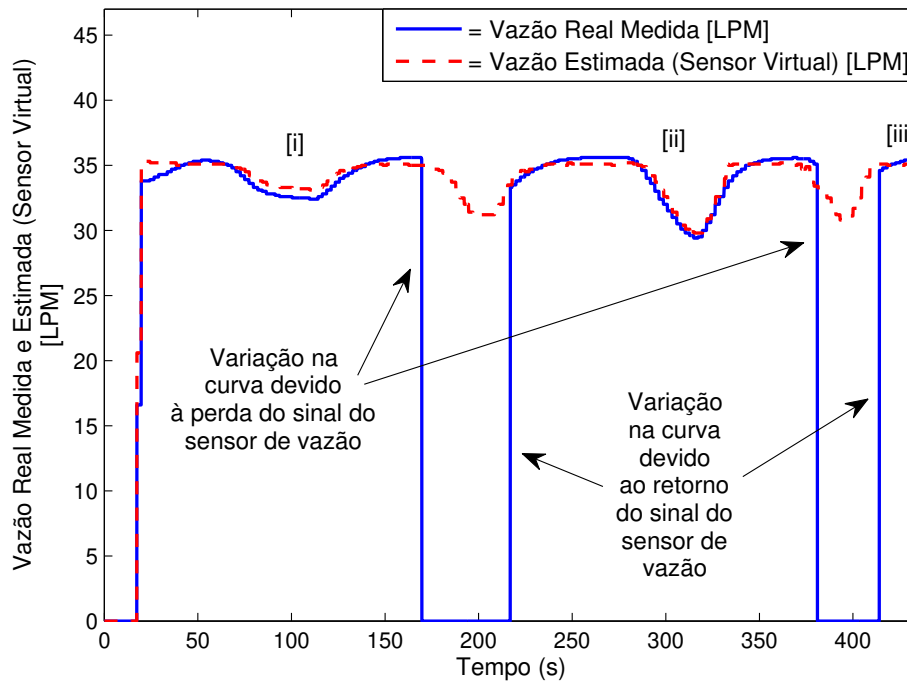
Os experimentos realizados para testar o monitoramento na plataforma utilizando os sensores virtuais foram a mudança do valor de frequência na entrada do processo, a retirada de um sensor por vez e a variação da válvula de controle. Com a perda de um sensor por vez, para que o processo não seja paralisado, as redes neurais, com base na entrada que está sendo aplicada (frequência) e em outras variáveis medidas do processo, estimam os valores de pressão e vazão, fazendo com que o processo possa ser monitorado utilizando os valores estimados (sensores virtuais) enquanto os sinais dos sensores reais não são recuperados.

Na Figura 5.13 pode ser observado o gráfico com o monitoramento dos valores de pressão (Figura 5.13a) e valores de vazão (Figura 5.13b) durante o mesmo processo (observe que o eixo 'Tempo (s)' dos gráficos são os mesmos).

Figura 5.13: Monitoramento dos Valores de Pressão (a) e Vazão (b) Durante o Mesmo Processo (Redes 1 e 2).



(a)



(b)

A sequência de ações para realização deste teste foi: inicialmente aplicou-se 58 Hz de frequência, fazendo com que os valores de pressão e vazão alterassem. Em seguida, o sensor de pressão foi removido (antes de 50 segundos) enquanto a posição da válvula era alterada de 100% aberta para 60% aberta e logo após retornada novamente para seu estado 100% aberta (primeiro declive na curva). Após o retorno do sinal do sensor de pressão, foi realizada a retirada do sinal do sensor de vazão (após 150 segundos) enquanto a posição da válvula era variada de 100% aberta para 45% aberta e logo após retornada novamente para seu estado 100% aberta (segundo declive na curva). O terceiro passo foi retirar novamente o sinal do sensor de pressão enquanto a válvula manual era aberta durante um determinado tempo para causar distúrbio no processo (terceiro declive na curva). E por fim, após o retorno do sinal do sensor de pressão, foi realizada a retirada do sinal do sensor de vazão novamente (antes de 400 segundos) enquanto a válvula manual era aberta durante um determinado tempo para causar distúrbio no processo (quarto declive na curva). Os gráficos da Figura 5.13 foram divididos em partes ([i], [ii] e [iii]) para a apresentação dos erros calculados nos momentos em que existiam ambos os sinais (real e estimado), para ser possível o cálculo do erro. Na Tabela 5.5 podem ser observados os erros calculados.

Tabela 5.5: Erros Obtidos nas Estimativas dos Valores de Pressão e Vazão Aplicados na Plataforma Utilizando as Redes 1 e 2.

Erros Pressão (%)		
Parte i	Parte ii	Parte iii
4,49	1,02	1,62
Erros Vazão (%)		
Parte i	Parte ii	Parte iii
1,81	1,33	0,69

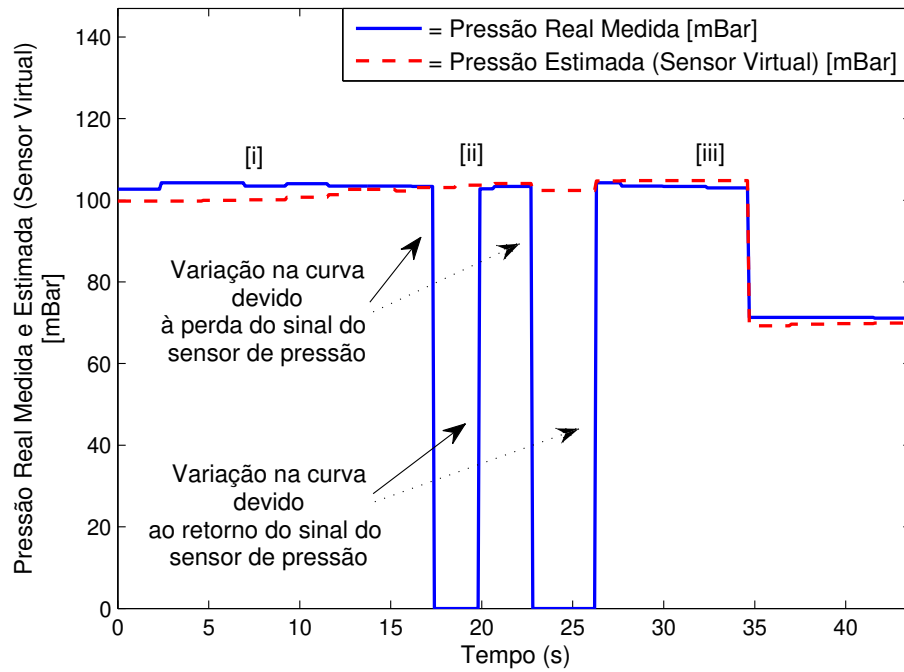
Como apresentado, foi possível realizar o monitoramento do processo estimando os valores de pressão e vazão utilizando redes neurais na implementação dos sensores virtuais. Mesmo com a perda dos sinais dos sensores, variação de frequência na entrada e variação na posição da válvula de controle, foi possível evitar que o processo fosse paralisado, continuando sua operação, com os valores estimados (sensores virtuais) enquanto os sinais dos sensores reais não eram recuperados.

Resultados com a Retirada dos Dois Sensores Simultaneamente (Caso 3, com redes 3 e 4)

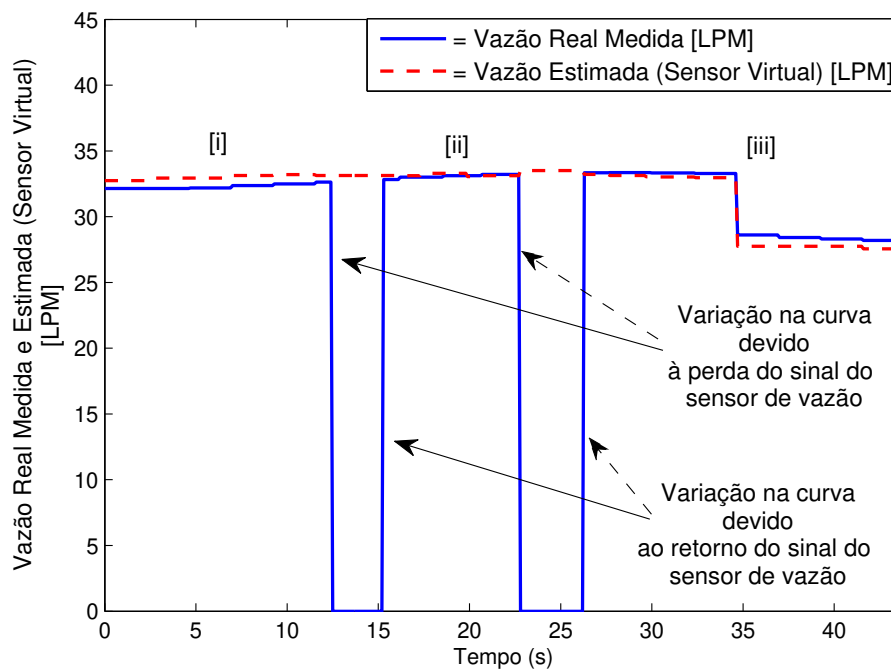
Outro experimento para o monitoramento da plataforma utilizando os sensores virtuais foi a retirada dos dois sensores reais simultaneamente. Com a perda dos sinais dos sensores reais, para que o processo não seja paralisado, as redes neurais, com base na entrada que está sendo aplicada (frequência), estimam os valores de pressão e vazão, fazendo com que o processo possa ser monitorado, utilizando os valores estimados (sensores virtuais).

Na Figura 5.14 pode-se observar o gráfico com o monitoramento dos valores de pressão (a) e vazão (b) durante o mesmo processo (observe que o eixo 'Tempo (s)' dos gráficos são os mesmos).

Figura 5.14: Monitoramento dos Valores de Pressão (a) e Vazão (b) Durante o Mesmo Processo (Redes 3 e 4).



(a)



(b)

Com o processo em andamento, sendo aplicada um valor de frequência de 54 Hz, foi realizada a retirada dos sinais dos sensores, em um primeiro momento o de vazão (após 10 segundos e retornando em aproximadamente 15 segundos), depois o de pressão

(após 15 segundos e retornando em aproximadamente 20 segundos), e por fim os dois simultaneamente (setas pontilhadas em (a) e (b)). Em seguida, foi alterado o valor de frequência que estava sendo aplicado de 54 Hz para 45 Hz, provocando a queda dos valores de pressão e vazão (é possível ser observada em aproximadamente 35 segundos no gráfico). Os gráficos da Figura 5.14 foram divididos em partes ([i], [ii] e [iii]) para a apresentação dos erros calculados nos momentos em que existiam ambos os sinais (real e estimado), para ser possível o cálculo do erro. Na Tabela 5.6 é possível observar os erros calculados.

Tabela 5.6: Erros Obtidos nas Estimativas dos Valores de Pressão e Vazão Aplicados na Plataforma Utilizando as Redes 3 e 4.

Erros Pressão (%)		
Parte i	Parte ii	Parte iii
2,66	0,74	1,80
Erros Vazão (%)		
Parte i	Parte ii	Parte iii
2,18	0,51	1,62

Para o cálculo dos erros relativos aos valores de pressão (Figura 5.14 (a)), a parte [i] vai do início do gráfico até o momento em que o sinal do sensor de pressão é perdido pela primeira vez, a parte [ii] vai do primeiro retorno do sinal do sensor (em aproximadamente 20 segundos) até o momento em que é perdido novamente, e a parte [iii] do momento em que o sinal do sensor é novamente retomado (após 25 segundos) até o fim do processo.

Para o cálculo dos erros relativos aos valores de vazão (Figura 5.14 (b)), a parte [i] vai do início do gráfico até o momento em que o sinal do sensor de vazão é perdido pela primeira vez, a parte [ii] vai do primeiro retorno do sinal do sensor (em aproximadamente 15 segundos) até o momento em que é perdido novamente, e a parte [iii] do momento em que o sinal do sensor é novamente retomado (após 25 segundos) até o fim do processo.

Observa-se que o momento de perda e retorno dos sinais dos sensores apontados com as setas pontilhadas é o mesmo. É o momento em que o processo fica sem a medição dos sensores reais, mas com as estimativas (sensores virtuais) das redes neurais.

Capítulo 6

Conclusões e Sugestões para Trabalhos Futuros

6.1 Conclusões Gerais

O objetivo principal nesta tese foi apresentar o desenvolvimento de sensores virtuais para monitoramento de variáveis de processos complexos não lineares e com múltiplas entradas e múltiplas saídas, utilizando redes neurais nas estimativas das variáveis. O intuito é ser possível ter um sistema monitorado utilizando o conceito de sensor virtual aliado às redes neurais nas estimativas, o que ainda é pouco difundido na literatura técnico científica, sendo uma área promissora de pesquisa.

Com a realização do estudo sobre os assuntos relacionados ao trabalho, foi possível observar a utilidade do conceito de sensor virtual no monitoramento de processos em geral, assim como um cenário da evolução e do estado atual do estudo da modelagem e identificação de sistemas não lineares aliado às redes neurais, verificando que são áreas de estudo promissoras e de grande importância para diversas aplicações, buscando entender e explicar fenômenos que ocorrem, por exemplo, na natureza, em sistemas sociais, biomédicos, sistemas de controle e monitoração, estimação, simulação e treinamento.

Além disso, com os experimentos realizados, foi possível utilizar sensores virtuais no monitoramento dos valores de pressão e vazão em uma plataforma de teste experimental (sistema de transporte fluídico) para validação dos mesmos. O objetivo é a possibilidade de utilização em diversas plantas, sem a necessidade de um modelo específico, qualquer sistema com valores de entradas e saídas que se relacionam no processo, ser possível implementar sensores virtuais para uma determinada aplicação, sendo necessário apenas adequações para cada caso. Assim, com um conjunto de redes neurais, ser possível obter sensores virtuais para processos multivariáveis.

Existem algumas áreas de atuações dos sensores virtuais que são importantes e podem

ser abordadas no estudo para monitoramento de processos em geral. Pode-se citar casos em que se deseja obter medidas de sensores em ambientes que são inacessíveis fisicamente, ou de complicado acesso. Outro caso seria o de confiabilidade, realizar a verificação do estado do sensor real, usar o sensor virtual monitorando o sensor real para ser possível identificar se existe algum problema. Um outro caso, abordado nesta tese, é o cenário em que se deseja que o sensor virtual acompanhe o mesmo comportamento do sensor real, que funcione idealmente igual ao sensor real de interesse. Com isso, pode-se ter cenários com plantas altamente complexas que precisam de uma instrumentação ampla, em que se tem gastos elevados pela necessidade de uma grande quantidade de sensores, verificando-se a necessidade de se ter processos monitorados com um menor número de sensores, consequentemente reduzindo custos.

Com os resultados obtidos utilizando os sensores virtuais desenvolvidos foi possível realizar o monitoramento dos valores das variáveis de interesse na plataforma de teste experimental utilizada garantindo o funcionamento do processo mesmo quando ocorre perda dos sinais dos sensores reais. O menor erro médio entre as medições dos sensores reais e dos sensores virtuais foi para o caso 3 (rede 4), com valor médio de 0,51%. E o maior erro médio obtido foi para o caso 1 (estimativa de pressão quando apenas um sinal de sensor real é perdido), com média de 4,49%. Os erros médios obtidos entre as medições foram considerados satisfatórios, por ser possível realizar o monitoramento sem comprometer o funcionamento do processo mesmo quando há perda de sinal de sensor real.

Como apresentado, mesmo com a perda dos sinais dos sensores, variação de frequência na entrada e variação na posição da válvula de controle, foi possível evitar que o processo fosse paralisado, continuando sua operação, com os valores estimados (sensores virtuais) enquanto os sinais dos sensores reais não eram recuperados.

Assim, a contribuição do trabalho foi o desenvolvimento de sensores virtuais para monitoramento de processos não lineares multivariáveis. O fato de não precisar do modelo matemático, das equações que regem a planta, é um ponto positivo, porque evita a complexidade inerente a processos não lineares multivariáveis. Com dados de entradas e saídas que se relacionam no processo é possível desenvolver sensores virtuais para a aplicação de interesse.

6.2 Sugestões para Trabalhos Futuros

A seguir são apresentadas as sugestões para continuação deste trabalho:

- utilizar outros tipos de redes neurais ou outros métodos de estimativas de variáveis. Com isso é possível realizar comparações de resultados, podendo obter sensores com

melhores desempenhos;

- utilizar mais de um sensor real da mesma variável de interesse a ser monitorada, de forma redundante, para ser possível a validação dos sensores virtuais, realizando o cálculo do erro quando houvesse perda do sinal do sensor real. Por exemplo, no momento em que são perdidos os sensores reais de pressão e vazão ao mesmo tempo, se houvesse um segundo sensor de pressão ou vazão, seria possível calcular o erro das estimativas para validação.
- implementar sensores virtuais que identifiquem quando o sensor real está defeituoso, não só quando o sinal é perdido completamente, mas também quando esteja fornecendo valores que não condizem com os valores corretos referentes à variável de interesse. Para isso seria necessário treinar a rede utilizando amostras coletadas de um sensor falho (descalibrado, por exemplo), para que houvesse o aprendizado de como o sensor defeituoso funciona.
- realizar o controle do processo dada uma perturbação no sistema, no caso da plataforma utilizada, essa perturbação pode ser feita utilizando as válvulas manuais.

Referências Bibliográficas

- [1] A. Abusnina. *Adaptive Soft Sensors in Industry*. Department of Computer Science Large Scale Complex IT Systems, Janeiro 2013. EngD Qualifying Dissertation.
- [2] Luis Antonio Aguirre. *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora da UFMG, Belo Horizonte, Brasil, 1 edition, 2000.
- [3] Ivan Arsie, Andrea Cricchioa, Matteo De Cesareb, Francesco Lazzarinia, Cesare Pianeasa, and Marco Sorrentinoa. Neural network models for virtual sensing of NOx emissions in automotive diesel engines with least square-based adaptation. *Control Engineering Practice*, 61:11–20, 2017.
- [4] Ashcroft. *Ashcroft Digital Industrial Gauge - Operating Instructions*, 2006. Disponível em: http://www.ashcroft.com/installationandmaintenance_pdf/upload/manual-digital-gauge-2074-2174-2274.pdf.
- [5] Christos G. Cassandras and Stéphane Lafortune. *Introduction to Discrete Event Systems*. (Springer Science Business Media, LLC, Nova York, USA, 2 edition, 2008.
- [6] Antonio Augusto Rodrigues Coelho and Leandro dos Santos Coelho. *Identificação de Sistemas Dinâmicos Lineares*. Editora da UFSC, Florianópolis, Brasil, 2004.
- [7] George Cybenko. *Approximation by superpositions of a sigmoid function*. Mathematics of Control Signals and Systems, 2 edition, 1989.
- [8] Jaidilson Jó da Silva. *Desenvolvimento de Técnicas de Detecção de Incrustação Utilizando Métodos Não-Invasivos*. Universidade Federal de Campina Grande, Campina Grande, Brasil, Dezembro 2008. Tese de Doutorado em Engenharia Elétrica.
- [9] Adriano da Silve Guahyba. *Utilização de Inteligência Artificial (Redes Neurais Artificiais) no Gerenciamento de Reprodutoras Pesadas*. Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil, 2001. Tese de Doutorado em Ciências Veterinárias.

- [10] Thamiles Rodrigues de Melo, Marcus Marinho Bezerra, Jaidilson Jó da Silva, and José Sérgio da Rocha Neto. Investigaç o de set-points para controle de uma plataforma de detecç o de incrusta o. *Anais do XIX Congresso Brasileiro de Autom tica, CBA.*, 2012.
- [11] Jos  Alfeu A. de S  Marques and Joaquim J. O. Sousa. F rmula de *Colebrook - White*: velha mas actual. soluç es expl citas. *III Silusba, S mposio de Hidr ulica e Recursos H dricos dos Pa ses de L ngua Oficial Portuguesa*, 1997.
- [12] Jos  Eli Santos dos Santos. *Controle Preditivo N o-Linear para Sistemas de Hammerstein*. Universidade Federal de Santa Catarina, Florian polis, Brasil, Abril 2007. Tese de Doutorado em Engenharia El trica.
- [13] Yuri A.W. Shardt e Xu Yang. Development of Soft Sensors for the Case Where the Time Delay is Random. *International Federation of Automatic Control (IFAC)*, 49:1193–1198, 2016.
- [14] Sandra Mauren Ell and Aldie Trabachini. Perda de carga em condutos forçados, 2011. Dispon vel em: <http://pt.scribd.com/doc/72710149/Perda-de-Carga-Tubulacao-Singular-Ida-Des>.
- [15] Andries P. Engelbrecht. *Computational Intelligence - An Introduction*. Wiley, University of Pretoria South Africa, 2 edition, 2007.
- [16] Robert W. Fox, Alan T. McDonald, and Philip J. Pritchard. *Introduç o   Mec nica dos Fluidos*. LTC, Rio de Janeiro, 7 edition, 2010.
- [17] G. D. Gonzalez. Soft sensors for processing plants. *Intelligent Processing and Manufacturing of Materials (IPMM)*, 1, 1999.
- [18] Simon Haykin. *Neural Networks and Learning Machines*. Prentice Hall, Ontario, Canada, 3 edition, 2009.
- [19] Miriam L. Galindo Hernandez, Ra l R. Leal Ascencio, and Cuauhtemoc Aguilera Galicia. An artificial neural network on a complex programmable logic device as a virtual sensor. In *International Symposium on Robotics*, pages 12–14, 1998.
- [20] Guang-Bin Huang. Learning capability and storage capacity os two-hidden-layer feedforward networks. *Transactions on Neural Networks*, 14, 2003.
- [21] Babu Joseph and Coleman Brosilow. Inferential control of processes: Part i, ii e iii. *American Institute of Chemical Engineers (AIChE Journal)*, 24:485–509, 1978.

- [22] Petr Kadlec, Bogdan Gabrys, and Sibylle Strandt. Data-driven soft sensors in the process industry. *Computers and Chemical Engineering*, pages 795–814, 2008.
- [23] Paul Kah, Pavel Layus, Esa Hiltunen, and Jukka Martikainen. Real-time weld process monitoring. *Advanced Materials Research*, 933:117–124, 2014.
- [24] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. J. Hudspeth. *Principles of Neural Science*. McGraw-Hill, 5 edition, 2013.
- [25] Bahram G. Kermani, Susan S. Schiffman, and H. Troy Nagle. Performance of the levenberg marquardt neural network training method in electronic nose applications. *Sensors and Actuators B: Chemical*, 110:13–22, 2005.
- [26] Hassan K. Khalil. *Nonlinear Systems*. Pearson Education Limited, Harlow, England, 3 edition, 2014.
- [27] K.P.Oliveira-Esquerre, M. Mori, and R.E. Bruns. Simulation of an industrial wastewater treatment plant using artificial neural networks and principal components analysis. *Brazilian Journal of Chemical Engineering*, 19:365–370, 2002.
- [28] KSB. *KSB - Pumps, Valves and Service*. KSB Bombas Hidráulicas S/A, 3 edition, 1991. Disponível em: <http://www.ksb.com/>.
- [29] Jianlin Li and Hongyun Zhang. Application of bp neural network to determine of mine water inrush sources based on matlab. *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, 2011.
- [30] Linhui Liu, Jie Chen, and Lixin Xu. Realization and application research of BP neural network based on matlab. *International Seminar on Future BioMedical Information Engineering*, 2008.
- [31] Lennart Ljung. System identification. *W.S. The Control Handbook, CRC Press & IEEE Press*, 1996.
- [32] Lennart Ljung and Torkel Glad. *Modeling of Dynamic Systems*. Prentice Hall, Englewood Cliffs, NJ, 1 edition, 1994.
- [33] A. M. Lopes, J. P. Lapa, and L. A. Oliveira. Unidade de transição de regime laminar a turbulento - guia de trabalho prático, 2006. Disponível em: <https://woc.uc.pt/dem/getFile.do?tipo=6&id=362>.
- [34] Francisco Antonio Lotufo. *Desenvolvimento de um Sensor Virtual para Processos Não-Lineares e Variantes no Tempo, com Aplicação em Planta de Neutralização de*

- pH*. Universidade Estadual Paulista, Guaratinguetá, Brasil, 2010. Tese de Doutorado em Engenharia Mecânica.
- [35] Salvatore Graziani e Maria Gabriella Xibilia Luigi Fortuna. *Soft Sensor for Monitoring and Control of Industrial Processes*. Editora Springer, 2007.
- [36] Raul K. Mansano, Eduardo P. Godoy, and Arthur J. V. Porto. The benefits of soft sensor and multi-rate control for the implementation of wireless networked control systems. *Sensors*, 14:24441–24461, 2014.
- [37] G.Ali Mansoori. Deposition and fouling of heavy organic oils and other compounds. *9th International Conference on Properties and Phases Equilibria for Product and Process Design*, 2001. Disponível em: www.gbhenterprises.com/japanconfpaper.pdf.
- [38] Angelos P. Markopoulos, Sotirios Georgiopoulos, and Dimitrios E. Manolakos. On the use of back propagation and radial basis function neural networks in surface roughness prediction. *Journal of Industrial Engineering International*, 12:389–400, Setembro 2016.
- [39] U. F. Monitors. *Turbine Flowmeters*, 2003. Disponível em: <http://www.flowmeters.com/ufm/index.cfm?task=turbine>.
- [40] J. A. N. Neto. Padrão 4-20 ma e Protocolo HART. 2011. Disponível em: <http://www.dee.ufcg.edu.br/~alves/4-20texto.pdf>.
- [41] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, São Paulo, Brasil, 5 edition, 2009.
- [42] Enrique Ortega. Cálculo da energia de atrito, 2012. Disponível em: http://www.unicamp.br/fea/ortega/aulas/aula05_fator_atrito.ppt.
- [43] Drazen Sliskovic e Petr Kadlec Ratko Grbic. Adaptive soft sensor for online prediction based on moving window gaussian process regression. *11th International Conference on Machine Learning and Applications*, 2012.
- [44] Sherief Reda, Ryan J. Cochran, and Abdullah Nazma Nowroz. Improved thermal tracking for processors using hard and soft sensor allocation techniques. *Transactions on Computers*, 60, Junho, 2011.
- [45] Sandhya Samarasinghe. *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. Auerbach Publications, Taylor and Francis Group, 1 edition, 2006.

- [46] Agus Saptoro. State of the art in the development of adaptive soft sensors based on just-in-time models. *International Conference and Workshop on Chemical Engineering*, 9:226–234, 2014.
- [47] Mitch Serpas, Yunfei Chu, and Juergen Hahn. Fault detection approach for systems involving soft sensors. *Journal of Loss Prevention in the Process Industries*, 26:443–452, Maio 2013.
- [48] Kristen Severson, Paphonwit Chaiwatanodom, and Richard D. Braatz. Perspectives on process monitoring of industrial systems. *International Federation of Automatic Control (IFAC)*, 48:931–939, 2015.
- [49] Georg Fischer Signet. *Signet 8550-1 Flow Transmitter*, 2011. Disponível em: <http://50.244.15.10/techlib/Signet/Signet%20Flow%20Transmitter%208550-1%20.pdf?>
- [50] Seema Singh, J. Harini, and B. R. Surabhi. A novel neural network based automated system for diagnosis of breast cancer from real time. In *International Conference on Circuits, Communication, Control and Computing (I4C)*. ACM, 2014.
- [51] Jiong Tang, Rajamani Doraiswami, and Chris P. Diduch. Identification of a linear model for nonlinear systems. *International Conference on Control and Automation*, pages 3959–3967, December 2009.
- [52] Hao Wang, Youngseok Oh, and En Sup Yoon. Strategies for modeling and control of nonlinear chemical processes using neural networks. *Computers & Chemical Engineering*, 22:832–862, 1998.
- [53] Leandro Osmar Werle. *Analisadores Virtuais Baseados em Modelo Neural para Monitoramento e Controle de Colunas de Destilação com Aquecimento Distribuído*. Universidade Federal de Santa Catarina, Florianópolis, 2012. Tese de Doutorado em Engenharia Química.
- [54] Bernard Widrow and Samuel D. Stearns. *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 2 edition, 1985.
- [55] Jon Wilson. Pressure measurement: Principles and practice, 2003. Disponível em: <http://www.sensormag.com/sensors/pressure/pressure-measurement-principles-and-practice-969>.
- [56] Furong Gao e Zhihuan Song Zhiqiang Ge. Mixture probabilistic pcr model for soft sensing of multimode processes. *Chemometrics and Intelligent Laboratory Systems*, 105:91–105, 2011.

Apêndice A

Erros Médios Obtidos nas Estimativas dos Valores de Pressão (Rede 1)

Tabelas completas com os erros médios obtidos para cada configuração testada para as redes do tipo *feedforward* e *feedforward* em Cascata nas estimativas dos valores de pressão quando apenas um dos sinais de sensores são perdidos.

- A.1: *feedforward* com 70% para treino e 30% para teste.
- A.2: *feedforward* em cascata com 70% para treino e 30% para teste.
- A.3: *feedforward* com 75% para treino e 25% para teste.
- A.4: *feedforward* em cascata com 75% para treino e 25% para teste.
- A.5: *feedforward* com 80% para treino e 20% para teste.
- A.6: *feedforward* em cascata com 80% para treino e 20% para teste.

Tabela A.1: Erros obtidos com Rede *Feedforward* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 1).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		4,46	4,62	5,31	4,47	4,82	5,23	4,44	4,46	4,53	4,44	4,46	4,53
2		4,38	4,79	4,68	4,79	4,43	4,80	4,51	4,45	4,43	4,51	4,45	4,43
3		4,40	4,64	4,56	4,43	4,62	4,41	4,90	4,54	4,40	4,90	4,54	4,40
4		4,49	4,58	5,60	4,70	4,44	4,85	5,05	4,47	4,91	5,05	4,47	4,91
5		4,85	4,71	4,63	4,79	4,65	4,49	4,90	4,48	4,59	4,90	4,48	4,59
6		4,86	4,57	4,36	5,10	4,61	4,71	4,77	4,61	4,79	4,77	4,61	4,79
7		4,67	4,77	4,82	4,66	4,43	4,78	5,80	4,83	4,60	5,80	4,83	4,60
8		4,74	5,10	5,03	4,99	4,69	4,66	4,99	4,58	4,59	4,99	4,58	4,59
9		5,03	4,50	4,76	4,77	4,98	4,69	5,46	4,65	4,64	5,46	4,65	4,64
10		4,77	4,57	4,96	5,20	4,71	5,17	4,94	4,83	4,60	4,94	4,83	4,60

Tabela A.2: Erros obtidos com Rede *Feedforward em Cascata* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 1).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		4,53	4,59	4,50	4,55	4,43	4,55	4,44	4,61	4,44	4,44	4,61	4,63
2		4,53	4,45	4,86	4,56	4,49	4,44	4,56	4,48	4,56	4,56	4,48	4,52
3		4,93	4,49	4,49	4,49	4,47	4,66	4,58	4,61	4,58	4,58	4,61	4,64
4		4,47	4,54	4,61	4,68	4,47	4,46	4,53	4,53	4,53	4,53	4,53	4,46
5		4,49	4,43	4,55	4,87	4,60	4,50	4,49	4,52	4,49	4,49	4,52	4,45
6		4,94	4,57	4,71	4,89	4,54	4,75	4,99	4,55	4,99	4,99	4,55	4,41
7		5,01	4,59	4,65	5,02	4,46	4,68	5,06	4,63	5,06	5,06	4,63	4,52
8		4,99	4,81	4,50	5,11	4,52	4,63	5,06	4,51	5,06	5,06	4,51	4,59
9		4,83	4,48	4,32	4,49	4,59	4,67	4,91	4,76	4,67	4,91	4,76	4,61
10		4,54	4,55	4,81	4,77	4,76	4,53	4,87	4,52	4,53	4,87	4,52	4,83

Tabela A.3: Erros obtidos com Rede *Feedforward* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 1).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		3,74	4,32	4,28	3,70	4,02	4,19	3,69	3,72	3,87			
2		3,74	4,80	6,23	3,67	3,56	3,78	3,70	3,66	3,91			
3		3,66	3,72	3,54	3,63	3,74	3,66	3,81	3,59	4,12			
4		3,65	4,96	5,96	3,80	3,67	3,78	3,71	3,71	3,60			
5		3,99	3,55	3,77	3,72	3,77	3,99	3,78	3,66	3,80			
6		3,67	4,35	4,29	3,80	3,67	3,72	5,14	3,66	3,71			
7		3,97	3,80	3,79	3,94	3,82	3,81	3,74	3,72	3,73			
8		3,78	3,86	3,71	4,07	3,63	3,83	4,13	3,66	3,69			
9		3,87	3,87	3,95	4,06	3,72	3,98	4,16	3,75	3,96			
10		3,93	3,61	3,74	3,92	4,02	4,63	3,84	3,85	4,01			

Tabela A.4: Erros obtidos com Rede *Feedforward em Cascata* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 1).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		3,66	3,83	3,72	3,74	3,81	3,73	3,67	3,62	3,67	3,67	3,62	3,79
2		3,78	3,69	3,89	3,68	3,69	3,71	3,69	3,67	3,69	3,69	3,67	3,66
3		3,67	3,70	3,71	3,73	3,69	3,75	3,91	3,77	3,91	3,77	3,77	3,71
4		3,68	3,70	4,07	3,70	3,69	3,67	3,78	3,72	3,78	3,72	3,72	3,74
5		3,74	3,71	4,17	4,17	3,68	3,77	3,85	3,78	3,85	3,78	3,78	3,87
6		3,80	4,02	4,07	3,73	3,87	3,99	3,96	3,69	3,96	3,69	3,69	3,72
7		3,79	3,75	3,50	3,89	3,67	4,26	3,89	3,69	3,89	3,69	3,69	3,71
8		3,81	3,88	3,75	3,99	3,70	3,70	4,25	3,67	4,25	3,67	3,67	3,62
9		3,91	3,89	4,05	3,97	3,76	4,01	4,27	3,68	4,27	3,68	3,68	4,35
10		3,90	4,21	3,65	3,92	3,62	3,88	4,18	3,61	4,18	3,61	3,61	3,69

Tabela A.5: Erros obtidos com Rede *Feedforward* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 1).

		Erros (%)								
		<i>Feedforward</i>								
		30 Epochs			50 epochs			100 Epochs		
		Função de Treinamento			Função de Treinamento			Função de Treinamento		
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		3,03	3,42	5,38	2,95	3,23	3,70	2,92	3,17	3,75
2		2,85	3,83	3,59	2,71	2,70	3,68	2,90	2,90	2,88
3		2,77	3,05	3,26	2,85	3,01	3,18	2,70	2,71	2,92
4		2,68	7,88	3,59	2,67	2,78	2,75	2,76	2,90	2,86
5		2,70	2,78	3,00	2,69	2,97	2,91	2,65	2,63	3,01
6		2,67	2,75	2,96	2,82	2,74	3,10	2,75	2,70	2,86
7		2,69	2,87	3,16	2,68	2,82	3,29	2,63	2,71	2,74
8		2,70	2,96	5,06	2,63	2,70	2,78	2,68	2,71	2,70
9		2,66	3,00	2,84	2,70	2,68	2,90	2,69	2,94	2,73
10		2,76	2,75	3,02	2,62	2,92	2,75	2,79	2,92	2,98

Tabela A.6: Erros obtidos com Rede *Feedforward em Cascata* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 1).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		2,84	2,94	2,89	2,91	2,85	2,94	2,88	2,94	2,88	2,94	2,91	2,91
2		2,89	2,90	3,01	2,89	2,90	2,94	2,72	2,88	2,81	2,72	2,90	2,89
3		2,89	2,93	2,87	2,72	2,88	2,81	2,69	2,89	2,95	2,69	3,17	2,93
4		2,71	2,85	2,96	2,67	2,89	2,84	2,72	2,87	2,84	2,72	2,92	2,84
5		2,65	2,89	3,17	2,70	2,87	2,93	2,67	2,87	2,84	2,67	2,76	2,78
6		2,68	2,97	3,10	2,70	2,95	2,93	2,71	2,95	2,93	2,71	2,76	2,86
7		2,81	2,90	2,96	2,80	2,79	2,68	2,68	2,79	2,68	2,68	2,75	3,12
8		2,64	2,91	2,98	2,68	2,86	3,15	2,69	2,86	3,15	2,69	2,98	2,78
9		2,66	2,82	3,06	2,66	2,76	2,81	2,68	2,76	2,81	2,68	2,71	2,89
10		2,69	2,91	3,44	2,66	2,74	2,85	2,88	2,74	2,85	2,88	2,85	2,92

Apêndice B

Erros Médios Obtidos nas Estimativas dos Valores de Vazão (Rede 2)

Tabelas completas com os erros médios obtidos para cada configuração testada para as redes do tipo *feedforward* e *feedforward* em Cascata nas estimativas dos valores de vazão quando apenas um dos sinais de sensores são perdidos.

- B.1: *feedforward* com 70% para treino e 30% para teste.
- B.2: *feedforward* em cascata com 70% para treino e 30% para teste.
- B.3: *feedforward* com 75% para treino e 25% para teste.
- B.4: *feedforward* em cascata com 75% para treino e 25% para teste.
- B.5: *feedforward* com 80% para treino e 20% para teste.
- B.6: *feedforward* em cascata com 80% para treino e 20% para teste.

Tabela B.1: Erros obtidos com Rede *Feedforward* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 2).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		2,25	3,39	2,63	2,26	2,59	2,49	2,23	2,32	2,52	2,23	2,32	2,52
2		2,18	4,61	2,93	2,18	2,17	2,26	2,36	2,18	2,25	2,36	2,18	2,25
3		2,14	2,20	2,20	2,14	2,96	3,05	2,23	2,19	2,29	2,23	2,19	2,29
4		2,23	2,18	2,59	2,18	2,33	2,25	2,13	2,15	2,48	2,13	2,15	2,48
5		2,19	2,21	2,26	2,11	2,23	2,12	3,33	2,16	2,24	3,33	2,16	2,24
6		3,17	2,06	2,17	2,15	2,23	2,28	3,30	2,18	2,20	3,30	2,18	2,20
7		2,41	2,23	2,52	2,59	2,09	2,41	2,35	2,28	2,29	2,35	2,28	2,29
8		2,80	2,13	2,22	2,95	2,10	2,06	4,18	2,07	2,23	4,18	2,07	2,23
9		3,26	2,37	2,02	2,85	2,36	2,39	2,84	2,13	2,26	2,84	2,13	2,26
10		3,98	2,21	2,37	3,85	2,23	2,14	3,01	2,21	2,21	3,01	2,21	2,21

Tabela B.2: Erros obtidos com Rede *Feedforward em Cascata* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 2).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		2,17	2,39	2,27	2,16	2,31	2,46	2,23	2,20	2,25	2,23	2,20	2,25
2		2,13	2,35	2,25	2,19	2,25	2,39	2,17	2,24	2,16	2,17	2,24	2,16
3		2,21	2,20	2,25	2,17	2,21	2,34	3,60	2,21	2,19	3,60	2,21	2,19
4		2,18	2,22	2,18	3,68	2,17	2,39	3,90	2,24	2,18	3,90	2,24	2,18
5		2,98	2,21	2,50	3,27	2,17	2,20	2,79	2,14	2,22	2,79	2,14	2,22
6		2,18	2,25	2,31	2,15	2,16	2,37	3,53	2,12	2,15	3,53	2,12	2,15
7		2,10	2,21	2,29	2,77	2,05	2,17	2,84	2,37	2,18	2,84	2,37	2,18
8		3,33	2,16	2,20	2,83	2,07	2,23	2,16	2,13	2,17	2,16	2,13	2,17
9		2,64	2,21	2,07	3,45	2,14	2,26	3,22	2,07	2,26	3,22	2,07	2,26
10		2,78	2,19	1,97	3,39	2,28	2,30	2,98	2,34	2,29	2,98	2,34	2,29

Tabela B.3: Erros obtidos com Rede *Feedforward* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 2).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		1,74	1,99	2,20	1,75	1,87	2,10	1,75	1,87	2,10	1,75	1,81	1,91
2		1,73	1,91	3,00	1,76	1,73	1,85	1,74	1,73	1,85	1,74	1,89	1,76
3		1,75	1,79	1,84	1,68	1,75	1,72	1,70	1,75	1,72	1,70	1,79	1,75
4		1,70	1,96	1,99	1,77	1,76	1,78	1,68	1,76	1,78	1,68	1,71	1,74
5		2,69	2,37	1,83	1,79	1,74	1,79	1,97	1,74	1,79	1,97	1,82	1,75
6		1,65	1,79	1,82	1,74	1,79	1,91	1,70	1,79	1,91	1,70	1,81	1,79
7		1,97	1,75	1,93	1,87	1,67	1,82	1,77	1,67	1,82	1,77	1,73	1,73
8		2,57	1,80	1,90	1,86	1,93	1,70	3,23	1,93	1,70	3,23	1,84	1,77
9		1,96	1,77	1,93	2,95	1,67	2,03	2,90	1,67	2,03	2,90	1,86	1,80
10		1,98	2,24	1,79	1,87	1,70	2,18	2,68	1,70	2,18	2,68	1,68	1,77

Tabela B.4: Erros obtidos com Rede *Feedforward em Cascata* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 2).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		1,72	1,92	1,80	1,71	1,85	1,96	1,91	1,73	1,95	1,91	1,73	1,95
2		1,73	1,70	1,75	1,70	1,81	1,66	1,81	1,76	1,72	1,81	1,76	1,72
3		1,71	1,85	1,96	7,32	1,74	1,87	1,69	1,90	1,77	1,69	1,90	1,77
4		1,76	1,74	1,91	1,82	1,71	1,74	1,90	1,73	1,71	1,90	1,73	1,71
5		2,82	1,79	1,80	2,39	1,64	1,81	3,20	1,73	1,69	3,20	1,73	1,69
6		1,69	1,81	2,06	2,22	1,72	1,94	2,65	1,74	1,84	2,65	1,74	1,84
7		2,16	1,63	1,85	1,70	1,79	1,74	2,26	1,71	1,74	2,26	1,71	1,74
8		1,73	1,72	1,76	2,10	1,75	1,76	3,11	1,71	1,75	3,11	1,71	1,75
9		2,11	1,80	1,77	2,99	1,68	1,81	2,93	1,90	1,78	2,93	1,90	1,78
10		2,93	2,35	1,91	2,90	1,75	1,72	3,28	1,67	1,68	3,28	1,67	1,68

Tabela B.5: Erros obtidos com Rede *Feedforward* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 2).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		1,43	4,20	3,34	1,43	1,48	1,59	1,42	1,45	1,42	1,45	1,45	1,58
2		1,42	1,66	1,67	1,45	1,60	1,53	1,42	1,45	1,42	1,42	1,42	1,52
3		1,42	1,44	2,22	1,41	1,44	2,26	1,45	1,45	1,45	1,45	1,53	1,48
4		1,42	1,49	1,68	1,46	1,47	1,53	1,35	1,46	1,35	1,44	1,44	1,46
5		1,39	1,48	2,33	1,40	1,46	1,56	1,40	1,40	1,40	1,40	1,47	1,45
6		1,53	1,53	2,17	1,44	1,53	1,42	1,39	1,44	1,39	1,39	1,64	1,52
7		1,39	1,45	1,78	1,41	1,47	1,57	1,38	1,41	1,38	1,38	1,44	1,53
8		1,39	1,48	1,41	1,39	1,39	1,40	1,38	1,39	1,38	1,38	1,42	1,52
9		1,40	1,44	1,65	1,55	1,50	1,44	1,58	1,55	1,58	1,42	1,42	1,39
10		1,37	1,59	1,49	1,55	1,48	1,36	1,38	1,55	1,38	1,46	1,46	1,40

Tabela B.6: Erros obtidos com Rede *Feedforward em Cascata* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 2).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		1,37	1,52	1,52	1,43	1,38	1,49	1,39	1,39	1,39	1,39	1,39	1,49
2		1,42	1,40	1,44	1,40	1,43	1,41	1,49	1,42	1,42	1,49	1,42	1,38
3		1,44	1,43	1,56	1,45	1,43	1,53	1,44	1,43	1,43	1,44	1,43	1,38
4		1,40	1,40	1,46	1,57	1,42	1,39	1,40	1,42	1,42	1,40	1,42	1,43
5		1,48	1,38	1,50	1,40	1,40	1,44	1,49	1,40	1,44	1,49	1,43	1,43
6		1,40	1,40	1,91	1,38	1,40	1,44	1,42	1,40	1,44	1,42	1,40	1,44
7		1,41	1,40	1,43	1,38	1,39	1,48	1,55	1,38	1,48	1,55	1,38	1,41
8		1,39	1,40	1,39	1,55	1,49	1,43	1,64	1,49	1,43	1,64	1,42	1,59
9		1,42	1,46	1,56	1,42	1,36	1,46	1,56	1,42	1,46	1,56	1,39	1,55
10		1,50	1,38	1,47	1,41	1,44	1,41	1,38	1,44	1,41	1,38	1,38	1,43

Apêndice C

Medições para Obtenção do Conjunto de Dados (Redes 3 e 4)

C.1: Resultados dos Testes Realizados (Frequência (tensão) como Entrada) para a coleta dos valores de pressão e Vazão na obtenção do conjunto de dados utilizado no treinamento e teste das redes.

C.2: Resultado da média e do desvio Padrão dos valores obtidos com a repetição de medições apresentadas na Tabela C.1.

C.3: Valores de Pressão Reais Medidos e Valores Estimados com as Redes *Feedforward* e *Feedforward* em Cascata (Rede 3).

C.4: Valores de Vazão Reais Medidos e Valores Estimados com as Redes *Feedforward* e *Feedforward* em Cascata (Rede 4).

C.5: Erros Obtidos Entre Valores Reais e Estimados de Pressão e Vazão com as Redes 3 e 4.

Tabela C.1: Valores Obtidos nas Medições Para a Coleta do Conjunto de Dados.

Frequência (Hz)	Pressão (mBar)			Vazão (LPM)		
	Teste 1	Teste 2	Teste 3	Teste 1	Teste 2	Teste 3
20	6,30	6,28	6,32	11,18	10,99	11,10
23	10,58	10,60	10,59	13,16	12,95	13,04
24	12,00	12,15	12,19	13,60	14,02	13,93
25	14,54	14,49	14,50	15,00	14,61	13,92

Continua na próxima página

26	16,16	16,17	16,12	15,10	15,50	15,03
28	19,75	19,79	19,80	16,31	16,63	16,71
29	22,30	22,33	22,27	17,12	17,35	17,37
30	28,59	28,58	28,63	17,72	17,79	17,59
33	35,32	35,29	35,29	20,00	19,56	19,54
35	41,17	41,21	41,22	21,35	20,99	20,96
36	42,90	42,94	42,92	21,70	21,89	21,66
37	46,10	45,95	46,10	22,96	22,52	22,53
38	49,30	48,90	48,80	23,39	23,10	22,81
39	51,60	51,80	52,30	23,91	23,74	23,75
40	55,90	56,00	54,90	24,49	24,45	24,26
42	61,90	61,80	62,60	25,65	25,97	25,78
44	67,00	67,20	66,86	26,62	26,97	27,29
45	72,10	71,82	70,58	27,84	27,86	27,40
46	75,00	75,40	75,26	28,46	28,31	28,43
47	78,00	77,30	76,51	29,10	28,83	28,89
48	80,71	79,90	81,10	29,97	30,01	29,03
50	89,10	89,00	88,00	30,61	30,91	30,88
52	96,00	96,10	97,40	32,42	32,36	31,73
53	99,20	99,60	100,60	32,92	32,75	32,75
54	101,50	101,80	101,50	33,43	33,35	33,12
55	105,97	106,10	105,96	34,12	34,23	33,80
56	111,50	111,30	110,20	34,89	34,76	34,75
57	113,90	114,00	114,70	35,90	35,87	34,76
58	117,52	117,90	116,87	36,00	35,86	36,17
60	124,40	124,80	123,10	37,23	36,78	36,84

Tabela C.2: Média e Desvio Padrão dos Valores Obtidos com as Medições.

Frequência [Hz]	Média Aritmética Pressão [mBar]	Desvio Padrão (Pressão)	Média Aritmética Vazão [LPM]	Desvio Padrão (Vazão)
20	6,30	0,02	11,09	0,10
23	10,59	0,01	13,05	0,11
24	12,11	0,10	13,85	0,22
25	14,51	0,03	14,51	0,55
26	16,15	0,03	15,21	0,25
28	19,78	0,03	16,55	0,21
29	22,30	0,03	17,28	0,14
30	28,60	0,03	17,70	0,10
33	35,30	0,02	19,70	0,26
35	41,20	0,03	21,10	0,22
36	42,92	0,02	21,75	0,12
37	46,05	0,09	22,67	0,25
38	49,00	0,26	23,10	0,29
39	51,90	0,36	23,80	0,10
40	55,60	0,61	24,40	0,12
42	62,10	0,44	25,80	0,16
44	67,02	0,17	26,96	0,34
45	71,50	0,81	27,70	0,26
46	75,22	0,20	28,40	0,08
47	77,27	0,75	28,94	0,14
48	80,57	0,61	29,67	0,55
50	88,70	0,61	30,80	0,17
Continua na próxima página				

52	96,50	0,78	32,17	0,38
53	99,80	0,72	32,81	0,10
54	101,60	0,17	33,30	0,16
55	106,01	0,08	34,05	0,22
56	111,00	0,70	34,80	0,08
57	114,20	0,44	35,51	0,65
58	117,43	0,52	36,01	0,16
60	124,10	0,89	36,95	0,24

Tabela C.3: Valores de Pressão Reais Medidos e Valores Estimados com as Redes *Feedforward* e *Feedforward* em Cascata (Rede 3).

Frequência (Hz)	Valores Reais Medidos de Pressão (mBar)	Valores de Pressão Estimados Rede Feedforward (mBar)	Valores de Pressão Estimados Rede Feedforward em Cascata (mBar)
20	6,30	6,28	6,36
23	10,59	10,70	10,33
24	12,11	12,03	12,42
25	14,51	14,36	14,50
26	16,15	16,35	16,07
28	19,78	19,23	19,75
29	22,30	22,95	22,85
30	28,60	28,16	27,95
33	35,30	34,50	35,84
35	41,20	40,98	41,21
36	42,92	43,90	43,01

Continua na próxima página

37	46,05	45,39	45,69
38	49,00	49,12	48,94
39	51,90	52,12	51,95
40	55,60	55,07	54,99
42	62,10	61,58	61,25
44	67,02	67,87	68,24
45	71,50	70,95	72,00
46	75,22	72,67	73,97
47	77,27	78,67	78,39
48	80,57	81,97	80,58
50	88,70	86,06	87,77
52	96,50	96,22	94,70
53	99,80	100,43	99,01
54	101,60	102,43	103,11
55	106,01	104,46	106,38
56	111,00	110,70	109,73
57	114,20	112,57	111,75
58	117,43	117,45	117,67
60	124,10	124,14	124,11

Tabela C.4: Valores de Vazão Reais Medidos e Valores Estimados com as Redes *Feedforward* e *Feedforward* em Cascata (Rede 4).

Frequência (Hz)	Valores Reais Medidos de Vazão (LPM)	Valores de Vazão Estimados Rede Feedforward (LPM)	Valores de Vazão Estimados Rede Feedforward em Cascata (LPM)
20	11,09	11,11	11,07
23	13,05	13,05	13,11
24	13,85	13,76	13,80
25	14,51	14,54	14,48
26	15,21	15,16	15,16
28	16,55	16,49	16,49
29	17,28	17,13	17,15
30	17,70	17,77	17,80
33	19,70	19,67	19,75
35	21,10	21,09	21,14
36	21,75	21,63	21,68
37	22,67	22,29	22,33
38	23,10	23,12	23,13
39	23,80	23,62	23,61
40	24,40	24,28	24,25
42	25,80	25,60	25,53
44	26,96	26,90	26,81
45	27,70	27,74	27,71
46	28,40	28,16	28,09
47	28,94	28,98	28,99
48	29,67	29,40	29,37
Continua na próxima página			

50	30,80	30,61	30,64
52	32,17	31,83	31,91
53	32,81	32,45	32,54
54	33,30	33,10	33,16
55	34,05	33,75	33,79
56	34,80	34,79	34,78
57	35,51	35,09	35,03
58	36,06	35,98	35,98
60	36,95	36,93	36,86

Tabela C.5: Erros Obtidos Entre Valores Reais e Estimados de Pressão e Vazão com as Redes 3 e 4.

Frequência (Amostras de Teste (Hz))	Erro para a Rede 3 Pressão (%)		Erro para a Rede 4 Vazão (%)	
	<i>Feedforward</i>	<i>Feedforward em Cascata</i>	<i>Feedforward</i>	<i>Feedforward em Cascata</i>
20	0,32	0,95	0,18	0,18
23	1,06	2,41	0,01	0,48
24	0,67	2,55	0,63	0,33
25	1,03	0,07	0,21	0,21
26	1,24	0,47	0,30	0,33
28	2,78	0,13	0,37	0,37
29	2,91	2,47	0,86	0,77
30	1,56	2,27	0,37	0,58

Continua na próxima página

33	2,26	1,53	0,13	0,26
35	0,53	0,02	0,05	0,19
36	2,28	0,21	0,56	0,30
37	1,44	0,79	1,69	1,51
38	0,24	0,12	0,09	0,13
39	0,42	0,09	0,78	0,80
40	0,96	1,09	0,49	0,61
42	0,83	1,37	0,78	1,03
44	1,26	1,81	0,24	0,55
45	0,77	0,70	0,14	0,04
46	3,38	1,66	0,83	1,09
47	1,81	1,45	0,14	0,17
48	1,73	0,01	0,91	1,01
50	1,85	1,05	0,62	0,52
52	0,29	1,87	1,06	0,82
53	0,63	0,79	1,08	0,83
54	0,81	1,48	0,62	0,41
55	1,46	0,35	0,88	0,77
56	0,27	1,14	0,03	0,06
57	1,43	2,14	1,20	1,36
58	0,02	0,20	0,22	0,22
60	0,03	0,01	0,06	0,25

Apêndice D

Erros Médios Obtidos nas Estimativas dos Valores de Pressão (Rede 3)

Tabelas completas com os erros médios obtidos para cada configuração testada para a rede do tipo *feedforward* e *feedforward* em Cascata nas estimativas dos valores de pressão quando os dois sinais de sensores são perdidos simultaneamente.

- D.1: *feedforward* com 70% para treino e 30% para teste.
- D.2: *feedforward* em cascata com 70% para treino e 30% para teste.
- D.3: *feedforward* com 75% para treino e 25% para teste.
- D.4: *feedforward* em cascata com 75% para treino e 25% para teste.
- D.5: *feedforward* com 80% para treino e 20% para teste.
- D.6: *feedforward* em cascata com 80% para treino e 20% para teste.

Tabela D.1: Erros obtidos com Rede *Feedforward* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 3).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsec</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsec</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsec</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsec</i>	<i>trainoss</i>
1		3,21	11,59	8,46	3,39	6,13	5,31	3,29	4,62	9,97			
2		4,60	93,70	83,92	12,74	5,83	76,25	1,09	50,86	115,46			
3		3,03	49,60	29,81	5,60	14,99	11,40	0,78	30,07	22,35			
4		1,34	38,00	40,37	0,98	8,36	37,00	1,91	35,38	16,63			
5		1,61	15,79	6,91	0,99	14,37	9,44	4,07	9,31	35,20			
6		0,84	14,24	17,46	1,04	6,25	4,18	1,73	10,33	8,79			
7		6,46	4,90	48,38	2,81	3,27	26,84	1,29	27,89	11,30			
8		2,48	15,79	21,85	12,24	6,78	5,87	2,00	3,24	14,96			
9		2,34	4,80	21,20	2,38	6,10	7,35	1,39	33,64	1,87			
10		38,50	6,79	22,50	8,11	41,57	40,35	8,57	14,62	2,38			

Tabela D.2: Erros obtidos com Rede *Feedforward em Cascata* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Pressão com Rede 3).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		2,84	6,18	9,36	6,16	3,74	4,38	10,92	5,51	3,59			
2		1,34	3,38	1,42	1,08	7,13	7,66	1,99	2,40	19,41			
3		5,52	22,54	2,90	2,31	10,93	14,82	2,58	6,08	34,72			
4		1,39	37,31	8,07	1,68	4,37	13,95	1,23	16,89	6,51			
5		1,62	6,78	16,15	1,22	5,41	69,27	1,28	24,23	28,13			
6		1,24	10,14	46,41	1,12	10,60	30,03	0,87	8,29	13,10			
7		0,92	13,28	61,11	9,34	37,34	4,05	1,06	10,02	59,54			
8		1,75	7,58	16,60	19,67	12,49	11,79	2,26	15,93	12,54			
9		1,32	4,28	8,24	3,02	13,95	7,21	27,27	9,86	4,95			
10		0,82	12,04	29,09	1,93	3,13	32,65	1,51	9,77	35,27			

Tabela D.3: Erros obtidos com Rede *Feedforward* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 3).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		3,65	3,11	9,72	3,62	3,59	23,06	4,03	13,67	5,15			
2		7,03	11,72	43,52	4,23	26,24	50,13	3,10	9,57	69,05			
3		2,06	37,66	33,80	2,40	58,52	12,00	2,15	23,95	38,36			
4		1,66	12,10	15,25	1,25	16,22	94,95	1,10	7,60	27,12			
5		0,84	3,61	21,68	1,70	17,21	18,96	1,23	10,85	3,72			
6		21,05	7,03	46,50	36,99	21,56	17,36	4,44	5,06	18,40			
7		7,22	16,16	9,32	1,37	43,16	18,77	47,42	20,23	9,71			
8		28,00	18,48	22,62	4,45	8,10	5,63	1,58	5,24	61,23			
9		2,25	8,80	20,13	8,48	3,84	13,98	6,79	2,44	14,87			
10		1,34	20,27	11,76	0,61	3,16	24,50	5,36	9,93	18,22			

Tabela D.4: Erros obtidos com Rede *Feedforward em Cascata* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Pressão com Rede 3).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		9,60	3,00	3,35	4,15	3,19	6,28	7,68	9,40	24,44			
2		2,84	50,68	12,56	1,38	9,36	26,38	1,17	101,67	27,53			
3		5,11	8,72	5,60	4,36	32,87	6,62	11,91	3,74	4,09			
4		1,55	10,56	14,13	3,04	3,52	6,42	7,31	11,38	16,48			
5		1,95	5,68	57,77	1,07	4,82	10,10	1,75	75,86	14,71			
6		4,02	10,48	10,76	12,55	9,86	35,20	0,92	4,33	16,32			
7		11,59	17,94	88,43	1,24	3,37	55,17	7,20	11,46	24,45			
8		25,23	6,57	45,99	0,59	20,12	3,29	0,78	4,33	7,21			
9		3,10	2,51	31,82	1,41	3,52	5,69	2,68	2,92	19,80			
10		0,80	20,93	23,93	6,61	40,08	6,97	1,75	2,19	10,07			

Tabela D.5: Erros obtidos com Rede *Feedforward* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 3).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		4,15	34,91	14,29	5,08	58,34	20,93	4,34	14,81	10,41			
2		1,68	11,05	13,60	1,35	11,00	32,56	15,55	54,33	46,03			
3		3,03	25,72	56,84	1,36	8,81	23,46	1,78	30,97	57,59			
4		1,63	19,25	84,04	1,57	9,96	26,00	1,65	16,94	26,08			
5		0,85	9,00	70,66	1,35	43,01	16,21	1,09	15,75	17,92			
6		5,57	9,44	18,95	1,39	14,20	29,86	5,86	12,41	39,92			
7		1,21	7,74	14,93	27,78	12,77	3,25	1,02	20,36	55,68			
8		2,22	54,05	57,70	4,95	10,21	4,16	6,63	15,58	22,27			
9		2,03	20,84	17,27	9,95	18,17	33,11	4,66	16,24	1,73			
10		6,77	32,35	72,40	1,11	10,26	35,05	0,67	21,67	25,78			

Tabela D.6: Erros obtidos com Rede *Feedforward em Cascata* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Pressão com Rede 3).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		9,47	5,00	6,31	7,81	5,81	9,56	7,06	5,99	19,90			
2		1,24	48,07	18,08	1,08	12,41	5,50	36,05	2,11	8,84			
3		9,10	55,78	18,90	1,69	12,45	95,02	1,52	3,87	6,76			
4		1,59	9,49	16,16	5,10	27,48	16,98	1,58	12,33	42,98			
5		1,81	23,94	15,49	1,27	8,09	12,43	5,05	1,69	8,25			
6		0,79	10,16	22,46	1,72	2,69	72,30	5,85	79,73	40,05			
7		1,04	21,88	13,51	1,21	19,41	40,34	1,12	41,48	5,95			
8		2,05	3,70	15,40	19,24	9,60	30,35	1,69	23,87	112,20			
9		1,50	14,46	120,33	3,72	1,26	21,16	33,87	49,95	12,45			
10		10,41	30,86	27,82	1,30	18,85	34,09	1,05	39,23	9,40			

Apêndice E

Erros Médios Obtidos nas Estimativas dos Valores de Vazão (Rede 4)

Tabelas completas com os erros médios obtidos para cada configuração testada para a rede do tipo *feedforward* e *feedforward* em Cascata nas estimativas dos valores de vazão quando os dois sinais de sensores são perdidos simultaneamente.

- E.1: *feedforward* com 70% para treino e 30% para teste.
- E.2: *feedforward* em cascata com 70% para treino e 30% para teste.
- E.3: *feedforward* com 75% para treino e 25% para teste.
- E.4: *feedforward* em cascata com 75% para treino e 25% para teste.
- E.5: *feedforward* com 80% para treino e 20% para teste.
- E.6: *feedforward* em cascata com 80% para treino e 20% para teste.

Tabela E.1: Erros obtidos com Rede *Feedforward* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 4).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		0,75	1,75	5,22	2,47	3,24	2,64	2,23	3,25	8,07			
2		0,31	22,11	13,96	0,69	2,88	6,70	0,32	20,97	4,87			
3		0,73	10,75	12,36	0,24	6,96	4,77	0,62	9,66	5,09			
4		0,20	3,89	2,92	1,30	5,06	2,82	0,32	11,90	2,62			
5		0,76	1,42	3,56	1,21	1,60	2,22	0,23	3,72	3,53			
6		5,13	2,14	8,43	0,27	1,34	7,57	1,48	3,71	5,72			
7		3,63	2,99	1,68	6,02	18,02	1,42	1,20	1,91	5,70			
8		1,96	9,51	3,72	0,45	3,41	2,04	1,99	1,02	2,30			
9		0,47	2,91	5,48	2,13	9,39	2,13	1,31	3,29	2,92			
10		3,18	2,81	4,05	1,97	1,48	12,97	0,74	17,20	4,86			

Tabela E.2: Erros obtidos com Rede *Feedforward em Cascata* com 70% das Amostras para Treino e 30% pra Teste (Estimativa de Vazão com Rede 4).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		0,20	0,28	1,12	0,18	0,30	5,20	0,23	1,26	1,52	0,23	1,26	1,52
2		0,16	0,43	0,21	0,36	0,26	1,93	0,18	0,19	1,29	0,18	0,19	1,29
3		0,16	8,98	2,45	0,22	0,76	4,51	0,25	1,62	4,93	0,25	1,62	4,93
4		0,36	0,67	2,55	0,14	5,21	1,14	0,21	0,90	1,03	0,21	0,90	1,03
5		0,20	4,53	3,78	3,04	0,75	3,70	0,20	1,46	1,74	0,20	1,46	1,74
6		1,86	1,74	2,64	0,22	2,33	5,12	0,20	5,15	8,12	0,20	5,15	8,12
7		0,22	19,59	0,80	2,31	0,82	4,56	0,78	0,45	5,66	0,78	0,45	5,66
8		0,26	12,17	18,44	0,36	10,77	9,25	1,95	4,01	7,22	1,95	4,01	7,22
9		0,65	0,80	5,43	0,27	2,56	3,69	0,55	2,00	8,95	0,55	2,00	8,95
10		0,39	22,64	17,15	0,87	0,96	2,68	0,52	3,08	15,58	0,52	3,08	15,58

Tabela E.3: Erros obtidos com Rede *Feedforward* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 4).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		3,18	5,25	3,95	0,79	5,80	2,31	0,30	1,25	4,46			
2		0,84	13,10	12,89	0,22	15,99	9,40	0,23	2,50	6,99			
3		0,10	5,34	11,58	0,52	6,04	5,53	0,17	3,87	19,09			
4		0,24	13,42	6,00	0,29	20,41	4,62	0,29	5,22	5,15			
5		0,18	7,48	5,53	0,14	0,99	4,21	0,35	1,81	8,16			
6		3,91	6,02	1,56	0,31	1,81	3,82	1,64	9,32	4,07			
7		4,07	3,80	2,66	0,64	6,51	4,67	3,29	3,02	1,93			
8		7,78	5,11	5,08	1,00	2,59	11,69	4,23	6,69	6,28			
9		0,92	2,74	3,29	2,06	11,28	9,27	1,22	3,00	1,03			
10		0,70	9,68	5,20	2,95	2,34	17,96	1,93	3,48	2,22			

Tabela E.4: Erros obtidos com Rede *Feedforward em Cascata* com 75% das Amostras para Treino e 25% pra Teste (Estimativa de Vazão com Rede 4).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		0,16	0,23	1,57	0,17	0,74	3,18	0,31	0,19	0,77			
2		0,17	0,65	1,38	0,12	0,27	4,91	0,66	0,28	0,96			
3		0,16	0,94	4,83	0,19	4,62	2,42	0,16	1,53	0,68			
4		0,24	2,84	1,85	0,24	10,54	8,13	0,25	0,63	9,61			
5		0,25	3,35	1,91	0,28	1,39	2,57	0,13	0,16	1,22			
6		0,26	1,24	3,28	0,56	3,33	5,64	0,83	3,94	3,34			
7		0,25	0,68	2,86	0,18	0,41	6,90	0,49	3,78	5,47			
8		1,16	1,64	3,86	0,89	0,63	3,28	0,53	5,86	2,95			
9		0,92	16,54	2,62	0,97	7,81	5,72	0,24	4,63	6,25			
10		0,44	1,94	7,11	0,73	1,85	2,84	1,58	8,35	3,53			

Tabela E.5: Erros obtidos com Rede *Feedforward* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 4).

		Erros (%)											
		<i>Feedforward</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		1,06	4,37	5,52	0,96	1,79	3,51	1,03	6,13	6,26			
2		0,47	4,76	5,49	0,24	12,59	6,33	0,26	4,87	16,58			
3		0,15	5,59	12,41	1,55	4,63	5,57	0,55	6,51	9,04			
4		0,36	3,45	4,89	0,23	5,63	17,29	0,26	5,22	4,98			
5		2,45	0,97	3,85	0,22	7,92	4,70	0,11	5,65	5,04			
6		0,96	10,28	6,00	0,18	39,28	5,19	2,53	8,21	2,38			
7		5,37	1,90	13,94	3,89	2,90	10,32	5,02	1,88	5,88			
8		5,93	4,81	12,49	1,95	1,45	4,50	2,58	9,26	6,52			
9		1,88	1,23	2,36	6,92	1,78	39,79	3,58	0,77	3,43			
10		1,71	4,50	3,77	0,74	2,60	3,62	0,79	1,57	1,37			

Tabela E.6: Erros obtidos com Rede *Feedforward em Cascata* com 80% das Amostras para Treino e 20% pra Teste (Estimativa de Vazão com Rede 4).

		Erros (%)											
		<i>Feedforward em Cascata</i>											
		30 Epochs				50 epochs				100 Epochs			
		Função de Treinamento				Função de Treinamento				Função de Treinamento			
Número de Neurônios		<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>	<i>trainlm</i>	<i>trainsecg</i>	<i>trainoss</i>
1		0,56	0,76	0,84	0,42	0,35	1,47	0,19	0,41	0,84	0,19	0,71	1,45
2		2,38	0,34	1,06	0,23	0,25	7,00	0,19	0,71	1,45	0,19	0,71	1,45
3		0,26	3,21	2,39	0,14	0,25	0,56	0,38	1,85	0,71	0,38	1,85	0,71
4		0,13	2,25	1,31	0,38	4,26	2,86	0,26	3,00	5,51	0,26	3,00	5,51
5		0,14	4,63	9,41	0,14	0,20	4,70	0,17	4,05	2,77	0,17	4,05	2,77
6		0,53	2,38	9,89	0,17	1,19	2,87	0,64	2,49	18,53	0,64	2,49	18,53
7		0,92	1,94	3,51	0,27	8,79	34,28	7,63	10,06	5,65	7,63	10,06	5,65
8		0,16	2,86	14,37	0,25	1,01	8,31	0,44	8,50	4,07	0,44	8,50	4,07
9		0,14	1,17	4,86	0,32	3,81	16,01	0,52	0,16	1,97	0,52	0,16	1,97
10		0,92	1,28	9,36	0,45	2,42	26,05	2,56	13,44	14,83	2,56	13,44	14,83