



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



DETECÇÃO E ISOLAMENTO DE FALHAS EM SISTEMAS DINÂMICOS BASEADOS EM REDES NEURAIIS

Raphaela Galhardo Fernandes

Orientador: Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Número de ordem PPgEE: M187
Natal, RN, Fevereiro de 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Detecção e Isolamento de Falhas em Sistemas Dinâmicos Baseados em Redes Neurais

Raphaela Galhardo Fernandes

Dissertação de Mestrado aprovada em 08 de fevereiro de 2007 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira (orientador)
DCA/UFRN

Prof^a Dr. Walmir Matos Caminhas..... DEE/UFMG

Prof. Dr. Adrião Duarte Dória Neto DCA/UFRN

Prof. Dr. Jorge Dantas de Melo DCA/UFRN

*Dedico esta dissertação de mestrado
ao meu avô José Fernandes, que
onde quer que esteja deve estar
orgulhoso da neta engenheira e
agora recebendo o título de mestre.*

Agradecimentos

Agradeço a Deus por ter me dado sabedoria para conquistar este título em minha vida.

Aos meus pais, Fernando e Valéria, por terem me propiciado uma boa educação, permitindo a minha formação intelectual.

Ao meu namorado Gleydson, por ter estado sempre ao meu lado com carinho e me dando apoio para que eu conseguisse suportar as dificuldades encontradas neste trabalho.

À minha irmã Aline e aos meus amigos que me incentivaram a concluir mais esta jornada. Em especial a Diego, que me ajudou muito no desenvolvimento do trabalho apresentado nesta dissertação de mestrado.

Aos meus queridos professores Affonso (meu orientador) e Adrião que contribuíram bastante na minha formação acadêmica, acreditando na minha capacidade e sempre me incentivando.

Resumo

Esta dissertação de mestrado apresenta o desenvolvimento de um sistema de detecção e isolamento de falhas (DIF) baseado em redes neurais. O sistema é dividido em duas etapas: uma de identificação neural do sistema e outra de detecção e classificação de falhas. Ambos subsistemas usam técnicas de redes neurais com o algoritmo *Backpropagation* para redes *Perceptron* de Múltiplas Camadas. Duas abordagens para identificação neural foram testadas e uma delas selecionada para fazer parte do sistema DIF. O classificador de falhas utiliza apenas valores residuais para a classificação das mesmas. Todos os testes foram realizados tanto em ambiente simulado quanto em ambiente real, no intuito de comprovar dificuldades encontradas em testes reais não existentes quando se trabalha apenas com simulações.

Palavras-chave: Detecção e Isolamento de Falhas, Diagnóstico de Falhas, Resíduos, Redes Neurais.

Abstract

This master dissertation presents the development of a fault detection and isolation system based in neural network. The system is composed of two parts: an identification subsystem and a classification subsystem. Both of the subsystems use neural network techniques with multilayer perceptron training algorithm. Two approaches for identification stage were analyzed. The fault classifier uses only residue signals from the identification subsystem. To validate the proposal we have done simulation and real experiments in a level system with two water reservoirs. Several faults were generated above this plant and the proposed fault detection system presented very acceptable behavior. In the end of this work we highlight the main difficulties found in real tests that do not exist when it works only with simulation environments.

Keywords: Fault Detection and Isolation, Fault Diagnosis, Residual, Neural Networks.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	vi
Lista de Símbolos e Abreviaturas	vii
1 Introdução	1
1.1 Organização do texto	3
2 Fundamentação Teórica	4
2.1 Redes Neurais Artificiais	4
2.1.1 Neurônio Computacional	4
2.1.2 Perceptron de Múltiplas Camadas	6
2.2 Identificação Neural de Sistemas	9
2.3 Detecção e Diagnóstico de Falhas	11
2.3.1 O Processo de Detecção e Diagnóstico de Falhas	12
2.4 Algumas Técnicas de Diagnósticos de Falhas	13
2.4.1 Redundância Física	13
2.4.2 Redundância Analítica	14
2.4.3 Análise de Sinais	16
2.4.4 Inteligência Artificial	16
3 Proposta do Trabalho	19
3.1 Processo Utilizado: Sistema de Níveis	20
3.2 Identificação Neural do Sistema	22
3.3 Detecção e Isolamento de Falhas	24
4 Resultados da Identificação Neural	29
4.1 Resultados com Modelo Matemático - Simulação	30

4.1.1	Identificação em Um Único Passo	31
4.1.2	Identificação em Dois Passos	34
4.2	Resultados Reais	38
4.2.1	Identificação em Um Único Passo	39
4.2.2	Identificação em Dois Passos	42
4.3	Comentários - Identificação Neural	45
5	Resultados da Detecção e Isolamento de Falhas	48
5.1	Falhas que Podem Ocorrer no Sistema de Níveis	49
5.2	Resultados com Modelo Matemático - Simulação	53
5.3	Resultados Reais	54
5.4	Comentários Finais	62
6	Conclusões e Perspectivas	64
	Referências bibliográficas	66

Lista de Figuras

2.1	Modelo Não-Linear de um Neurônio	5
2.2	Funções de Ativação de um Neurônio	6
2.3	Exemplo de Arquitetura de uma rede MLP	7
2.4	Estrutura Geral de Identificação de Sistemas com RNA	11
2.5	Estrutura Geral para Geração de Resíduos	15
3.1	Esquema Geral do Sistema DIF	19
3.2	Arquitetura do Ambiente Real	20
3.3	Planta de Níveis com Controlador PID	21
3.4	Fotografia do Ambiente Real Montado no LAMP	22
3.5	Treino para Determinar Nível do Tanque 2 com Única RNA	23
3.6	1ª Etapa: Determinação do Nível do Tanque 1 a partir da Entrada do Sistema	23
3.7	2ª Etapa: Determinação do Nível do Tanque 2 a partir da Entrada do Sistema	24
3.8	Esquema Geral para Classificação das Falhas	26
4.1	Arquitetura da RNA para Identificação em um Único Passo	29
4.2	Arquitetura da RNA 1 para Identificação em Dois Passos	29
4.3	Arquitetura da RNA 2 para Identificação em Dois Passos	30
4.4	Exemplo de Conjunto de Dados para Treinamento	30
4.5	Evolução do Erro Quadrático Médio Durante o Treinamento da Rede para Identificação em Um Único Passo	31
4.6	Validação do Treinamento - Identificação em Único Passo	32
4.7	Exemplo 1 - Identificação em Único Passo - Simulação	33
4.8	Exemplo 2 - Identificação em Único Passo - Simulação	33
4.9	Exemplo 3 - Identificação em Único Passo - Simulação	34
4.10	Evolução dos Erros Quadráticos Médios Durante o Treinamento das Re- des 1 e 2 para Identificação em Dois Passos	34
4.11	Validação do Treinamento - Identificação em Dois Passos	35
4.12	Exemplo 1 - Identificação em Dois Passos - Simulação	36
4.13	Exemplo 2 - Identificação em Dois Passos - Simulação	37

4.14	Exemplo 3 - Identificação em Dois Passos - Simulação	37
4.15	Exemplo de Conjunto de Dados para Treinamento - Ambiente Real	38
4.16	Evolução do Erro Quadrático Médio Durante o Treinamento da Rede para Identificação em Um Único Passo - Ambiente Real	39
4.17	Validação do Treinamento - Identificação em Único Passo - Ambiente Real	40
4.18	Exemplo 1 - Identificação em Único Passo - Ambiente Real	40
4.19	Exemplo 2 - Identificação em Único Passo - Ambiente Real	41
4.20	Exemplo 3 - Identificação em Único Passo - Ambiente Real	41
4.21	Evolução dos Erros Quadráticos Médios Durante o Treinamento das Re- des 1 e 2 para Identificação em Dois Passos - Ambiente Real	42
4.22	Validação do Treinamento - Identificação em Dois Passos - Ambiente Real	43
4.23	Exemplo 1 - Identificação em Dois Passos - Ambiente Real	43
4.24	Exemplo 2 - Identificação em Dois Passos - Ambiente Real	44
4.25	Exemplo 3 - Identificação em Dois Passos - Ambiente Real	44
4.26	Diferenças do Nível do Tanque 1: Modelo Matemático, Real e Gerado pela RNA	46
4.27	Diferenças do Nível do Tanque 2: Modelo Matemático, Real e Gerado pela RNA	46
4.28	Diferenças do Sinal de Controle: Modelo Matemático e Real	47
5.1	Simulação das 13 Falhas Seleccionadas	52
5.2	Evolução do Erro Quadrático Médio Durante o Treinamento da Rede para Classificação das Falhas	53
5.3	Interface Gráfica do Supervisório	54
5.4	Falha 1: Novo Orifício no Tanque 1. Água Não Cai no Tanque 2 - Sistema Simulado	55
5.5	Falha 2: Diminuição do Orifício do Tanque 1 - Sistema Simulado	55
5.6	Falha 3: Diminuição do Orifício do Tanque 2 - Sistema Simulado	56
5.7	Falha 4: Aumento do Orifício do Tanque 1 - Sistema Simulado	56
5.8	Falha 5: Aumento do Orifício do Tanque 2 - Sistema Simulado	57
5.9	Falha 6: Aumento do Sinal de Controle - Sistema Simulado	57
5.10	Falha 1: Novo Orifício no Tanque 1. Água Não Cai no Tanque 2 - Sistema Real	58
5.11	Falha 2: Diminuição do Orifício do Tanque 1 - Sistema Real	58
5.12	Falha 3: Diminuição do Orifício do Tanque 2 - Sistema Real	59
5.13	Falha 4: Aumento do Orifício do Tanque 1 - Sistema Real	59

5.14 Falha 5: Aumento do Orifício do Tanque 2 - Sistema Real	60
5.15 Falha 7: Redução do Sinal de Controle - Sistema Real	60
5.16 Falha 12: Fluxo Extra de Água no Tanque 1 - Sistema Real	61
5.17 Falha 13: Fluxo Extra de Água no Tanque 2 - Sistema Real	61

Lista de Tabelas

3.1	Parâmetros da Planta	21
3.2	Disposição dos Resíduos para as 13 Falhas	26
3.3	Possíveis Saídas para Classificação de 3 Falhas	27
3.4	Sete Situações: Seis Conjuntos de Falhas e Ausência de Falhas	27
5.1	Simulação das Falhas: Novas Variáveis Incluídas Nas Equações Matemáticas	50
5.2	Mensagens Exibidas Após a Ocorrência de Falhas	62

Lista de Símbolos e Abreviações

DDF	-	Detecção e Diagnóstico de Falhas
DIF	-	Detecção e Isolamento de Falhas
FF	-	Foundation Fieldbus
IA	-	Inteligência Artificial
MLP	-	Perceptron de Múltiplas Camadas (Multilayer Perceptron)
OPC	-	Ole for Process
PID	-	Proporcional Integral Derivativo
RNA	-	Rede Neural Artificial

Capítulo 1

Introdução

Com o advento da automação industrial, cada vez mais cresce a necessidade de se garantir segurança e confiabilidade dos equipamentos utilizados na execução dos processos. Em um sistema dinâmico, as falhas podem causar alterações em parâmetros críticos do sistema, ou até mesmo mudanças em sua dinâmica. É bastante importante que os estados indesejados ou não permitidos do sistema sejam percebidos, de forma que se possa tomar ações apropriadas, evitando situações de perigo ou algum acidente.

Dessa forma, o ideal é que as falhas possam ser detectadas, isoladas e toleradas, proporcionando uma melhor confiabilidade do sistema. Detectar uma falha no sistema significa dizer se ela está ocorrendo ou não. Já isolar, significa classificar qual o tipo da falha, por exemplo, informar em que componente do sistema ela aconteceu, e a tolerância da falha pode ser definida como ações apropriadas que devem ser tomadas para minimizar o seu efeito. Em outras palavras, um sistema que possui um comportamento tolerante a falhas deve ser capaz de detectar e classificar automaticamente as falhas ocorridas, bem como ser capaz de continuar funcionando após as suas ocorrências. Todo este processo pode ser também definido como Detecção e Diagnóstico de Falhas (DDF).

As falhas que possam vir a ocorrer em um sistema dinâmico podem gerar prejuízos tanto danificando os equipamentos pertencentes ao processo quanto pondo em risco as pessoas envolvidas no mesmo. A necessidade de se evitar estes tipos de prejuízos aumenta a importância da existência de um sistema de detecção e diagnóstico de falhas eficiente.

Para detectar e diagnosticar falhas eficientemente, é importante conhecer antecipadamente os padrões de falhas e as tendências do sinal para que elas ocorram, possibilitando uma detecção e diagnóstico de falhas mais exatas e uma redução nos custos de manutenção do processo.

Alguns métodos de detecção e diagnóstico de falhas são baseados na redundância física, ou seja, na duplicação de componentes de *hardware*, como sensores, atuadores e controladores. Nestes casos, as saídas dos pares de componentes equivalentes são com-

paradas para a detecção de falhas [Weber 2002]. Se os valores das saídas dos pares forem próximos, significa que não está ocorrendo falha, caso haja uma divergência entre eles, há falha. A principal desvantagem desses métodos é a necessidade de equipamentos extras o que pode tornar o custo de manutenção bastante elevado, além de requerer um maior espaço físico para suas acomodações.

Outras técnicas, surgidas a partir da década de 70, são baseadas no paradigma da redundância analítica. Com essa abordagem, pode-se utilizar um modelo matemático do sistema, baseado nas leis da física, para gerar sinais redundantes que são comparados aos sinais medidos pelos sensores físicos [Persin et al. 2002]. Semelhantemente à redundância física, as saídas dos sensores são comparadas às calculadas através do modelo matemático e, havendo uma diferença significativa entre elas, caracteriza-se a ocorrência de alguma falha. Uma das dificuldades desta abordagem é a necessidade da existência de um modelo matemático bem próximo da realidade.

Nos últimos anos, as pesquisas feitas no contexto dos Sistemas de Detecção e Diagnóstico de Falhas apresentam abordagens que utilizam os conceitos da Inteligência Artificial (IA), como por exemplo, Lógica *Fuzzy* [Uppal et al. 2005] ou Redes Neurais Artificiais (RNA's) [Shin et al. 2005], de modo a evitar a utilização de modelos matemáticos explícitos.

Neste trabalho foi desenvolvido um sistema de Detecção e Isolamento de Falhas (DIF) aplicado a um sistema dinâmico em tempo real, a saber, uma planta composta por dois tanques (Sistema de Níveis), onde um dos tanques tem o seu nível controlado. Um sistema DIF corresponde a uma parte de um sistema DDF, ou seja, apenas às etapas de detectar e classificar o tipo da falha, deixando de lado as fases de análise das consequências das falhas e ações a serem tomadas. Para isso, foram utilizadas redes neurais artificiais (Perceptron de Múltiplas Camadas - MLP [Haykin 2001]). Este sistema de níveis real está montado no Laboratório de Automação da Medição em Petróleo (LAMP) presente na Universidade Federal do Rio Grande do Norte (UFRN).

Basicamente, o sistema DIF foi dividido em duas partes: a primeira corresponde à identificação neural do modelo da planta, utilizada para detectar a ocorrência ou não de falhas no sistema de níveis. Já a segunda, consiste no isolamento de falhas no processo, ou seja, em uma classificação do tipo de falha dentre um conjunto pré-definido.

A principal contribuição deste trabalho consiste em não se ter apenas aplicado o sistema DIF a um ambiente simulado, utilizando um modelo matemático baseado nas leis da física, mas sim também em aplicá-lo ao sistema de níveis real ligado a uma rede industrial. Dessa maneira, em tempo real, o sistema consegue monitorar o funcionamento do sistema de níveis, detectando e isolando as falhas no momento em que as mesmas

ocorrem.

1.1 Organização do texto

Neste documento, o Capítulo 2 apresenta a fundamentação teórica relacionada a sistemas de detecção e diagnóstico de falhas, bem como sobre redes neurais, que é uma das ferramentas utilizadas no desenvolvimento do trabalho apresentado. O Capítulo 3 apresenta toda a metodologia adotada, detalhando o procedimento efetuado. Nos Capítulos 4 e 5 estão os resultados obtidos tanto em simulações quanto em situações reais. Por fim, o Capítulo 6 apresenta as conclusões e perspectivas futuras para o trabalho e por último as referências bibliográficas utilizadas como base de estudo para o desenvolvimento do trabalho.

Capítulo 2

Fundamentação Teórica

Neste capítulo será apresentada a fundamentação teórica que será a base do desenvolvimento do trabalho. Primeiramente, serão explanados alguns conceitos relacionados às redes neurais artificiais, bem como seria o esboço do processo de identificação de um sistema com as mesmas. Por fim, serão mostrados alguns conceitos presentes no contexto do processo de detecção e diagnóstico de falhas em um sistema, assim como algumas técnicas existentes.

2.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) podem ser consideradas como uma abordagem à solução de problemas na área de inteligência artificial. Nessa abordagem, ao invés de se tentar fazer com que um computador imite um comportamento inteligente, tentaria-se construir um computador com um comportamento inteligente, de forma que ele erre, aprenda novas tarefas, generalize, faça descobertas e algumas vezes, possa ser melhor do que seu professor [Haykin 2001].

2.1.1 Neurônio Computacional

Os neurônios biológicos e o sistema nervoso são a inspiração das redes neurais artificiais. Porém, as RNA's são bem diferentes das redes neurais biológicas e muitas vezes, as semelhanças são mínimas. Modelar o sistema nervoso é um trabalho que vem sendo desenvolvido há muito tempo. Os primeiros trabalhos que impulsionaram o interesse foram o de McCulloch e Pitts [McCulloch & Pitts 1943], o de Hebb [Hebb 1949], e o de Rosenblatt [Rosenblatt 1958].

As RNA's são formadas por várias unidades de processamento conhecidas como neurônios. O neurônio artificial possui uma estrutura bem mais simples do que a de um neurônio bio-

lógico. A Figura 2.1 ilustra um modelo não-linear de um neurônio computacional.

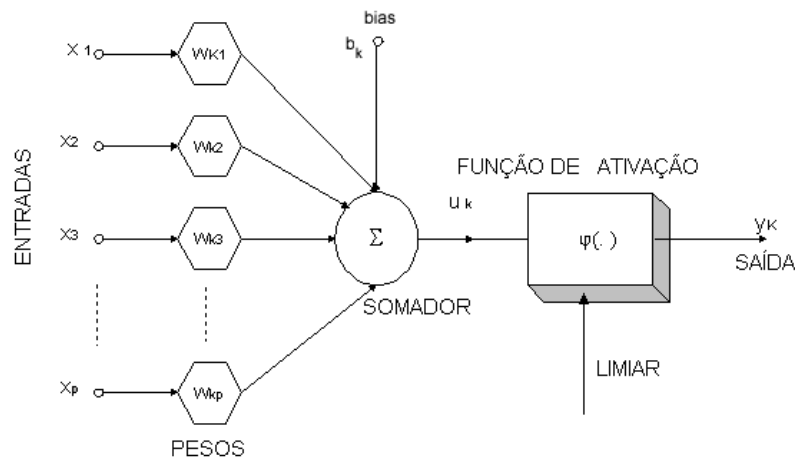


Figura 2.1: Modelo Não-Linear de um Neurônio

Um neurônio artificial é basicamente constituído por um conjunto de sinapses, um somador e uma função de ativação, geralmente não-linear. Além disso, os conjuntos de entradas e saídas são análogos, respectivamente, aos dendritos e aos axônios do neurônio biológico [Haykin 2001].

As entradas do neurônio artificial, antes de serem propagadas até a saída do mesmo, são ponderadas pelos pesos sinápticos. O somador tem o papel de somar esses sinais de entrada ponderados, tendo a função de um combinador linear. Já a função de ativação restringe a amplitude da saída de um neurônio e aplica a não linearidade.

No modelo da Figura 2.1, existe um bias, b_k , que aumenta ou diminui a entrada líquida da função de ativação.

Uma modelagem matemática para um neurônio artificial k pode ser obtida através das equações 2.1 e 2.2,

$$u_k = \sum_{j=1}^p w_{kj} x_j \quad (2.1)$$

$$y_k = \varphi(u_k + b_k) \quad (2.2)$$

onde x_1, x_2, \dots, x_p , são os sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{kp}$ 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; $\varphi(\cdot)$ é a função de ativação e y_k é o sinal de saída do neurônio.

A função de ativação, $\varphi(v)$, define a saída de um neurônio em relação ao campo local induzido v . As principais funções de ativação são: função de limiar, função linear por partes e função sigmóide. A escolha dessas funções depende da aplicação da rede neural. A Figura 2.2 apresenta alguns exemplos de funções de ativação.

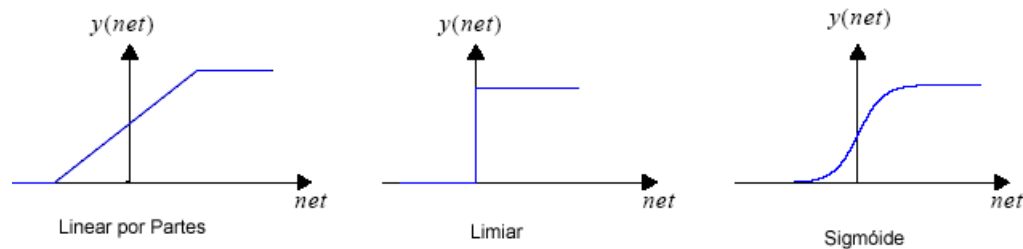


Figura 2.2: Funções de Ativação de um Neurônio

A subseção seguinte apresenta um pouco sobre as RNA's Percéptron de Múltiplas Camadas que foram escolhidas para o desenvolvimento do trabalho apresentado.

2.1.2 Perceptron de Múltiplas Camadas

As redes Perceptron de Múltiplas Camadas (MLP) têm sido usadas com sucesso em diversas aplicações nas mais variadas áreas, como por exemplo, em reconhecimento de padrões, em processamento de sinais e em controle.

A estrutura de uma MLP consiste de nós em uma camada de entrada, de um conjunto de neurônios dispostos em uma ou mais camadas ocultas (ou intermediárias) e na camada de saída. Uma rede MLP pode ser definida como uma generalização do Perceptron [Braga et al. 2000].

A Figura 2.3 mostra a arquitetura de uma rede MLP com uma camada de entrada, duas camadas intermediárias e uma camada de saída.

Uma rede MLP é dita progressiva, ou *feedforward*, quando as saídas dos neurônios em qualquer camada se conectam apenas com os neurônios de entrada da camada seguinte, ou seja, não possuem nenhum laço de realimentação. Dessa forma, o sinal de entrada é propagado camada a camada até chegar na camada de saída, ou seja, de forma progressiva.

A quantidade de nós na camada de entrada é determinada pela dimensão do espaço de observação. Já o número de neurônios da camada de saída é igual ao número requerido de saídas desejadas.

Para o projeto de uma MLP, deve-se determinar o número de camadas ocultas, o número de neurônios em cada uma dessas camadas e definir os pesos sinápticos dos

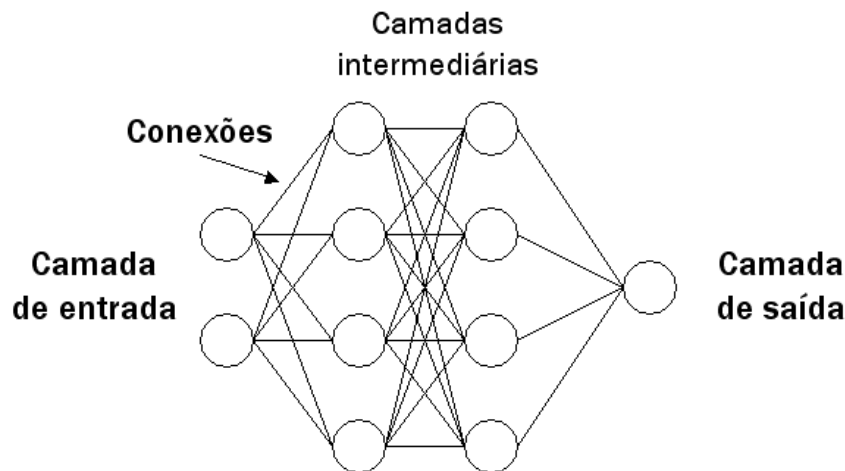


Figura 2.3: Exemplo de Arquitetura de uma rede MLP

neurônios que constituem a rede.

Definir a quantidade de camadas ocultas e os neurônios que as compõem é uma tarefa bastante complexa, pois não há regras determinadas para defini-los. Já a definição dos pesos sinápticos é feita utilizando algoritmos de treinamento supervisionado. O algoritmo de treinamento mais conhecido na literatura é o Algoritmo *Backpropagation*, baseado na minimização do erro usando métodos do tipo gradiente, em que o erro é retropropagado da camada de saída para as camadas intermediárias.

Esse algoritmo é constituído basicamente por dois passos: computação no sentido direto e computação no sentido reverso.

No primeiro passo do aprendizado (computação para frente), aplica-se um vetor de entrada aos nós de entrada da rede e seu efeito é propagado camada por camada até chegar à camada de saída, onde produz uma resposta ao vetor de entrada. Neste passo, os pesos sinápticos são mantidos constantes.

Na computação para trás, ajusta-se os pesos sinápticos através das regras de correção de erro. Basicamente, subtrai-se o padrão de saída desejado da resposta à excitação de entrada da MLP, que corresponde ao sinal de erro. Então, propaga-se esse sinal de erro através dos neurônios no sentido contrário ao que o vetor de entrada foi propagado no passo anterior, por isso o nome *backpropagation*. À medida que o erro é propagado, os pesos sinápticos são ajustados de forma que a resposta obtida pela MLP se aproxime ao máximo do padrão de resposta desejada.

Resumo do Algoritmo Backpropagation

O algoritmo *backpropagation* pode ser considerado como o mais popular no contexto do aprendizado das redes MLP. Ele é bastante simples de implementar e provê uma grande capacidade de armazenamento de informação nos pesos sinápticos da rede.

Se a qualidade de dados apresentada durante o treinamento da rede for suficiente para representar o ambiente em que a rede está inserida, a rede MLP treinada usando o algoritmo *backpropagation* desenvolverá a capacidade de generalizar. Dessa forma, a rede MLP pode apresentar ótimos desempenhos quando submetida a dados de entrada que não foram apresentados durante o treinamento.

Considere que um exemplo de treinamento possa ser representado através de um vetor de entrada $x(n)$ e um vetor de saída desejada $d(n)$ no instante n . O vetor de entrada é apresentado aos nós da camada de entrada e o vetor de saída desejada é comparado aos neurônios da camada de saída da rede.

Resumidamente, o algoritmo *backpropagation* pode ser definido da seguinte maneira:

1. Inicialização aleatória dos pesos sinápticos e limiares, utilizando uma distribuição uniforme cuja média é nula.
2. Apresentação de uma época de conjuntos de exemplo para treinamento. Para cada conjunto de treinamento $(x(n), d(n))$, realize as etapas 3 e 4.
3. Propagação. Cálculo dos campos locais induzidos e dos sinais funcionais da rede camada por camada prosseguindo para frente. A equação 2.3 representa o campo local induzido para o neurônio j da camada l , em que no instante n , $y_i^{(l-1)}(n)$ é o sinal de saída do neurônio i da camada anterior $l - 1$ e $w_{ji}^{(l)}(n)$ é o peso sináptico do neurônio j da camada l alimentado pelo neurônio i da camada $l - 1$, e m_0 representa o número total de entradas.

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \quad (2.3)$$

Se $i = 0$, $y_0^{(l-1)}(n) = +1$ e o bias aplicado ao neurônio j na camada l corresponde a $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$.

Considerando o uso de uma função sigmóide como de ativação para o sinal de saída do neurônio j na camada l , $y_j^{(l)}(n)$ é definido de acordo com a equação 2.4.

$$y_j^{(l)} = \Phi_j(v_j(n)) \quad (2.4)$$

Caso o neurônio j faça parte da primeira camada escondida ($l = 1$), então o sinal de

saída dessa camada é dado pela equação 2.5 , em que $x_j(n)$ é o j -ésimo elemento do vetor de entrada $x(n)$. Assim,

$$y_j^{(0)} = x_j(n) \quad (2.5)$$

Caso o neurônio j pertença à camada de saída ($l = L$), então o sinal de saída é dado pela equação 2.6.

$$y_j^{(L)} = o_j(n) \quad (2.6)$$

Portanto, o sinal de erro da rede para um dado conjunto de treinamento $(x(n), d(n))$ pode ser calculado através da equação 2.7.

$$e_j(n) = d_j(n) - o_j(n) \quad (2.7)$$

4. Retropropagação. Cálculo dos gradientes locais definidos pela equação 2.8.

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n) \phi_j'(v_j^{(L)}(n)), & \text{para a camada de saída } L \\ \phi_j'(v_j^{(l)}(n)) \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n), & \text{para a camada oculta } l \end{cases} \quad (2.8)$$

Em seguida, os pesos sinápticos da rede devem ser ajustados de acordo com a equação 2.9, em que α é a constante de momento e η a taxa de aprendizado.

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha \left[w_{ji}^{(l)}(n-1) \right] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \quad (2.9)$$

5. Os passos de treinamento 3 e 4 devem ser executados usando novas épocas de treinamento até que o critério de parada seja atingido, ou seja, até que o limite de erro mínimo ou o número máximo de épocas seja atingido.

2.2 Identificação Neural de Sistemas

A identificação de sistemas constitui-se em um dos problemas básicos na teoria de controle. Para o caso em que a planta possui natureza linear, a identificação pode ser feita através da construção de uma função de transferência que represente o comportamento da mesma, em tempo discreto ou contínuo. Técnicas para identificação de sistemas lineares estão bem consolidadas e com vários resultados teóricos e práticos [Ljung 1987].

No mundo real, a maioria das plantas é de natureza não-linear e sua identificação se

torna bem mais difícil [Billings 1980], podendo ser feita analiticamente ou computacionalmente.

Identificar analiticamente significa analisar a dinâmica do sistema físico e desenvolver um modelo matemático para ele. Já a identificação computacional é feita a partir de dados observados na entrada e na saída da planta e deve ser representativa do ponto de vista estatístico. Ela pode ser realizada a partir dos seguintes passos:

1. Planejamento Experimental: determinar qual será o método de amostragem para a coleta dos dados;
2. Seleção de Uma Estrutura de Modelo: selecionar a estrutura do modelo e em seguida, determinar os parâmetros livres (parâmetros a serem ajustados);
3. Estimação de Parâmetros: ajustar os parâmetros livres a partir das estatísticas dos dados obtidas;
4. Validação: avaliar o desempenho do modelo para os dados de testes.

Uma das aplicações existentes para a identificação de sistema é relacionada ao projeto de controladores. A idéia geral no projeto de controladores é encontrar um controlador que modifica um dado comportamento de um sistema dinâmico com a finalidade de alcançar alguns objetivos. Mesmo que a estratégia de controle possa ser obtida diretamente, vários métodos de controle exigem que a planta seja identificada, ou seja, há a necessidade de um modelo de planta capaz de prever as saídas futuras para sinais de testes, permitindo o ajuste dos parâmetros do controlador [Nepomuceno et al. 2004], [Coelho & Coelho 1999].

Pode-se também aplicar a identificação de plantas em sistemas de Detecção e Diagnósticos de Falhas [Naughton et al. 1996], [Shin et al. 2005]. Nesses trabalhos, em particular, utilizou-se uma Rede Neural Artificial (RNA) para reproduzir o comportamento dinâmico do sistema livre de falhas e a cada instante, a saída do sistema é comparada à saída estimada da RNA, possibilitando informar a ocorrência ou não de falhas.

Como os projetistas e engenheiros tendem a lidar com sistemas mais complexos, viabilizou-se a identificação e o controle desses sistemas com o uso de técnicas inteligentes [Nürnberg et al. 2001]. Uma das estratégias mais populares para a identificação inteligente de sistemas não-lineares é utilizar RNA's [Haykin 2001], afinal elas são ferramentas notáveis para aproximação de funções. As RNA's são utilizadas para identificar sistemas, principalmente, quando: pouco se conhece sobre a planta; a estrutura da planta apresenta não-linearidades significativas; somente dados de entrada-saída estão disponíveis.

A característica mais importante das redes neurais está em sua habilidade de aproximar funções. Em outras palavras, dada uma certa função desconhecida a qual se deseja aproximar, a rede tem seus parâmetros ajustados de forma a produzir a mesma resposta em relação à função, para entradas semelhantes.

A identificação de sistemas utilizando redes neurais pode ser feita com redes estáticas ou dinâmicas. Também não há a necessidade de conhecimento detalhado prévio de características da planta, pois a identificação é baseada nos dados de entrada e saída. Dessa forma, a RNA tem a tarefa não trivial de construir uma representação para a planta usando pares entrada-saída extraídos da planta real. Depois que um modelo “ótimo” é identificado para a planta real, simulações computacionais podem ser realizadas utilizando o modelo ao invés da planta real.

A estrutura de identificação de sistemas utilizando redes neurais é exibida na Figura 2.4. Os pares de vetores $x(k)$ e $y(k)$, que correspondem às entradas e saídas da planta, respectivamente, são padrões apresentados à RNA. No caso de identificação de sistemas dinâmicos, as entradas da RNA devem conter também valores atrasados, possibilitando que a rede aprenda a dinâmica do sistema. O sinal de erro é calculado e os pesos da rede são ajustados de forma que $y'(k)$ aproxime $x(k)$ para todo $u(k)$. O treinamento é finalizado quando o erro é significativamente pequeno e a RNA obtida pode ser usada como um modelo da planta real.

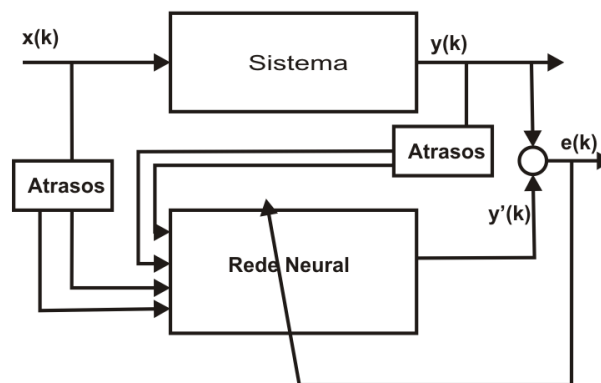


Figura 2.4: Estrutura Geral de Identificação de Sistemas com RNA

2.3 Detecção e Diagnóstico de Falhas

Nesta seção, conceitos e técnicas ligados ao processo de detecção e diagnóstico de falhas serão apresentados. Antes de explicar todo esse processo é importante conhecer

alguns conceitos fundamentais utilizados nesta área, entre eles [Witczak 2005]:

- **Fracasso:** uma condição de mau funcionamento ou de colapso. Um mau funcionamento causa uma queda de desempenho funcional do sistema. Já o colapso provoca uma parada total da operação;
- **Degradação:** provoca uma leve queda no desempenho do sistema, mas se mantida por um longo período, pode causar falhas severas. O desempenho global do sistema só será alterado caso a degradação atinja um nível crítico;
- **Falha:** representa uma propriedade inaceitável do sistema ou de algum componente. Em geral, todos os fracassos são falhas, mas nem toda falha é um fracasso;
- **Defeito:** corresponde a uma causa imediata de degradação do sistema que o leva a um mau funcionamento;
- **Sintoma:** alguma mudança do comportamento normal do processo que pode ser percebida, indicando a presença de falha ou defeito;
- **Ruído:** uma perturbação indesejada que implica na deturpação, distorção da informação transmitida de forma imprevisível;
- **Resíduo:** também conhecido como desvio, corresponde à diferença entre um valor medido em um sistema real e o valor esperado em sistema simulado.

2.3.1 O Processo de Detecção e Diagnóstico de Falhas

Em um sistema, detectar e diagnosticar falhas significa identificar a ocorrência de falhas, saber classificar que tipo da falha ocorreu e a partir desta percepção, ser possível aplicar medidas corretivas ou preventivas no sistema.

A Detecção e Diagnóstico de Falha (DDF) engloba os seguintes passos [Witczak 2005]:

- Detecção da Falha;
- Isolamento da Falha;
- Identificação da Falha;
- Acomodação da Falha.

Detectar uma falha significa determinar que uma falha está ocorrendo no sistema. Essa operação pode ser feita através da observação de alguma variável do sistema, por exemplo, pode-se dizer que uma falha ocorreu caso esta variável exceda algum valor pré-determinado. Vale salientar que o sistema de detecção deve ser robusto para impedir que falsos alarmes aconteçam.

Isolar uma falha corresponde à classificação do tipo da falha que está ocorrendo, por exemplo, o sensor x deixou de funcionar ou a bomba y queimou, ou seja, determinar o tipo e a localização da falha. A identificação do tipo e de onde a falha ocorre depende, primeiramente, da detecção da mesma.

A identificação da falha significa estimar a extensão e a magnitude da falha durante o período que ela ocorre, ou seja, avaliar o impacto da falha no desempenho do sistema.

Já o último conceito, acomodação da falha, refere-se às ações realizadas para minimizar os efeitos da falha, que podem ser, por exemplo: a tolerância da falha, a realização do conserto rapidamente ou até mesmo parar a operação do sistema até que todos os reparos necessários sejam realizados.

O conceito de DDF pode ser utilizado para definir o processo completo que vai desde a detecção até a acomodação da falha.

Neste trabalho, foi desenvolvido um sistema de Detecção e Isolamento de Falhas (DIF) em um sistema dinâmico formado por uma planta de níveis. Portanto, os passos de identificação e acomodação de falhas não foram abordados.

Na próxima seção serão apresentadas algumas técnicas existentes para o processo de detecção e diagnósticos de falhas.

2.4 Algumas Técnicas de Diagnósticos de Falhas

Na literatura há uma diversidade de trabalhos relacionada à detecção e diagnóstico de falhas, utilizando um conjunto de ferramentas diferentes, algumas delas baseadas em redundância física, outras utilizando lógica *fuzzy*, redes neurais, redes *bayesianas*, transformadas *wavelets*, etc. Como já citado, neste trabalho, as redes neurais artificiais foram escolhidas para o desenvolvimento de um sistema de detecção e isolamento de falhas.

Esta seção, destina-se a mostrar algumas das estratégias existentes utilizadas no processo de DDF, de forma a fazer uma breve revisão bibliográfica.

2.4.1 Redundância Física

Os métodos para detecção de falhas que utilizam a redundância física (ou redundância de hardware [Weber 2002]) necessitam de equipamentos de *hardware* extras, como sensores, atuadores ou controladores. Esses métodos comparam as saídas de componentes redundantes (em geral, sensores ou subsistemas redundantes) com as saídas equivalentes no sistema real. Se houver uma divergência considerável entre as saídas correspondentes, considera-se a ocorrência de falha e um alarme pode ser disparado. Caso contrário,

considera-se que o sistema está operando normalmente.

Havendo redundância em mais de um componente do sistema, ocorrendo uma única falha, o local em que a falha ocorre é facilmente identificado, pois haverá apenas um divergência significativa entre os pares de componentes redundantes.

Uma vantagem desta técnica, é que se os componentes redundantes apresentarem falhas, eles podem ser simplesmente ignorados, não afetando a operação do sistema. Assim, os componentes redundantes com falhas podem ser substituídos quando for apropriado.

Por outro lado, a redundância física pode representar um custo adicional ao projeto do sistema ao ter que adquirir equipamentos extras. Também pode haver problemas devido à necessidade de espaços extras para acomodação dos equipamentos redundantes.

Outra desvantagem está no fato de que se a falha ocorrer em componentes do sistema que não possuem redundância, ela poderá não ser detectada facilmente, pois o sistema de detecção estaria indicando que tudo está operacionando normalmente. Como em geral, a redundância é feita em sensores, podem acontecer falhas em atuadores e controladores que não serão detectadas.

2.4.2 Redundância Analítica

As técnicas que utilizam a redundância analítica [Armengol et al. 2003], [Simani 2005], [Persin et al. 2002] utilizam modelos simulados (por exemplo, modelos matemáticos) do sistema para a geração dos sinais redundantes, que são comparados aos sinais medidos pelos sensores físicos. Em outras palavras, as medidas dos sensores são comparadas com os valores correspondentes obtidos analiticamente [Gertler 1998] do modelo matemático que reproduz o comportamento do sistema livre de falhas.

Para a realização de seus passos, a maioria dos sistemas DDF considera a comparação de uma ou mais variáveis do sistema real com a sua respectiva variável em um sistema simulado (que deve corresponder ao real). Essa comparação é feita a partir da diferença entre a variável obtida a partir do sistema real e a variável gerada a partir da simulação. Como já citado, esta diferença é denominada resíduo. Se esse resíduo ultrapassar algum limite pré-estabelecido, um alarme pode ser acionado, detectando que alguma falha está ocorrendo. O esquema geral para a geração de resíduo a partir da observação dessas duas variáveis é apresentado na Figura 2.5.

Na Figura 2.5, as entradas do sistema real também alimentam um sistema que o representa, o sistema simulado. Essas entradas são processadas por ambos sistemas e, por fim, são comparadas, gerando um valor de resíduo. Este valor de resíduo é analisado e pode ser utilizado como base para informar a ocorrência ou não de falha no sistema real.



Figura 2.5: Estrutura Geral para Geração de Resíduos

A análise do resíduo isoladamente geralmente é utilizada para a detecção da falha. A porção do resíduo que corresponde ao sistema real pode ser obtida a partir da leitura de diversas variáveis do sistema real, como por exemplo, temperaturas, pressões que ainda não foram processadas, ou até mesmo o resultado do pré-processamento dessas variáveis medidas. Já a porção que corresponde ao sistema simulado pode ser obtida, por exemplo, através da predição dessas variáveis segundo um modelo matemático do sistema real.

Uma das desvantagens de se utilizar esta metodologia é o fato do sistema DDF ser bastante sensível a erros de modelagem, podendo provocar situações de falsos alarmes. Dessa forma, para se construir um sistema DDF o mais robusto possível, é importante conhecer o modelo nominal do sistema e um conjunto abrangente de possíveis falhas, observando os seus comportamentos.

As condições em que os sistemas reais operam não são ideais, ou seja, há a presença de ruídos, distúrbios, etc. Como o modelo matemático geralmente é construído considerando condições ideais, ele apresenta alguns erros. Dessa maneira, o resíduo (diferença entre a saída medida e a saída do modelo matemático) não é exatamente igual a zero quando não ocorre nenhuma falha. Assim, não se deve considerar que há falha caso o resíduo possua um valor diferente de zero, ou seja, o sistema DDF baseado na redundância analítica deve ser sensível às falhas e insensível às incertezas da modelagem e dos ruídos do sistema.

Uma maneira de minimizar os efeitos dos erros de modelagens e da presença de ruídos é utilizar uma faixa de valores assumida pelo resíduo que não indica a ocorrência de falhas. Por outro lado, essa abordagem pode apresentar alguns problemas, como por exemplo, pode haver comprometimento da sensibilidade do sistema ou pode ser difícil determinar essa faixa de valores, já que o resíduo varia com o sinal de entrada, com a magnitude e natureza dos distúrbios no sistema. Faixas grandes de valores podem diminuir a sensibilidade do sistema, já faixas menores podem provocar falsos alarmes mais frequen-

temente.

Existem várias abordagens para DDF baseadas na redundância analítica [Persin et al. 2002], [Ding et al. 2005]. Em vários casos, o modelo simulado não é necessariamente um modelo baseado na física do processo, podendo ser qualquer forma de mapeamento entrada-saída que estime a saída medida, permitindo a geração do resíduo. Por exemplo, um modelo pode ser representado por uma rede neural artificial treinada a partir de dados de entrada-saída do sistema operando normalmente [Tinós 1999].

2.4.3 Análise de Sinais

Algumas técnicas de detecção e diagnóstico de falhas são baseadas na análise conjunta em tempo e frequência da decomposição de sinais [Jones & Corbin 1989], [Mallet et al. 1997]. Com a análise espectral do sinal, consegue-se detectar a periodicidade de suas características, porém pode não funcionar bem quando utilizada para a detecção de falhas, por não responder bem a perturbações de altas frequências (associadas a certos tipos de falhas). Também podem existir situações em que a falha não ocorre inicialmente de forma repetitiva, não sendo detectada rapidamente a partir da análise espectral. Em algumas situações, pode ser que após algum tempo considerável a falha que vem ocorrendo passe a ser perceptível na análise do espectro do sinal. Dessa forma, a abordagem para a detecção de falhas deveria utilizar tanto informações espectrais quanto temporais.

Uma ferramenta bastante utilizada na área de análise de sinais é a Transformada de Wavelet [Haykin & Veen 2001], pois é bastante eficiente na análise não-estacionária dos sinais, na detecção de descontinuidades suaves [Mallet et al. 1997] e na filtragem de ruídos de medição.

Vários trabalhos sobre detecção de falhas são desenvolvidos utilizando wavelets para a detecção de falhas [de Souza et al. 2000], [Galvão & Yoneyama 1998], [Zhang & Yan 2001], [Paiva 2003]. Semelhantemente à redundância analítica, não há necessidade de equipamentos extras e se for aplicada somente aos sinais de saídas, e não aos sinais de resíduos, pode ser que não haja necessidade de um modelo matemático do sistema bastante próximo da realidade. Como desvantagem, tem-se as limitações relacionadas pela baixa resolução espectral em alta frequência e a latência na obtenção de informações relevantes em baixa frequência.

2.4.4 Inteligência Artificial

Em geral, as técnicas para detecção e diagnóstico de falhas baseadas em inteligência artificial (IA) são utilizadas para a geração de classificadores, que se baseiam nas variáveis

medidas do processo, ou para a reprodução do comportamento dinâmico do sistema, a partir do qual um vetor de resíduos pode ser gerado. As ferramentas de IA mais utilizadas para DDF são: sistemas baseados em conhecimento (ou sistemas especialistas), lógica nebulosa e redes neurais artificiais.

Sistemas Baseados em Conhecimento

Os sistemas baseados em conhecimento ou sistemas especialistas pertencem ao campo da Inteligência Artificial aplicada. Eles utilizam uma base de conhecimento heurístico, geralmente representada através de um conjunto de expressões condicionais qualitativas e com significado verbal, cujo mérito é serem semanticamente claras. Também têm a capacidade de ampliar sua base de conhecimento inicial, a partir de um processo de inferência ou “aprendizado”.

Os sistemas especialistas são formados por um conjunto de regras que analisam informações sobre um classe específica do problema. Geralmente são utilizados para a solução de problemas que normalmente são resolvidos por especialistas humanos e, para isso, é necessário que possuam uma base significativa de conhecimento do domínio do problema, com mecanismos de raciocínio variados através de regras pré-definidas [Rich & Knight 1994].

Existem vários trabalhos que utilizam sistemas especialistas na implementação de sistemas DDF [Hu et al. 1999], [Tatara et al. 2000], [Al-Taani 2005]. No contexto dos sistemas de DDF, um sistema especialista voltado para o diagnóstico de falhas é basicamente composto por uma série de eventos antecedentes e uma parte conseqüente aos quais é associada uma falha conhecida [Evsukoff et al. 1997]. Em outras palavras, as informações referentes à história do processo participam do sistema na forma de eventos antecedentes e conseqüentes, envolvendo um mapeamento explícitos dos sintomas conhecidos para a raiz das causas.

Lógica Nebulosa ou Lógica Fuzzy

A lógica nebulosa ou lógica fuzzy [Rich & Knight 1994] é aquela que suporta os modos de raciocínio que são aproximados, ao invés de exatos, como se naturalmente trabalha. Ela está baseada na teoria dos conjuntos nebulosos e difere dos sistemas lógicos tradicionais em suas características e detalhes.

A lógica fuzzy pode ser utilizada em sistemas de DDF [Koscielny et al. 1999], [Fries & Graham 2003], [Monsef et al. 1997] por diversos motivos, entre eles: possui conhecimento transparente em forma de regras, especialistas podem formular um conhecimento

formal usando valores linguísticos e possui habilidade na simulação de incertezas que deve ser consideradas para proporcionar um diagnóstico de falhas seguro.

Muitos trabalhos utilizam abordagens híbridas para diagnóstico de falhas, por exemplo, utilizando sistemas neuro-fuzzy [Palade et al. 2002], [da Silva Lira & de Carvalho Jr. 1999], [Uppal et al. 2005].

Redes Neurais Artificiais

As redes neurais apresentam uma alternativa para o diagnóstico automático de possíveis falhas que venham a ocorrer em um sistema em operação. A capacidade de aprender a partir de uma base de informações possibilita que as redes neurais adquiram o conhecimento necessário para interferir no funcionamento do sistema original, acusando a presença de uma dada falha. Algumas características de robustez e generalização das redes neurais [Haykin 2001] fazem com que elas sejam uma ferramenta bastante utilizada para detecção e classificação de falhas mesmo sob a presença de ruídos.

As RNAs podem ser usadas de duas formas para detecção e diagnóstico de falhas. Na primeira ela é usada para identificar o sistema que vai ser monitorado, para isso, é treinada com um conjunto entrada/saída que representa o sistema sob condições normais. Essa etapa pode ser executada coletando dados diretamente do sistema em funcionamento ou através de um modelo simulado. Este último deve ser o mais realista possível. Durante a reprodução do comportamento do sistema via a RNA, gera-se um vetor de resíduos [Köppen-Seliger & Frank 1995] a partir do qual será possível informar a ocorrência ou não de falhas.

No segundo caso, pode-se usar uma rede neural como um classificador, ou seja, para identificar qual o tipo de falha que está ocorrendo [Tinos & Terra 1998], [Naughton et al. 1996], [Shin et al. 2005], [J. 1994], [Srinivasan & Batur 1994], [Shin et al. 2005], [Linarié & Koroman 2003]. Suas entradas são sinais que podem ser analisados para decidir a existência e a classificação das falhas, ou seja, sinais de saída do sistema real e da rede neural que o identifica. Vale salientar que esta classificação depende de um conhecimento prévio que será passado como saída desejada do classificador.

Uma das vantagens de se utilizar RNAs no diagnóstico de falhas é não ser necessária a existência de um modelo analítico exato do sistema a ser diagnosticado. Além disso, é uma ótima ferramenta para tratar com problemas não lineares e possui grande capacidade de aproximação de funções. Por outro lado, há a necessidade de se ter um conjunto de dados representativos para o seu treinamento.

Capítulo 3

Proposta do Trabalho

Neste trabalho foi desenvolvido um sistema de detecção e isolamento de falhas utilizando redes neurais artificiais, aplicado a um sistema de níveis. Ele foi testado utilizando o modelo matemático do sistema e também a planta física real.

O esquema geral do funcionamento do sistema está exibido na Figura 3.1. No caso, ao mesmo tempo que o sistema de níveis está em execução, um sistema elaborado a partir de RNAs procura fazer a sua identificação utilizando suas mesmas entradas ($x(k)$). A cada instante, a saída do sistema de níveis ($y(k)$) é comparada à saída do sistema de identificação ($y'(k)$), gerando um valor de resíduo ($r(k) = y(k) - y'(k)$) que será posteriormente utilizado no sistema de detecção e isolamento/classificação de falhas. Este último analisa os valores residuais e indica a ocorrência ou não de falhas e caso ocorra, qual o seu tipo.

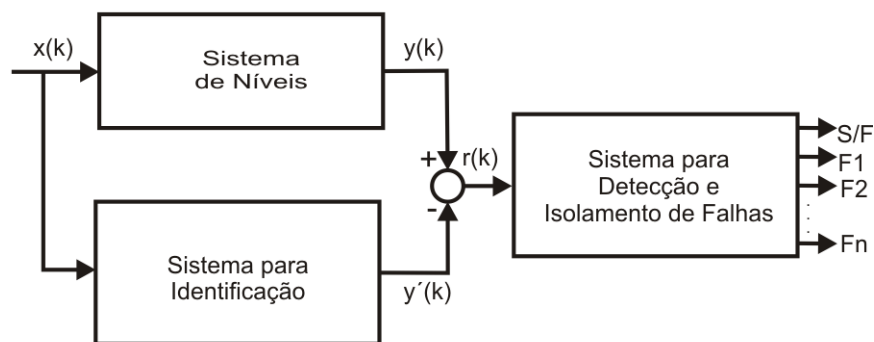


Figura 3.1: Esquema Geral do Sistema DIF

Dois abordagens para identificação neural do sistema de níveis foram desenvolvidas e testadas, porém apenas uma delas foi aplicada ao sistema para detecção e isolamento de falhas. Na próxima seção, o sistema de níveis será apresentado e, posteriormente, os passos seguidos para a conclusão do sistema DIF aplicado ao mesmo.

3.1 Processo Utilizado: Sistema de Níveis

A Figura 3.2 apresenta a planta de níveis composta por dois tanques em cascata, representando um modelo de segunda ordem com uma entrada. Nessa planta é aplicado uma estratégia de controle Proporcional Integrativo Derivativo (PID) [Ogata 1998], para controlar o nível do tanque 2.

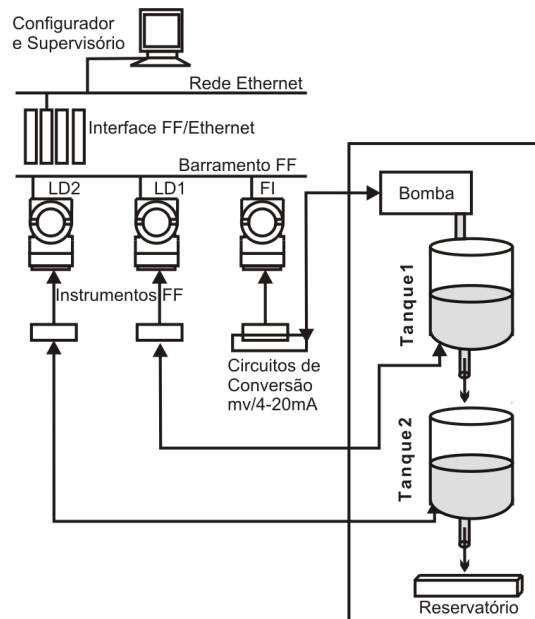


Figura 3.2: Arquitetura do Ambiente Real

A água que sai pelo orifício do tanque 1 cai dentro do tanque 2. Este, por sua vez, também tem um orifício onde a água cai diretamente para o reservatório. Uma bomba é responsável por levar a água do reservatório para o tanque 1. Em cada tanque existe um sensor de pressão *Foundation Fieldbus* (FF) [Lima et al. 2004], usado para medir os níveis correspondentes, ligados à rede *Fieldbus*. Além dos sensores de pressão, um conversor (FF/loop de corrente 4 a 20 mA) é usado para injetar sinais na bomba d'água.

A rede industrial *Foundation Fieldbus* é ligada a um computador supervisor através de interfaces de rede *Ethernet*. A partir deste computador é realizado todo o processo de configuração dos dispositivos, e o posterior monitoramento.

Os dados reais extraídos da planta para treinamento das redes neurais também foram captados da rede com o auxílio de softwares rodando neste computador. O padrão utilizado para comunicação entre o dispositivo mestre da rede FF e o computador configurador e supervisor é o OPC (*OLE for Process Control*) [Zheng & Nakagawa 2002]. O OPC é responsável por disponibilizar informações sobre o nível dos dispositivos no

nível de supervisão para qualquer computador da rede. Além disso, permite a alteração de alguns parâmetros de configuração do sistema FF.

A identificação neural foi feita em dois ambientes: simulado e real. Para a simulação do comportamento da planta, utilizou-se o modelo matemático não linear descrito pelas equações 3.1 e 3.2, presentes no manual do fabricante da planta de níveis [Qua n.d.].

$$\dot{L}_1 = -\frac{a_1}{A_1}\sqrt{2gL_1} + \frac{K_m}{A_1}V_p \quad (3.1)$$

$$\dot{L}_2 = -\frac{a_2}{A_2}\sqrt{2gL_2} + \frac{a_1}{A_2}\sqrt{2gL_1} \quad (3.2)$$

A Tabela 3.1 apresenta as descrições e valores dos parâmetros mostrados nas equações 3.1 e 3.2, usadas para estimar os níveis dos tanques 1 e 2, respectivamente.

Tabela 3.1: Parâmetros da Planta

Nome	Significado	Valor
K_m	Constante da bomba	$4,6(cm^3/s)/V$
V_p	Tensão aplicada à bomba	$-22 < V_p < 22$
a_1	Diâmetro da saída do tanque 1	$0,178139cm$
a_2	Diâmetro de saída do tanque 2	$0,178139cm$
A_1	Área do tanque 1	$15,5179cm^2$
A_2	Área do tanque 2	$15,5179cm^2$
g	Aceleração da gravidade	$980cm/s^2$

Por fim, o modelo da planta de níveis juntamente com o controlador PID está ilustrado na Figura 3.3. Como é possível observar, o sinal de referência é o nível desejado para o tanque 2. O sinal desejado ($d(k)$) é comparado ao nível atual do tanque 2 ($L_2(k)$), gerando um erro ($e(k)$) que é utilizado para o controlador PID para gerar um sinal de controle (V_p) para a bomba que injeta água no tanque 1.

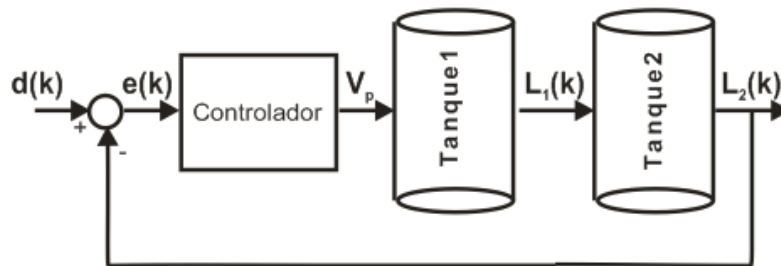


Figura 3.3: Planta de Níveis com Controlador PID

Finalizando esta seção, a Figura 3.4 mostra o ambiente em que se encontra a planta física de níveis no Laboratório de Automação da Medição em Petróleo (LAMP) da UFRN.



Figura 3.4: Fotografia do Ambiente Real Montado no LAMP

Na seção seguinte, serão apresentadas as abordagens desenvolvidas para a identificação neural do sistema de níveis.

3.2 Identificação Neural do Sistema

Como já citado, duas estratégias foram usadas para a identificação neural do sistema de níveis e ambas aplicadas ao ambiente simulado e ao ambiente real. Além disso, todas as RNAs utilizadas são do tipo Percéptron de Múltiplas Camadas (MLP) e foram treinadas com o algoritmo *Backpropagation*. O número de neurônios de suas camadas ocultas foi escolhido heurísticamente e os neurônios estão totalmente conectados.

Inicialmente, a identificação do sistema foi feita considerando-se o sistema como um todo, ou seja, uma única RNA foi treinada para gerar o nível do tanque 2. Essa primeira estratégia foi denominada como identificação em um único passo. Assim, a saída da RNA deve corresponder à saída estimada da planta, no caso, o nível do segundo tanque. O treinamento da RNA para essa abordagem está ilustrado na Figura 3.5.

A RNA utilizada na identificação em um único passo possui a seguinte arquitetura:

- Quatro nós na camada de entrada, onde as entradas são: $V_p(k)$, $V_p(k-1)$, $L_2(k-1)$ e $L_2(k-2)$;
- Uma camada oculta com quatro neurônios, com função de ativação sigmóide;
- Um neurônio na camada de saída (função de ativação linear), representando o nível do tanque 2 estimado no instante atual k , $L_2(k)$.

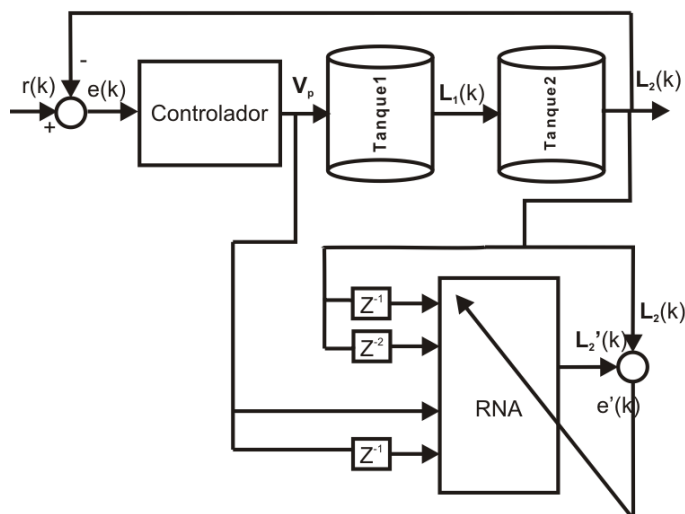


Figura 3.5: Treino para Determinar Nível do Tanque 2 com Única RNA

Como já citado, a identificação da planta de níveis descrita neste trabalho é a primeira etapa de um sistema maior de detecção e isolamento de falhas na própria planta. Pensando nesta segunda etapa, elaborou-se uma segunda estratégia de identificação que tem função de auxiliar a classificação de falhas. Dessa forma, verificou-se que a identificação neural do sistema seria melhor realizada se fosse feita em dois passos. Em dois passos significa que deve existir uma RNA para estimar o nível do tanque 1 e uma outra para estimar o nível do tanque 2. Os treinamentos das RNAs podem ser ilustrados pelas Figuras 3.6 e 3.7.

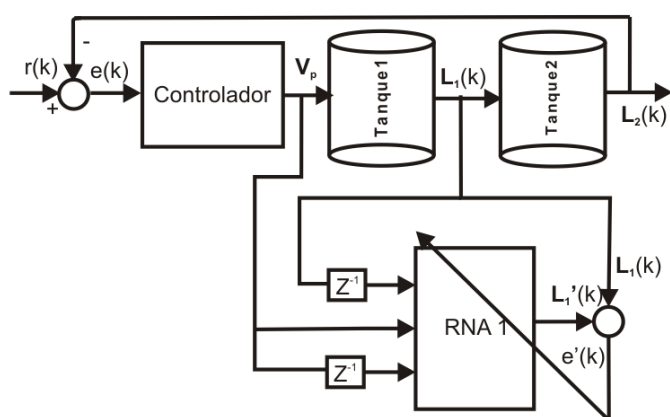


Figura 3.6: 1ª Etapa: Determinação do Nível do Tanque 1 a partir da Entrada do Sistema

A arquitetura da RNA 1, usada para estimar o nível do tanque 1, foi:

- Três nós na camada de entrada, onde as entradas são: $V_p(k)$, $V_p(k-1)$ e $L_1(k-1)$;

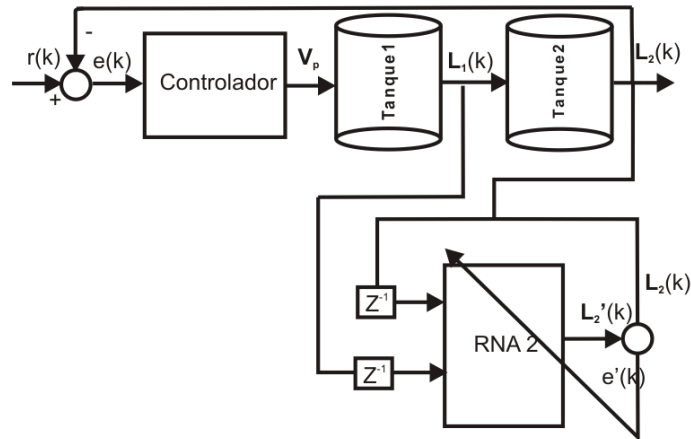


Figura 3.7: 2ª Etapa: Determinação do Nível do Tanque 2 a partir da Entrada do Sistema

- Uma camada oculta com três neurônios com função de ativação sigmóide;
- Um neurônio na camada de saída (função de ativação linear), representando o nível do tanque 1 estimado no instante atual k , $L_1(k)$.

A arquitetura da RNA 2, usada para estimar o nível do tanque 2, foi:

- Dois nós na camada de entrada, onde as entradas são: $L_1(k-1)$ e $L_2(k-1)$;
- Uma camada oculta com três neurônios com função de ativação sigmóide;
- Um neurônio na camada de saída (função de ativação linear), representando o nível do tanque 2 estimado no instante atual k , $L_2'(k)$.

Na próxima seção será apresentado o desenvolvimento do sistema DIF na planta de níveis, que utiliza como etapa inicial a identificação neural em dois passos apresentada anteriormente.

3.3 Detecção e Isolamento de Falhas

A partir de uma análise de possíveis falhas no sistema de níveis, verificou-se que elas poderiam ocorrer em sensores, atuadores ou no próprio processo. Um conjunto de trezes falhas foi identificado e elas foram classificadas como:

1. Vazamento no tanque 1, mas água que vaza não cai no tanque 2. Simulado a partir de um novo orifício no tanque 1, jogando parte da água para fora do mesmo;
2. Entupimento do orifício de saída do tanque 1. Simulado a partir da diminuição do orifício de saída do tanque 1, acumulando água no tanque 1 e reduzindo o fluxo de água de entrada para o tanque 2;

3. Entupimento do orifício de saída do tanque 2. Simulado a partir da diminuição do orifício de saída do tanque 2, aumentando o nível do tanque 2 e reduzindo o fluxo de água de entrada para o reservatório;
4. Vazamento no tanque 1, com água que vaza caindo no tanque 2. Simulado a partir do aumento do orifício de saída do tanque 1, reduzindo a água no tanque 1 e elevando o fluxo de água de entrada para o tanque 2;
5. Vazamento no tanque 2. Simulado com o aumento do orifício de saída do tanque 2, reduzindo a água no tanque 2 e elevando o fluxo de água de entrada para o reservatório;
6. Ruído positivo no sinal de controle. Simulado a partir da introdução de um *bias* que aumenta o valor do sinal de controle aplicado à bomba;
7. Ruído negativo no sinal de controle. Simulado a partir da introdução de um *bias* que diminui o valor do sinal de controle aplicado à bomba;
8. Erro de leitura no sensor 1, a partir da introdução de um *bias* positivo;
9. Erro de leitura no sensor 1, a partir da introdução de um *bias* negativo;
10. Erro de leitura no sensor 2, a partir da introdução de um *bias* positivo;
11. Erro de leitura no sensor 2, a partir da introdução de um *bias* negativo;
12. Fonte de água entrando no tanque 1 sem ser a partir da mangueira que joga água a partir da bomba, aumentando a quantidade de água no mesmo;
13. Fonte de água entrando no tanque 2 sem ser a partir do orifício de entrada de água do tanque 1, aumentando a quantidade de água no mesmo;

Depois de escolhida a maneira que o sistema de níveis vai ser representado (identificação em dois passos), e de treinar as redes neurais responsáveis pela identificação do mesmo, treinou-se o classificador das falhas. Com esta estratégia de identificação, é possível obter dois resíduos, R_1 e R_2 , onde $R_1 = L_1 - L'_1$ e $R_2 = L_2 - L'_2$. Através de alguns testes realizados, que serão apresentados no Capítulo 5, foi possível observar que as 13 falhas observadas apresentavam os comportamentos para os resíduos mostrados na Tabela 3.2, onde + representa o resíduo com valor positivo, - o resíduo com valor negativo e 0 o resíduo igual ou muito próximo a zero.

Observando a disposição dos resíduos para as falhas presente na Tabela 3.2, elaborou-se a estratégia para a classificação das falhas ilustrada na Figura 3.8. Neste caso, uma RNA, denominada RNA 3, é treinada recebendo como dados de entrada os valores de R_1 e R_2 . A saída da rede corresponde a um vetor de $N + 1$ números, onde N é a quantidade de falhas que a rede será capaz de classificar. Por exemplo, considerando $N = 3$, ter-se-ia as possíveis saídas para a rede mostradas na Tabela 3.3.

Tabela 3.2: Disposição dos Resíduos para as 13 Falhas

Falha	Descrição	R_1	R_2
	Ausência de Falha	0	0
1	Novo orifício no tanque 1. Água não cai no tanque 2	-	0
2	Diminuição do orifício do tanque 1	+	-
3	Diminuição do orifício do tanque 2	-	+
4	Aumento do orifício do tanque 1	-	+
5	Aumento do orifício do tanque 2	0	-
6	Aumento do sinal de controle	+	0
7	Redução do sinal de controle	-	0
8	Erro de leitura no sensor 1. <i>Bias</i> positivo	+	0
9	Erro de leitura no sensor 1. <i>Bias</i> negativo	-	0
10	Erro de leitura no sensor 2. <i>Bias</i> positivo	0	+
11	Erro de leitura no sensor 2. <i>Bias</i> negativo	0	-
12	Fluxo extra de água no tanque 1	+	0
13	Fluxo extra de água no tanque 2	0	+

De acordo com a Tabela 3.2, observa-se que alguns tipos de falhas possuem o mesmo comportamento para os resíduos 1 e 2. As falhas que possuem resíduos semelhantes são:

- Falhas 1, 7 e 9 $\Rightarrow R_1$ negativo e R_2 próximo a zero.
- Falhas 3 e 4 $\Rightarrow R_1$ negativo e R_2 positivo.
- Falhas 5 e 11 $\Rightarrow R_1$ igual próximo a zero e R_2 negativo.
- Falhas 6, 8 e 12 $\Rightarrow R_1$ positivo e R_2 próximo a zero.
- Falhas 10 e 13 $\Rightarrow R_1$ próximo a zero e R_2 positivo.

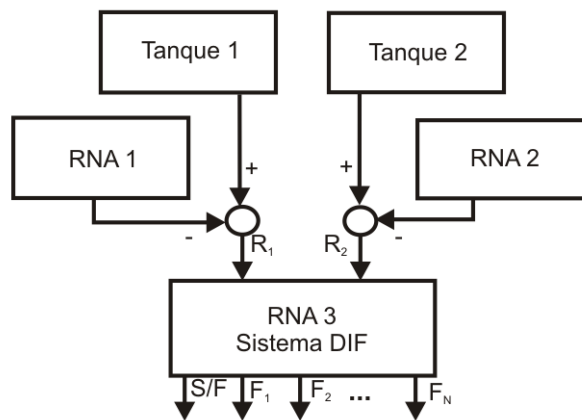


Figura 3.8: Esquema Geral para Classificação das Falhas

Tabela 3.3: Possíveis Saídas para Classificação de 3 Falhas

Situação Normal	1	0	0	0
Falha 1	0	1	0	0
Falha 2	0	0	1	0
Falha 3	0	0	0	1

Dessa maneira, a estratégia exibida na Figura 3.8 só funciona para classificar falhas com comportamentos de resíduos distintos, de forma que só é possível classificar falhas dentro de um conjunto de sete situações, como mostrado na Tabela 3.4.

Tabela 3.4: Sete Situações: Seis Conjuntos de Falhas e Ausência de Falhas

Situação	Falhas	R_1	R_2	Saída da RNA 3
1	Sem Falha	0	0	1000000
2	1, 7 e 9	-	0	0100000
3	2	+	-	0010000
4	10 e 13	0	+	0001000
5	3 e 4	-	+	0000100
6	5 e 11	0	-	0000010
7	6 e 8 e 12	+	0	0000001

A RNA 3 também é do tipo MLP, foi treinada com o algoritmo da *Backpropagation*, para a qual o número de neurônios foi escolhido heurísticamente e estão totalmente conectados. Com base na situação descrita, a arquitetura da RNA 3 ficou da seguinte maneira:

- Dois nós na camada de entrada, onde as entradas são: R_1 e R_2 ;
- Uma camada oculta com 6 neurônios com função de ativação sigmóide;
- Sete neurônios na camada de saída (função de ativação linear), representando uma das situações ilustradas na Tabela 3.4 no instante atual k .

Como o classificador neural trabalha em cima de valores positivos, negativos e próximos a zero para os resíduos, ele poderia ter sido traduzido em um conjunto de regras com sentenças condicionais. Por exemplo, se R_1 for positivo e R_2 negativo, bastaria fazer um teste condicional e indicar que a falha 2 está ocorrendo. Por outro lado, se um outro sistema dinâmico pudesse ser modelado por N resíduos, onde N é um valor muito grande, ficaria um pouco trabalhoso codificar diversas sentenças condicionais para informar a

falha. Além disso, o neural pode classificar falhas com resíduos semelhantes (duas falhas diferentes com mesmos R_1 e R_2 , por ex., $R_1 = +$ e $R_2 = 0$), porém com amplitudes em faixas de valores distintas, sem a necessidade de um novo teste condicional verificando o limiar da falha. A principal vantagem de se utilizar a RNA como classificador é o fato dela ser uma ótima generalizadora de funções, por exemplo, dificilmente os resíduos assumirão valores iguais a zero, então quando alguma situação for considerada com valor de R_1 ou R_2 igual a zero, significará que ele estará dentro de uma faixa de valores (positivos e negativos) muito próximos a zeros. Assim, com o classificador baseado em regras, seria necessária a definição de um limiar que identificasse resíduos nulos ou próximos de zero.

É importante ressaltar que o classificador foi treinado apenas considerando o modelo simulado (matemático), e o mesmo foi aplicado para classificar falhas no sistema real. Essa abordagem foi escolhida devido ao fato de que em situações reais, onde não se tem uma base de dados com informações de falhas ocorridas ao longo do tempo, poderia ficar complicado gerar falhas no sistema para assim poder desenvolver um aplicativo que pudesse detectá-las. O ideal seria a aquisição de uma base de dados do sistema que possuísse informações sobre suas variáveis durante a ocorrência de falhas no seu funcionamento normal.

Capítulo 4

Resultados da Identificação Neural

Neste capítulo serão apresentados os resultados de identificação neural do sistema de níveis para ambas abordagens aplicadas, tanto a partir do modelo matemático da planta quanto da execução da planta real de níveis.

Como já citado, duas estratégias para identificação neural foram implementadas, a primeira denominada Identificação em um Único Passo e a segunda, Identificação em Dois Passos.

As Figuras 4.1, 4.2 e 4.3 ilustram as arquiteturas da RNA utilizada para a identificação em um único passo (Ver Figura 3.5), das RNA 1 e RNA 2 utilizadas na identificação em dois passos (Ver Figuras 3.6 e 3.7), respectivamente.

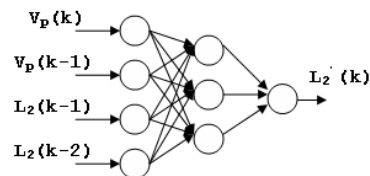


Figura 4.1: Arquitetura da RNA para Identificação em um Único Passo

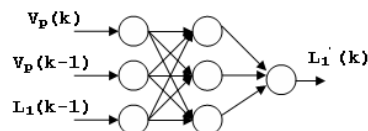


Figura 4.2: Arquitetura da RNA 1 para Identificação em Dois Passos

Nas duas seções seguintes serão apresentados os resultados referentes à identificação neural, obtidos a partir de simulações (via modelo matemático da planta) e do ambiente real onde se encontra a planta física de níveis.

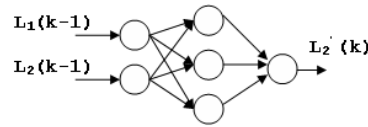


Figura 4.3: Arquitetura da RNA 2 para Identificação em Dois Passos

4.1 Resultados com Modelo Matemático - Simulação

Para ambas estratégias, os dados para treinamento das redes neurais foram captados aplicando sinais de referências à planta de níveis que tem o nível do tanque 2 controlado por um controlador PID, com os seguintes valores para os ganhos: $K_p = 2$, $T_i = 25$ e $T_d = 0,05$.

Na Figura 4.4, observa-se gráficos que exemplificam a planta simulada em funcionamento. O primeiro gráfico corresponde ao valor do nível do tanque 1 ao longo do tempo. Já o segundo, mostra o nível do tanque 2 em azul e a referência aplicada em vermelho. Por fim, o terceiro gráfico exibe o sinal de controle que está sendo aplicado à bomba, gerado a partir do controlador PID.

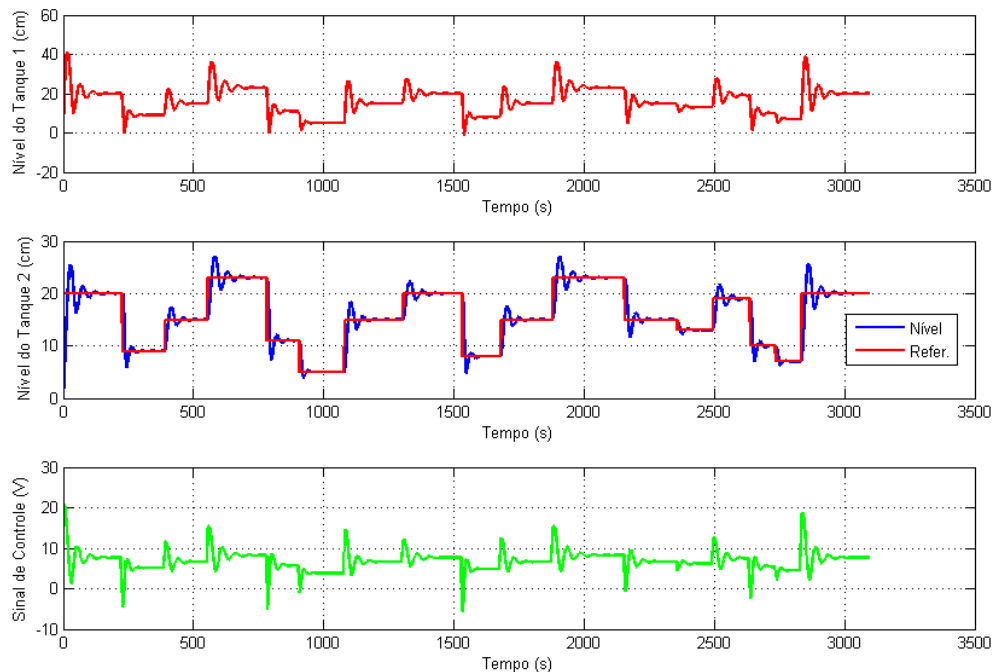


Figura 4.4: Exemplo de Conjunto de Dados para Treinamento

Durante a captação para dados de treinamento das redes e testes, os sinais de referên-

cias aplicados sempre foram no estilo degrau ou senóide [Haykin & Veen 2001], com a amplitude variando ao longo do tempo. Assim, a partir do funcionamento normal da planta, foi possível extrair os dados para treinamento das redes, que dependendo dela, poderia ser o nível do tanque 1, o nível do tanque 2 e o sinal de controle aplicado à bomba. Um exemplo de conjunto de dados para treinamento está ilustrado na Figura 4.4.

4.1.1 Identificação em Um Único Passo

Na identificação neural em um único passo, apenas uma rede foi utilizada e possui as seguintes entradas $V_p(k)$ e $V_p(k-1)$, $L_2(k-1)$ e $L_2(k-2)$ e uma única saída que é o valor estimado do nível do tanque 2 no instante k , $L_2(k)$. A Figura 4.5 apresenta a evolução do erro quadrático médio durante o treinamento desta rede, que chega em torno de 10^{-7} cm^2 .

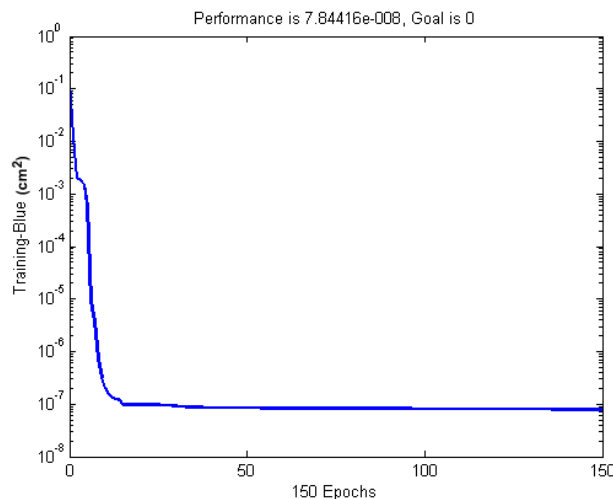


Figura 4.5: Evolução do Erro Quadrático Médio Durante o Treinamento da Rede para Identificação em Um Único Passo

Após o treinamento, realizou-se a validação do treinamento, aplicando o mesmo sinal de referência para o nível do tanque 2 utilizado no treinamento. A Figura 4.6 apresenta dois gráficos utilizados para validação do treinamento, onde o primeiro mostra o nível do tanque 2 gerado pela simulação (em azul), o mesmo gerado pela RNA treinada (em vermelho) e o sinal de referência para controlador (em verde). O segundo gráfico apresenta o erro entre a saída do modelo matemático e a saída gerada pela rede, mostrando-se dentro de uma faixa aceitável. O erro quadrático médio para esta validação foi em torno de $0,04 \text{ cm}^2$.

Após o treinamento e a validação da rede neural para identificação do sistema de níveis, alguns testes foram realizados, dentre eles os presentes nas Figuras 4.7, 4.8 e 4.9.

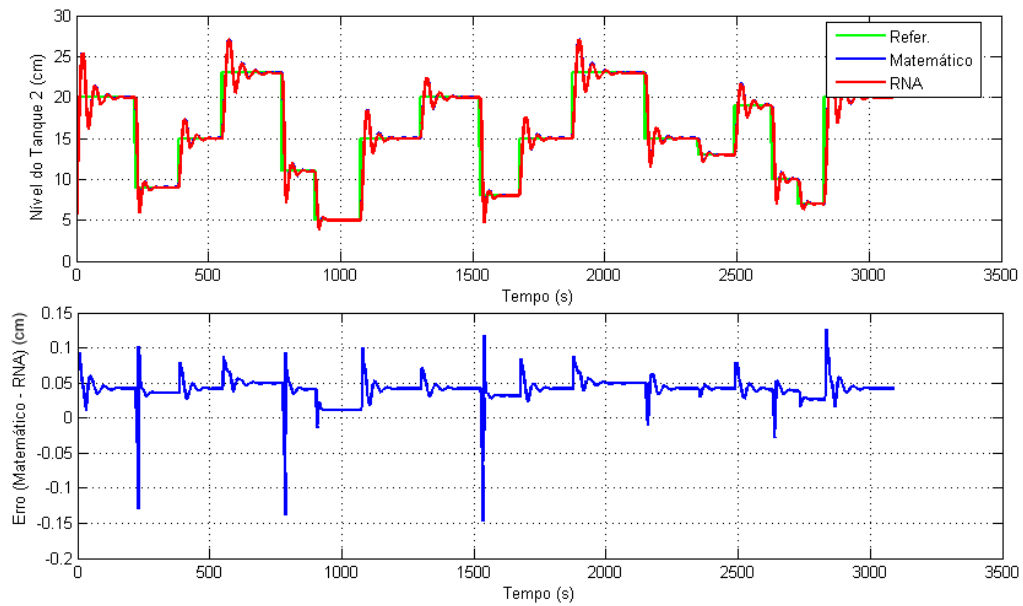


Figura 4.6: Validação do Treinamento - Identificação em Único Passo

Para cada exemplo destas três figuras, os erros quadráticos médios foram de $0,035 \text{ cm}^2$, $0,038 \text{ cm}^2$ e $0,0378 \text{ cm}^2$, respectivamente. A partir destes resultados, pode-se ver que a rede neural conseguiu acompanhar o comportamento do nível do tanque 2.

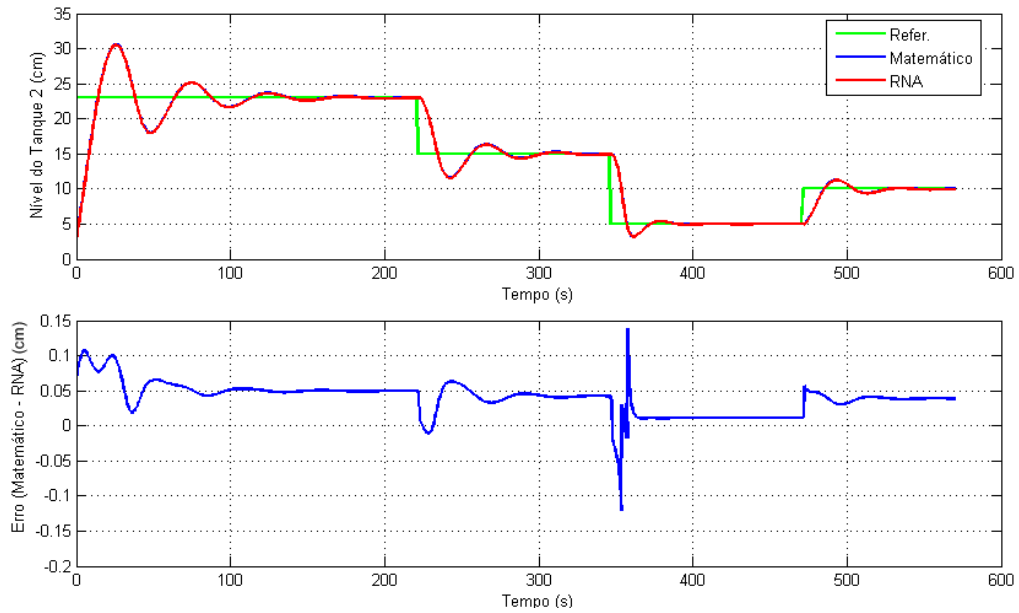


Figura 4.7: Exemplo 1 - Identificação em Único Passo - Simulação

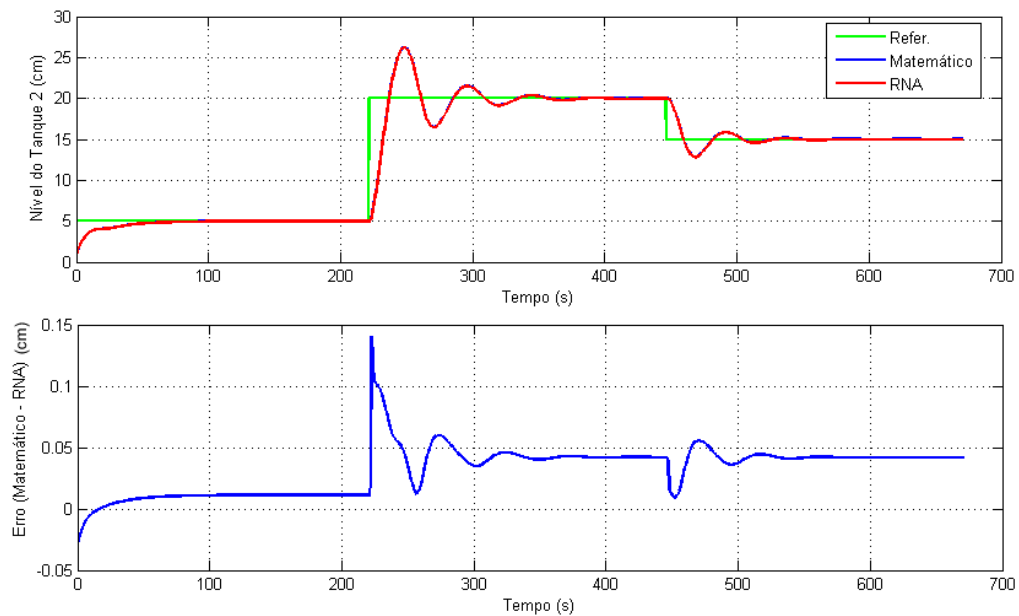


Figura 4.8: Exemplo 2 - Identificação em Único Passo - Simulação

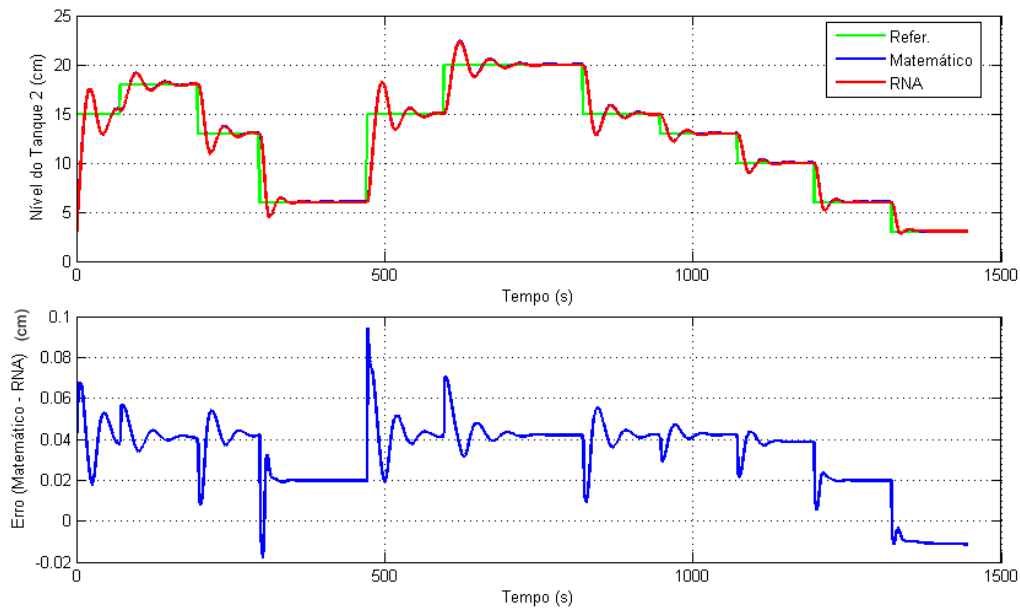


Figura 4.9: Exemplo 3 - Identificação em Único Passo - Simulação

4.1.2 Identificação em Dois Passos

Com a identificação neural em dois passos, duas redes foram utilizadas. Relembrando, a RNA 1 possui como entradas $V_p(k)$ e $V_p(k-1)$ e $L_1(k-1)$ e como saída o valor estimado do nível do tanque 1 no instante k , $L_1(k)$ e a RNA 2 tem como entradas $L_1(k-1)$ e $L_2(k-1)$ e como saída o valor estimado do nível do tanque 2 no instante k , $L_2(k)$. As Figuras 4.10 (a) e 4.10 (b) apresentam as evoluções dos erros quadráticos médios durante o treinamento das duas redes, respectivamente, ambos também em torno de 10^{-7} cm^2 .

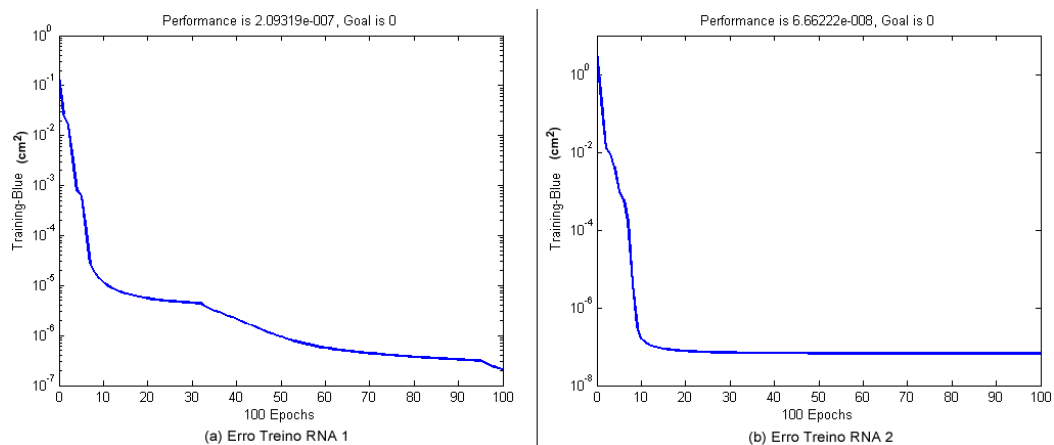


Figura 4.10: Evolução dos Erros Quadráticos Médios Durante o Treinamento das Redes 1 e 2 para Identificação em Dois Passos

Semelhantemente à estratégia anterior, também foi feita uma validação do treinamento. No caso, a Figura 4.11 apresenta esta validação para um dos conjuntos de dados utilizado para treinamento. O primeiro gráfico mostra o nível do tanque 1 gerado pelo modelo matemático (em azul) e o mesmo gerado pela RNA 1 (em vermelho). O segundo, mostra o nível do tanque 2 gerado pelo modelo matemático (em azul) e o mesmo gerado pela RNA 2 (em vermelho), além do sinal de referência (em verde) utilizado. Já o terceiro e último gráfico apresenta as diferenças entre os saídas dos tanques 1 e 2 geradas pelo modelo matemático e as geradas pelas RNA 1 e 2, denominadas Erro 1 (em azul) e Erro 2 (em vermelho), respectivamente. Como é possível observar, os valores dos erros apresentados para esta abordagem foram menores do que os apresentados pela abordagem de identificação explicada anteriormente. Para esta validação, os erros quadráticos médios foram de $0,0007 \text{ cm}^2$, considerando o nível do tanque 1 e de $0,0023 \text{ cm}^2$, considerando o nível do tanque 2.

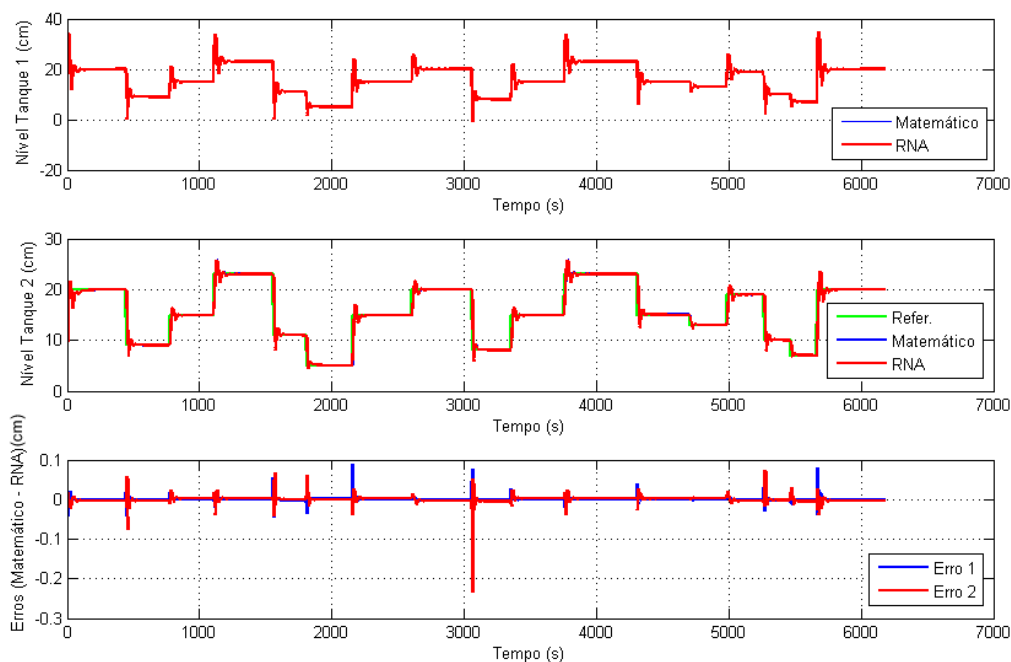


Figura 4.11: Validação do Treinamento - Identificação em Dois Passos

Após a validação do treinamento das redes neurais 1 e 2, executou-se alguns testes para ver como as mesmas se comportavam identificando o processo do sistema de níveis. As Figuras 4.12, 4.13 e 4.14 apresentam alguns dos resultados obtidos. Para cada um desses exemplos, os erros quadráticos médios foram ($0,00691 \text{ cm}^2$, $0,0066 \text{ cm}^2$), ($0,0082 \text{ cm}^2$, $0,0108 \text{ cm}^2$) e ($0,0065 \text{ cm}^2$ e $0,0087 \text{ cm}^2$), respectivamente, onde o formato (E1, E2), corresponde a:

- E1 = erro quadrático médio entre nível do tanque 1 simulado e o mesmo gerado pela RNA 1;
- E2 = erro quadrático médio entre nível do tanque 2 simulado e o mesmo gerado pela RNA 2.

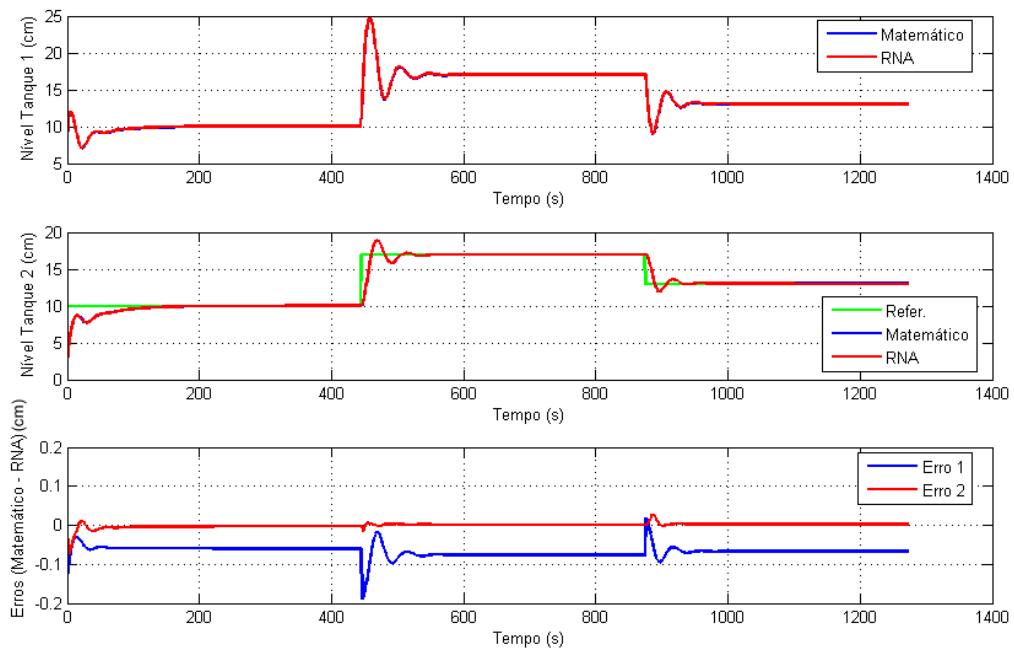


Figura 4.12: Exemplo 1 - Identificação em Dois Passos - Simulação

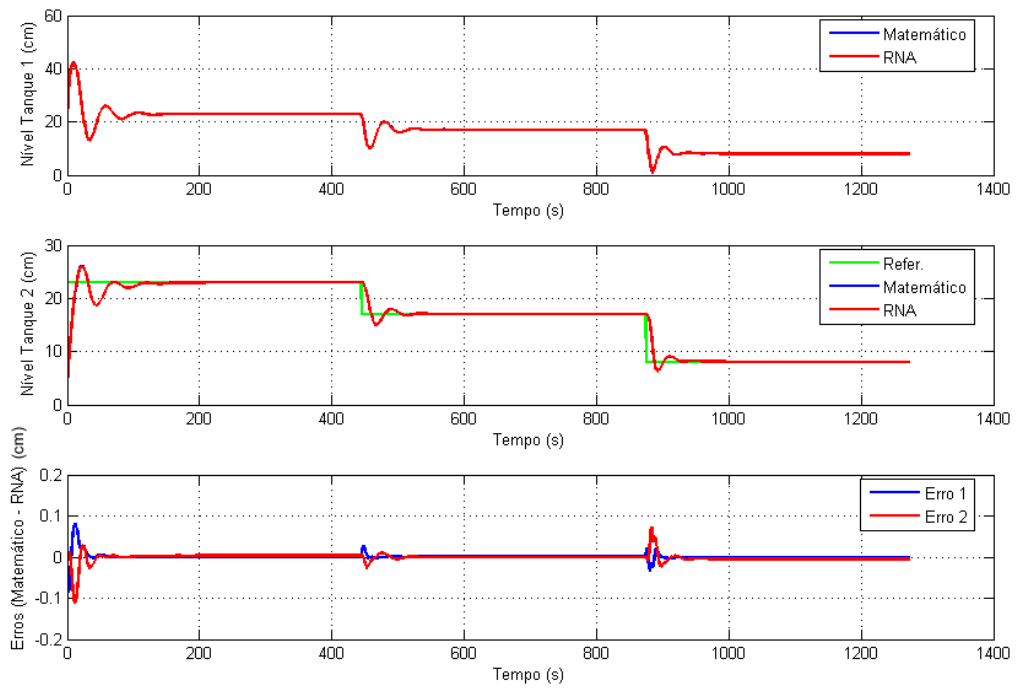


Figura 4.13: Exemplo 2 - Identificação em Dois Passos - Simulação

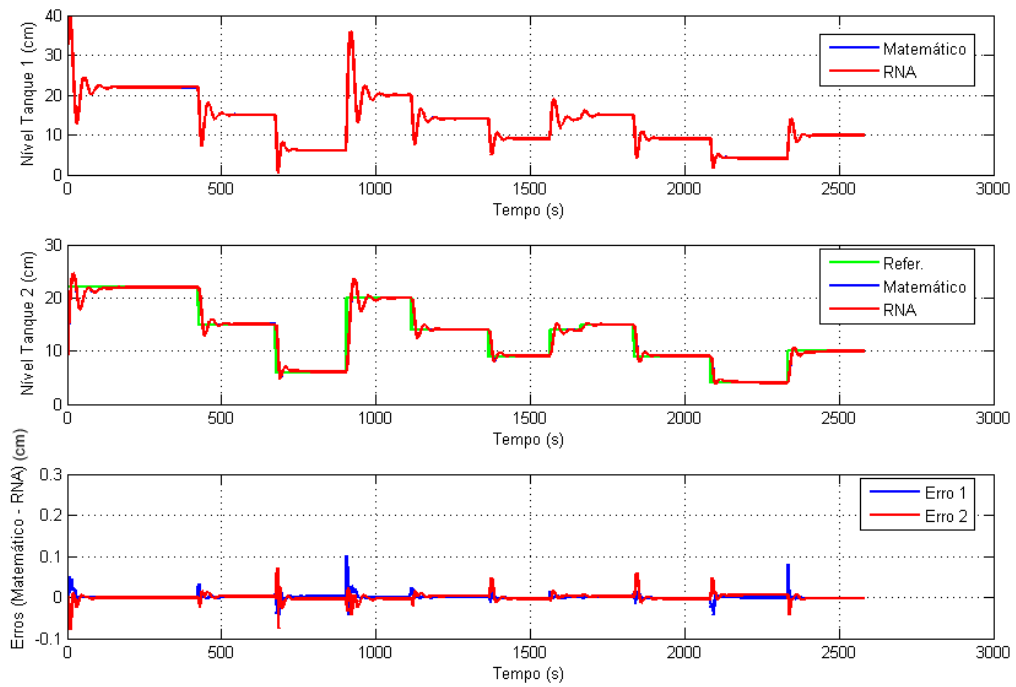


Figura 4.14: Exemplo 3 - Identificação em Dois Passos - Simulação

4.2 Resultados Reais

Da mesma forma que nos testes realizados com o modelo matemático da planta, os dados captados para treinamento das redes para identificação neural da planta real de níveis foram gerados a partir do funcionamento normal da mesma. Os sinais de referência aplicados à planta também eram dos tipos degrau ou senóide. Vale salientar que as mesmas arquiteturas das redes neurais usadas na simulação foram utilizadas nos testes na planta real.

A Figura 4.15 mostra um dos conjuntos de dados utilizados para treinamento. No primeiro gráfico, tem-se o valor do nível do tanque 1 ao longo do tempo. Já no segundo, exibe-se o nível do tanque 2 em azul e a referência utilizada pelo controlador PID em vermelho. O terceiro gráfico exibe o sinal de controle que está sendo aplicado à bomba, gerado pelo controlador, que utiliza os mesmos ganhos usados com o modelo matemático, ou seja, $K_p = 2$, $T_i = 25$ e $T_d = 0,05$.

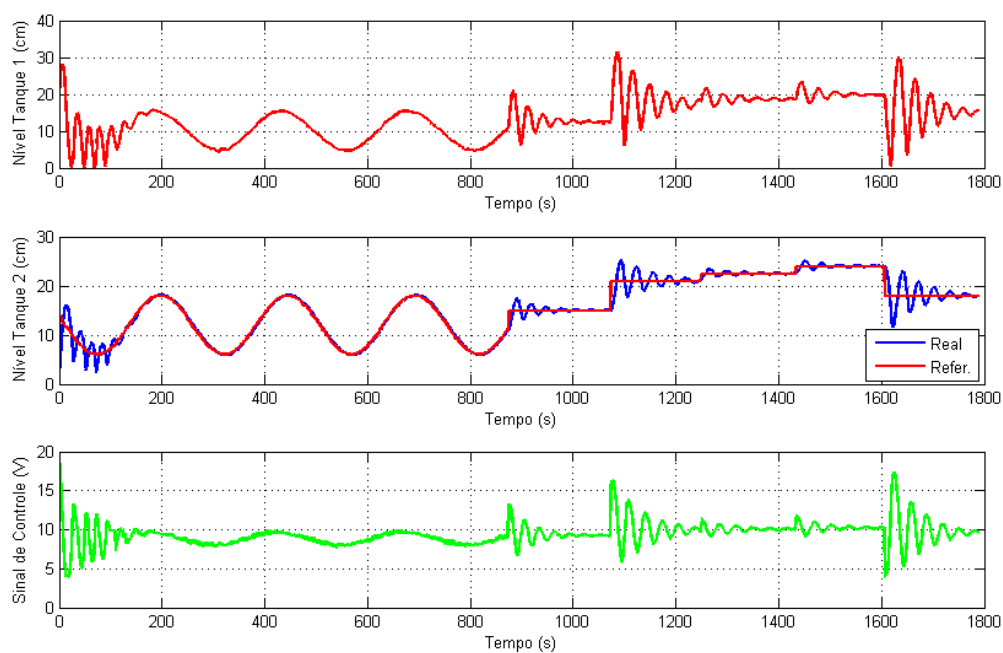


Figura 4.15: Exemplo de Conjunto de Dados para Treinamento - Ambiente Real

Nas subseções seguintes, alguns resultados obtidos na identificação neural da planta de níveis real serão apresentados.

4.2.1 Identificação em Um Único Passo

Depois do treinamento da RNA para identificação em um único passo, conseguiu-se uma evolução do erro ilustrada na Figura 4.16, onde o erro quadrático médio ficou em torno de 10^{-5} cm^2 .

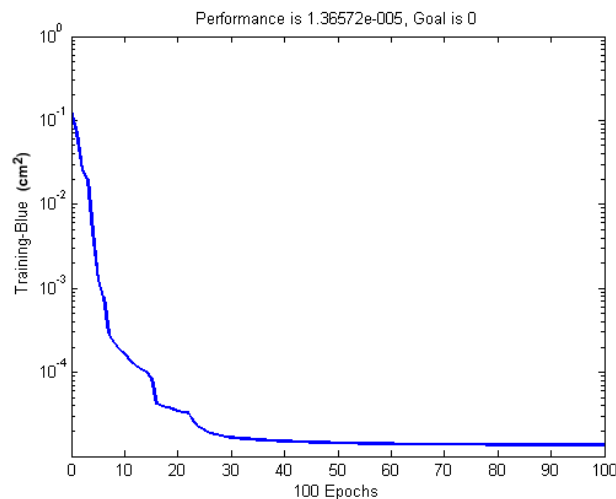


Figura 4.16: Evolução do Erro Quadrático Médio Durante o Treinamento da Rede para Identificação em Um Único Passo - Ambiente Real

Da mesma forma que com os dados simulados, após o treinamento, realizou-se a validação da rede empregando-se os mesmos dados utilizados para treinamento da rede. A Figura 4.17 exibe a validação após o treinamento para o conjunto de dados presente na Figura 4.15, onde o erro quadrático médio ficou em torno de $0,1352 \text{ cm}^2$. Nesta figura, o primeiro gráfico exibe o sinal em verde corresponde ao sinal de referência para o controlador PID; o em azul ao nível do tanque 2 ao longo do tempo; e o vermelho o nível do tanque 2 estimado pela rede neural. O segundo gráfico corresponde à diferença entre o nível real e estimado do tanque 2.

Validada a rede neural para identificação em único passo, alguns testes foram executados para ver se a mesma estava identificando o funcionamento do sistema da planta de níveis real. Os dados utilizados foram diferentes dos utilizados durante o treinamento. As Figuras 4.18, 4.19 e 4.20 apresentam alguns dos resultados obtidos. Para cada um desses exemplos, os erros quadráticos médios entre o nível do tanque 2 gerado pela RNA e o gerado pela execução do sistema foram de $0,1438 \text{ cm}^2$, $0,2453 \text{ cm}^2$ e $0,0749 \text{ cm}^2$, respectivamente.

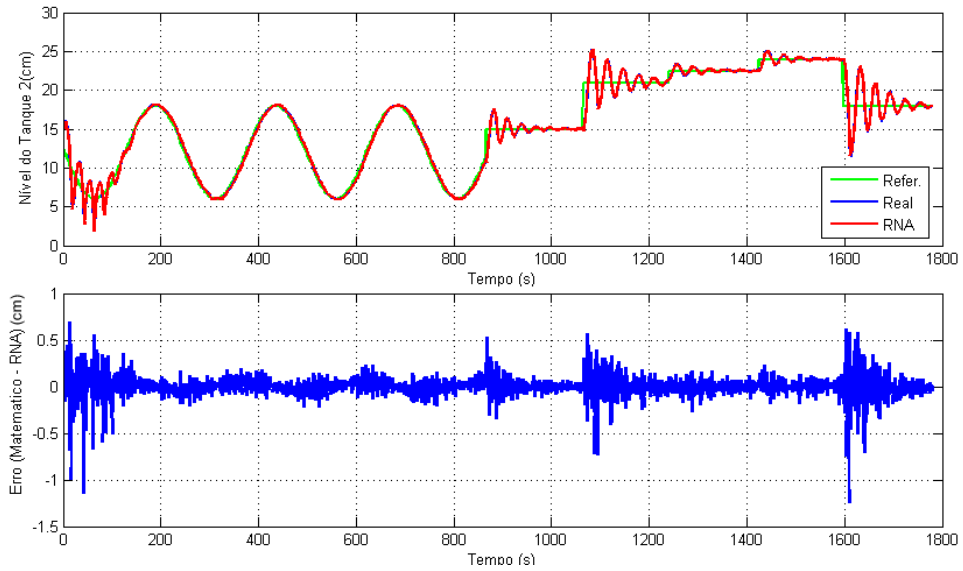


Figura 4.17: Validação do Treinamento - Identificação em Único Passo - Ambiente Real

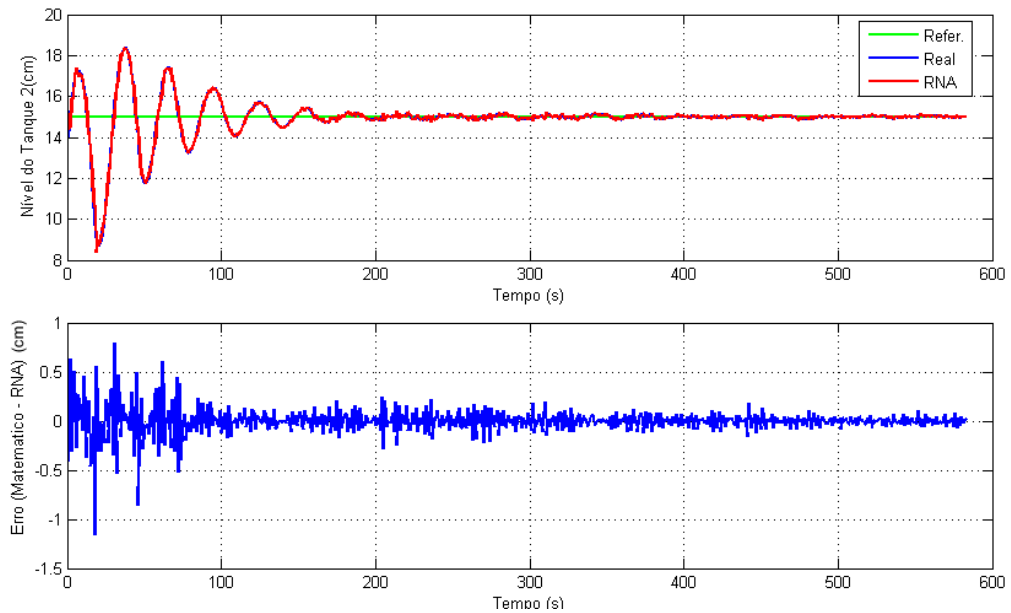


Figura 4.18: Exemplo 1 - Identificação em Único Passo - Ambiente Real

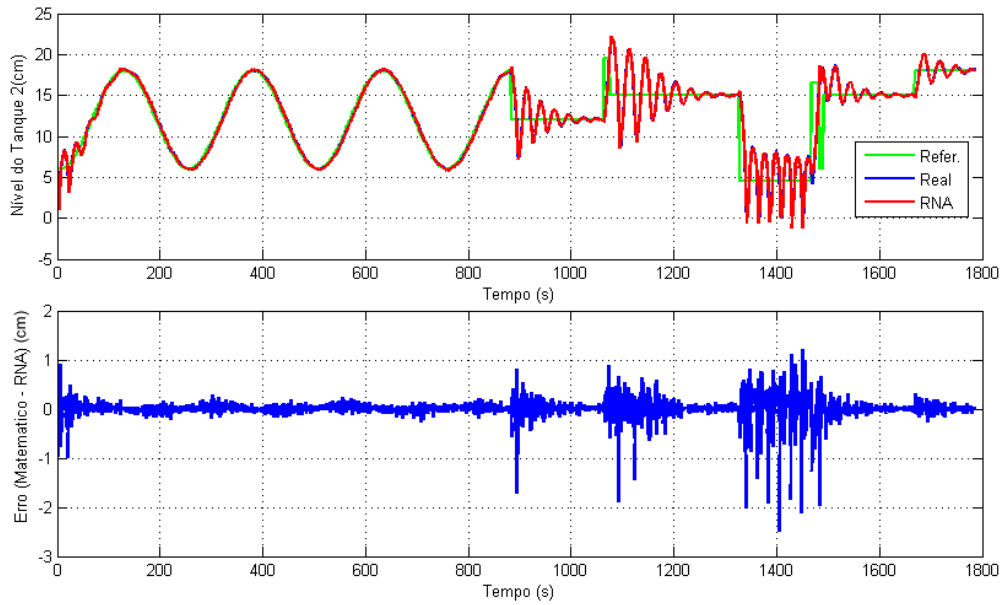


Figura 4.19: Exemplo 2 - Identificação em Único Passo - Ambiente Real

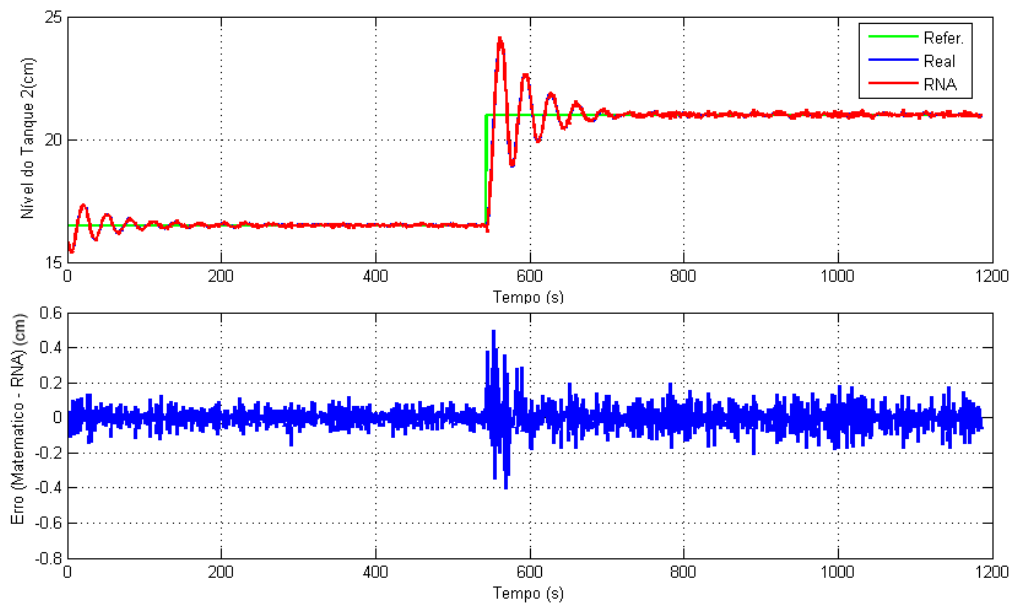


Figura 4.20: Exemplo 3 - Identificação em Único Passo - Ambiente Real

4.2.2 Identificação em Dois Passos

Para a identificação neural do sistema de níveis em dois passos, as arquiteturas das redes RNA 1 e RNA 2 também foram as mesmas utilizadas na simulação. As Figuras 4.21 (a) e 4.21 (b) exibem as evoluções dos erros quadráticos médios durante treinamento, chegando em torno de 10^{-4} cm^2 e 10^{-5} cm^2 para as RNA 1 e 2, respectivamente.

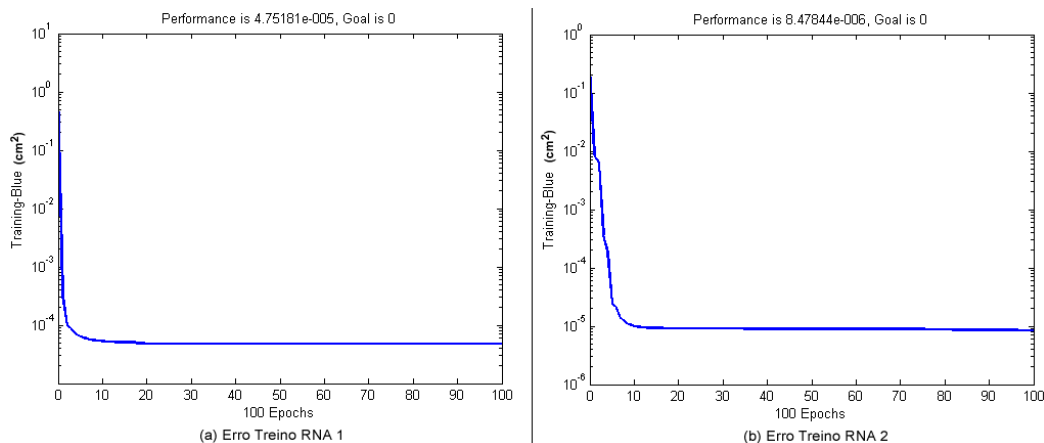


Figura 4.21: Evolução dos Erros Quadráticos Médios Durante o Treinamento das Redes 1 e 2 para Identificação em Dois Passos - Ambiente Real

A Figura 4.22 exibe a validação do treinamento para o conjunto de dados da Figura 4.15. No primeiro gráfico é possível visualizar o nível do tanque 1 (em azul) e o mesmo gerado pela RNA 1 (em vermelho). No segundo, mostra-se o nível do tanque 2 (em azul), o mesmo gerado pela RNA 2 (em vermelho), e a referência para o nível do tanque 2 a ser seguida (em verde). Já o terceiro gráfico apresenta as diferenças entre as saídas dos tanques 1 e 2 geradas pela planta real e as geradas pelas RNA 1 e 2, também denominadas Erro 1 (em azul) e Erro 2 (em vermelho), respectivamente. Nesta validação, os erros quadráticos médios foram de $0,2236 \text{ cm}^2$ considerando o nível do tanque 1 e de $0,1032 \text{ cm}^2$ considerando o nível do tanque 2.

Alguns testes também foram executados após a validação das redes. As Figuras 4.23, 4.24 e 4.25 apresentam alguns dos resultados obtidos. Para cada um desses exemplos, os erros quadráticos médios foram $(0,2431 \text{ cm}^2, 0,1405 \text{ cm}^2)$, $(0,2050 \text{ cm}^2, 0,1268 \text{ cm}^2)$ e $(0,1985 \text{ cm}^2, 0,1039 \text{ cm}^2)$, respectivamente. Da mesma forma que explicado anteriormente, o formato (E1, E2), corresponde a E1 = erro quadrático médio entre nível do tanque 1 e o mesmo gerado pela RNA 1 e E2 = erro quadrático médio entre nível do tanque 2 e o mesmo gerado pela RNA 2.

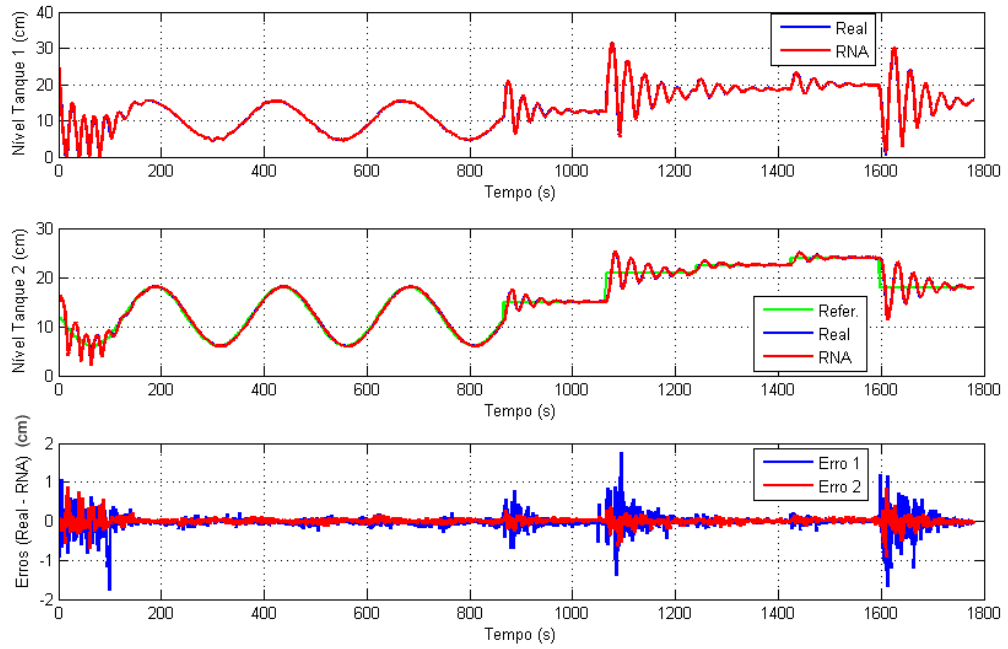


Figura 4.22: Validação do Treinamento - Identificação em Dois Passos - Ambiente Real

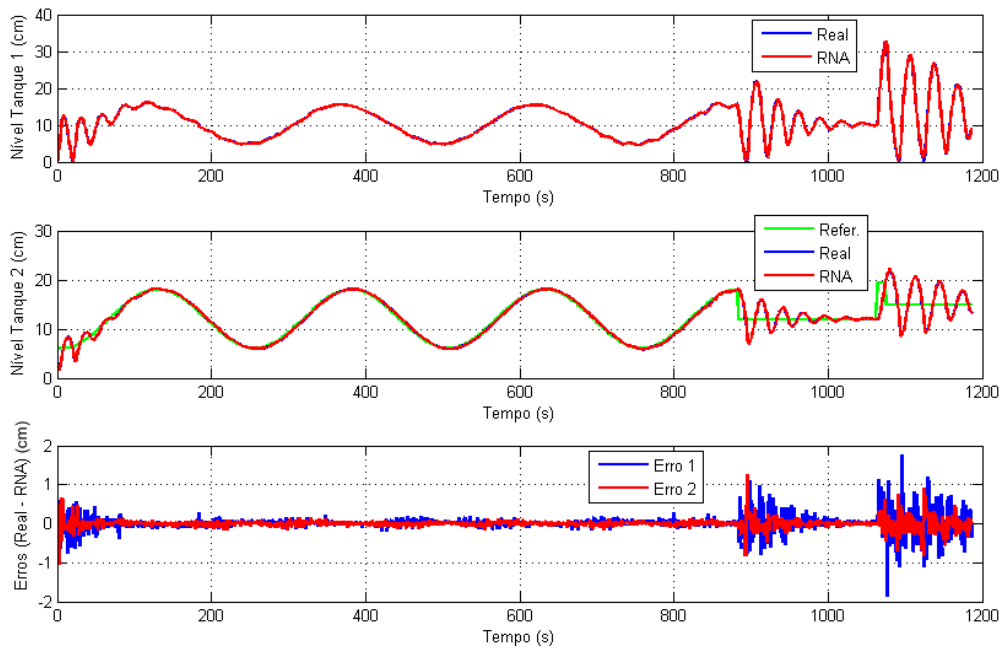


Figura 4.23: Exemplo 1 - Identificação em Dois Passos - Ambiente Real

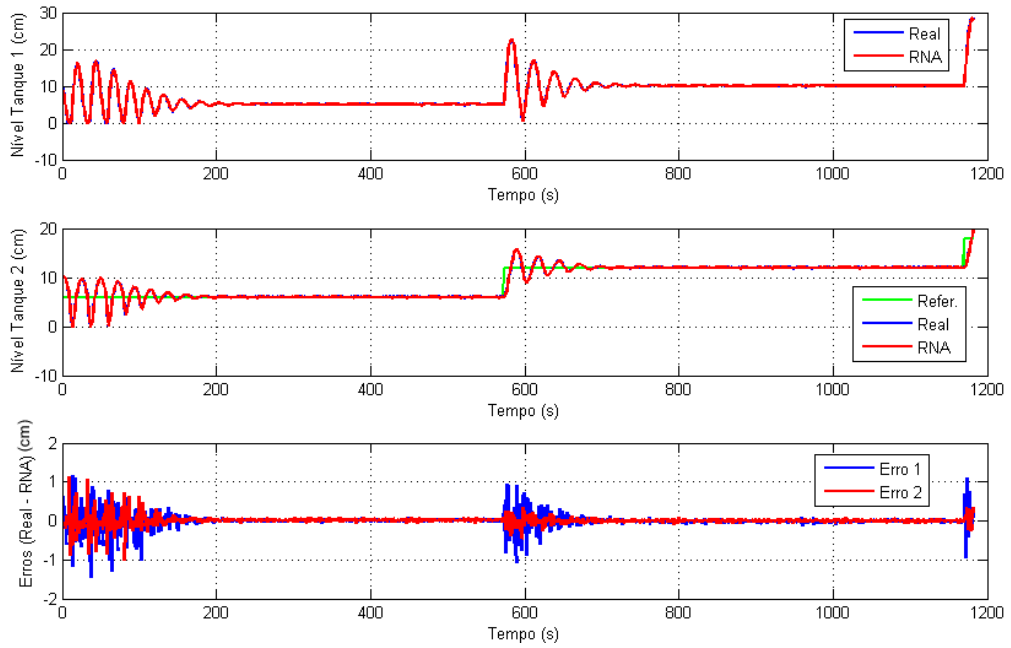


Figura 4.24: Exemplo 2 - Identificação em Dois Passos - Ambiente Real

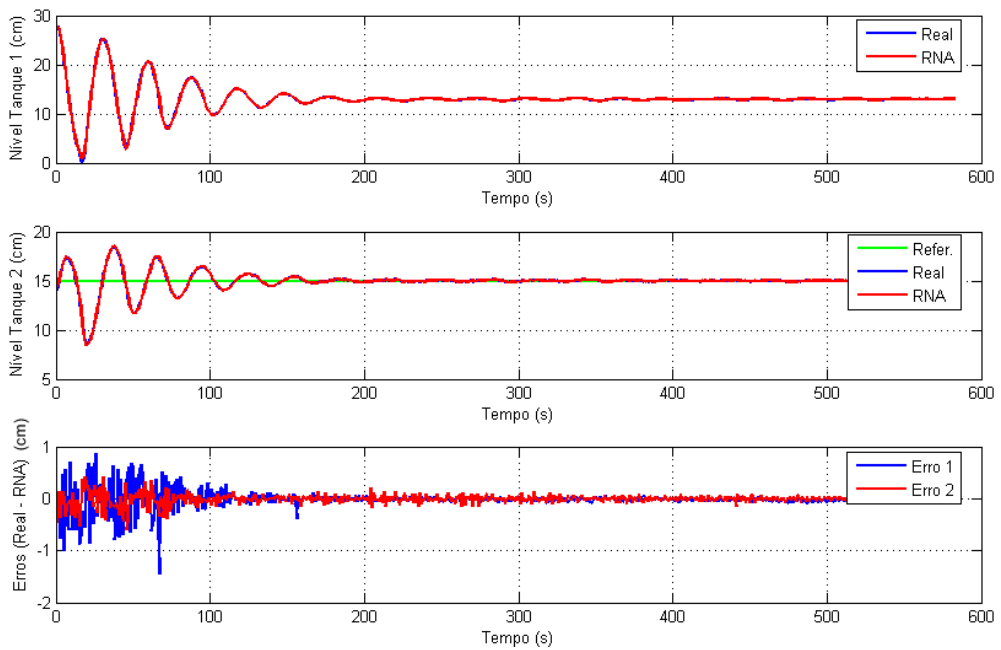


Figura 4.25: Exemplo 3 - Identificação em Dois Passos - Ambiente Real

4.3 Comentários - Identificação Neural

Como já foi citado anteriormente, a planta de níveis tem o nível do tanque 2 controlado por um controlador PID. Para a realização de todos os testes, sempre foram definidos sinais de referências para o nível do tanque 2 que variavam ao longo do tempo. Essas referências são utilizadas pelo controlador PID para gerar um sinal de controle (tensão em *Volts*) que é aplicado à bomba, fazendo com que a água caia no tanque 1 a partir de uma mangueira.

Observando os resultados gerados para a identificação do sistema de níveis, pode-se dizer que os resíduos gerados para os resultados da simulação são bem menores do que os gerados durante o funcionamento da planta real de níveis. Uma possível explicação para a diferença entre os resultados deve-se ao fato do modelo matemático do sistema de níveis ser uma simplificação do comportamento real, não modelando vários aspectos existentes no sistema real, como a dinâmica da bomba e a presença de ruídos.

Os três conjuntos de gráficos seguintes presentes nas Figuras 4.26, 4.27 e 4.28 permitem uma análise sobre o que é mais próximo do funcionamento real do sistema, se é o modelo matemático ou a RNA treinada a partir dos dados reais. Vale salientar que os mesmos ganhos para o controlador PID foram aplicados tanto ao controlador da planta real quanto ao do modelo matemático.

Para esta análise, foi considerada apenas a identificação neural em dois passos. O primeiro gráfico da Figura 4.26 apresenta o nível do tanque 1 do sistema real (em azul), o mesmo gerado pela RNA 1 de identificação (em vermelho) e o nível gerado pelo modelo matemático considerando a mesma referência. O segundo gráfico mostra os erros entre o nível real e o gerado pelo modelo matemático (em azul) e o nível real e o gerado pela RNA 1 de identificação (em vermelho).

Já na Figura 4.27, o primeiro gráfico apresenta o nível do tanque 2 do sistema real (em azul), o mesmo gerado pela RNA 2 de identificação (em vermelho) e o nível gerado pelo modelo matemático considerando a mesma referência. Da mesma forma que anteriormente, o segundo gráfico mostra os erros entre o nível real e o gerado pelo modelo matemático (em azul) e o nível real e o gerado pela RNA 2 de identificação (em vermelho).

Continuando esta análise, a Figura 4.28 exhibe os sinais de controle gerados pelo funcionamento real da planta (primeiro gráfico, em azul) e pelo modelo matemático (primeiro gráfico, em vermelho), bem como a diferença entre eles (segundo gráfico).

A partir desses resultados, pode-se concluir que os dados gerados pela identificação neural são bem mais próximos da realidade do que os gerados pelo modelo matemático

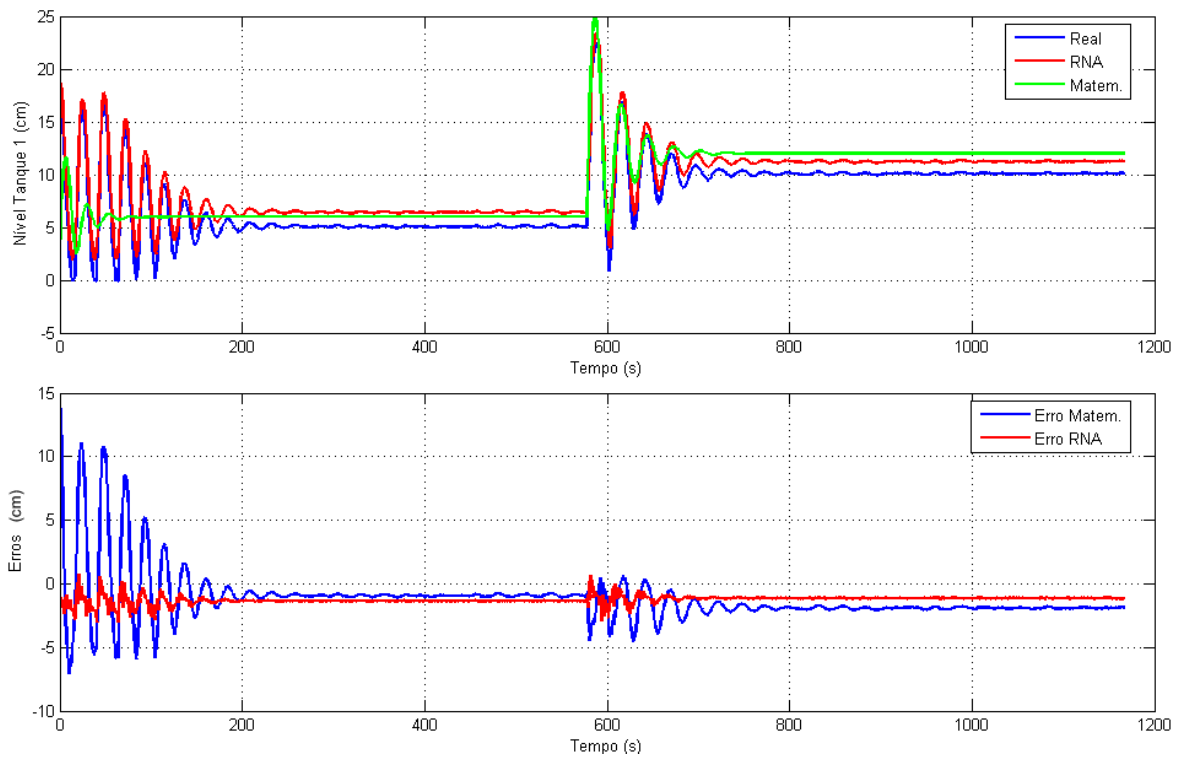


Figura 4.26: Diferenças do Nível do Tanque 1: Modelo Matemático, Real e Gerado pela RNA

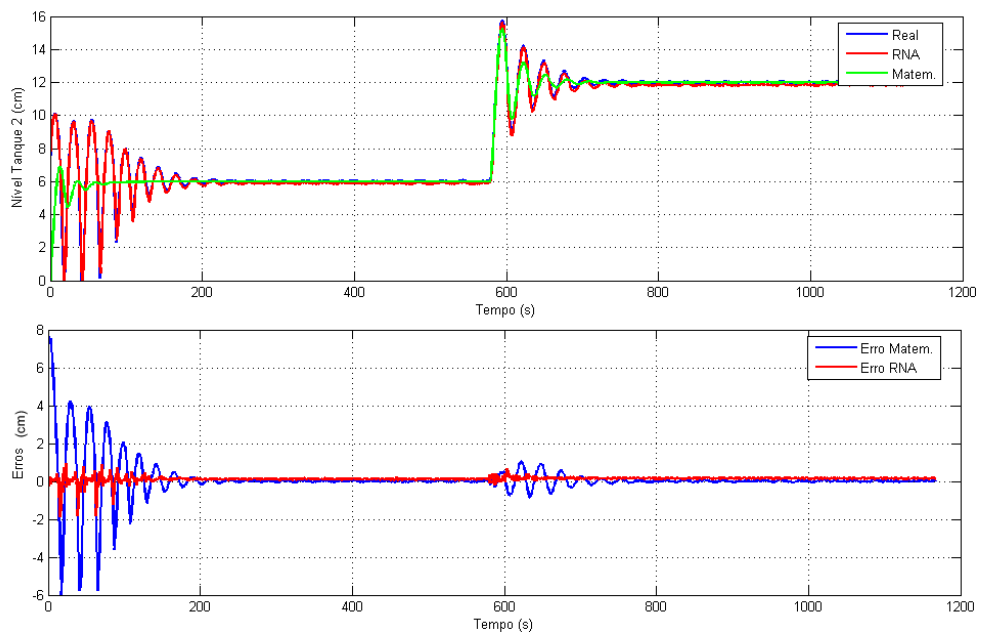


Figura 4.27: Diferenças do Nível do Tanque 2: Modelo Matemático, Real e Gerado pela RNA

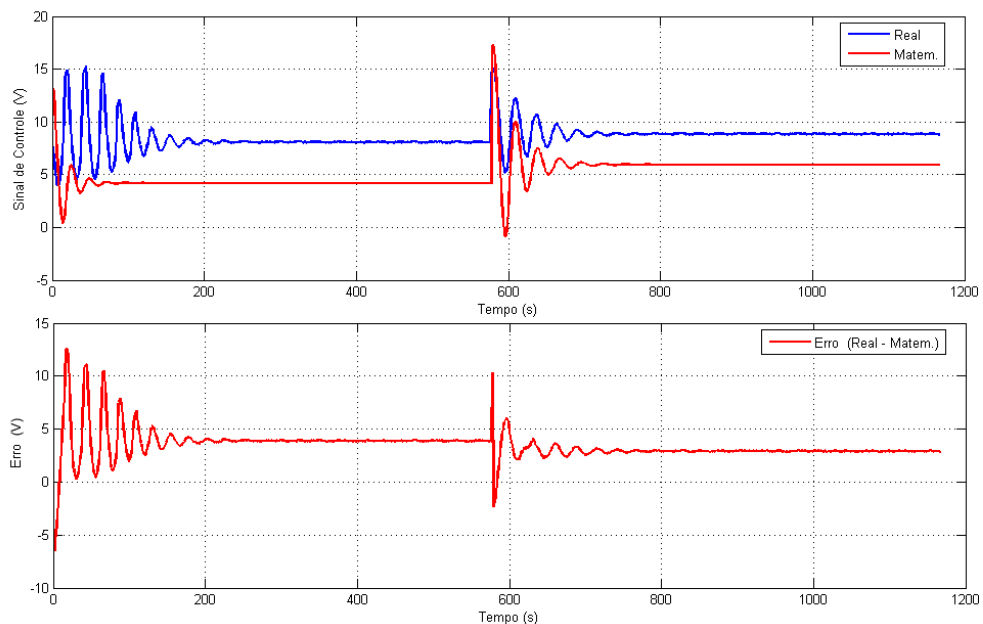


Figura 4.28: Diferenças do Sinal de Controle: Modelo Matemático e Real

disponível no manual da planta de níveis. Observa-se também que com o modelo matemático o nível do tanque 2 estabiliza mais rapidamente, provavelmente devido ao fato das equações matemáticas modelarem uma situação ideal, sem considerar ruídos e sem considerar a dinâmica da bomba (o que acontece no caso real).

Com base nos resultados demonstrados da identificação neural do modelo matemático e do sistema real, pode-se inferir que, para ambas situações, os erros entre os níveis dos tanques e os gerados nas identificações são maiores durante o período transitório do nível do tanque 2. Esses erros são maiores considerando a identificação neural no caso real, já que muitos ruídos estão presentes durante os testes e a bomba possui uma dinâmica bem mais não-linear do que no caso modelado matematicamente.

No próximo capítulo resultados do isolamento de falhas no sistema de níveis serão apresentados.

Capítulo 5

Resultados da Detecção e Isolamento de Falhas

Neste capítulo serão apresentados resultados do sistema DIF aplicado à planta de níveis considerando o modelo matemático e o seu funcionamento real.

Como apresentado no Capítulo 3, a detecção e classificação das falhas são feitas por uma rede neural, a RNA 3. Esta rede foi treinada apenas utilizando os resultados de identificação neural utilizando o modelo matemático do sistema. De posse do modelo matemático, falhas foram simuladas e os resíduos R_1 e R_2 captados, permitindo a geração de dados que significassem funcionamento normal do sistema ou ocorrência de falha. Essa abordagem foi escolhida por não haver uma base de dados armazenada ao longo do tempo com informações de falhas ocorridas no sistema de níveis. Dessa forma, todas as 13 falhas listadas anteriormente puderam ser simuladas com a utilização do modelo matemático. A partir delas, foi possível extrair os dados para treinamento do classificador.

Na hora de utilizar o classificador de falhas considerando o sistema real, apenas algumas falhas puderam ser simuladas e possíveis de serem testadas. Outro fato importante a ser citado, é que, atualmente, o classificador funciona apenas para detectar e isolar falhas quando o sistema de níveis estiver em regime (nível do tanque 2 estável). Isto apenas para o caso real, pois o classificador neural foi treinado com base na identificação do sistema simulado a partir do modelo matemático. Como as RNAs que identificam o modelo matemático geram resíduos muito próximos a zero também em situações de transitório, não correspondem ao comportamento real do sistema nas mesmas situações, como pôde ser visto na seção 4.3 do Capítulo 4.

5.1 Falhas que Podem Ocorrer no Sistema de Níveis

No Capítulo 3, foram mostrados 13 tipos de falhas (ver Tabela 3.2) como possíveis de ocorrer no sistema de níveis. Essas falhas tiveram classificações para valores dos resíduos R_1 e R_2 (relacionados aos níveis do tanque 1 e tanque 2, respectivamente) gerados pela identificação neural do sistema em dois passos, considerando apenas o funcionamento do sistema a partir da simulação matemática do mesmo.

Para simular as falhas, realizou-se uma modificação nas equações matemáticas (Equações 3.1 e 3.2) que simulam o comportamento do sistema de níveis. Para facilitar a compreensão, as mesmas estão novamente apresentadas nas equações 5.1 e 5.2 abaixo.

$$\dot{L}_1 = -\frac{a_1}{A_1} \sqrt{2gL_1} + \frac{K_m}{A_1} V_p \quad (5.1)$$

$$\dot{L}_2 = -\frac{a_2}{A_2} \sqrt{2gL_2} + \frac{a_1}{A_2} \sqrt{2gL_1} \quad (5.2)$$

As modificações nas equações originais são feitas a partir da adição de novas variáveis, que podem simular: alargamento ou decréscimo dos orifícios dos tanques, orifício extra no tanque 1 lançando a água para fora do tanque 2, aumento ou diminuição do sinal de controle e erro de leitura nos sensores.

As equações modificadas utilizadas para a simulação das falhas estão apresentadas nas equações 5.3 e 5.4. Observa-se a presença de variáveis extras que se assumirem o valor zero (ou 1 no caso da variável V_{p1}) fazem com que as equações representem o funcionamento normal do tanque, ou seja, sem a presença de falhas.

$$\dot{L}_1 = -\frac{(a_1 + a_{11} + a_{12})}{A_1} \sqrt{2gL_1} + \frac{K_m}{A_1} V_p V_{p1} + bias1 \quad (5.3)$$

$$\dot{L}_2 = -\frac{(a_2 + a_{21})}{A_2} \sqrt{2gL_2} + \frac{(a_1 + a_{11})}{A_2} \sqrt{2gL_1} + bias2 \quad (5.4)$$

A Tabela 5.1 apresenta uma descrição do que as variáveis incluídas nas equações do sistema de níveis podem simular.

Tabela 5.1: Simulação das Falhas: Novas Variáveis Incluídas Nas Equações Matemáticas

Variável	Descrição
a_{11}	Utilizada para simular um alargamento (valores positivos) ou diminuição (valores negativos) no orifício do tanque 1.
a_{12}	Utilizada para simular um novo orifício no tanque 1 por onde a água sai e não cai no tanque 2 (valores positivos).
a_{21}	Utilizada para simular um alargamento (valores positivos) ou diminuição (valores negativos) no orifício do tanque 2.
V_{p1}	Utilizada para simular ruídos no sinal de controle. Simulando um incremento (valores positivos) ou decréscimo (valores negativos) no mesmo.
$bias1$	Utilizada para simular ruídos na leitura feita pelo sensor que capta o valor atual do nível do tanque 1. Simulando um incremento (valores positivos) ou decréscimo (valores negativos) no mesmo.
$bias2$	Utilizada para simular ruídos na leitura feita pelo sensor que capta o valor atual do nível do tanque 2. Simulando um incremento (valores positivos) ou decréscimo (valores negativos) no mesmo.

Com base na manipulação dos valores dessas variáveis, foi possível simular as falhas. A seguir, apresenta-se para cada uma das 13 falhas apresentadas no Capítulo 3 como as mesmas foram simuladas:

1. Vazamento no tanque 1, mas água que vaza não cai no tanque 2.
Simulação: $a_{12} > 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.
2. Entupimento do orifício de saída do tanque 1.
Simulação: $a_{11} < 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.
3. Entupimento do orifício de saída do tanque 2.
Simulação: $a_{21} < 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.
4. Vazamento no tanque 1, com água que vaza caindo no tanque 2.
Simulação: $a_{11} > 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.
5. Vazamento no tanque 2.
Simulação: $a_{21} > 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.
6. Ruído positivo no sinal de controle.

Simulação: $V_{p1} > 1$, demais variáveis iguais a zero.

7. Ruído negativo no sinal de controle.

Simulação: $V_{p1} < 1$, demais variáveis extras iguais a zero.

8. Erro de leitura no sensor 1, a partir da introdução de um *bias* positivo.

Simulação: $bias1 > 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.

9. Erro de leitura no sensor 1, a partir da introdução de um *bias* negativo.

Simulação: $bias1 < 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.

10. Erro de leitura no sensor 2, a partir da introdução de um *bias* positivo.

Simulação: $bias2 > 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.

11. Erro de leitura no sensor 2, a partir da introdução de um *bias* negativo.

Simulação: $bias2 < 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.

12. Fonte de água entrando no tanque 1 sem ser a partir da mangueira que joga água a partir da bomba, aumentando a quantidade de água no mesmo.

Simulação: $bias1 > 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.

13. Fonte de água entrando no tanque 2 sem ser a partir do orifício de entrada de água do tanque 1, aumentando a quantidade de água no mesmo.

Simulação: $bias2 > 0$, $V_{p1} = 1$, demais variáveis extras iguais a zero.

Simuladas as falhas, extraiu-se os gráficos presentes na Figura 5.1 que apresentam os valores para os resíduos durante o funcionamento normal do sistema e durante a ocorrência das 13 falhas, permitindo a captação dos dados para treinamento da RNA 3. O eixo das abcissas corresponde ao instante atual em segundos e o eixo das ordenadas aos valores dos resíduos R_1 e R_2 naquele instante.

A partir dos gráficos com simulações das 13 falhas, é possível observar que nos instantes iniciais, onde o sistema está funcionando normalmente, os resíduos possuem valores bem próximos a zero. No momento em que as falhas ocorrem, instantaneamente eles alteram para valores distantes de zero. Outra observação é que os gráficos das falhas não foram ordenados pela classificação de seus tipos (Falha 1, Falha 2, Falha 3, etc.), mas sim,

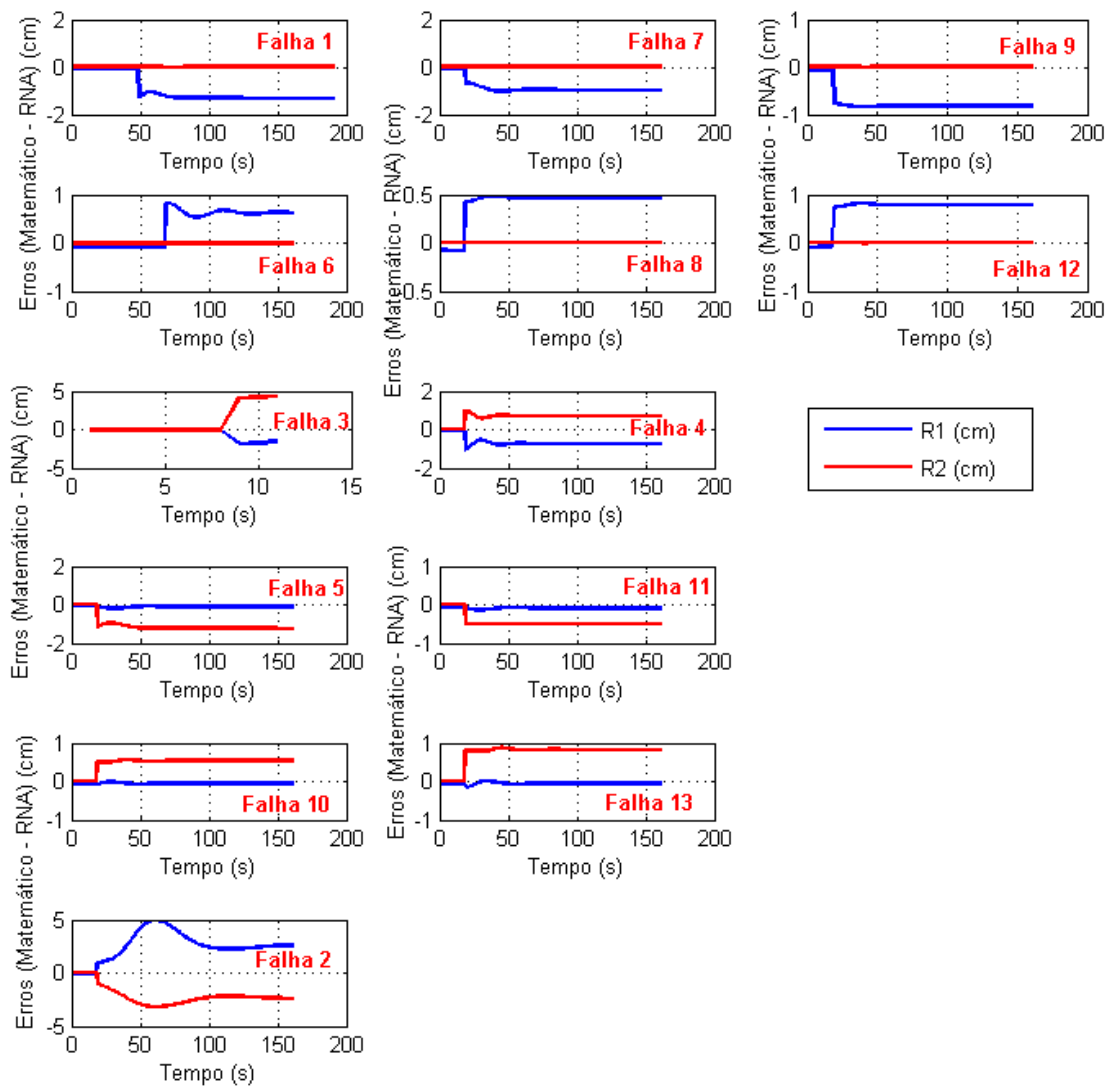


Figura 5.1: Simulação das 13 Falhas Seleccionadas

cada linha da Figura 5.1 apresenta gráficos de falhas com comportamentos de resíduos semelhantes durante a ocorrência da falha, por exemplo, na primeira linha, apresenta-se gráficos das falhas 1, 7 e 9 que geram valores para R_1 distantes de zero e negativos e valores para R_2 bem próximos a zero.

Com essas simulações, possibilitou-se a extração dos dados de treinamento para o classificador de falhas. Os resultados obtidos serão apresentados nas seções posteriores.

5.2 Resultados com Modelo Matemático - Simulação

Com a simulação das falhas utilizando o modelo matemático do sistema, captou-se os dados para treinamento da RNA 3 (classificador de falhas). Após o treinamento, o erro quadrático médio chegou em torno de 10^{-4} cm^2 , como pode ser visto a partir de sua evolução na Figura 5.2.

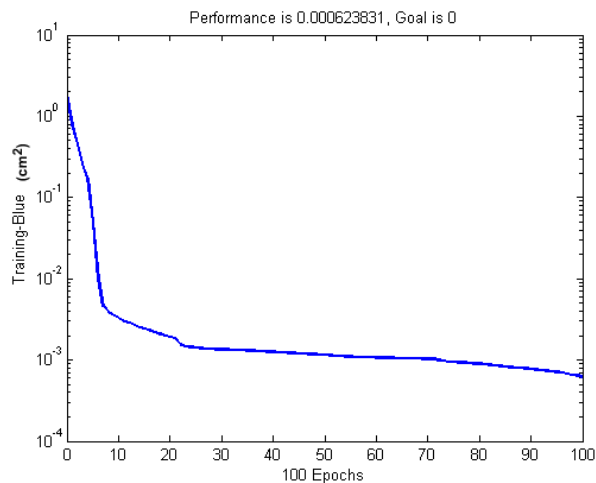


Figura 5.2: Evolução do Erro Quadrático Médio Durante o Treinamento da Rede para Classificação das Falhas

Como o objetivo final deste trabalho é fazer com que um sistema de detecção e isolamento de falhas execute em tempo real para um sistema de níveis, elaborou-se um supervisor de interface gráfica mostrado na 5.3. Neste supervisor, em tempo real, são exibidos um gráfico com os resíduos R_1 (em azul) e R_2 (em vermelho); os níveis atuais do tanque 1 (L_1) e do tanque 2 (L_2); os níveis estimados pelas RNA 1 (L'_1) e RNA 2 (L'_2); e os valores instantâneos de R_1 e R_2 . Os 13 tipos de falhas estão representados por botões em cor branca no supervisor, ocorrendo uma delas, o botão correspondente à falha passa a ter a cor vermelha e uma mensagem correspondendo à consequência da falha é exibida. Durante o funcionamento normal do sistema, todos os botões de falhas ficam em branco e é exibida a mensagem “Funcionamento Normal”.

As mensagens exibidas devido à ocorrência de alguma falha estão exibidas na Tabela 5.2.

As Figuras seguintes (5.4 - 5.9) apresentam situações em que ocorreram falhas no sistema simulado e as mesmas foram detectadas. Para fins de demonstração, apenas as falhas de 1 a 6 foram simuladas.

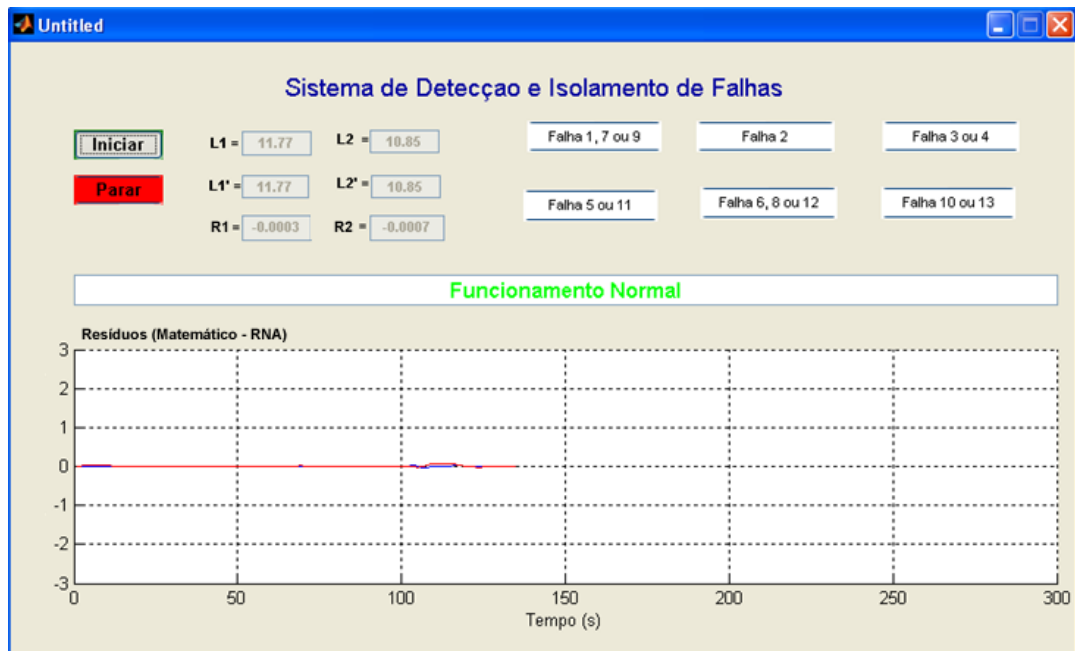


Figura 5.3: Interface Gráfica do Supervisório

5.3 Resultados Reais

Como já citado, o classificador neural de falhas foi treinado apenas considerando os dados residuais do modelo matemático. Nesta seção, serão apresentados resultados de classificação de falhas provocadas no sistema real de níveis.

O sistema DIF, em tempo real, acompanha o funcionamento do sistema, indicando funcionamento normal ou ocorrência de falhas. Nem todas as falhas puderam ser provocadas no sistema real. Os gráficos das Figuras 5.10 a 5.17 das falhas que foram possíveis de ser realizadas no sistema real. No momento em que são detectadas, o sistema ativa o botão da falha correspondente, mudando sua cor para vermelho e exibe uma mensagem informando o que está ocorrendo no sistema com a presença da falha.

Por motivos de segurança dos equipamentos envolvidos no sistema de níveis, os gráficos captados e apresentados mostram o instante em que a falha ocorreu e foi detectada. Dessa forma, evitando que os tanques transbordassem ou que a bomba fosse forçada por variações indevidas de tensões.

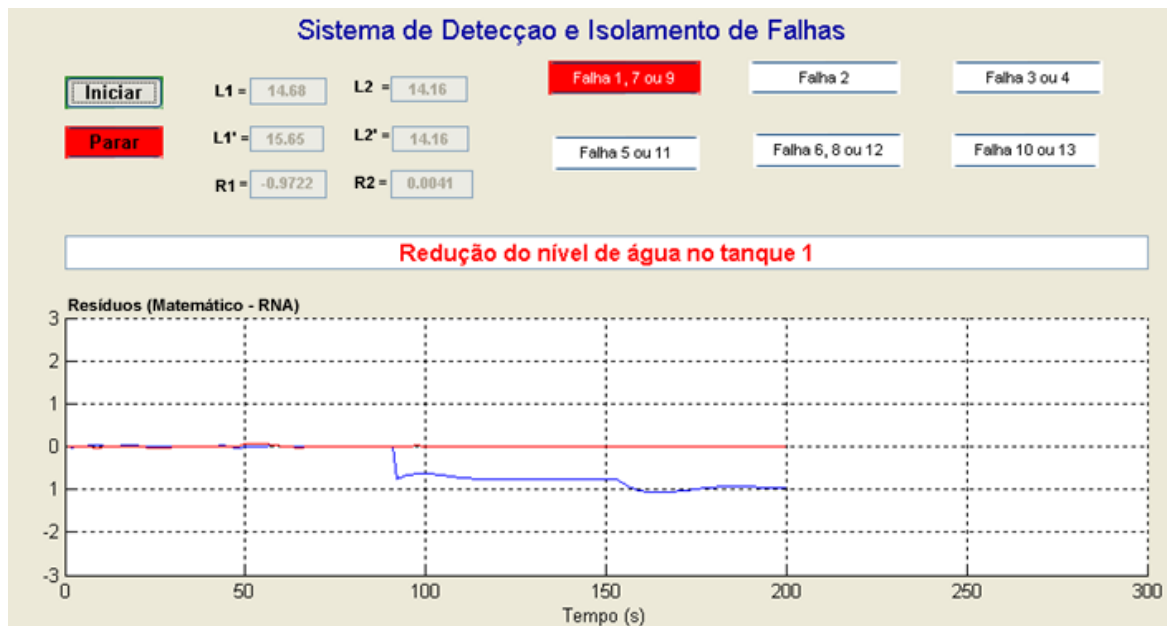


Figura 5.4: Falha 1: Novo Orifício no Tanque 1. Água Não Cai no Tanque 2 - Sistema Simulado

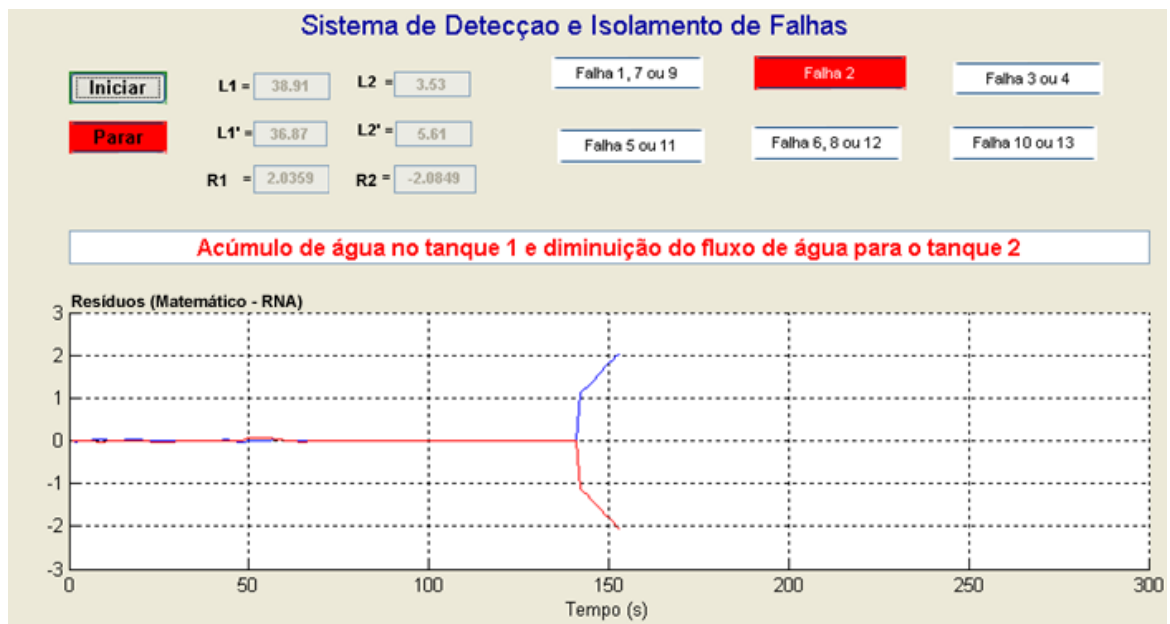


Figura 5.5: Falha 2: Diminuição do Orifício do Tanque 1 - Sistema Simulado

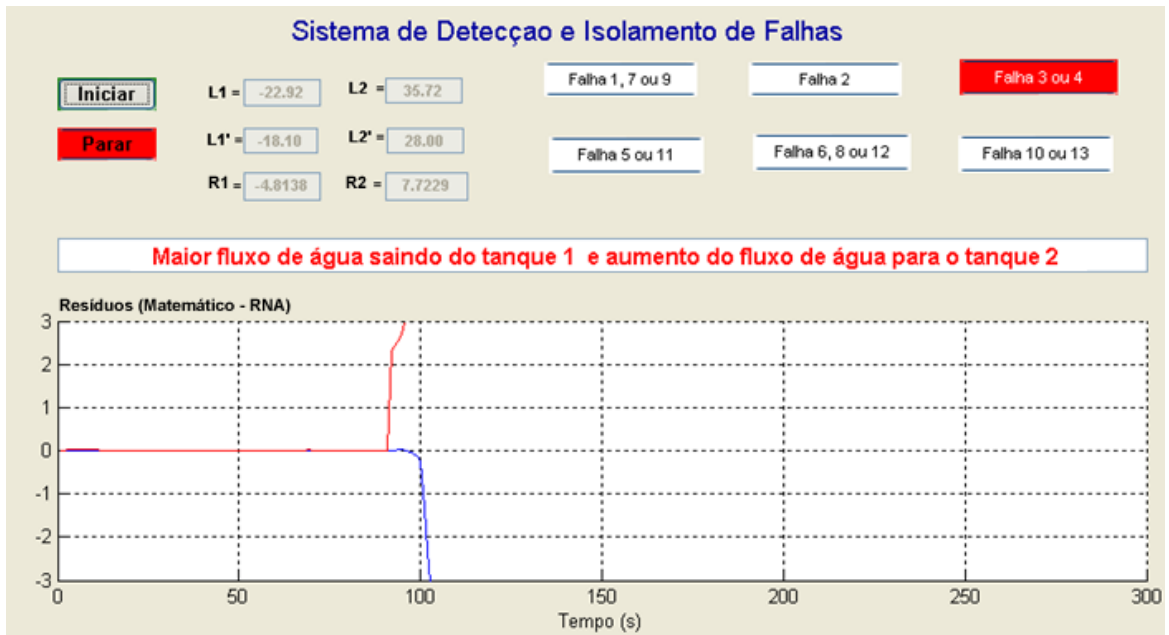


Figura 5.6: Falha 3: Diminuição do Orifício do Tanque 2 - Sistema Simulado

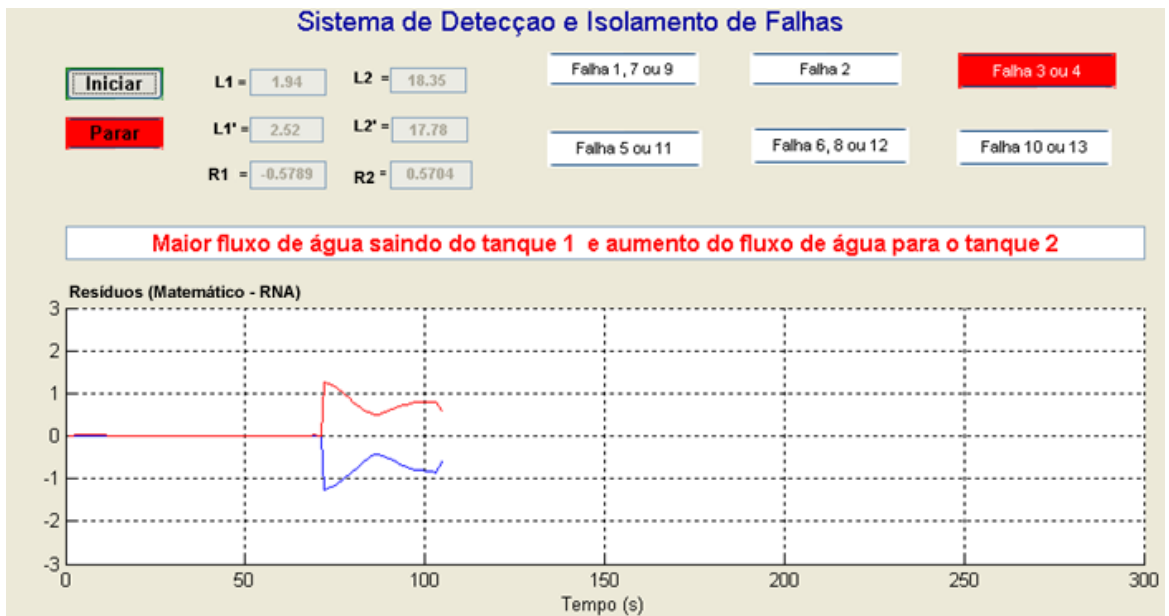


Figura 5.7: Falha 4: Aumento do Orifício do Tanque 1 - Sistema Simulado

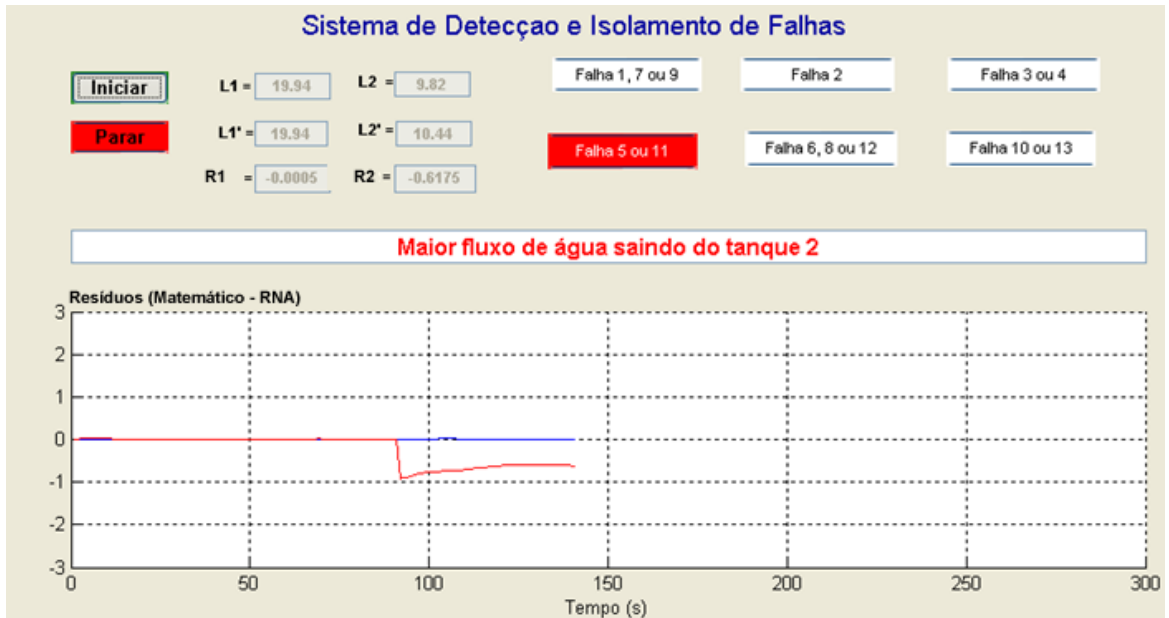


Figura 5.8: Falha 5: Aumento do Orifício do Tanque 2 - Sistema Simulado

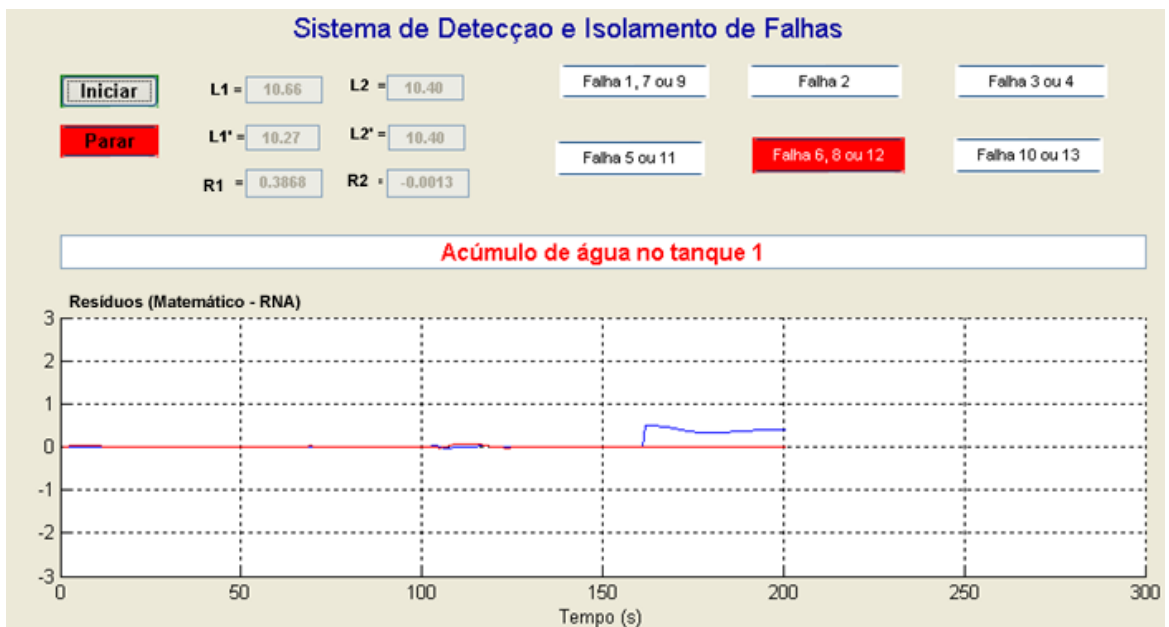


Figura 5.9: Falha 6: Aumento do Sinal de Controle - Sistema Simulado

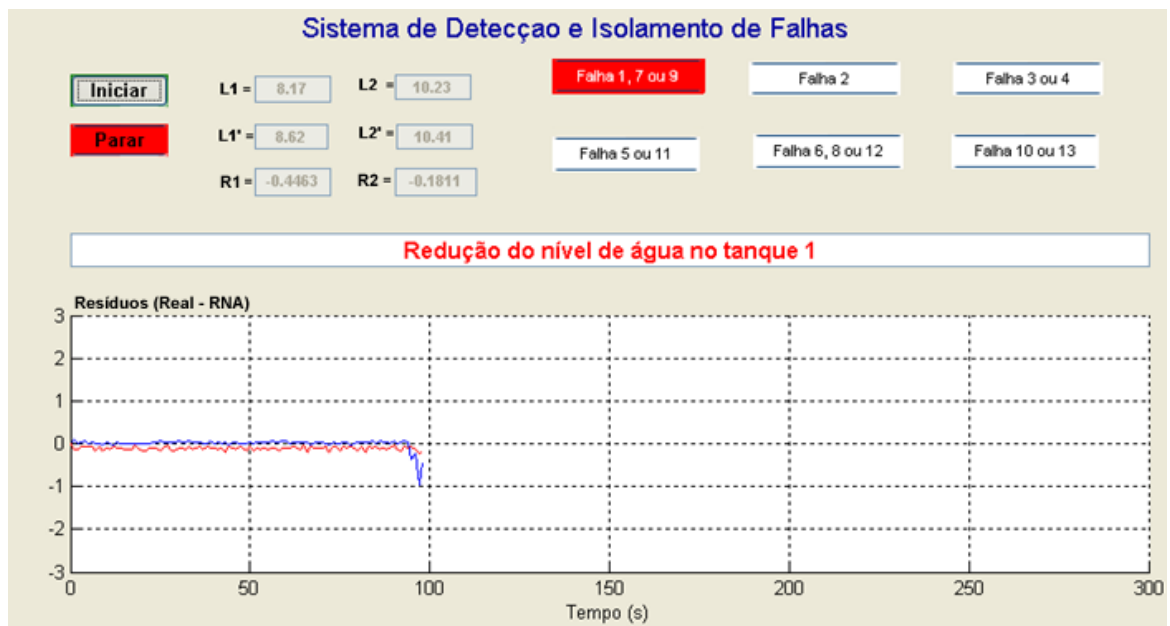


Figura 5.10: Falha 1: Novo Orifício no Tanque 1. Água Não Cai no Tanque 2 - Sistema Real

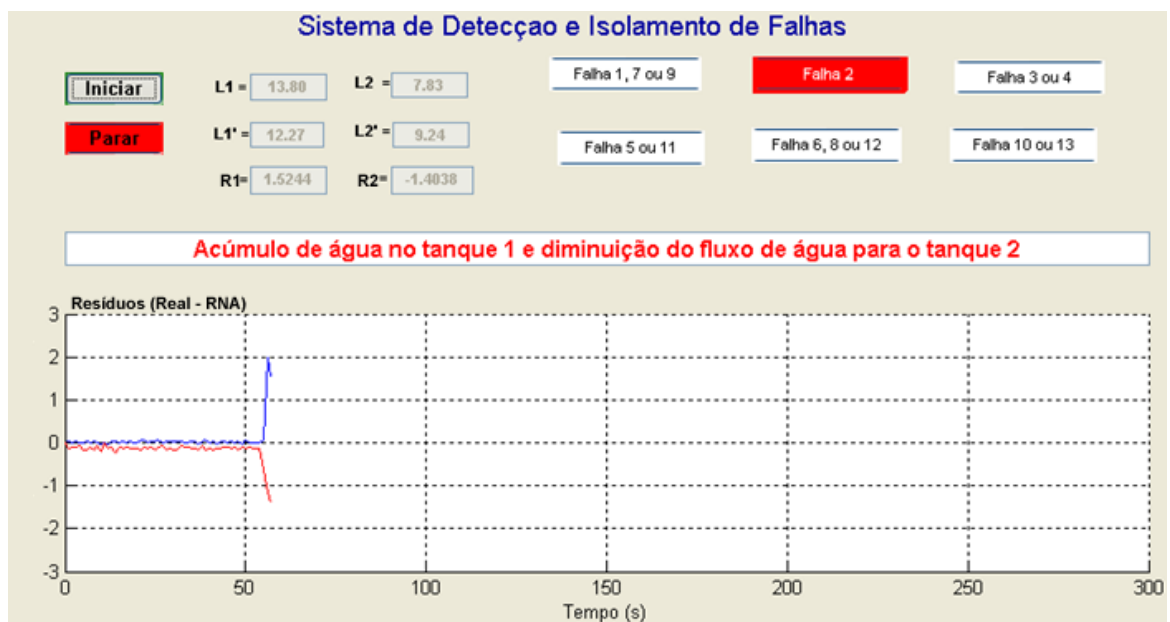


Figura 5.11: Falha 2: Diminuição do Orifício do Tanque 1 - Sistema Real

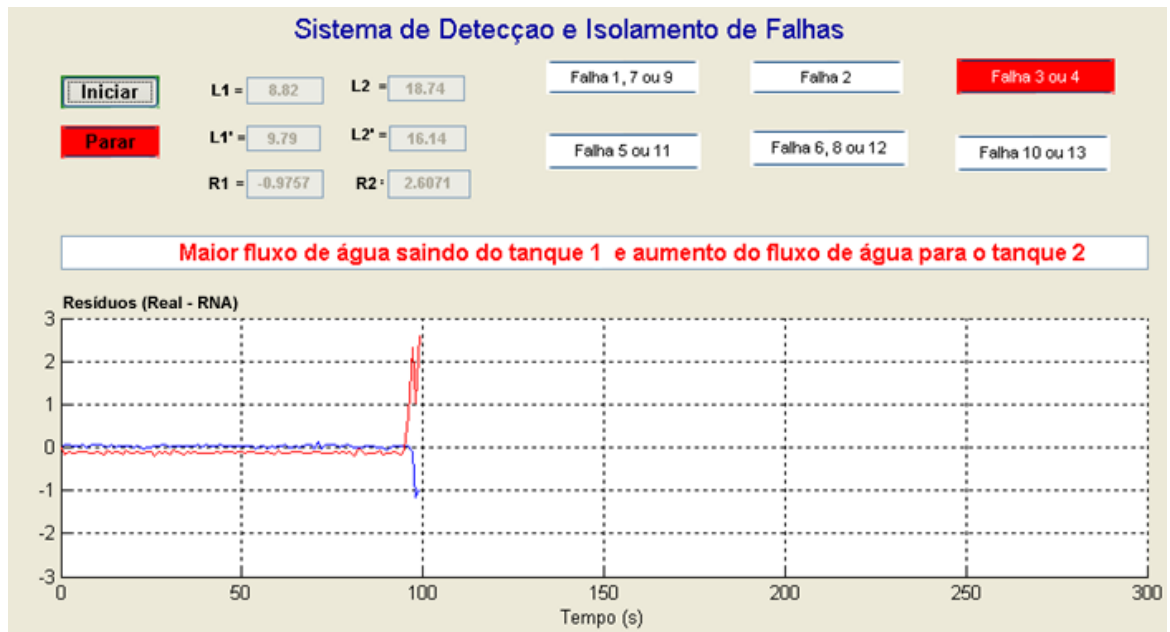


Figura 5.12: Falha 3: Diminuição do Orifício do Tanque 2 - Sistema Real

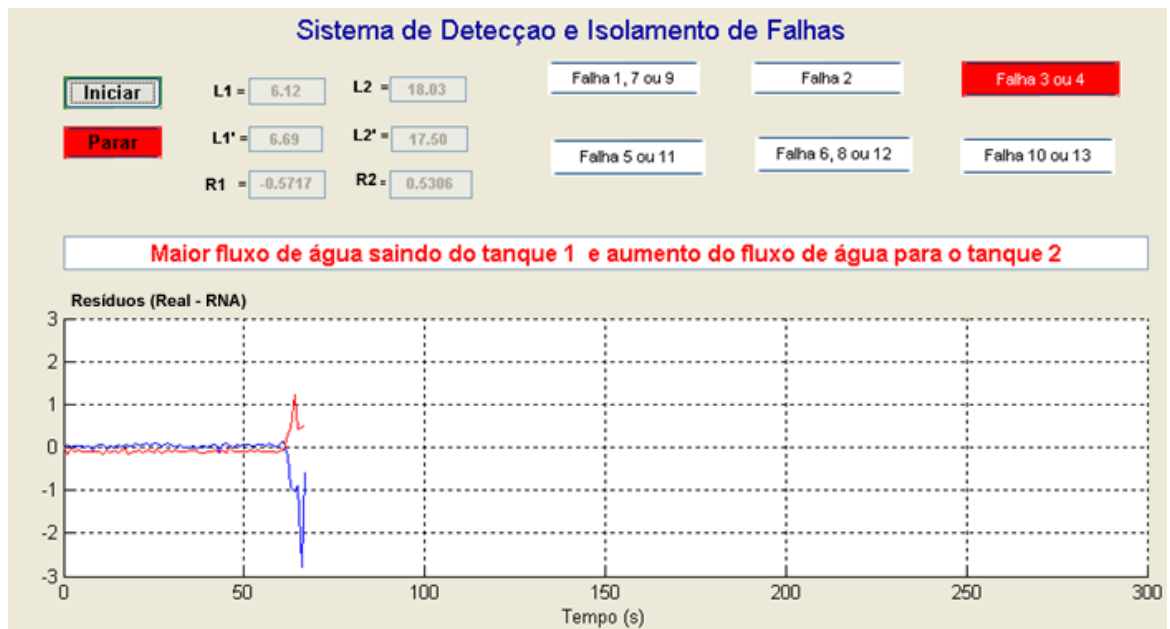


Figura 5.13: Falha 4: Aumento do Orifício do Tanque 1 - Sistema Real

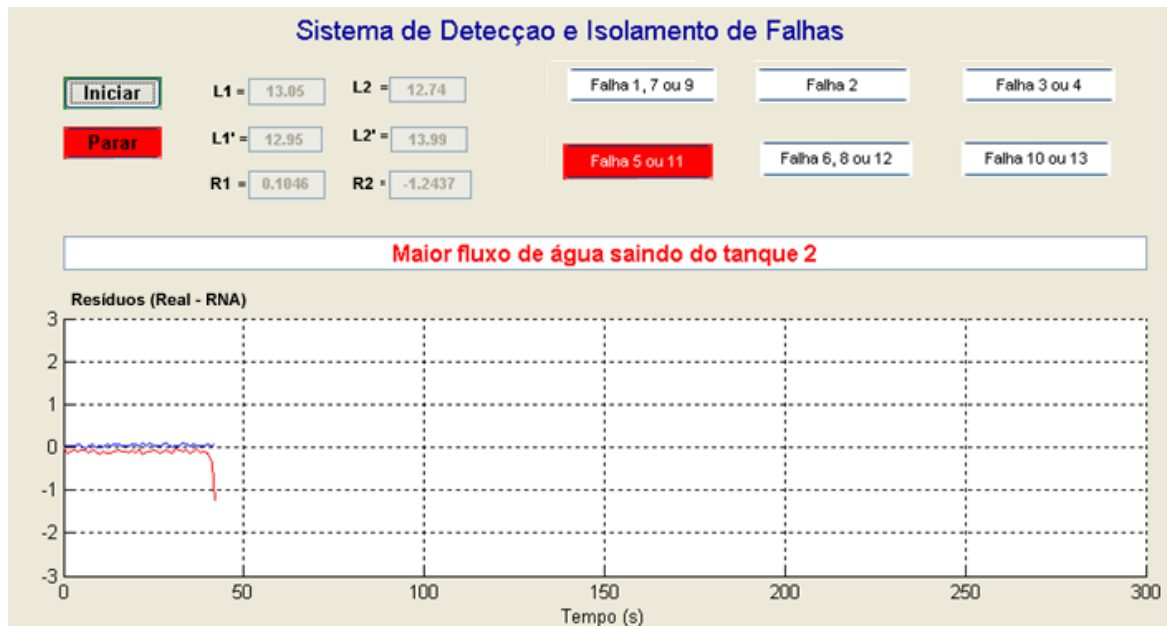


Figura 5.14: Falha 5: Aumento do Orifício do Tanque 2 - Sistema Real

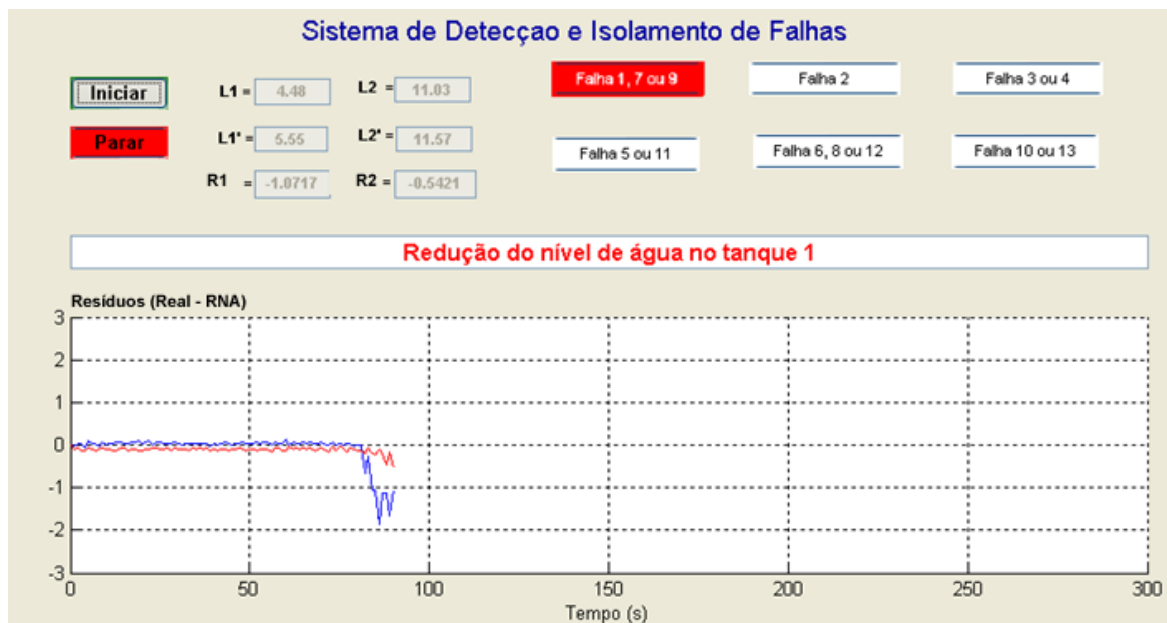


Figura 5.15: Falha 7: Redução do Sinal de Controle - Sistema Real

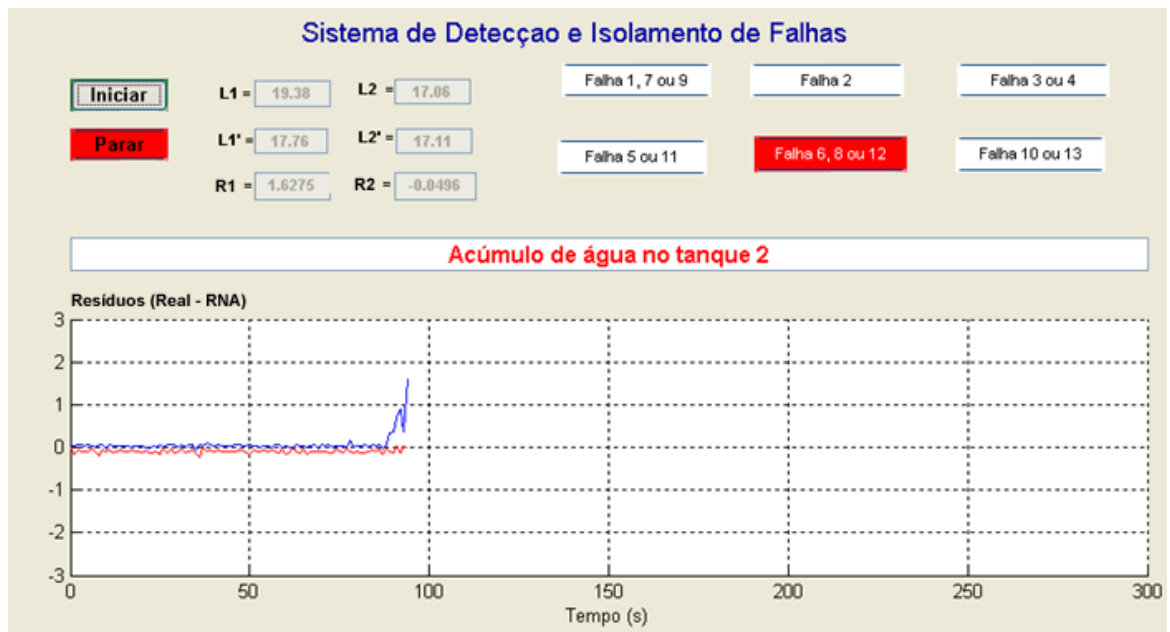


Figura 5.16: Falha 12: Fluxo Extra de Água no Tanque 1 - Sistema Real

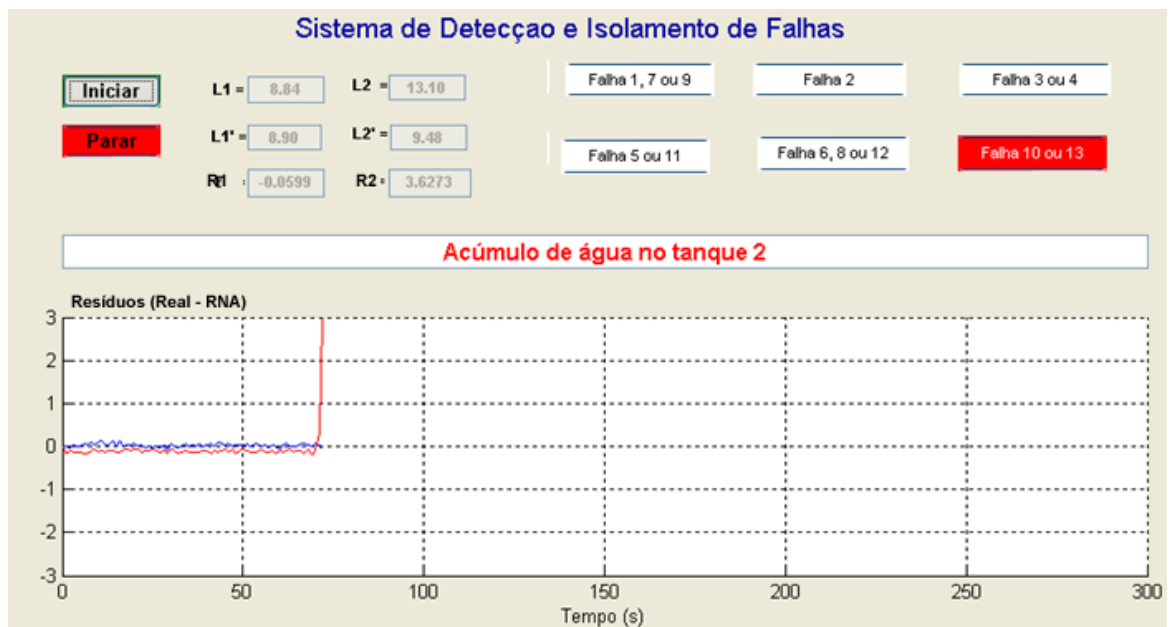


Figura 5.17: Falha 13: Fluxo Extra de Água no Tanque 2 - Sistema Real

Tabela 5.2: Mensagens Exibidas Após a Ocorrência de Falhas

Falhas	Mensagem
	Funcionamento Normal
1, 7 ou 9	Redução do nível de água no tanque 1
2	Acúmulo de água no tanque 1 e diminuição do fluxo de água para o tanque 2
3 ou 4	Maior fluxo de água saindo do tanque 1 e aumento do fluxo de água para o tanque 2
5 ou 11	Maior fluxo de água saindo do tanque 2
6, 8 ou 12	Acúmulo de água no tanque 1
10 ou 13	Acúmulo de água no tanque 2

5.4 Comentários Finais

A estratégia do sistema DIF foi implementada tanto no ambiente simulado quanto no real. A partir dos testes realizados e resultados obtidos, pode-se identificar claramente a diferença em se trabalhar apenas com dados simulados ao invés de dados reais. No caso real, muitas necessidades e dificuldades podem ser e foram encontradas, entre elas:

- Os testes deveriam ser feitos considerando que a água dos tanques não transbordasse;
- Necessidade do uso de equipamentos extras (fonte, módulo amplificador, etc.);
- Ocorrência de vazamentos nas conexões entre as mangueiras e os medidores de pressão;
- Dificuldades na identificação neural em momentos transitórios devido à grande não-linearidade da bomba (modelada matematicamente de forma ideal);
- Necessidades de circuitos extras para conversão do sinal aplicado à bomba;
- Sensibilidade a movimentos das mangueiras por parte dos sensores, causando erros de leitura;
- O módulo que interliga os equipamentos da rede industrial à rede Ethernet (módulo DFI) apresentou defeito e teve que ser trocado.

Outra diferença diz respeito ao fato do classificador de falhas apenas ser treinado com dados da identificação neural do modelo matemático. Dessa forma, ele não classifica muito bem falhas no transitório do sistema, já que na simulação os valores de R_1 e R_2 , nesta situação, são bem próximos à zero, o que não acontece no caso real.

Uma sugestão para detectar e classificar falhas no caso real, usando o classificador do sistema DIF treinado com dados simulados, seria fazer ajustes na equação da modelagem

matemática, de forma que o comportamento do sistema se aproximasse mais do real, inclusive em situações de transitório. Assim, resíduos em falhas mais próximos da situação real poderiam ser gerados e utilizados no treinamento da RNA 3.

Talvez um melhor ajuste dos parâmetros do controlador PID aplicado à planta real, de forma que reduzisse o período do transitório do sistema, pudesse melhorar a sua identificação neural nestes períodos.

Além disso, o classificador das falhas apenas detecta falhas pertencentes a conjuntos de resíduos (R_1 e R_2) distintos. Por exemplo, ele não consegue distinguir as falhas 1, 7 e 9 uma das outras, já que ambas possuem, durante a ocorrência da falha, valores de R_1 distantes de zero e negativos e valores de R_2 bem próximos a zero. Tentou-se incluir informações extras como entrada do classificador, como por exemplo, sinal de referência ou sinal de controle, porém não foram obtidos resultados satisfatórios para a distinção de falhas com resíduos semelhantes.

Capítulo 6

Conclusões e Perspectivas

Neste trabalho apresentou-se o desenvolvimento de um sistema de detecção e isolamento de falhas em um sistema de níveis real ligado à uma rede industrial Foundation Fieldbus. Este sistema foi dividido em duas partes: identificação neural e classificação/isolamento de falhas.

Além do desenvolvimento de um sistema aplicado ao problema de detecção e isolamento de falhas em um sistema dinâmico, o trabalho proposto e realizado objetivou aplicar o sistema DIF a situações simuladas e reais. Dessa forma, podendo estudar sua viabilidade prática quando aplicada a situações reais.

Duas abordagens para identificação neural foram apresentadas e uma delas escolhida (Identificação em Dois Passos) para fazer parte do Sistema de Detecção e Isolamento de Falhas. O sistema DIF, quando utilizado em um sistema real, mostrou-se capaz de acusar as falhas no momento em que as anormalidades ocorrem, superando todas as dificuldades não encontradas em ambientes simulados.

Resultados obtidos, considerando o modelo matemático do sistema e o seu funcionamento real, foram mostrados, apresentando algumas dificuldades em se trabalhar com sistemas reais e que não estão presentes em simulações. Algumas destas dificuldades foram: cuidado para que a água presente nos tanques não transbordasse, danificação de equipamentos físicos, necessidade da utilização de equipamentos extras, entre outros.

Também foi realizada uma comparação entre as saídas geradas pela identificação neural do sistema de níveis e as geradas pelo modelo matemático (desenvolvido com base nas leis da física), considerando as mesmas condições. Verificou-se que a abordagem neural apresentou resultados mais satisfatórios, devendo-se ao fato do modelo matemático apresentar uma situação ideal do funcionamento do sistema, o que não ocorre na realidade, onde ruídos e outras não linearidades não modeladas estão presentes.

Como perspectivas, pretendem-se melhorar o identificador neural para situações em que o controlador PID ainda não tenha estabilizado o nível do tanque 2 (transitório). Além

disso, pode-se ajustar o modelo matemático para que ele gere um comportamento mais próximo possível da realidade, de forma que o detector e classificador de falhas possa, a partir desses dados, funcionar adequadamente para situações de regime e transitório quando aplicado ao sistema real.

Outra sugestão, seria incrementar o classificador neural para que ele seja capaz de detectar e isolar falhas que possuam comportamentos semelhantes para os resíduos R_1 e R_2 . Então, provavelmente deva-se incluir informações adicionais como entrada da RNA 3 e usar outras ferramentas para analisar dados capazes de diferenciar as falhas de resíduos semelhantes.

O classificador também pode ter sua capacidade aumentada a fim de detectar e classificar falhas diferentes que ocorrem simultaneamente.

Com um módulo de detecção e isolamento de falhas robusto, o próximo passo seria o desenvolvimento de um módulo que realizasse a acomodação das falhas. Acomodar uma falha significa realizar ações para minimizar os seus efeitos. Por exemplo, no caso do sistema de níveis, a bomba poderia ser desligado no momento em que o sistema detectasse que um dos dois tanques estaria no limiar de transbordar.

Referências Bibliográficas

- Al-Taani, Ahmad T. (2005), An expert system for car failure diagnosis., *em* 'IEC (Prague)', pp. 457–560.
- Armengol, Joaquim, Josep Vehf, Miguel Àngel Sainz & Pau Herrero (2003), Fault detection in a pilot plant using interval models and multiple sliding time windows., *em* 'SAFEPROCESS'.
- Billings, S. A. (1980), 'Identification of nonlinear systems - a survey', *IEE Proceedings-D Control Theory and Applications* **127**(6), 272–285.
- Braga, A. P., A. C. P. L. F. Carvalho & T. B. Ludermir (2000), *Redes Neurais Artificiais: Teoria e Aplicações*, LTC.
- Coelho, Leandro S. & Antônio A. R. Coelho (1999), 'Algoritmos evolutivos em identificação e controle de processos: Uma visão integrada e perspectivas.', *SBA Controle e Automação* **10**(1).
- da Silva Lira, Milde Maria & Manoel Afonso de Carvalho Jr. (1999), Sistemas híbridos aplicados à filtragem de sinais de alarme de proteção de uma subestação telecomandada, *em* 'Proceedings of the IV Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais'.
- de Souza, João A. M. F., R. K. H. Galvão & T. Yoneyama (2000), A novel wavelet-based scheme for fault detection, *em* 'Proceedings of the UK Control Conference'.
- Ding, S. X., S. Schneider, E. L. Ding, A. Rehm & Robert Bosch GmbH (2005), Advanced model-based diagnosis of sensor faults in vehicle dynamics control systems, *em* '16th IFAC World Congress'.
- Evsukoff, A., J. Mountmain & S. Gentil (1997), Dynamic model and causal knowledge-based fault detection and isolation., *em* 'Proceeding of the IFAC Symposium of Fault Detection, Supervision and Safety for Technical Processes'.

- Fries, T.P. & J.H. Graham (2003), A fuzzy intelligent agent approach to fault diagnosis., *em* 'Proceeding (388) Intelligent Systems and Control'.
- Galvão, R. K. H. & T. Yoneyama (1998), A method for fault detection in control systems based on the wavelet transform, *em* 'Anais do XII Congresso Brasileiro de Automática', pp. 1741–1746.
- Gertler, J. (1998), A survey of model-based failure detection and isolation in complex plants., *em* 'IEEE Control Systems Magazine', Vol. 8, pp. 3–11.
- Haykin, Simon (2001), *Redes neurais: princípios e prática*, 2nd^a edição, Bookman.
- Haykin, Simon & Barry Van Veen (2001), *Sinais e Sistemas*, Bookman.
- Hebb, D. O. (1949), *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York.
- Hu, W., M. Schroeder & A.G. Starr (1999), A knowledge-based real-time diagnostic system for plc controlled manufacturing systems, *em* 'IEEE SMC '99 Conference Proceedings'.
- J., Patton R. (1994), Robust model based fault diagnosis: The state of the art, *em* 'Proc. of the IFAC Simp. On Fault Detection Supervision and Safety for Tech. Processes, Helsinki, Finland', pp. 1–24.
- Jones, J. G. & M. J. Corbin (1989), *Band-limiting filter approach to fault detection. Fault Diagnosis in Dynamics Systems - Theory and Application*, Prentice Hall.
- Koscielny, J. M., M. Syfert & M. Bartys (1999), Fuzzy-logic fault diagnosis of industrial process actuators., *em* 'Journal of Applied Mathematics and Computer Science'.
- Köppen-Seliger, B. & P. M. Frank (1995), Fault detection and isolation in technical processes with neural networks, *em* 'Proc. of the 34th Conference on Decision and Control'.
- Lima, Fábio, Luiz Affonso Guedes, Andres Ortiz & André Maitelli (2004), Hybrid environment for tests and training in fieldbuses, *em* 'VI Conferência Internacional de Aplicações Industriais - Induscon', pp. 1–6.
- Linarié, Davor & Vladimir Koroman (2003), Fault diagnosis of a hydraulic actuator using neural network, *em* 'IEEE, ICIT / Karel Jezernik'.

- Ljung, L. (1987), *System Identification: Theory for the user*, Prentice-Hall, Englewood Cliffs, NJ.
- Mallet, Yvette, Danny Coomans, Jerry Kautsky & Olivier De Vel (1997), Classification using adaptive wavelets for feature extraction, *em* 'IEEE Transactions on Pattern Analysis and Machine Intelligence', Vol. 19, pp. 1058–1066.
- McCulloch, W. S. & W. Pitts (1943), 'A logical calculus of the ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics* 5 pp. 115–153.
- Monsef, H., A. M. Ranjbar & S. Jadid (1997), Fuzzy rule-based expert system for power system fault diagnosis, *em* 'IEE Proceedings - Generation, Transmission and Distribution'.
- Naughton, J. M., Y. C. Chen & J. Jiang (1996), A neural networks application to fault diagnosis for robotic manipulator, *em* 'Ryerson Polytechnic University Toronto and University of Western Ontario London'.
- Nepomuceno, E.G., R. H. C. Takahashi, L. A. Aguirre, O. M. Neto & E. M. A. M. Mendes (2004), 'Multiobjective nonlinear system identification: a case study with thyristor controlled series capacitor (TCSC).', *International Journal of Systems Science* 35(9), 537–546.
- Nürnberger, Andreas, Arne Radetzky & Rudolf Kruse (2001), Using recurrent neuro-fuzzy techniques for the identification and simulation of dynamic systems, *em* 'Elsevier Science B.V. Neurocomputing 36', pp. 123–147.
- Ogata, Katsuhiko (1998), *Engenharia de Controle Moderno*, LTC.
- Paiva, Henrique Mohallem (2003), Detecção de falhas em sistemas dinâmicos empregando transformadas wavelets adaptativas, Dissertação de mestrado, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil.
- Palade, Vasile, Ron J. Patton, Faisal J. Uppal, Joseba Quevedo & S. Daley (2002), Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods, *em* 'Proc. of the 15th IFAC World Congress', pp. 2477–2482.
- Persin, S., B. Tovornik, N. Muskinja & D. Valh (2002), Increasing process safety using analytical redundancy, *em* 'Electrotechnical Review', pp. 240–246.
- Qua (n.d.), *Coupled Water Tank Experiments*.

- Rich, Elaine & Kevin Knight (1994), *Inteligência Artificial*, Makron Books.
- Rosenblatt, F. (1958), 'The perceptron: A probabilistic model for information storage and organization in the brain', *Psychological Review* 65 pp. 386–408.
- Shin, Hyun Joon, Dong-Hwan Eom & Sung-Shick Kim (2005), 'One-class support vector machines - an application in machine fault detection and classification', *Computer and Industrial Engineering* pp. 395–408.
- Simani, Silvio (2005), Identification and fault diagnosis of a simulated model of an industrial gas turbine, *em* 'IEEE Transactions on Industrial Informatics', Vol. 1, pp. 202–216.
- Srinivasan, Arvind & Celal Batur (1994), Hopfield/art-1 neural network-based fault detection and isolation, *em* 'IEEE Transactions on Neural Networks', Vol. 5.
- Tatara, Eric, Cenk Ündey, Bruce Williams, Gülnur Birol & Ali Çinar (2000), A real - time knowledge-based system for automated monitoring and fault diagnosis of batch processes, *em* 'AIChE'.
- Tinos, Renato & Marco H. Terra (1998), Fault detection and isolation in robotic manipulators and the radial basis function network trained by the kohonen's self-organizing map., *em* 'SBRN', pp. 85–90.
- Tinós, Renato (1999), Detecção e diagnóstico de falhas em robôs manipuladores via redes neurais artificiais, Dissertação de mestrado, Escola de Engenharia de São Carlos, Universidade de São Paulo.
- Uppal, Faisal J., Ron J. Patton & Marcin Witczak (2005), A neuro-fuzzy multiple-model observer approach to robust fault diagnosis based on the damadics benchmark problem, *em* 'Control Engineering Practice'.
- Weber, Taisy Silva (2002), Um roteiro para exploração dos conceitos básicos de tolerância a falhas, Relatório técnico, Instituto de Informática UFRGS.
- Witczak, Marcin (2005), 'Soft computing in fault detection and isolation'.
- Zhang, O. & Y. Yan (2001), A wavelet-based approach to abrupt fault detection and diagnosis of sensors, *em* 'IEEE Transactions on Instrumentation and Measurement', Vol. 50, pp. 1389–1396.

Zheng, Li & H. Nakagawa (2002), Opc (ole for process control) specification and its developments, *em* '41st SICE Annual Conference', Vol. 2, pp. 917–920.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)