



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E ENGENHARIA
DE PETRÓLEO



Estudo e Implementação de Algoritmos Inteligentes para Detecção e Classificação de Falhas na Medição de Gás Natural

Juliana Pegado de Medeiros

Orientador: Prof. Dr. Jorge Dantas de Melo

Co-orientador: Prof. Dr. Adrião Duarte Dória Neto

Natal, RN, junho de 2009

Livros Grátis

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

Milhares de livros grátis para download.



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E ENGENHARIA
DE PETRÓLEO



Estudo e Implementação de Algoritmos Inteligentes para Detecção e Classificação de Falhas na Medição de Gás Natural

Juliana Pegado de Medeiros

Orientador: Prof. Dr. Jorge Dantas de Melo

Co-orientador: Prof. Dr. Adrião Duarte Dória Neto

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência e Engenharia de Petróleo da UFRN (área de concentração: Automação na Indústria de Petróleo e Gás Natural) como parte dos requisitos para obtenção do título de Mestre em Ciência e Engenharia de Petróleo.

Natal, RN, junho de 2009

Estudo e Implementação de Algoritmos Inteligentes para Detecção e Classificação de Falhas na Medição de Gás Natural

Juliana Pegado de Medeiros

Dissertação de Mestrado aprovada em junho de 2009 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Jorge Dantas de Melo (orientador) DCA/UFRN

Prof. Dr. Adrião Duarte Dória Neto (co-orientador) DCA/UFRN

Prof. Dr. Luiz Affonso H. Guedes DCA/UFRN

Prof. Dr. Eduardo Oliveira Freire NEL/UFS

*Dedico este trabalho à minha avó,
Gercina Gentil de Medeiros.*

Agradecimentos

Aos meus orientadores Jorge Dantas de Melo e Adrião Duarte Dória Neto, sou grata pela orientação e pelo seu reconhecimento.

Aos meus pais, Rui e Fátima, a quem eu agradeço eternamente pelo carinho, apoio e educação.

À toda a minha família pelo apoio durante esta jornada.

Aos amigos do LAMP e DCA pelas sugestões e contribuições.

A todos os meus amigos, pelo incentivo para que eu seguisse em frente sem desaminar.

A Ronaldo Tavares do CTGÁS pelo fornecimento dos dados utilizados neste trabalho.

Resumo

Esta dissertação apresenta o estudo e implementação de algoritmos inteligentes para o monitoramento da medição de sensores envolvidos em processos de transferência de custódia de gás natural. Para a criação destes algoritmos são investigadas arquiteturas de Redes Neurais Artificiais devido a características particulares, tais como: aprendizado, adaptação e predição. Um preditor é implementado com a finalidade de reproduzir o comportamento dinâmico da saída de um sensor de interesse, de tal forma que sua saída seja comparada à saída real do sensor. Uma rede recorrente é utilizada para este fim, em virtude de sua capacidade em lidar com informação dinâmica. A saída real do sensor e a saída estimada do preditor formam a base para a criação das estratégias de detecção e identificação de possíveis falhas. Duas arquiteturas de redes neurais competitivas são investigadas e suas potencialidades são utilizadas para classificar tipos diferentes de falhas. O algoritmo de predição e as estratégias de detecção e classificação de falhas, bem como os resultados obtidos, serão apresentados.

Palavras-chave: Redes Neurais, sensores, detecção e classificação de falhas, predição.

Abstract

This master dissertation presents the study and implementation of intelligent algorithms to monitor the measurement of sensors involved in natural gas custody transfer processes. To create these algorithms Artificial Neural Networks are investigated because they have some particular properties, such as: learning, adaptation, prediction. A neural predictor is developed to reproduce the sensor output dynamic behavior, in such a way that its output is compared to the real sensor output. A recurrent neural network is used for this purpose, because of its ability to deal with dynamic information. The real sensor output and the estimated predictor output work as the basis for the creation of possible sensor fault detection and diagnosis strategies. Two competitive neural network architectures are investigated and their capabilities are used to classify different kinds of faults. The prediction algorithm and the fault detection classification strategies, as well as the obtained results, are presented.

Keywords: Neural Networks, sensor, fault detection and classification, prediction.

Sumário

Sumário	i
Lista de Figuras	ii
Lista de Tabelas	iii
Lista de Símbolos e Abreviaturas	iv
1 Introdução	1
1.1 Apresentação do problema	2
1.2 Objetivos	3
1.3 Trabalhos Relacionados	4
1.4 Publicação Científica	5
1.5 Organização do Texto	5
2 Redes Neurais Artificiais	7
2.1 Introdução	8
2.2 Perceptron de Múltiplas Camadas	9
2.2.1 Predição de Séries Temporais com Redes MLP	11
2.2.2 O Filtro de Kalman Estendido	15
2.3 Redes Neurais com Aprendizado Auto-Supervisionado	17
2.3.1 Classificação de Padrões com Aprendizado Auto-Supervisionado	17
2.3.2 Mapas Auto-Organizáveis	18
2.3.3 Teoria da Ressonância Adaptativa	21
3 Detecção e Diagnóstico de Falhas em Sensores	26
3.1 Introdução	27
3.2 Técnicas de detecção e diagnóstico de falhas	28
3.2.1 Redundância Física	28
3.2.2 Redundância Analítica	29
3.2.3 Análise de Sinais	30

3.2.4	Inteligência Artificial	30
4	Algoritmos Inteligentes	32
4.1	Introdução	33
4.2	Preditor Neural	33
4.3	Algoritmos para Classificação de Falhas	37
5	Resultados e Discussões	41
5.1	Algoritmo de Predição	42
5.2	Detecção e Classificação de Falhas	46
5.2.1	Detecção de Falhas pela Análise do Erro	50
5.2.2	Classificação de Falhas com Arquiteturas auto-organizadas	54
6	Considerações Finais e Perspectivas Futuras	58
	Referências bibliográficas	61

Lista de Figuras

1.1	Cadeia de transporte e de distribuição de gás natural no Brasil	6
2.1	Representação gráfica de um neurônio artificial	8
2.2	Perceptron de múltiplas camadas	10
2.3	Modelo de espaço de estados	11
2.4	Predição h passos à frente	12
2.5	Rede NARX	14
2.6	Grade de neurônios	19
2.7	Rede ART	22
3.1	Esquema de DDF hierárquico	29
4.1	Preditor construído a partir de uma rede recorrente	34
4.2	Treinamento de uma rede neural para predição k passos adiante	36
4.3	Predição de $\hat{y}(n)$	37
4.4	Esquema geral de diagnóstico de falhas	39
4.5	Arquitetura do mapa auto-organizável	40
5.1	Sinal de temperatura utilizado	42
5.2	Erro médio quadrático de treinamento	43
5.3	Sinal obtido pela predição de 1 passo	43
5.4	Erro médio quadrático de treinamento	44
5.5	Sinal obtido pela predição de 1 passo com a presença de ruído	45
5.6	Erro médio de teste X horizonte de predição	45
5.7	Sinal obtido em um horizonte de predição de 6 passos	47
5.8	Sinal obtido em um horizonte de predição de 5 passos com a presença de ruído	47
5.9	Comportamento do preditor de passo simples na presença de falhas	49
5.10	Comportamento do preditor de passos múltiplos ($h = 3$) na presença de falhas	50
5.11	Exemplo de sinal utilizado para teste	51

5.12	Tempo de detecção com preditor de passo simples ($T_{Janela} = 10$) . . .	52
5.13	Tempo de detecção com preditor de passos múltiplos $h = 2$ ($T_{Janela} = 10$)	53
5.14	Tempo de detecção com preditor de passos múltiplos $h = 3$ ($T_{Janela} = 10$)	53
5.15	Sinal de temperatura com falhas simuladas	55

Lista de Tabelas

5.1	Correlações	46
5.2	Percentual de Falhas Detectadas ($EPA > 6\%$, $T_Janela = 3$)	54
5.3	Percentual de alarmes falsos em relação ao conjunto de testes	54
5.4	Parâmetros da arquitetura SOM	56
5.5	<i>Clusters</i> formados pelo SOM	56
5.6	Parâmetros da arquitetura <i>Fuzzy ART</i>	56
5.7	<i>Clusters</i> formados pela rede <i>Fuzzy ART</i>	57
5.8	Percentual de Falhas Classificadas Corretamente ($T_Janela = 3$)	57

Lista de Símbolos e Abreviaturas

ART Adaptive Resonance Theory

DSP Digital Signal Processor

EPA Erro Percentual Absoluto

EPAM Erro Percentual Absoluto Médio

FF Foundation Fieldbus

FTDNN Focused Time Delay Neural Network

MLP MultiLayer Perceptron

NARX Nonlinear AutoRegressive model with eXogenous input

SOM Self-Organizing Map

CAPÍTULO 1

Introdução

1.1 Apresentação do problema

Desde a etapa de produção até sua distribuição para consumo, o gás natural é um produto constantemente monitorado através de instrumentos de medição. Estes instrumentos são utilizados para monitorar uma série de variáveis, dentre elas, vazão, temperatura e pressão.

O gás natural é usualmente distribuído através de gasodutos, tubulações que conduzem o produto introduzido sob pressão, através de compressores. Por força do fluxo, há uma perda de energia por atrito e a pressão cai gradativamente ao longo da tubulação, sendo necessárias estações de compressão (intercaladas ao longo do percurso) para elevar a pressão e permitir a continuidade do fluxo do produto.

Durante sua distribuição, o gás natural é movimentado sob a responsabilidade de empresas diferentes. Cada empresa recebe o gás natural e o repassa à próxima empresa, por meio de um processo denominado de *Transferência de Custódia*. A transferência de custódia é realizada em estações de controle e de medição específicas, localizadas em entroncamentos de gasodutos, ou em *citygates* onde a transportadora entrega o gás para a concessionária local (CTGÁS 2007).

A Figura 1.1 apresenta de forma esquemática a cadeia de transporte e de distribuição do gás natural no Brasil, mostrando o trajeto percorrido pelo gás natural da produção até o consumidor final. Em cada estação, mede-se a quantidade de gás natural, mais comumente medida em unidades de volume. O volume de gás natural que passa por uma estação é medido a partir de sua vazão e calculado nas condições de base de pressão e temperatura (pressão de 1 atm e temperatura de 20°C), através de um equipamento chamado computador de vazão.

Os instrumentos de medição são chamados de sensores ou medidores. Um sensor é um elemento transdutor que recebe como entrada uma grandeza física (vazão, pressão, temperatura, etc.) e fornece um sinal elétrico, analógico ou digital, com valor proporcional ao da grandeza medida.

É muito importante que as medições fornecidas por cada medidor sejam acuradas, pois, o custo do gás natural recebido estará de acordo com o volume fornecido. A ocorrência de erros de medição acarreta em cálculos de preços incompatíveis com os dados reais e, conseqüentemente, em perdas econômicas. Como a comercialização de gás natural é feita em grandes quantidades, pequenos erros de medição, ao serem acumulados, podem levar a prejuízos elevados.

Medidas de sensores estão sujeitas a uma série de incertezas que podem ser intrínsecas aos instrumentos, como descalibração e falhas mecânicas, elétricas ou eletrônicas ou

ocorrem devido a influências externas, como o ruído. O ideal é que os erros de medição estejam dentro de um limite aceitável e, caso este limite seja ultrapassado, devem ser tomadas providências para a correção destas medidas.

É muito importante que os estados indesejados nos sensores sejam detectados e corrigidos em um curto período de tempo, de forma que se evitem prejuízos para as empresas e para o consumidor do gás natural. Um mecanismo capaz de detectar e identificar falhas nestes sensores deve ter a capacidade de melhorar a confiabilidade dos sistemas de medição.

A atividade de detecção de falhas tem sido assunto em diversas pesquisas ao longo dos últimos anos. As técnicas mais convencionais são baseadas em redundância física ou no conhecimento de modelos matemáticos do processo em questão. A redundância física está associada ao aumento de custos e pode não ser economicamente viável. Já os métodos baseados em modelos matemáticos requerem um profundo conhecimento do processo que se está analisando (Yung 1993). Em estações de medição de gás natural, não há informações acerca do processo de escoamento do produto que permitam a formulação de um modelo matemático, ou seja, não há condições de se determinar de forma analítica os valores de vazão, temperatura e pressão do gás natural que chega à estação de medição e compará-los às medições dos sensores.

1.2 Objetivos

Este trabalho é resultado da investigação e implementação de arquiteturas de redes neurais cuja finalidade é desempenhar as atividades de detecção e classificação de falhas em sensores. Os algoritmos inteligentes apresentados neste trabalho têm como objetivo explorar de forma eficaz a informação agregada às medições de um sensor, superando a indisponibilidade de informações de todo o processo no qual o dispositivo está envolvido. Sabe-se que a inclusão de elementos de Inteligência Artificial em um sistema é capaz de melhorar seu desempenho, confiabilidade e robustez, pois eles têm condições de agregar um tipo de conhecimento que não pode ser resgatado nos modelos analíticos tradicionais. As Redes Neurais Artificiais têm esta característica, além da sua capacidade de aproximação de funções e interpolação, vantagens que as tornam bastante adequadas para realizar as tarefas apresentadas neste trabalho.

Em geral, as técnicas para detecção e diagnóstico de falhas baseadas em redes neurais artificiais são utilizadas para a reprodução do comportamento dinâmico do sistema, gerando um vetor de resíduos a partir do qual será possível informar a ocorrência ou não de falhas. Em outro caso pode-se usar uma rede neural como classificador, ou seja, para

identificar qual o tipo de falha que está ocorrendo. 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 (Fernandes 2007).

No trabalho desenvolvido, um algoritmo de predição, baseado em uma rede *Perceptron* de Múltiplas Camadas Recorrente é utilizado para reproduzir o comportamento da medição do sensor, através de medições adquiridas em instantes no tempo presente e passados. A arquitetura de rede recorrente é utilizada devido à sua capacidade de agregar conhecimento dinâmico das informações que lhe são apresentadas. Com estas informações é possível gerar um vetor de erros cujos valores indicam a presença ou ausência de falhas no sensor. Se o erro de predição, ou seja, se a diferença entre a saída do sensor e a saída do preditor estiver acima de um valor considerado aceitável, um alarme de falha deve ser gerado.

Para desempenhar a tarefa de classificação de falhas são investigadas arquiteturas de redes neurais não-supervisionadas ou auto-organizadas. A vantagem destes modelos é que não há necessidade de um conhecimento prévio em relação às falhas que possam ocorrer nos sensores. Estas redes são capazes de gerar agrupamentos, conforme seja detectado algum comportamento diferente do usual. A informação real do sensor, com falhas simuladas, e a estimativa do preditor são utilizados na geração de padrões de entrada para treinamento destas arquiteturas.

A contribuição principal deste trabalho é tornar possível o monitoramento local do sensor, independente do conhecimento dos modelos matemáticos do processo no qual o sensor está inserido, além de não aumentar custos com o uso de redundância dos equipamentos e fornecer respostas rápidas ao usuário, para que as devidas providências sejam tomadas em um tempo hábil. Associando a possibilidade de monitoramento em nível local com o avanço nas pesquisas voltadas à sensores inteligentes, estas novas funcionalidades são apresentadas com a finalidade de, futuramente, serem implementadas de forma embarcada junto aos dispositivos no nível de campo.

1.3 Trabalhos Relacionados

Dentre as aplicações das redes neurais, destacam-se aquelas voltadas para a predição de séries temporais. Em uma proposta voltada para melhorias em medição, (Sachenko et al. 2000) propôs o uso de uma rede neural recorrente para prever os desvios de calibração de um sensor com base nas curvas históricas de descalibração de sensores do mesmo tipo, sob as mesmas condições de uso. No trabalho de (Júnior & Barreto 2006) uma rede neural com realimentação global e entradas externas é explorada para tarefas de

predição de séries temporais não-lineares.

Em um trabalho recente, (Pinto et al. 2005) propõem uma estratégia de monitoramento de sensores e verificação da calibração utilizando redes neurais e métodos estatísticos. Em (Moustapha & Selmic 2007) utiliza-se redes neurais recorrentes aplicadas à detecção de falhas em redes de sensores sem fio.

Diversos trabalhos foram realizados no sentido de prover funcionalidades inteligentes para instrumentação de campo, como uma pesquisa destinada a implementar redes neurais artificiais em ambiente FF a partir de blocos funcionais (Silva 2005). Outro trabalho apresentou a criação de algoritmos de auto-calibração, auto-compensação e auto-validação de medidas em sensores (Pereira 2006) e a implementação do algoritmo de auto-compensação com uma rede recorrente utilizando blocos funcionais (Pereira et al. 2005). Em (Costa 2006) um algoritmo para extração de ruídos usando filtragem estocástica é proposto, e implementado em blocos funcionais *Foundation Fieldbus* e em uma placa baseada em DSP. Atualmente pesquisas vêm sendo desenvolvidas na área de multi-agentes e no estudo de sensores de *software* aplicados ao ambiente *Foundation Fieldbus* (Machado et al. 2008).

1.4 Publicação Científica

Durante o desenvolvimento desta dissertação o seguinte artigo científico foi publicado:

- Juliana P. Medeiros, Jorge D. Melo, Adrião D. Dória Neto, Vinícius P. Machado (2008): Algoritmo de Predição Baseado em Redes Neurais Recorrentes Aplicado ao Monitoramento de Medidas em Sensores, *XVII Congresso Brasileiro de Automática* (CBA 2008), Juiz de Fora (MG).

1.5 Organização do Texto

Neste documento, o Capítulo 2 apresenta uma fundamentação teórica relacionada a redes neurais artificiais e direciona suas aplicações para o contexto deste trabalho. O Capítulo 3 apresenta conceitos e técnicas referentes à atividade de Detecção e Diagnóstico de Falhas. No Capítulo 4, é explanada a implementação do algoritmo de predição e das estratégias de classificação de falhas, seguido do Capítulo 5 no qual os resultados obtidos são apresentados. Por fim, o Capítulo 6 apresenta as conclusões e perspectivas futuras deste trabalho.

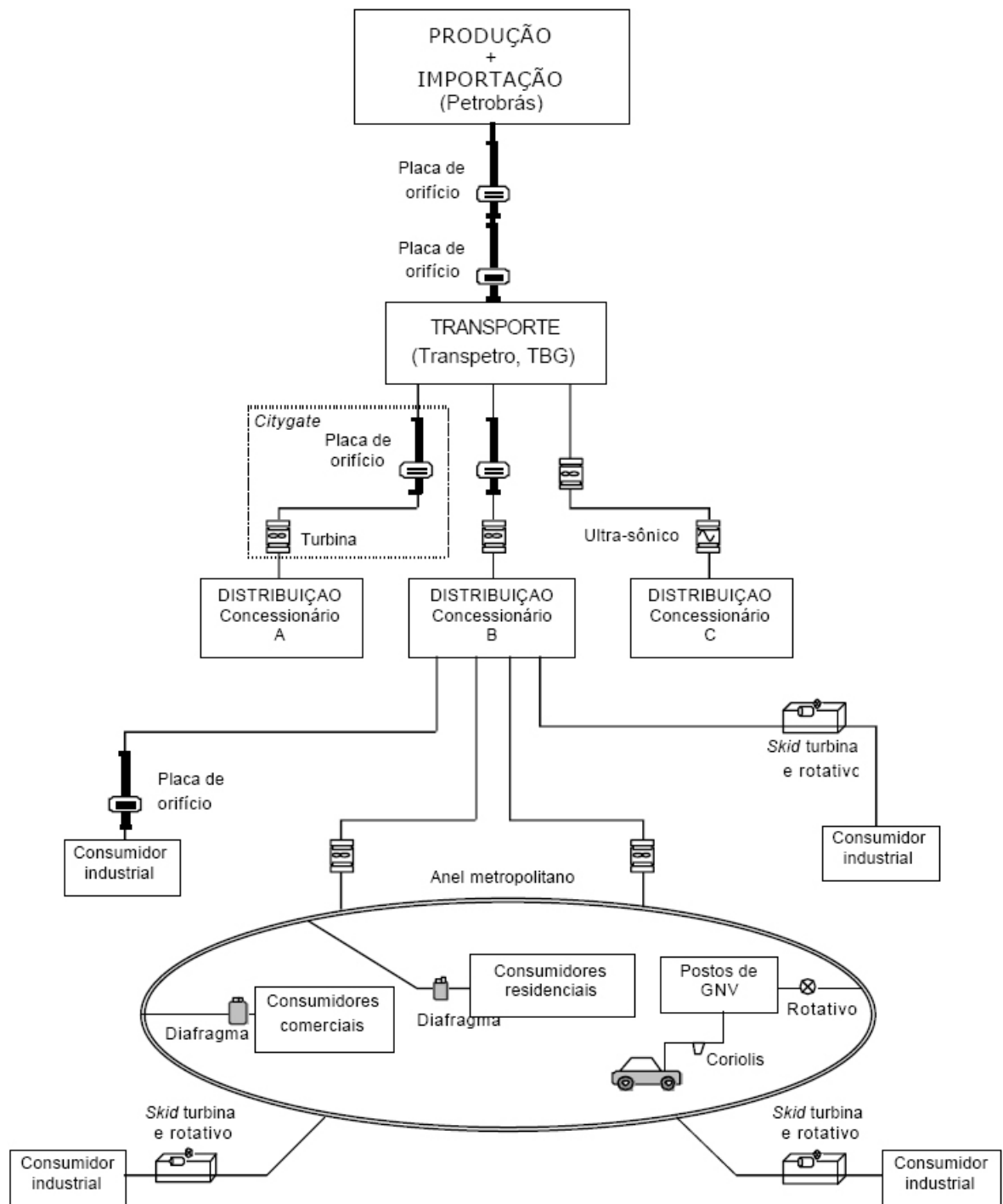


Figura 1.1: Cadeia de transporte e de distribuição de gás natural no Brasil

CAPÍTULO 2

Redes Neurais Artificiais

2.1 Introdução

Uma rede neural artificial (Haykin 2001b) é um sistema paralelamente distribuído cuja modelagem é inspirada no funcionamento dos neurônios biológicos. As redes neurais são constituídas por uma interligação de células computacionais simples denominadas "neurônios" ou "unidades de processamento", que são uma abstração do neurônio biológico. Os neurônios têm a característica de armazenar conhecimento experimental e torná-lo disponível para uso.

O modelo de um neurônio artificial pode ser visto na Figura 2.1 onde se observam os seguintes elementos:

- Pesos sinápticos: correspondem às forças de ligação entre os neurônios da rede e são representados por w_{kj} , onde k corresponde ao neurônio e j à uma de suas entradas.
- Combinador Linear: realiza o somatório das entradas do neurônio ponderadas pelos pesos sinápticos correspondentes gerando o campo local induzido, v_k .
- Função de Ativação: limita o campo local induzido entre 0 e 1 ou -1 e 1 para assegurar que sua amplitude tenha um valor finito. Pode ser linear ou não-linear.

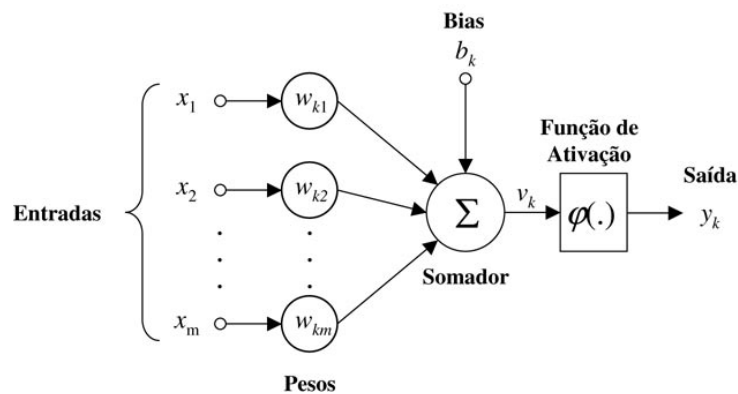


Figura 2.1: Representação gráfica de um neurônio artificial

A partir da estrutura acima pode-se formular um modelo matemático do neurônio artificial descrito pelas equações 2.1 e 2.2:

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

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

onde x_1, x_2, \dots, x_m representam os sinais de entrada, w_{kj} corresponde a cada peso sináptico, v_k representa o campo local induzido, e y_k é a saída do neurônio. O termo b_k corresponde ao *bias*, aplicado externamente e tem o efeito de aumentar ou diminuir o nível DC de entrada da função de ativação.

A maneira pela qual os neurônios de uma rede neural estão estruturados está intimamente ligada com o algoritmo de aprendizagem usado para treinar a rede. Pode-se classificar os algoritmos de treinamento de acordo com o paradigma de aprendizagem abordado. Sob este ponto de vista, o processo de treinamento pode ser supervisionado ou não-supervisionado.

O treinamento supervisionado é caracterizado pela presença de um "professor". O conhecimento se encontra na forma de conjuntos de exemplos de entrada-saída e é através deles que o professor consegue treinar a rede neural, alterando os pesos sinápticos e diminuindo o erro de saída. Uma das arquiteturas de rede neural mais utilizadas é a rede Perceptron de Múltiplas Camadas, ou *MultiLayer Perceptron* (MLP), na qual os neurônios são organizados em camadas e conectados entre si.

Na aprendizagem não-supervisionada, não há um "professor" externo para supervisionar o processo de aprendizado. A rede neural extrai informação somente dos padrões de entrada sem a necessidade de uma resposta desejada. Dentre os modelos de redes neurais auto-organizadas, duas arquiteturas serão investigadas, com o objetivo de diagnosticar falhas em sensores: as redes SOM (*Self-Organizing Map*) (Kohonen 1990) e as redes ART (*Adaptive Resonance Theory*) (Carpenter & Grossberg 1987).

2.2 Perceptron de Múltiplas Camadas

Como ilustra a Figura 2.2, uma rede neural MLP consiste de uma camada de entrada, através da qual os sinais de entrada são recebidos pela rede, uma ou mais camadas ocultas e uma camada de saída. A arquitetura da rede neural define a quantidade de camadas, quantos neurônios são dispostos em cada camada e se há, ou não, conexões de realimentação.

As redes MLP podem ser divididas em duas categorias: as redes alimentadas adiante, ou redes *feedforward*, e as redes recorrentes, ou recursivas.

As redes *feedforward* são aquelas cuja a informação é propagada a partir da camada de entrada, passando pelas camadas ocultas, até a camada de saída sem realimentação da informação, ou seja, as saídas de cada camada são conectadas somente às entradas da camada posterior, como previamente ilustrada na Figura 2.2.

As redes *feedforward* são estáticas e podem não ser capazes de capturar a dinâmica de

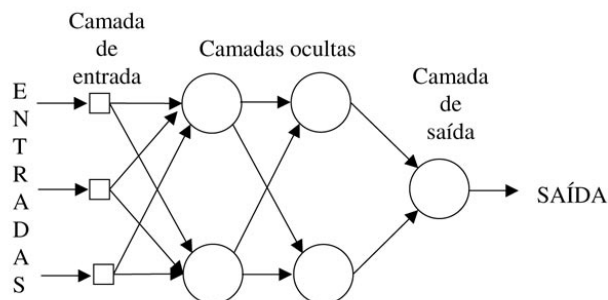


Figura 2.2: Perceptron de múltiplas camadas

um sistema. Para capturar a dinâmica de um sistema com mais eficácia, é mais apropriado o uso das redes recorrentes, devido à sua capacidade de agregar memória a rede neural (Principe & Kuo 1995). As redes com realimentação são objetos de estudo deste trabalho, devido à essa característica.

As redes neurais recorrentes possuem conexões de realimentação em sua arquitetura, ou seja, as saídas dos neurônios podem ser alimentadas de volta para suas entradas ou para os neurônios das camadas anteriores.

Estas redes são facilmente adaptadas a partir das redes *feedforward* bastando introduzir conexões de realimentação em suas camadas. O uso de realimentação com atrasos introduz memória à rede tornando-a apropriada para predição.

A realimentação em redes recorrentes pode ser tanto local como global. No caso local, as saídas das camadas ocultas são realimentadas entre si enquanto que no caso global a resposta da camada de saída é inserida na camada de entrada. De acordo com o tipo de realimentação podemos distinguir arquiteturas diferentes das redes recorrentes.

Uma propriedade importante de uma rede recorrente é que ela pode ser descrita pelo modelo de espaço de estados e pode aproximar uma ampla classe de sistemas dinâmicos não-lineares (Haykin 2001*b*).

A noção de estado desempenha um papel vital na formulação matemática de um sistema dinâmico. O estado de um sistema dinâmico é formalmente definido como um conjunto de quantidades que resumem toda a informação sobre o comportamento passado que é necessária para descrever unicamente o seu comportamento futuro, exceto pelos efeitos puramente externos que surgem devido à entrada aplicada. Em termos matemáticos, o comportamento dinâmico do sistema ilustrado pela rede recorrente da Figura 2.3, assumido livre do ruído, é descrito pelo seguinte par de equações não-lineares:

$$\mathbf{x}(n+1) = \varphi(\mathbf{W}_a \mathbf{x}(n) + \mathbf{W}_b \mathbf{u}(n)) \quad (2.3)$$

$$\mathbf{y}(n) = \mathbf{C}\mathbf{x}(n) \quad (2.4)$$

onde $\mathbf{x}(n)$ é um vetor q -por-1 representando o estado do sistema, $\mathbf{u}(n)$ é um vetor m -por-1, correspondente às entradas aplicadas ao sistema e $\mathbf{y}(n)$ é um vetor p -por-1 que representa a saída. A função $\varphi(\cdot)$ é a função de ativação não-linear. A dimensionalidade do espaço de estados, ou seja q , é a ordem do sistema, assim, o modelo de espaço de estados da Figura 2.3 é um modelo recorrente de ordem q , com m entradas e p saídas. A matriz \mathbf{W}_a , de dimensão q -por- q , representa os pesos sinápticos dos q neurônios na camada oculta que estão conectados aos nós de realimentação na camada de entrada. A matriz \mathbf{W}_b é uma matriz q -por- $(m+1)$ que representa os pesos sinápticos destes neurônios ocultos que estão conectados aos nós fontes na camada de entrada. Assume-se que os termos de bias para os neurônios ocultos estão incorporados na matriz de pesos \mathbf{W}_b . A matriz \mathbf{C} representa os pesos sinápticos dos p neurônios lineares na camada de saída que estão conectados aos neurônios ocultos e tem dimensão p -por- q . Assume-se que os termos de bias para os neurônios de saída estão incorporados na matriz de pesos \mathbf{C} .

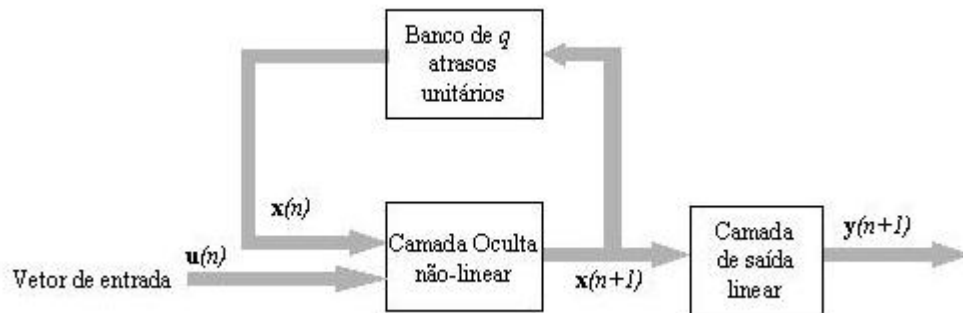


Figura 2.3: Modelo de espaço de estados

2.2.1 Predição de Séries Temporais com Redes MLP

Série Temporal é uma seqüência de dados ordenados no tempo que consistem nas saídas geradas por um determinado sistema o qual se deseja analisar.

A análise de séries temporais pode ser feita sob diferentes aspectos, tais como:

- Investigação do mecanismo gerador da série temporal;

- Descrição do comportamento da série;
- Predição de valores futuros.

A predição de valores futuros de uma série temporal consiste em estimar o comportamento da série em um momento futuro com base nas informações obtidas a partir de suas observações passadas.

Obtendo-se um conjunto de observações de uma série temporal até um determinado instante de tempo t e um modelo capaz de representar o mecanismo gerador destas informações, a previsão do valor da série no tempo $t + h$ pode ser obtida. A Figura 2.4, ilustra as amostras reais da série temporal, até o instante t , e as amostras estimadas por algum mecanismo de predição, desde $t + 1$ até $t + h$, em que h é conhecido como horizonte de predição.

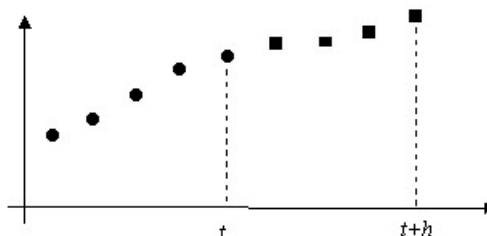


Figura 2.4: Predição h passos à frente

A previsão de valores futuros de uma série temporal é utilizada, principalmente nas áreas de economia, controle e otimização de processos industriais, planejamento comercial, entre outros (Carvalho 2005).

De acordo com o horizonte de predição desejado, podem existir duas maneiras de se prognosticar os valores futuros de uma série temporal (Mueller 1996):

- **Predição de Passo Simples:** a previsão é realizada para um período de tempo imediatamente posterior ao atual, a partir de observações passadas da série. Considera-se um horizonte de curto prazo.
- **Predição de Passos Múltiplos:** adotada para horizontes de médio e longo prazo. Na previsão múltiplos passos, o conjunto de valores correntes é empregado na realização da previsão para determinado instante; esta previsão é, então, introduzida entre as observações passadas, compondo, desta forma, um novo conjunto de dados, sobre o qual será obtida a previsão do tempo subsequente.

Idealmente, a realização de previsões seria feita a partir do conhecimento das equações que modelam os mecanismos responsáveis pela geração das séries. No entanto, em muitas

situações reais, não é possível obter as informações necessárias para a construção destas equações. Nestes casos, utiliza-se um modelo que possa se aproximar do sistema responsável pela geração da série de tempo observada e, a partir desse modelo, predizer os valores futuros da mesma (Gomes 2005).

A predição realizada através de um modelo é baseada na extrapolação de características de observações passadas e do inter-relacionamento entre elas, levando em consideração que tais observações contém todas as informações sobre o padrão de comportamento da série temporal e o que o comportamento desta série, no futuro, é similar ao passado.

A predição de séries de tempo vêm sendo realizada com o uso dos tradicionais modelos Autoregressivo e/ou Médias Móveis (AR, MA e ARMA) e modelos Autoregressivo Integrado de Médias Móveis (ARIMA), cujas siglas vêm dos termos em inglês, *AutoRegressive, Integrate, Moving Average*. Tratam-se de modelos paramétricos através dos quais a previsão das futuras observações é obtida a partir da combinação linear de valores passados e/ou componentes de ruído da série de interesse ponderadas por um conjunto de parâmetros. Estes modelos são utilizados em (Yung 1993) como estratégia de detecção local de falhas em sensores.

A partir do final da década de 80 as redes neurais começaram a ser consideradas para realizar atividades de predição de séries temporais (Mandic & Chambers 2001). Devido à sua capacidade de agregar conhecimento em sua estrutura, a partir de exemplos, um preditor baseado em rede neural tem condições de estimar o comportamento futuro de uma série temporal apenas a partir de suas amostras passadas. Um modelo de predição deste tipo é chamado de não-paramétrico, uma vez que não há a necessidade de se conhecer os parâmetros do processo que gera o sinal. O modelo do processo é estimado através de um algoritmo de aprendizagem onde os exemplos são apresentados à rede neural e seus pesos são atualizados de acordo com o erro de predição.

Além da capacidade de aprendizagem, as redes neurais tem capacidade de lidar com sistemas cujas séries temporais apresentam maiores dificuldades para a predição, como não-linearidades e ruído. Os modelos estatísticos tradicionais são lineares e não são capazes de lidar com a natureza não-linear e a não-estacionariedade de certos sinais. Já as redes neurais têm condições de realizar um mapeamento entrada-saída não-linear devido às suas funções de ativação.

Um exemplo típico de rede neural aplicada à predição consiste de uma rede treinada a partir de janelas de tempo. A entrada da rede consiste em uma memória com a observação atual e as passadas do sinal e a saída desejada para esta entrada é a observação obtida no próximo passo de tempo.

O modelo utilizado neste trabalho é baseado no modelo NARX, que nada mais é

do que uma rede MLP cuja entrada consiste da própria saída realimentada com atrasos no tempo e uma entrada exógena, também com atrasos. Nesta arquitetura, as saídas estimadas da rede são introduzidas novamente às entradas, permitindo implementar a predição de passos múltiplos (Principe & Kuo 1995), (Menezes 2006). Esta rede neural é um equivalente do modelo estatístico NARX (*Nonlinear AutoRegressive model with eXogenous input*), que realiza o seguinte mapeamento entrada-saída:

$$y(n) = g(y(n-1), \dots, y(n-l+1), u(n), \dots, u(n-q+1)), \quad (2.5)$$

onde $u(n)$ e $y(n)$ correspondem à entrada e saída da rede no tempo t e l e q são, respectivamente, as ordens da saída e da entrada. A função $g(\cdot)$ é uma função não-linear geralmente desconhecida que pode ser aproximada, justamente, por uma rede MLP resultando na topologia apresentada na Figura 2.5.

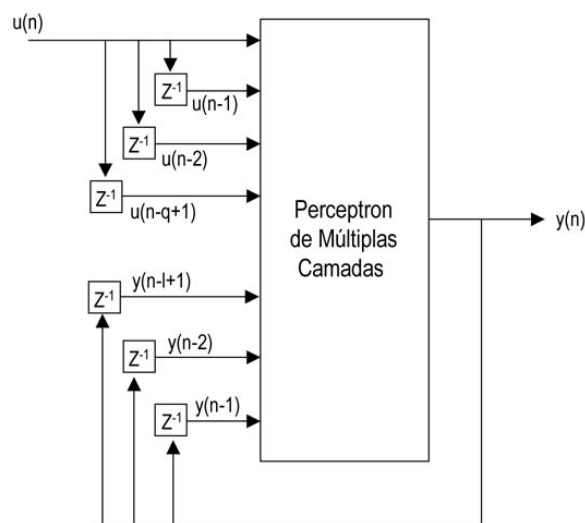


Figura 2.5: Rede NARX

O modelo tem uma única entrada que é aplicada a uma memória de linha de atraso derivada com q unidades e também uma única saída que é realimentada para a entrada através de uma outra memória de linha de atraso derivada, também com q unidades. Os conteúdos destas duas memórias são utilizados para alimentar a camada de entrada de um perceptron de múltiplas camadas. O valor presente da entrada do modelo é representado por $u(n)$, e o valor correspondente da saída do modelo é representado por $y(n)$.

Em (Haykin 2001b) mostra-se que o modelo NARX, representado pela Figura 2.5,

com sua realimentação limitada ao neurônio de saída, é capaz de simular o modelo de espaço de estados das equações 2.3 e 2.4, representado na Figura 2.3, sem nenhuma diferença entre os seus comportamentos de entrada-saída.

A rede NARX pode ser treinada e utilizada em dois modos de operação diferentes (Narendra & Parthasarathy 1990):

- Modo de Identificação Paralelo: a saída estimada é realimentada e incluída no regressor da saída;

$$\hat{y}(n) = \hat{g}(\hat{y}(n-1), \dots, \hat{y}(n-l+1), u(n), u(n-1), \dots, u(n-q+1)) \quad (2.6)$$

- Modo de Identificação Série-Paralelo: o regressor de saída é formado somente por valores atuais do sistema;

$$\hat{y}(n) = \hat{g}(y(n-1), \dots, y(n-l+1), u(n), \dots, u(n-q+1)) \quad (2.7)$$

2.2.2 O Filtro de Kalman Estendido

O Filtro de Kalman Estendido se enquadra no método de treinamento supervisionado, caracterizado pela presença de um professor. O conhecimento se encontra na forma de conjuntos de exemplos de entrada-saída e é através deles que o professor consegue treinar a rede neural, alterando os pesos sinápticos e diminuindo o erro de saída.

O Filtro de Kalman Estendido é uma alternativa aos algoritmos baseados no método do gradiente. Ele provê um componente estocástico que permite uma busca mais efetiva no espaço de pesos. Este algoritmo é uma adaptação ao Filtro de Kalman para sistemas não-lineares.

O Filtro de Kalman (Kalman 1960) é uma ferramenta matemática poderosa que provê uma solução ótima para o problema da filtragem linear que consiste em estimar os estados de um determinado sistema linear cujo comportamento é conhecido somente pelas suas entradas e saídas, que podem ser ruidosas.

Um sistema dinâmico pode ser modelado através das equações de estados, como segue:

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + \mathbf{q}_k \quad (2.8)$$

$$\mathbf{y}_{k+1} = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k \quad (2.9)$$

O filtro de Kalman, na sua formulação original, não pode ser utilizado para estimar os estados de sistemas não-lineares, como as Redes Neurais. Para este propósito utiliza-se o Filtro de Kalman Estendido (FKE).

Considere uma Rede Neural cujos pesos sinápticos são representados pelo vetor \mathbf{w}_k . Seu comportamento pode ser descrito através de um sistema discreto não-linear modelado pelas equações de estado a seguir (Haykin 2001a):

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \boldsymbol{\omega}_k \quad (2.10)$$

$$\mathbf{y}_k = h(\mathbf{w}_k, \mathbf{u}_k, \mathbf{v}_k) + \vartheta_k \quad (2.11)$$

A equação 2.10, conhecida como equação de processo, especifica que o estado de uma rede neural ideal (após o treinamento) é caracterizado por um processo estacionário corrompido por um ruído de processo $\boldsymbol{\omega}_k$, branco e de média zero, onde o estado do sistema é dado pelos pesos sinápticos da rede neural \mathbf{w}_k . A equação 2.11, usualmente denominada equação de medida, representa o vetor de saídas desejadas \mathbf{y}_k , como uma função não-linear do vetor de entrada \mathbf{u}_k , do vetor de pesos sinápticos \mathbf{w}_k e, se a rede for recorrente, do vetor de atividades recorrentes \mathbf{v}_k . Esta equação é acrescida de um ruído de medida ϑ_k , branco e de média zero.

Para se aplicar a teoria do filtro de Kalman no modelo não-linear, deve-se linearizar a equação de medida para que ela fique na seguinte forma:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{w}_k + \vartheta_k \quad (2.12)$$

A linearização é feita calculando-se as derivadas parciais das p saídas da rede em relação aos W pesos da rede, como mostra a equação 2.13:

$$\mathbf{H}_k = \begin{bmatrix} \frac{\partial h_1}{\partial w_1} & \frac{\partial h_1}{\partial w_2} & \cdots & \frac{\partial h_1}{\partial w_n} \\ \frac{\partial h_2}{\partial w_1} & \frac{\partial h_2}{\partial w_2} & \cdots & \frac{\partial h_2}{\partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial w_1} & \frac{\partial h_p}{\partial w_2} & \cdots & \frac{\partial h_p}{\partial w_n} \end{bmatrix} \quad (2.13)$$

O cálculo da matriz \mathbf{H}_k pode ser realizado com o algoritmo de retropropagação. Com a linearização da matriz de medida, prossegue-se com a aplicação do algoritmo da filtragem de Kalman, descrito de forma a encontrar a melhor estimativa para o estado \mathbf{w} . O Filtro de Kalman Estendido aplicado ao processo de aprendizagem da rede neural é dado através das seguintes equações:

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (2.14)$$

$$\hat{\mathbf{w}}_{k+1} = \hat{\mathbf{w}}_k + \mathbf{K}_k \xi_k \quad (2.15)$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k + \mathbf{Q}_k \quad (2.16)$$

O vetor $\hat{\mathbf{w}}_k$ corresponde ao valor estimado dos pesos no k-ésimo passo de treinamento. A atualização deste vetor é uma função do erro de saída $\xi_k = \mathbf{y}_k - \hat{\mathbf{y}}_k$, ponderado pela matriz de ganho de Kalman, \mathbf{K}_k . A matriz \mathbf{P}_k , corresponde à matriz de covariância do erro de aproximação dos estados. Na prática, esta matriz é inicializada assumindo $\mathbf{P}_0 = \varepsilon^{-1} \mathbf{I}$ (na ordem de 0.001-0.01). A matriz de ganho de Kalman é uma função da matriz de covariância do erro de aproximação \mathbf{P}_k , da matriz de medida linearizada \mathbf{H}_k e da matriz de covariância do ruído de medida \mathbf{R}_k . A matriz \mathbf{P}_k evolui recursivamente junto com as estimativas dos pesos e é acrescida da matriz de covariância do ruído de processo \mathbf{Q}_k .

O treinamento da rede neural é realizado de forma seqüencial. Cada passo de treinamento é composto dos seguintes subpassos:

1. Um vetor de entradas \mathbf{u}_k é propagado através da rede para produzir um vetor de saídas \mathbf{y}_k . Calcula-se o erro de saída ξ_k .
2. A matriz de derivadas parciais \mathbf{H}_k é obtida a partir do algoritmo *backpropagation*.
3. Calcula-se a matriz de ganho de Kalman, \mathbf{K}_k .
4. Os pesos da rede neural são atualizados.
5. Atualiza-se a matriz de covariância do erro de aproximação, \mathbf{P}_k .

2.3 Redes Neurais com Aprendizado Auto-Supervisionado

2.3.1 Classificação de Padrões com Aprendizado Auto-Supervisionado

Uma rede MLP treinada com algoritmo *BackPropagation* é um exemplo clássico de rede neural atuando como classificador. Cada padrão de entrada apresentado à rede deve ser mapeado para uma classe de saída. Esta associação entrada-saída é previamente conhecida e é justamente a informação utilizada para o treinamento supervisionado da arquitetura. Em alguns problemas de classificação, no entanto, a relação entre padrões de entrada e suas respectivas classes não é explícita de tal forma que não seja possível criar um conjunto entrada-saída para uma abordagem supervisionada.

As redes neurais apresentadas nesta seção têm a característica de detectar entradas semelhantes e associá-las ao mesmo neurônio ou agrupamento de neurônios (*cluster*), de forma que cada um deles represente uma classe ou categoria após o treinamento. No contexto deste trabalho, estas redes têm a função de classificar padrões de sinal de um sensor. Ao final do treinamento, o padrão de sinal que corresponde ao funcionamento normal do sensor deve estar associado a um determinado neurônio ou agrupamento, enquanto padrões que representem falhas devem estar associados a outros agrupamentos distintos.

2.3.2 Mapas Auto-Organizáveis

O Mapa Auto-Organizável (SOM) (Kohonen 1990) foi idealizado a partir da analogia com a região do córtex cerebral humano. Descobriu-se que esta parte do cérebro aloca regiões específicas para atividades específicas e que, para uma determinada ativação cerebral, o grau de ativação dos neurônios diminuía à medida que se aumentava a distância da região de ativação inicial. Neste contexto, surgiu o interesse na construção de mapas topográficos artificiais que aprendem através de uma forma de auto-organização inspirada na neurobiologia. Este estudo levou ao *princípio da formação de mapas topográficos*, que pode ser formulado como: "A localização espacial de um neurônio de saída em um mapa topográfico corresponde a um domínio ou característica particular do dado retirado do espaço de entrada".

Em um mapa auto-organizável, os neurônios estão colocados em nós de uma grade que é normalmente uni- ou bidimensional. Os neurônios se tornam *seletivamente sintonizados* a vários padrões de entrada (estímulos) ou classes de padrões de entrada no decorrer de um processo de aprendizagem. As localizações dos neurônios assim sintonizados (i.e., os neurônios vencedores) se tornam ordenadas entre si de forma que um sistema de coordenadas significativo para diferentes características de entrada é criado sobre a grade. Um mapa auto-organizável é, portanto, caracterizado pela formação de um mapa topográfico dos padrões de entrada no qual as localizações espaciais dos neurônios na grade (linha e coluna) são indicativas das características estatísticas intrínsecas contidas nos padrões de entrada.

Este princípio forneceu a motivação neurobiológica para dois modelos de mapeamento de características, que diferem entre si no modo como os padrões de entrada são especificados.

O modelo em estudo neste trabalho foi introduzido por Kohonen (Kohonen 1990). Este modelo busca capturar as características essenciais dos mapas computacionais do cérebro e ainda se manter tratável do ponto de vista computacional. O mapa auto-organizável

de Kohonen consiste de uma única grade de neurônios. O modelo produz um mapeamento topológico que localiza otimamente um número fixo de vetores de peso associados a neurônios que representam um espaço de entrada de dimensionalidade mais elevada, e desse modo facilita a compressão de dados de entrada.

A Figura 2.6 mostra o diagrama esquemático de uma grade bidimensional de neurônios, representando um mapa discreto. Cada neurônio da grade está totalmente conectado a todos os nós de fonte da camada de entrada. Esta grade representa uma estrutura alimentada adiante com uma única camada computacional consistindo de neurônios arranjados em linhas e colunas. Cada padrão de entrada apresentado à grade consiste tipicamente de uma região localizada ou "foco" de atividade contra um fundo em repouso. A localização e a natureza deste foco será retratada por um neurônio vencedor, que varia de um padrão de entrada para outro.

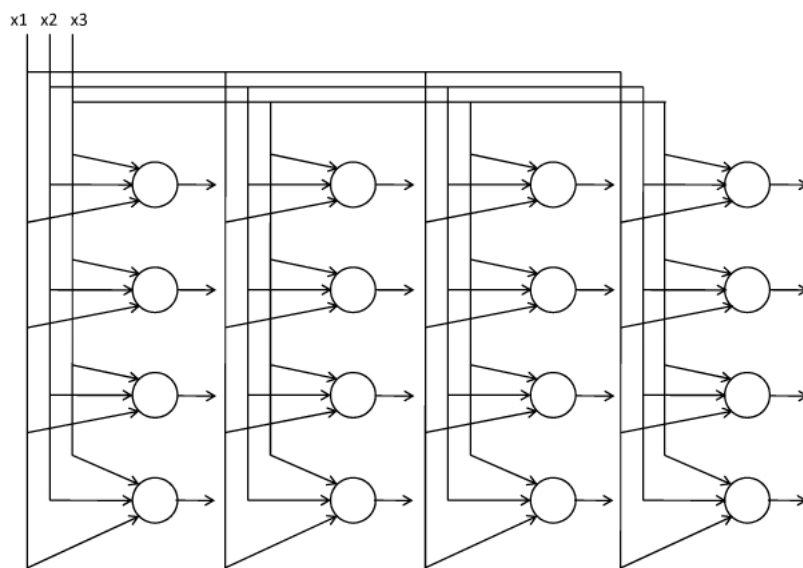


Figura 2.6: Grade de neurônios

O algoritmo responsável pela formação do mapa auto-organizável começa inicializando os pesos sinápticos da grade atribuindo-lhes, usualmente, valores pequenos e aleatórios, de forma que nenhuma organização prévia é imposta ao mapa de características.

Antes de iniciar o algoritmo primeiramente deve-se definir os seguintes parâmetros:

- Um espaço de entrada contínuo composto por padrões de ativação que são gerados

de acordo com uma certa distribuição de probabilidade.

- Uma topologia na forma de uma grade de neurônios, que define um espaço de saída discreto.
- Uma função de vizinhança variável no tempo $h_{j,i(\mathbf{x})}(n)$, definida em torno de um neurônio vencedor $i(\mathbf{x})$, em que j corresponde a cada neurônio da grade.
- Um parâmetro da taxa de aprendizagem $\eta(n)$ que começa em um valor inicial $\eta(n=0)$ e então diminui gradualmente com o tempo discreto n , mas nunca vai a zero.

Uma vez que a grade tenha sido propriamente inicializada, há três passos básicos envolvidos na formação do mapa auto-organizável:

1. Competição

Para cada padrão de entrada, os neurônios da grade calculam seus respectivos valores de uma função discriminante, que fornece a base para a competição entre os neurônios. O neurônio particular com o maior valor da função discriminante é declarado vencedor da competição. Considere que \mathbf{x} represente uma amostra do espaço de entrada. O vetor \mathbf{x} representa um padrão de ativação de dimensão igual a m , aplicado à grade. O vetor peso sináptico \mathbf{w}_j de cada neurônio j da grade tem a mesma dimensão que o espaço de entrada. Para encontrar o melhor casamento, selecione o neurônio com o maior produto interno $\mathbf{w}_j^T \mathbf{x}$ usando o critério da mínima distância euclidiana:

$$i(\mathbf{x}) = \operatorname{argmin}_j \|\mathbf{x}(n) - \mathbf{w}_j\|, j = 1, 2, \dots, l \quad (2.17)$$

2. Cooperação

O neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios excitados, fornecendo assim a base para a cooperação entre os neurônios vizinhos. A função de vizinhança $h_{j,i(\mathbf{x})}(n)$ deve inicialmente incluir quase todos os neurônios da grade centrados no neurônio vencedor i e então diminuir lentamente durante a aprendizagem.

3. Adaptação Sináptica

Este último mecanismo permite que os neurônios excitados aumentem seus valores individuais da função discriminante em relação ao padrão de entrada através de ajustes aplicados aos seus pesos sinápticos. Os ajustes feitos são tais que a resposta do neurônio vencedor à aplicação subsequente de um padrão de entrada similar é melhorada. A atualização dos pesos sinápticos é realizada com a seguinte fórmula:

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n)h_{j,i(\mathbf{x})}(\mathbf{x}(n) - \mathbf{w}_j(n)) \quad (2.18)$$

onde, $\eta(n)$ é o parâmetro da taxa de aprendizagem e $h_{j,i(\mathbf{x})}(n)$ é a função de vizinhança centrada em torno do neurônio vencedor $i(\mathbf{x})$.

Os passos acima são repetidos até que não sejam observadas modificações significativas no mapa de características. Uma vez que o algoritmo SOM tenha convergido, o *mapa de características* calculado pelo algoritmo mostra características estatísticas importantes do espaço de entrada. Dado um vetor de entrada \mathbf{x} , o algoritmo SOM identifica um neurônio com o melhor casamento ou neurônio vencedor $i(\mathbf{x})$ na grade.

2.3.3 Teoria da Ressonância Adaptativa

A Teoria da Ressonância Adaptativa foi introduzida como uma teoria do processamento cognitivo da informação pelo ser humano (Grossberg 1976). Desde então, esta teoria serviu como base para uma série de modelos de rede neural com objetivo de descobrir agrupamentos em um conjunto de padrões de forma não supervisionada. A família de modelos ART (Adaptive Resonance Theory) foi proposta em arquiteturas capazes de gerar, de forma estável, agrupamentos em resposta a sequências de entrada arbitrárias, devido à suas características de auto-organização e auto-estabilização (Carpenter & Grossberg 1987), (Carpenter et al. 1991).

Os modelos ART foram desenvolvidos com o objetivo de evitar o chamado dilema da estabilidade-plasticidade em redes neurais que utilizam aprendizagem não-supervisionada. Este dilema demonstra como um sistema de aprendizagem pode preservar o conhecimento adquirido *a priori* enquanto se mantém capaz de agregar novos conhecimentos, ou seja, como aprender uma nova informação sem "esquecer" aquela já armazenada. A motivação básica seria o treinamento contínuo, por exemplo, em um ambiente onde as condições alteram-se ao longo do tempo.

A arquitetura de um sistema ART consiste de dois subsistemas como mostra a Figura 2.7 (Carpenter & Grossberg 1987), um subsistema de atenção e um subsistema de orientação. No subsistema de atenção ocorre a ativação dos pesos e a estabilização do treinamento, através da correspondência entre a entrada e a memória interna da arquitetura. O subsistema de orientação funciona como um detector de novidades que controla o subsistema de atenção quando não há correspondência entre a entrada e a memória. O mecanismo de aprendizagem destas arquiteturas engloba um esquema de busca paralela

que se atualiza adaptativamente à medida que o treinamento se desenvolve e dependerá do modelo utilizado.

A rede ART consiste de duas camadas, uma camada de entrada, F_1 e uma camada de saída, F_2 , que é uma camada competitiva. Estas camadas codificam padrões de ativação e os armazenam em memórias de curto prazo (STM - *Short Term Memory*), que existem apenas durante a apresentação de um vetor de entrada. A camada F_1 é totalmente conectada a F_2 através de conexões *bottom-up* e *top-down*, ponderadas por pesos, codificados em memórias de longo prazo (LTM - *Long Term Memory*). O conjunto de conexões com seus pesos é chamado de filtro adaptativo, pois os pesos podem ser modificados dinamicamente para se adaptar a novos vetores de entrada.

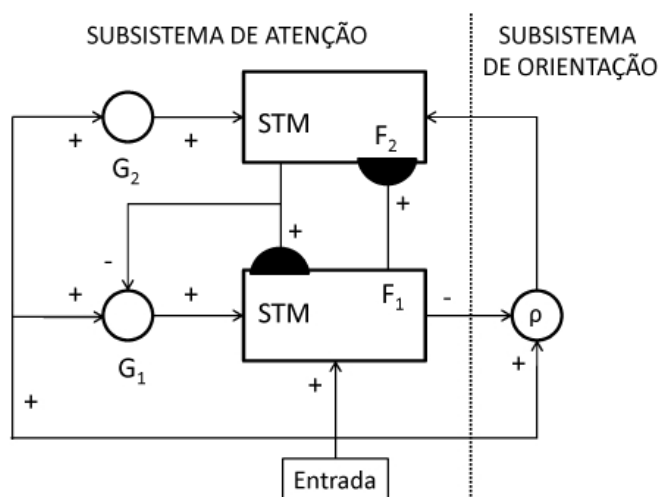


Figura 2.7: Rede ART

Vários modelos são descritos na literatura. No mais simples, ART1 apenas padrões binários são permitidos. ART2 é similar ao ART1 com a possibilidade de usar padrões reais, e ART3 apresenta princípios organizacionais (incorpora no modelo transmissores químicos) que permitem o processo de busca em uma estrutura hierárquica. *Fuzzy ART* é uma modificação do modelo ART para incorporar lógica nebulosa. O modelo ARTMAP é uma versão supervisionada que pode aprender mapeamentos arbitrários de padrões binários (Carpenter et al. 1991). Uma versão nebulosa deste último modelo é o *Fuzzy ARTMAP* que também possui mecanismo de aprendizagem supervisionado.

O modelo em estudo neste trabalho é o modelo *Fuzzy ART* (Carpenter et al. 1991), capaz de aprender de forma estável o reconhecimento de categorias em resposta a seqüências arbitrárias de padrões de entrada analógicos ou binários. O mecanismo de busca por

uma categoria, na rede *Fuzzy ART*, segue os 4 passos a seguir:

1. Apresentação de um padrão de entrada

Cada entrada \mathbf{I} é um vetor M -dimensional tal que,

$$\mathbf{I} = (I_1, \dots, I_M) \quad (2.19)$$

em que cada componente se encontra no intervalo $[0,1]$. Um padrão \mathbf{I} é apresentado à rede e é armazenado em uma memória de curto prazo STM na camada F_1 . Neste instante o subsistema de orientação permanece inibido. Cada categoria j corresponde a um vetor $\mathbf{w} = (w_{j1}, \dots, w_{jM})$ de pesos adaptativos, ou LTM. Inicialmente, nenhuma categoria é associada aos pesos. O vetor de pesos \mathbf{w} , no modelo *Fuzzy ART*, abrange tanto o filtro adaptativo *bottom-up* quanto o *top-down*. Um sinal T é gerado na STM da camada F_2 , através da seguinte equação:

$$T_j(\mathbf{X}) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \quad (2.20)$$

em que $\alpha > 0$ é o parâmetro de escolha e \wedge é o operador AND na lógica *Fuzzy*, representado por

$$(\mathbf{x} \wedge \mathbf{y})_i = \min(x_i, y_i), \quad (2.21)$$

e a norma $|\cdot|$ é definida por

$$|\mathbf{x}| = \sum_{i=1}^M |x_i|. \quad (2.22)$$

O nó j que tiver a maior entrada T_j representa a categoria que melhor se enquadra ao vetor de \mathbf{I} , ou seja, sua ativação pode ser interpretada como uma hipótese para o vetor \mathbf{I} . Cada nó nesta camada possui conexões inibitórias para os nós vizinhos de forma que apenas o nó com a maior entrada, seja ativado.

2. Teste de hipótese

A rede ART deve, então, confrontar o padrão de entrada \mathbf{I} com a categoria esperada, ou seja, verificar se a hipótese obtida no passo anterior corresponde com a entrada. Um vetor \mathbf{V} representa o padrão criado a partir dos pesos escolhidos. Se \mathbf{V} e \mathbf{I} forem iguais, então a busca é finalizada, e a entrada \mathbf{I} é, então, associada àquela categoria j , ou seja, àquele neurônio vencedor. Se \mathbf{V} não é exatamente igual a \mathbf{I} na camada F_1 , o subsistema de orientação é desinibido, para que se realize um teste de equivalência.

3. Teste de equivalência

O critério de uma equivalência aceitável é definido por um parâmetro adimensional chamado de *vigilância* (ρ). O parâmetro de vigilância indica o quão próximo a entrada \mathbf{I} está da categoria j esperada. O termo ressonância se refere ao momento em que função de equivalência da categoria escolhida atender o parâmetro de vigilância:

$$\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} \geq \rho. \quad (2.23)$$

Neste caso, o vetor de pesos é atualizado de acordo com a regra a seguir:

$$\mathbf{w}_j^{(new)} = \beta(\mathbf{I} \wedge \mathbf{w}_j^{(old)}) + (1 - \beta)\mathbf{w}_j^{(old)}, \quad (2.24)$$

onde β é a taxa de aprendizagem, $\beta \in [0, 1]$. Ao se escolher $\beta = 1$, tem-se a denominada aprendizagem rápida (*fast learning*), que permite a um sistema se adaptar rapidamente a entradas que são raramente apresentadas, mas que requeiram respostas acuradas.

Se o parâmetro de vigilância não for atendido, isto é, se:

$$\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} < \rho, \quad (2.25)$$

o valor da função de escolha T_j recebe um sinal de *reset* e assume o valor -1 durante a apresentação da mesma entrada, para evitar que o mesmo nó seja selecionado novamente.

4. Reapresentação do vetor de entrada

O sinal de *reset* para T_j em F_2 , resulta na eliminação do vetor \mathbf{V} na camada F_1 . O sinal T é novamente gerado para a camada F_2 . O nó anteriormente escolhido em F_2 permanece inibido, devido à ação do subsistema de orientação, de forma que um novo j seja escolhido. O processo de busca continua até que a categoria escolhida satisfaça a equação 2.23. A inibição dos nós só é removida quando um novo padrão de entrada é apresentado.

Se o padrão de entrada \mathbf{I} chegar a ser diferente de qualquer categoria previamente aprendida, um nó em F_2 não comprometido é selecionado, e a aprendizagem de uma nova categoria é iniciada com a atualização dos pesos conforme a equação 2.24. Quando a aprendizagem de uma categoria se estabiliza, todas as entradas codificadas por aquela categoria acessam-na diretamente. É possível que se realize a aprendizagem *online*, de forma que entradas não categorizadas continuem com buscas adaptativas por novas cate-

gorias, até que a capacidade de memória da rede seja totalmente utilizada.

Para aumentar o desempenho da rede *Fuzzy ART*, uma regra denominada *Complemento de Código* é utilizada para normalização das entradas prevenindo um efeito conhecido como proliferação de categorias (Carpenter et al. 1991). O complemento de uma entrada a é calculado como segue:

$$a^c = 1 - a \quad (2.26)$$

Em resumo, o treinamento da rede *Fuzzy ART*, segue os princípios do primeiro modelo ART1. Inicialmente não há nenhum agrupamento (i.e., neurônio ativo, representando um agrupamento). A aprendizagem ocorre comparando-se os estímulos da entrada com modelos internos, pesos sinápticos que representam agrupamentos. Caso a distância do padrão ao conjunto de pesos mais próximo exceda um limiar (parâmetro de vigilância), um novo agrupamento é formado, caso existam neurônios ainda não atribuídos a nenhum agrupamento. Caso a distância esteja dentro do limiar a algum agrupamento existente, os pesos sinápticos correspondentes são adaptados.

CAPÍTULO 3

Detecção e Diagnóstico de Falhas em Sensores

3.1 Introdução

Neste capítulo serão apresentados alguns conceitos e técnicas referentes à Detecção e Diagnóstico de Falhas (DDF), voltados a sensores. Em primeiro lugar será apresentada a terminologia básica sobre DDF, com alguns conceitos utilizados na área.

Entende-se como *falha*, um mau funcionamento de qualquer componente de um sistema, causando desde a sua perda de desempenho até a total parada da execução de suas funções.

Quando um sistema está com falha, este se comporta de forma diferente do esperado. Chama-se *erro* a discrepância observada entre uma variável medida e uma faixa aceitável para o valor esperado desta variável (do Amaral 2006).

O termo *sensor* é utilizado neste trabalho de acordo com a definição de (Zhang & Yan 2001), na qual o sensor representa todo sistema que comporta um dispositivo de sensoriamento, um transdutor, um condicionador de sinal e componentes de interface. A entrada do sensor é o mensurando $x(t)$, que é desconhecido, e sua saída é a medição $y(t)$, onde t é uma variável que representa o tempo.

Um sensor é considerado *com falha* se a medição $y(t)$ fornece uma representação incorreta do mensurando $x(t)$. A falha em um sensor pode surgir de qualquer parte do sistema.

Um entendimento dos tipos de falhas e de seus padrões de sinal associados é muito importante para os métodos de classificação de falhas em sensores. Para facilitar o estudo dos algoritmos será abordada a classificação encontrada em (Zhang & Yan 2001), que apresenta dois tipos de falhas possíveis em um sensor:

- Falha Abrupta: falha repentina de um componente, modelada como desvios do tipo degrau.
- Falha Incipiente: falha que ocorre lentamente em um componente, modelada por derivas na medição.

O processo de DDF tem a finalidade de identificar a causa do mau funcionamento de um sistema. Normalmente, o processo de detecção e diagnóstico de falhas pode ser dividido em três fases (do Amaral 2006):

- Detecção da Falha: determinar se uma falha de fato ocorreu num sistema;
- Isolamento da Falha: determinar onde esta falha ocorreu;
- Determinar o tipo e extensão da falha: consiste na classificação da falha.

Ao se considerar que a atividade de DDF é realizada em um sensor, o processo de isolamento de falha se torna muito complexo, pois seria necessário detalhar a relação entre as falhas de sinal e as falhas físicas, intrínsecas de cada dispositivo. Este seria um processo minucioso que requer o estudo aprofundado das características de cada tipo de sensor para identificar todas as possíveis falhas que podem ocorrer. Por esta razão, no contexto deste trabalho, o processo de detecção e diagnóstico de falhas se reduz às fases de detecção e classificação.

As atividades de detecção e classificação de falhas em sensores desempenhadas neste trabalho são baseadas em um esquema de *validação local de sensores* (Yung 1993). Trata-se de uma abordagem na qual as falhas em sensores são detectadas ou diagnosticadas no nível mais baixo de uma estrutura hierárquica e descentralizada de detecção e diagnóstico de falhas, apresentada na Figura 3.1. A detecção e diagnóstico de falhas pode ocorrer em três níveis: no nível de planta, um modelo do processo é utilizado para reproduzir a saída do sensor a partir das medições de outros sensores; no nível de processo são utilizadas medições redundantes para compará-las com as medições do sensor; no nível do sensor, ou seja, no nível de campo um estado da medição é estimado unicamente através da análise da própria saída do elemento de medição.

A idéia por trás da detecção e diagnóstico local de falhas é fornecer uma informação adequada e correta ao nível acima de modo que, quando uma falha ocorre no sensor, as ações apropriadas sejam tomadas. Um esquema de validação local do sensor deve ser abrangente (capaz de detectar e diagnosticar tipos de falhas diferentes), portátil (aplicável a sensores diferentes), simples (de forma que possa ser processado no próprio sensor), viável (provendo detecção acurada e taxas mínimas de alarmes falsos) e independente do processo (livre da necessidade de desenvolver um modelo detalhado do processo).

3.2 Técnicas de detecção e diagnóstico de falhas

3.2.1 Redundância Física

As técnicas baseadas em redundância física, ou redundância de *hardware*, consistem na construção de subsistemas físicos no sistema monitorado. Na indústria se utiliza, normalmente, redundância dupla ou tripla de sensores para se detectar falhas nos mesmos. As saídas dos sensores redundantes são comparadas entre si e em caso de divergência entre as saídas correspondentes considera-se que um deles se encontra com falha. A vantagem deste sistema é que, no caso de uma falha, o sistema redundante assume o monitoramento. A desvantagem desta técnica está no aumento do custo, espaço e complexidade

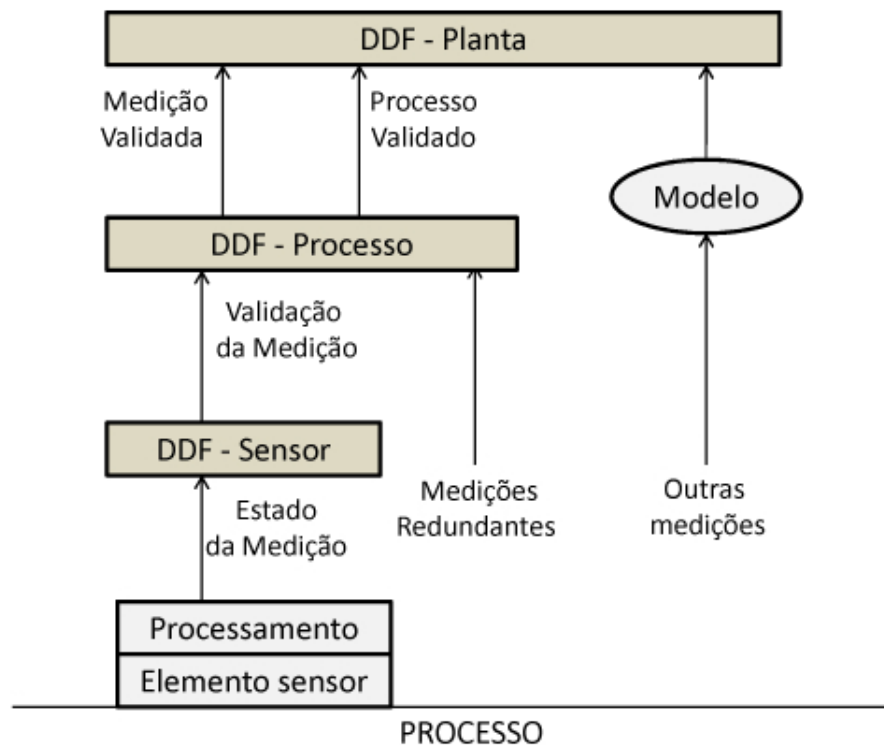


Figura 3.1: Esquema de DDF hierárquico

de incorporar *hardware* redundante.

3.2.2 Redundância Analítica

De forma semelhante à redundância física, técnicas baseadas em redundância analítica consistem na geração de um sinal redundante que seja comparado à saída real do sensor físico, mas, diferente da primeira, utiliza-se um modelo matemático do processo no qual o sensor está inserido para geração do sinal correspondente à sua medida (Gertler 1998). A diferença entre a saída real do sensor e o valor obtido através de simulação é denominada resíduo e se este resíduo ultrapassar um valor pré-estabelecido, uma falha é sinalizada.

A idéia por trás da redundância analítica, quando aplicada ao nível de sensor, é baseada no fato que as saídas de sensores medindo grandezas diferentes de um mesmo sistema dependem de um único estado dinâmico que pode ser representado por um modelo matemático próximo da realidade. A dependência entre as saídas de sensores diferentes é chamada redundância analítica ou funcional, ou seja, a medida de um sensor pode ser

inferida a partir das medidas dos outros sensores, associadas a um modelo matemático capaz de mapear estas medidas em uma medição redundante do sensor que se deseja analisar. Estes métodos, no entanto, esbarram na dificuldade em se encontrar relações funcionais entre sensores para formar a redundância analítica. Outra dificuldade prática deste método, no contexto deste trabalho é a indisponibilidade de se construir um modelo de entrada-saída para cada grandeza envolvida no processo de transporte de gás natural.

Outras desvantagens relacionadas à redundância analítica são os possíveis erros de modelagem que podem levar um sistema de detecção de falhas a gerar falsos alarmes. Os erros de modelagem podem estar associados à própria concepção do modelo matemático, que pode não contemplar todas informações inerentes ao sistema. Outro fator que leva a erros de modelagem é a construção do modelo de um sistema sob condições ideais, ou seja, não considerando a presença de ruídos ou distúrbios, por exemplo.

3.2.3 Análise de Sinais

A análise espectral do sinal de saída de sensor permite que suas características periódicas sejam detectadas. Uma ferramenta bastante utilizada na área de análise de sinais é a Transformada de Wavelet, pois é bastante eficiente na análise não-estacionária dos sinais tanto no domínio do tempo como na frequência (Zhang & Yan 2001).

Com a informação local derivada da transformada Wavelet, os instantes em que ocorre um estado anormal no sensor podem ser identificados. Uma vez que estas mudanças são detectadas, as diferenças na distribuição de energia do sinal, antes e depois das mudanças são utilizadas para diagnosticar falhas no sensor.

Se for aplicada somente à saída do sensor, não há necessidade de equipamentos extras ou 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 (Fernandes 2007).

3.2.4 Inteligência Artificial

As ferramentas de IA mais utilizadas para Detecção e Diagnóstico de Falhas em sensores são: sistemas baseados em conhecimento (ou sistemas especialistas), lógica nebulosa e redes neurais artificiais. Por ser a ferramenta utilizada neste trabalho, alguns trabalhos baseados em redes neurais serão apresentados nesta seção.

Devido a sua capacidade de aprender a partir de uma base de informações uma rede neural pode adquirir conhecimento suficiente sobre o funcionamento de um sensor de tal

forma que, na presença de falhas, seja possível identificar anomalias no mesmo. Algumas características de robustez e generalização das redes neurais (Haykin 2001*b*) fazem com que elas sejam uma ferramenta bastante utilizada para detecção e classificação de falhas mesmo sob a presença de ruídos.

Em geral, as técnicas para detecção e diagnóstico de falhas baseadas em inteligência artificial (IA) são utilizadas para a reprodução do comportamento dinâmico do sistema, a partir do qual um vetor de resíduos pode ser gerado (detecção), ou para geração de classificadores (diagnóstico).

Uma rede neural pode ser usada para reproduzir o comportamento do sistema a ser monitorado de modo a gerar um vetor de erros cujos valores indicam a presença ou ausência de falhas, ou fornecer informações a um classificador. Durante a reprodução do comportamento do sistema via a RNA, gera-se um vetor de resíduos a partir do qual será possível informar a ocorrência ou não de falhas. Técnicas de monitoramento de sensores baseadas em redes neurais são abordadas em (Wrest et al. 1997), (Xu et al. 1998) e (Hines et al. 2000), que apresentam mecanismos para calibração e detecção de falhas de sensores em plantas de geração de energia nuclear e em processos químicos envolvendo uma grande quantidade de sensores. Em (Moustapha & Selmic 2007) utiliza-se redes neurais recorrentes aplicadas à detecção de falhas em redes de sensores sem fio.

Além da sua aplicação em detecção de falhas, pode-se usar uma rede neural como um classificador, ou seja, para identificar qual o tipo de falha que está ocorrendo. 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 reproduz o seu comportamento (Simani & Fantuzzi 2000). Esta classificação pode depender de um conhecimento prévio que será passado como saída desejada para um classificador, ou pode se utilizar de mecanismos de aprendizagem não-supervisionados nos quais a própria rede neural cria agrupamentos à medida que "novidades" são apresentadas.

Devido à capacidade que as RNAs têm de aprender a partir de exemplos, não há a necessidade de se conhecer profundamente o sistema no qual um sensor está inserido. O modelo do processo é estimado através de um algoritmo de aprendizado onde os exemplos são apresentados à rede neural e seus pesos são atualizados de acordo com o erro da saída.

Além da capacidade de aprendizado, as redes neurais tem capacidade de lidar com não-linearidades e ruído. Por outro lado, há a necessidade de se ter um conjunto de dados representativos para o seu treinamento.

CAPÍTULO 4

Algoritmos Inteligentes

4.1 Introdução

Este capítulo descreve os algoritmos inteligentes implementados para a detecção e classificação de falhas em sensores envolvidos na medição de gás natural.

4.2 Preditor Neural

Como explanado anteriormente, uma rede neural pode ser usada para reproduzir o comportamento do sistema a ser monitorado de modo a gerar um vetor de erros cujos valores indicam a presença ou ausência de falhas, ou fornecer informações a um classificador. Para isso, a rede pode ser treinada com um conjunto entrada/saída que representa o sistema sob condições normais. No entanto, a entrada de um sensor é o mensurando que é desconhecido, pois os sensores são utilizados para converter grandezas que não podem ser explicitamente convertidas em parâmetros mensuráveis. Desta forma, a informação obtida do sensor é limitada à sua saída. Em vista disso, ao invés do conjunto entrada/saída, comumente usado em identificação de sistemas (Fernandes 2007), este trabalho propõe um mecanismo capaz de reproduzir o comportamento do sensor utilizando apenas o conjunto de saídas, em uma abordagem de predição.

O algoritmo de predição é baseado em uma rede neural recorrente do modelo NARX (Menezes 2006). Esta rede possui uma arquitetura neural recorrente usada originalmente para identificação entrada-saída de sistemas não-lineares. A entrada da rede NARX é formada por duas janelas deslizantes (*sliding time window*), uma que desliza sobre o sinal de entrada e outra que desliza sobre sinal de saída. Quando aplicada para predição de séries temporais, a rede NARX é projetada geralmente como um modelo autoregressivo não-linear (NAR), eliminando a janela de atraso da saída.

Para usar todo o poder computacional da rede NARX como uma arquitetura dinâmica para predição de séries temporais, a rede recorrente é definida de modo que seu regressor de entrada contenha d_u atrasos da variável observada $y(n)$, enquanto que o regressor de saída contém d_y estimativas da mesma variável, amostrados em instantes consecutivos, conforme mostra a Figura 4.1.

Um conjunto de medições coletadas pelo sensor e armazenadas em uma memória constituem a série temporal de interesse que se quer analisar. Para reproduzir de forma acurada a saída de um sensor em bom funcionamento, o treinamento da rede neural é realizado obedecendo os seguintes passos:

- O sensor em condição padrão (i.e. calibrado e em bom funcionamento) é inserido

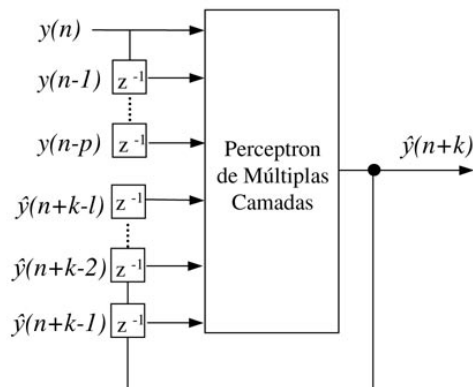


Figura 4.1: Preditor construído a partir de uma rede recorrente

em uma estação de medição de forma a coletar dados em condições de operação.

- As medições do sensor são coletadas e armazenadas durante um período de tempo. O número de amostras e a frequência de amostragem devem ser de tal forma que todas as informações contidas no sinal sejam preservadas (condições do teorema de Nyquist).
- A partir das amostras, cria-se uma janela de tempo deslizante, que formará vetores de treinamento apresentados ao longo do treinamento e teste da rede.
- As janelas de tempo são utilizadas para treinar e testar a rede neural.
- Após a fase de treinamento a rede neural tem suas respostas comparadas às saídas do sensor em função do tempo através do cálculo do erro de predição.

O algoritmo de predição proposto recebe como entrada uma memória contida em uma janela de tempo de tamanho q correspondente à quantidade de medições armazenadas até o tempo n e fornece como saída uma estimativa da medição no tempo $n + h$ onde h representa o horizonte de predição. Se $h = 1$, tem-se a predição de passo simples, enquanto $h > 1$ corresponde à predição de passos múltiplos. A arquitetura recorrente em estudo neste trabalho deve ser capaz de prover estimativas confiáveis de medição, tanto para um horizonte de predição de passo simples quanto para um horizonte de múltiplos passos. Como será visto adiante, a predição recursiva, apresenta melhor desempenho quando associada às atividades de detecção e classificação de falhas aqui apresentadas, em comparação à predição de passo simples.

Após uma série de simulações, variando a quantidade de camadas da rede e de neurônios em cada camada, a seguinte arquitetura apresentou o melhor desempenho:

- Regressor de Entrada: uma memória de linha de atraso de ordem 2.

- Regressor de Saída: uma memória de linha de atraso de ordem 3.
- Camadas Ocultas: uma camada com 3 neurônios Tangente Sigmóide.
- Camada de Saída: 1 neurônio do tipo linear puro.

A rede neural foi treinada para atuar como um preditor passos múltiplos, por esta razão, o modo de treinamento Paralelo, ou recursivo se mostrou o mais indicado para esta finalidade. No entanto, as incertezas propagadas no modo Paralelo devidas à realimentação da saída podem causar instabilidade à rede. O treinamento da rede NARX em modo Paralelo, ou recursivo, é muito mais instável que em modo Série-Paralelo quando se trabalha com identificação de sistemas (Narendra & Parthasarathy 1990). Na proposta apresentada em (Menezes 2006), a instabilidade no modo de identificação Paralelo é atenuada pelo fato dos regressores de entrada e saída serem compostos da mesma variável, ou de suas estimativas. Nas simulações realizadas neste trabalho tal atenuação não foi observada e foi necessário buscar uma alternativa para garantir a estabilidade no treinamento da arquitetura recorrente.

De forma a utilizar a rede recorrente proposta como preditor de passos múltiplos e, ao mesmo tempo, não permitir instabilidade durante o treinamento, o seguinte teste foi realizado: periodicamente o regressor de saída da rede recorrente é alimentado com um vetor de zeros, com o objetivo de evitar o aumento do erro de predição a ser propagado através da rede neural, responsável pela instabilidade durante o treinamento. Após a realização deste teste observou-se que houve redução do erro de saída a cada época do treinamento.

A Figura 4.2 ilustra como é realizado o treinamento da arquitetura selecionada. Uma janela de tempo $\bar{y}(n)$ de ordem 3 do conjunto de treinamento, é apresentada à rede de forma que sua resposta seja comparada com o valor do sinal desejado $y(n+1)$. As entradas correspondentes ao regressor de saída são iniciadas com um vetor de zeros (Figura 4.2(a)). O erro de predição é calculado e, então, os pesos da rede são atualizados, utilizando o Filtro de Kalman Estendido. No passo seguinte, uma nova janela do conjunto de treinamento $\bar{y}(n+1)$ é apresentada à memória de entrada da rede. A resposta da rede é realimentada para o regressor de saída, tal como no modo de treinamento Paralelo, e a próxima resposta obtida será comparada ao valor do sinal no instante de tempo subsequente $y(n+2)$, em que o erro é calculado e utilizado para atualizar os pesos sinápticos novamente (Figura 4.2(b)). Este processo se repete até que um número pré-definido de passos seja alcançado (Figura 4.2(c)), momento em que o regressor de saída é alimentado com um vetor de zeros e o vetor $\bar{y}(n+1)$ é novamente apresentado à rede (Figura 4.2(d)). Esta forma de treinamento permite que o erro de predição seja reduzido e, ao mesmo

tempo, a dinâmica da medição seja agregada à rede.

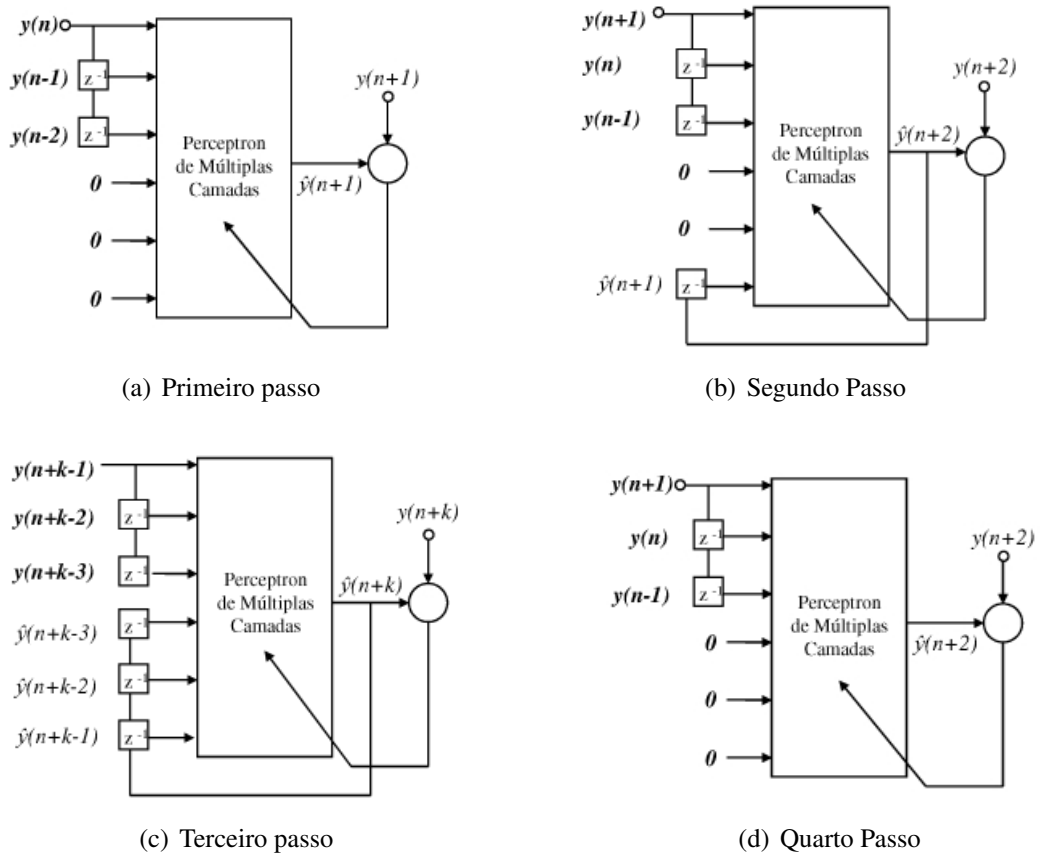


Figura 4.2: Treinamento de uma rede neural para predição k passos adiante

Observa-se que, durante os passos de treinamento, a rede neural se comporta inicialmente como um preditor de um passo, pois, há a apresentação de valores reais no sinal de temperatura. Por outro lado, as saídas da rede que são realimentadas para o regressor de saída tendem a replicar o comportamento da entrada no final do treinamento. Durante o teste, a rede neural pode assumir o papel, tanto de um preditor de passo simples, como de um preditor de passos múltiplos. O preditor de passo simples, tem as atividades recorrentes desativadas e se reduz a uma rede FTDNN (*Focused Time Delay Neural Network*), ou Rede Neural com Atraso de Tempo Focada (Menezes 2006), como mostra a Figura 4.3(a). O preditor de passos múltiplos, apresenta atividade recorrente, tanto no regressor de saída quanto no regressor de entrada, de tal modo que a rede seja alimentada com suas próprias estimativas até um horizonte de predição confiável, como exemplificado na Figura 4.3(b).

Após o processo de aprendizado, o objetivo é estabelecer um limite de confiabilidade para as estimativas realizadas pela rede neural, ou seja, deve-se garantir que o erro entre a

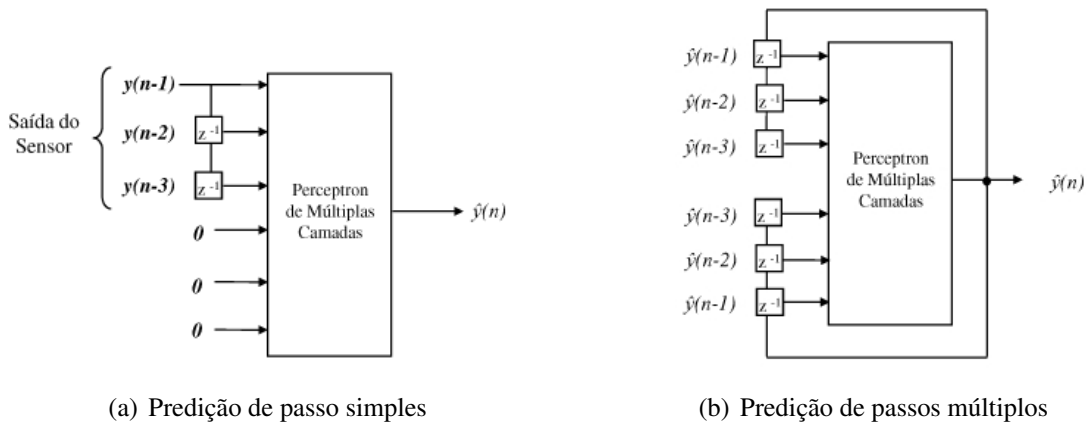


Figura 4.3: Predição de $\hat{y}(n)$

saída do sensor e a saída da rede neural permaneçam dentro de um limite aceitável. Sabe-se que, se uma medida é estimada através da predição de passo simples, a rede neural terá recebido como entradas as amostras reais do sinal. Estas observações são reais, no sentido que, são obtidas diretamente da saída do sensor e o erro de predição deverá ser pequeno para este caso. A predição de um sinal por passos múltiplos é diferente: a informação, na medida em que o horizonte de predição cresce, não provém do sensor, mas, das estimativas da própria rede neural. A confiabilidade destas estimativas tende a diminuir, à medida que o horizonte se distancia do ponto de partida da predição.

A análise do erro de predição, que corresponde à diferença entre o valor medido pelo sensor real e o valor esperado, obtido através da predição, é utilizada para a detecção da falha. Se esse resíduo ultrapassar algum limite pré-estabelecido, um alarme pode ser acionado, detectando que alguma falha está ocorrendo. O erro considerado para este trabalho é o Erro Percentual Absoluto (EPA) e sua média (EPAM) que são calculados nas equações (4.1) e (4.2), respectivamente.

$$EPA = \frac{abs(y(n) - \hat{y}(n))}{y(n)} \times 100 \quad (4.1)$$

$$EPAM = \frac{\sum_1^N \frac{abs(y(n) - \hat{y}(n))}{y(n)}}{N} \times 100 \quad (4.2)$$

4.3 Algoritmos para Classificação de Falhas

O algoritmo de predição proposto é uma ferramenta poderosa capaz de fornecer estimativas confiáveis da saída do sensor monitorado em cada instante de tempo. Através

do erro de predição, é possível detectar falhas, como citado anteriormente, caso este erro ultrapasse um limiar pré-estabelecido.

Detectar uma falha em um sensor significa indicar que o mesmo está em mal funcionamento, mas, determinar a causa do problema é uma atividade que requer uma análise mais detalhada do comportamento do sinal gerado pelo sensor. As arquiteturas de rede auto-organizadas têm a característica de associar padrões de entrada a agrupamentos, sem a necessidade de informação prévia sobre as tendências de sinal para cada padrão de falha ou sobre respostas esperadas para cada padrão. Por esta razão optou-se por investigar as potencialidades destas arquiteturas na classificação de falhas.

Dentro de cada um dos dois tipos de falha apresentados na seção 3.1, outras falhas podem ser definidas. Uma perda de conexão em um termopar, uma placa de orifício bloqueada ou um curto-circuito no amplificador operacional de um transmissor de pressão, por exemplo, podem ser considerados como falhas. Deve-se levar em consideração, no entanto, que falhas em instrumentos diferentes podem gerar um mesmo padrão de sinal na saída do sensor. A prática comum é classificar as falhas levando em consideração os padrões de comportamento da medição do sensor (Zhang & Yan 2001). Neste trabalho as falhas nas medições serão classificadas de acordo com os seguintes padrões de sinal (Teixeira 2005):

- Ausência de falha: o sensor está em funcionamento normal.
- Valor zero: o sensor passa a fornecer apenas um valor constante zero.
- Valor de Fundo de escala: o sensor passa a fornecer um valor de fundo de escala positivo, ou negativo.
- Deriva de fator de escala: o fator de escala do sensor se altera ao longo do tempo.
- Deriva de "Offset": um sinal de "offset" crescente ou decrescente é adicionado ao sinal de saída do sensor.

O erro de predição, por si só, pode ser utilizado para detectar a presença de uma falha, no entanto, não agrega informações sobre o comportamento dinâmico do sinal no momento da ocorrência desta falha. Após testes realizados foi observado que as entradas que permitem melhor exploração do classificador são as seguintes:

$$I = \begin{bmatrix} y(n) - y(n-1) \\ \hat{y}_h(n) - y(n-1) \end{bmatrix} = \begin{bmatrix} \Delta y(n) \\ \Delta \hat{y}_h(n) \end{bmatrix} \quad (4.3)$$

onde $\hat{y}_h(n)$ corresponde à estimativa de $y(n)$ obtida a partir da predição de h passos.

Desta forma, o classificador obtém duas informações: a variação da medição do sensor real e a variação calculada em função da medição prevista (Figura 4.4). A relação entre

estas variações muda de acordo com o comportamento da saída do sensor. Se a saída do sensor apresenta comportamento normal, tanto $\Delta y(n)$ como $\Delta \hat{y}_h(n)$ possuem valores semelhantes e pequenos, característicos do sinal de temperatura utilizado. De acordo com o sensor e a variável utilizados este comportamento pode mudar. Se a medição apresentar anomalias, o valor de $\Delta y(n)$ se modificará, e $\Delta \hat{y}_h(n)$ se mantém com o mesmo comportamento. Com estas informações a rede auto-organizada tem condições de interpretá-las para categorizá-las em neurônios ou *clusters* de neurônios.

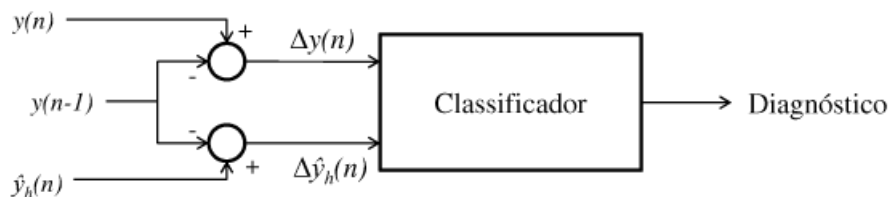


Figura 4.4: Esquema geral de diagnóstico de falhas

Duas arquiteturas são investigadas: o mapa auto-organizável e a rede *Fuzzy ART* e suas arquiteturas foram definidas empiricamente após uma série de testes. Idealmente, 5 neurônios deveriam ser suficientes para que cada um fosse capaz de representar um estado diferente, ou seja, 4 falhas e situação normal. No entanto, o sensor é um sistema dinâmico e, como será apresentado adiante, um padrão de sinal pode estar associado a neurônios diferentes em virtude da saída do sensor apresentar tendências diferentes tanto na ausência de falhas, como na presença da falha. Por exemplo, em uma falha abrupta, a tendência da medição do sensor é no primeiro momento, variar bruscamente e nos instantes que seguem, se manter constante. São dois comportamentos distintos relacionados a uma mesma falha, que dependem do instante de tempo em que a mesma é analisada. Por esta razão, um número maior de neurônios foi utilizado para abranger estas variações e, ao mesmo tempo, manter as arquiteturas de rede suficientemente simples para uma futura implementação embarcada. A quantidade de neurônios tanto para a rede SOM como para a rede *Fuzzy ART* são as mesmas, de forma a comparar o desempenho das duas arquiteturas.

A implementação da rede SOM foi realizada com o auxílio do *Neural Network Toolbox* do software MATLAB ©. A arquitetura da rede é apresentada na Figura 4.5, ilustrando um vetor de 2 entradas e um vetor de 9 saídas, em que cada uma representa um neurônio que assume o valor 1, caso seja o neurônio vencedor.

Observa-se que, não é fornecida à rede neural a classificação desejada dos tipos de

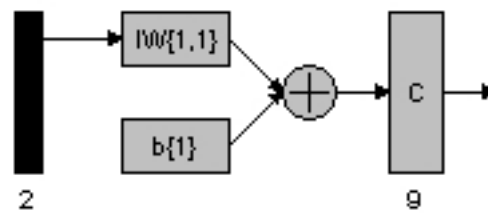


Figura 4.5: Arquitetura do mapa auto-organizável

falha. O próprio mapa auto-organizável deverá fornecer esta classificação, de acordo com os agrupamentos que o mesmo irá criar durante o processo de treinamento. Após o treinamento, é possível associar as ocorrências de falha aos agrupamentos correspondentes.

A arquitetura ART implementada, como citado anteriormente, é a rede *Fuzzy ART*. A rede recebe as mesmas entradas que o mapa auto-organizável, de modo que seus resultados sejam comparados. A implementação foi realizada através do *toolbox Fuzzy ART and Fuzzy ARTMAP Neural Networks* (Garrett 2003) com o *software MATLAB* ©.

As entradas para a rede *Fuzzy ART* foram reescaladas entre 0 e 1, como especificado no algoritmo e o complemento de código foi utilizado para calcular os complementos de cada entrada. O vetor de entradas para a rede ART é, então, composto com as entradas definidas na Equação 4.3 e seus complementos.

Diferente do mapa auto-organizável, que deve ser inicializado com um número fixo de agrupamentos possíveis, a rede ART é inicializada com apenas uma categoria e é criada para que possa ser expandida à medida que novos agrupamentos, ou categorias, sejam criados durante o treinamento até um limite de memória pré-estabelecido. A criação de novas categorias é controlada através do parâmetro de vigilância ρ , que foi ajustado de forma empírica, conforme a observação dos resultados obtidos.

CAPÍTULO 5

Resultados e Discussões

5.1 Algoritmo de Predição

A Figura 5.1, mostra a série temporal utilizada para a implementação e testes dos algoritmos. Trata-se de um sinal de temperatura, obtido em uma estação de medição de gás natural no período de um mês, com intervalo de amostragem de uma hora. Estes dados foram coletados em medições de auditoria realizadas pelo CTGÁS em pontos de medição fiscal de gás natural. Este conjunto de valores foi subdividido em conjunto de treinamento (450 amostras) e teste (217 amostras) e o Filtro de Kalman Estendido foi utilizado para o processo de aprendizado. As implementações foram feitas com a utilização do software MATLAB®.

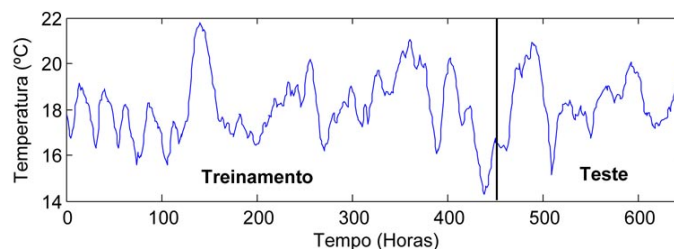


Figura 5.1: Sinal de temperatura utilizado

Após o treinamento da rede recorrente com o Filtro de Kalman Estendido o erro médio quadrático de predição ficou em torno de $0,0027 \text{ }^\circ\text{C}$. A Figura 5.2 ilustra a evolução do treinamento. Observa-se que o valor do erro convergiu com poucas épocas, vantagem que o Filtro de Kalman Estendido apresenta em relação a algoritmos baseados no gradiente descendente. O aumento no número de épocas de treinamento não apresentou melhorias significativas na performance do preditor.

A Figura 5.3 ilustra dois gráficos utilizados para analisar a predição de passo simples realizada pela rede neural, que se comporta nesse caso com uma FTDNN. O primeiro gráfico ilustra os sinais de saída do sensor e da rede NARX utilizada como preditor de passo simples. O sinal do preditor é representado pela linha vermelha, enquanto o sinal gerado pela saída do sensor de temperatura é representado pela linha azul. O segundo gráfico corresponde ao erro de predição. Calculou-se o EPA de predição para cada instante e obteve-se um erro médio de teste no valor de $0,71\%$, dentro de um intervalo de 200 horas. Isto significa que a saída estimada pela rede neural através da predição de passo simples é, em média, aproximadamente $99,3\%$ correta em relação à saída do sensor, durante este período.

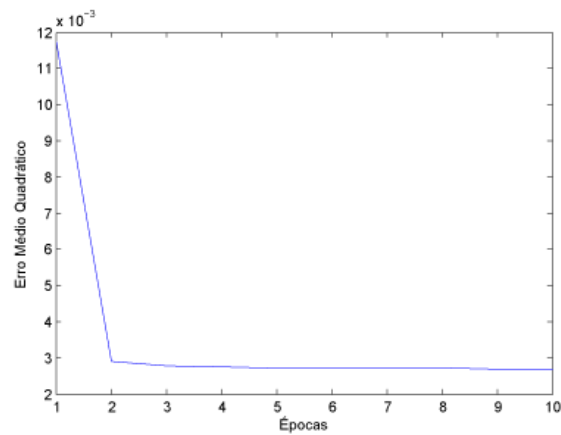


Figura 5.2: Erro médio quadrático de treinamento

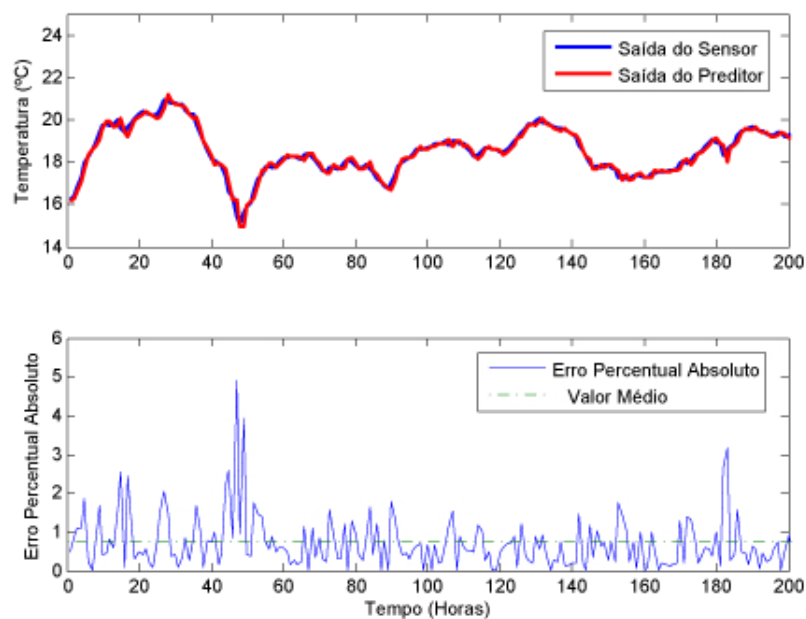


Figura 5.3: Sinal obtido pela predição de 1 passo

Para testar a robustez da rede neural frente ao ruído, foi somado ao sinal original, um sinal de ruído branco de média zero e desvio padrão $\sigma = 0.25$. A Figura 5.4 mostra a evolução do erro médio quadrático de treinamento, que chegou a $0,0116$ °C, maior que no treinamento com um sinal sem ruído, para a mesma quantidade de épocas. O aumento no número de épocas neste caso não trouxe melhorias significativas.

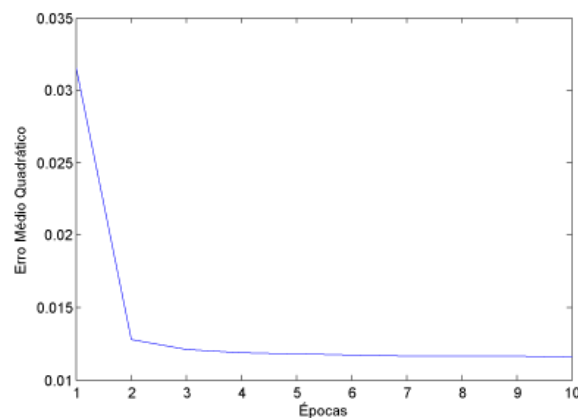


Figura 5.4: Erro médio quadrático de treinamento

A Figura 5.5 ilustra os dois gráficos utilizados para analisar a predição de passo simples realizada em um sinal ruidoso. O primeiro gráfico ilustra os sinais de saída do sensor e da rede NARX utilizada como preditor de passo simples. O sinal do preditor é representado pela linha vermelha, enquanto o sinal gerado pela saída do sensor de temperatura é representado pela linha azul. O segundo gráfico corresponde ao erro de predição. Calculou-se o erro absoluto percentual de predição para cada instante e obteve-se um erro médio de teste no valor de 1,453%, dentro de um intervalo de 200 horas. Isto significa que a saída estimada pela rede neural através da predição de passo simples é, em média, aproximadamente 98,5% correta em relação à saída do sensor, durante este período.

Na Figura 5.6 o valor médio do erro percentual absoluto (EPAM) é plotado em função do horizonte de predição, tanto para o sinal original quanto para o sinal ruidoso. Como esperado, o erro de predição é maior para o sinal que apresenta ruído e aumenta à medida que um maior horizonte é estabelecido, para os dois casos. Como pode ser observado, o erro para um horizonte $h = 10$ chega a 4,5% considerando a saída original do sensor e a 5%, considerando a presença de ruído.

O valor do erro é um parâmetro que indica o quão as medidas do sensor e do preditor se desviam, no entanto, a análise a partir do erro médio pode não ser suficiente para determinar se o sinal gerado pela rede neural tem correspondência com o sinal real. Por exemplo, no caso da predição de 10 passos adiante, um erro de 5 % pode ser considerado satisfatório, no entanto, o sinal gerado pode não estar correlacionado com o sinal do sensor. Por esta razão, foi calculada uma medida de correlação entre o sinal de saída do sensor e os sinais gerados por cada estratégia de predição (de 1 a 10 passos adiante), como mostra a Equação 5.1:

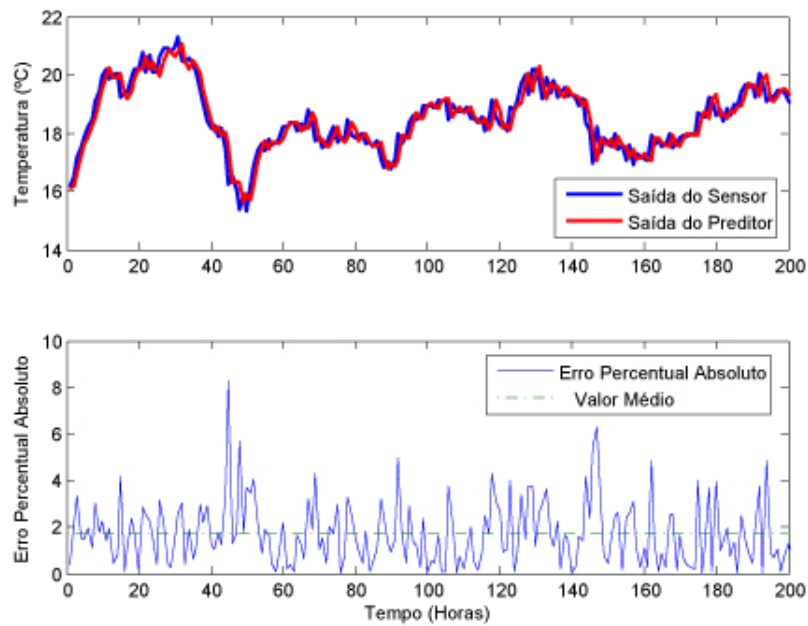


Figura 5.5: Sinal obtido pela predição de 1 passo com a presença de ruído

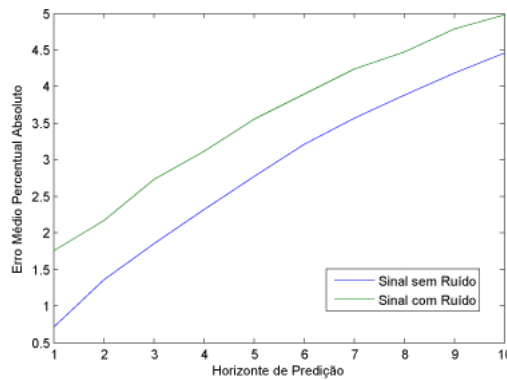


Figura 5.6: Erro médio de teste X horizonte de predição

$$R(i) = E(y(n) - \mu_y)(\hat{y}_i(n) - \mu_{\hat{y}_i})\sigma_y\sigma_{\hat{y}_i} \tag{5.1}$$

onde, $\hat{y}_i(n) = \hat{y}(j + i)$, sendo $j + i = n$, corresponde à estimativa de y no tempo n através da predição i passos adiante e medições reais do sensor obtidas até o tempo j . Os valores de μ_y e $\mu_{\hat{y}_i}$, correspondem aos valores médios dos sinais real e previsto, respectivamente enquanto σ_y e $\sigma_{\hat{y}_i}$ correspondem ao desvio-padrão de cada sinal.

A tabela 5.1 mostra, na segunda coluna, a medida de correlação entre o sinal original e os sinais estimados ($R(i)$) e, na terceira coluna, esta medida entre o sinal acrescido de ruído e o sinais estimados ($R_n(i)$). Como esperado, a correlação entre o sinal real e o previsto diminui, à medida que o horizonte de predição adotado cresce. Observa-se uma correlação forte (≥ 0.7) até a linha 7 que corresponde à correlação entre o sinal original $y(n)$ e o estimado a partir da predição 6 passos adiante $\hat{y}_6(n)$, ou seja, para horizontes de predição maiores, a correlação começa a ser considerada fraca e a confiabilidade das estimativas se torna baixa. O sinal gerado através da predição de 6 passos é ilustrado na Figura 5.7.

Observa-se que, para um sinal ruidoso a rede neural foi capaz de manter uma correlação forte das predições dentro de um horizonte de predição de 5 passos. A Figura 5.8 ilustra o sinal gerado.

Tabela 5.1: Correlações

Sinal	$R(i)$	$R_n(i)$
$y(n)$	1	1
$\hat{y}_1(n)$	0.988	0.936
$\hat{y}_2(n)$	0.960	0.899
$\hat{y}_3(n)$	0.923	0.845
$\hat{y}_4(n)$	0.872	0.783
$\hat{y}_5(n)$	0.810	0.711
$\hat{y}_6(n)$	0.747	0.642
$\hat{y}_7(n)$	0.688	0.574
$\hat{y}_8(n)$	0.626	0.508
$\hat{y}_9(n)$	0.557	0.430
$\hat{y}_{10}(n)$	0.479	0.356

5.2 Detecção e Classificação de Falhas

Os quatro tipos de falha, definidos na seção 4.3, foram simulados em um mesmo instante de tempo do sinal e observou-se o comportamento do preditor frente à apresentação da medição modificada. Seja $y_f(n)$ a medição do sensor modificada pela influência de uma falha, seus valores são estabelecidos da seguinte maneira:

- Falha 1: Valor zero - $y_f(n) = 0$.
- Falha 2: Valor de Fundo de escala - $y_f(n) = 60$.
- Falha 3: Deriva de fator de escala - $y_f(n) = y(n)e^{(-ak)}$, tal que $a = 0.06$; a variável k assume valor 1 no instante em que se inicia o estado de falha e é incrementada a

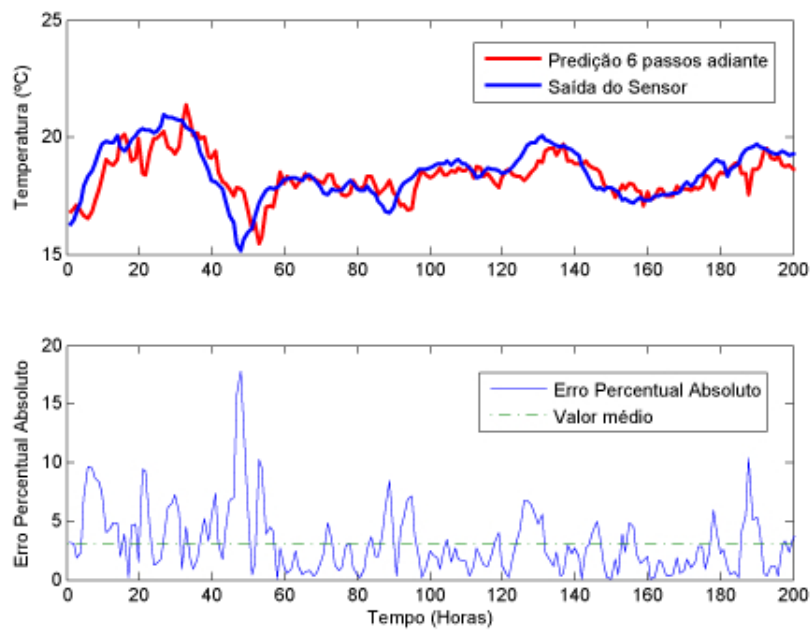


Figura 5.7: Sinal obtido em um horizonte de predição de 6 passos

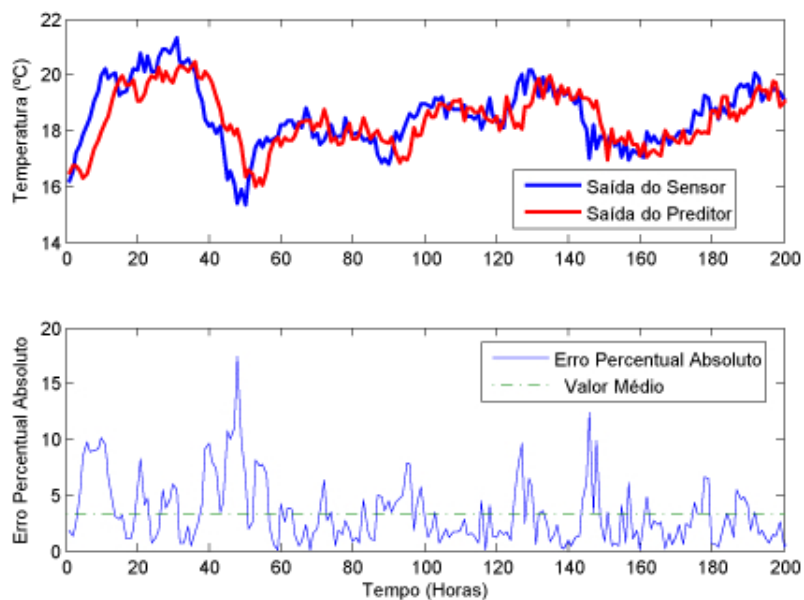


Figura 5.8: Sinal obtido em um horizonte de predição de 5 passos com a presença de ruído

cada instante de tempo subsequente.

- Falha 4: Deriva de "Offset- $y_f(n) = y(n) - ak$, tal que $a = 0.6$, correspondente a 1% do valor de Fundo de Escala; a variável k assume valor 1 no instante em que se inicia o estado de falha e é incrementada a cada instante de tempo subsequente.

Para cada falha simulada, o novo sinal obtido foi apresentado como entrada para a rede neural recorrente, previamente treinada como preditor, e suas respostas foram observadas. A arquitetura do preditor, bem como seu mecanismo de treinamento foram descritos anteriormente.

A rede neural recorrente pode assumir o papel, tanto de um preditor de passo simples, como de um preditor de passos múltiplos. Primeiramente, observou-se o comportamento do sinal de saída do preditor de passo simples frente à simulação de cada uma das quatro falhas, como ilustra a Figura 5.9. A linha preta e pontilhada representa a medição do sensor, caso o mesmo não estivesse sob efeito de falhas. A linha azul corresponde ao sinal obtido através da simulação de uma falha. A linha vermelha representa a estimativa do preditor neural a cada instante de tempo.

A rede neural em estudo, ao assumir o papel de um preditor de passo simples, desempenha a função de uma rede FTDNN. As janelas de tempo apresentadas a esta rede são compostas das medições reais do sensor de temperatura ordenadas no tempo. No primeiro instante da ocorrência de uma falha, a janela de tempo apresentada ao preditor consiste de medições livres de falhas e a predição para este instante é calculada sobre estas medições.

$$\hat{y}_1(n+1) = f(y(n), y(n-1), y(n-2)) \quad (5.2)$$

Observa-se na Figura 5.9, que a saída do preditor apresenta um valor aproximado da medição livre de falhas. No instante seguinte, a medição incorreta, afetada pela falha é agregada à janela de tempo apresentada ao preditor. A estimativa para o instante de tempo seguinte sofre a influência da falha e se distancia da medição esperada.

A mesma simulação, ilustrada na Figura 5.10 foi realizada considerando que o sinal gerado pelo preditor é resultado da predição de passos múltiplos, e o horizonte de predição é $h = 3$

Ao assumir o papel de um preditor de passos múltiplos, a rede neural se comporta de maneira diferente. As janelas de tempo apresentadas à rede, inicialmente, são compostas das medições reais do sensor de temperatura ordenadas no tempo. Com o aumento do horizonte de predição, a saída da rede neural passa a compor a janela de tempo dos regressores, tanto de entrada, como de saída. No primeiro instante da ocorrência de uma falha, a janela de tempo apresentada ao preditor, já é composta de estimativas obtidas nos

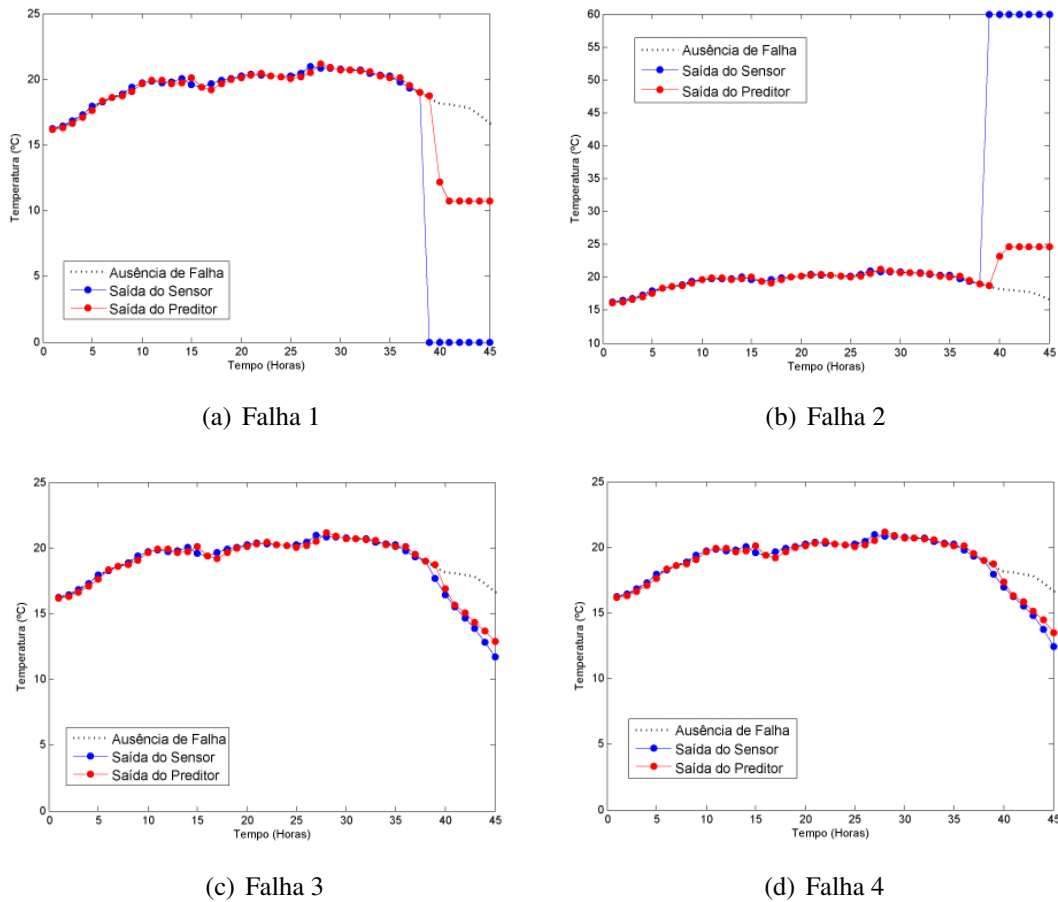


Figura 5.9: Comportamento do preditor de passo simples na presença de falhas

passos de tempo anteriores. No caso da predição de 3 passos tem-se:

$$\hat{y}_3(n+1) = f(\hat{y}_2(n), \hat{y}_1(n-1), y(n-2)) \quad (5.3)$$

O conjunto destas predições geradas ao longo do tempo formam o sinal de saída do preditor. No instante seguinte ao início da falha, a medição incorreta do sensor não é apresentada à entrada da rede. A influência da falha só se reflete na saída do preditor de passos múltiplos após alguns instantes de tempo.

A informação do sensor em bom funcionamento é utilizada para o estudo do comportamento do preditor na presença de uma falha e é conhecida, neste caso, em virtude das falhas serem simuladas. No entanto, o mecanismo de detecção de falhas terá apenas a informação do sensor, com ou sem falha, e a informação das predições, seja de passo simples ou de passos múltiplos. Em falhas abruptas a discrepância entre a saída do sensor e a saída do preditor pode ser detectada com facilidade, em ambos os esquemas de predição.

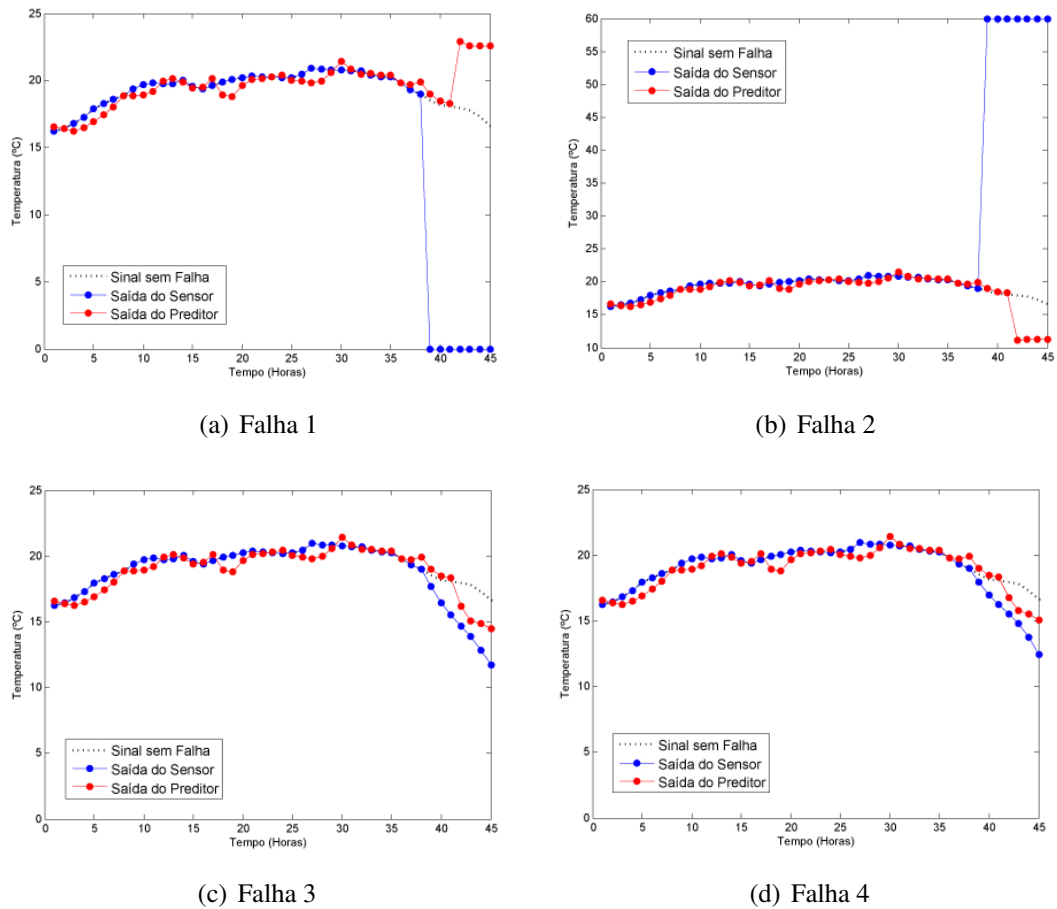


Figura 5.10: Comportamento do preditor de passos múltiplos ($h = 3$) na presença de falhas

Já no caso das falhas incipientes, as estimativas geradas através da predição de passo simples não apresentam diferenças significativas em relação às saídas do sensor. O erro gerado é pequeno, tal como o erro de predição calculado anteriormente para a predição de passo simples. No caso da predição recursiva, observa-se que o sinal gerado se aproxima do esperado para o sensor, durante os primeiros instantes de duração da falha. O erro de predição em relação à saída real do sensor será maior, se comparado ao erro calculado durante os testes do preditor, para predição de 3 passos adiante. Dessa forma é possível detectar com maior eficácia a presença de falhas incipientes no sensor, como apresentado a seguir.

5.2.1 Detecção de Falhas pela Análise do Erro

O erro entre a medição do sensor e a estimativa do preditor neural é a informação utilizada para detectar a presença de falhas no dispositivo. Caso o erro ultrapasse um

limite pré-estabelecido, um alarme deve ser acionado e ações devem ser tomadas para minimizá-lo.

Para avaliar a capacidade de detecção de falhas com o uso do erro de predição, 10 sinais com falha foram gerados a partir do conjunto de teste do sinal de temperatura. Em cada sinal, os 4 tipos de falha são simulados em instantes de tempo diferentes e escolhidos aleatoriamente, totalizando 40 falhas geradas. Cada falha é simulada uma vez no sinal e tem duração de tempo de 10 amostras. A Figura 5.11 ilustra um dos sinais gerados.

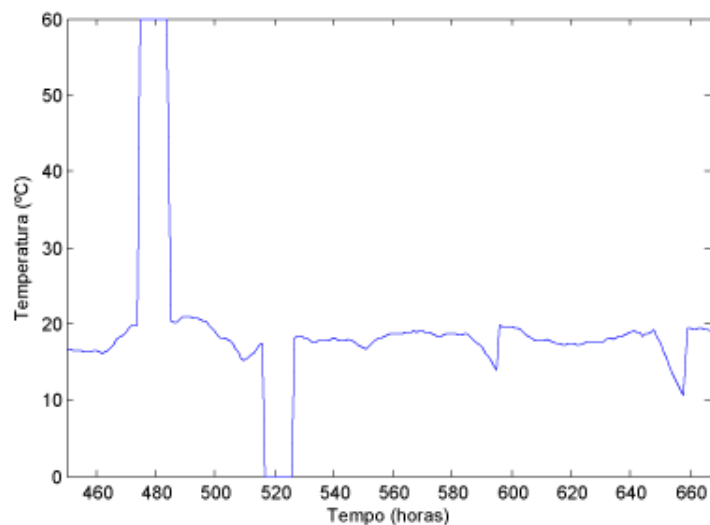


Figura 5.11: Exemplo de sinal utilizado para teste

Foi ilustrada, anteriormente, a diferença entre a resposta de um preditor de passo simples e a resposta de um preditor de passos múltiplos, quando se apresenta uma falha. Observou-se que, para falhas incipientes, a predição de passos múltiplos deverá apresentar um erro de predição maior, facilitando a detecção destas falhas. Para analisar esta situação, o sinal gerado foi apresentado aos seguintes preditores:

- Preditor 1: preditor de passo simples, $h = 1$;
- Preditor 2: preditor de passos múltiplos, $h = 2$;
- Preditor 3: preditor de passos múltiplos, $h = 3$.

O limite de erro aceitável deve ser estabelecido sob a premissa que, mesmo sem a presença de falhas existe erro na predição devido às incertezas do próprio preditor. Para cada sinal gerado, calcula-se o erro de predição entre este e a resposta de cada preditor. Após uma série de simulações foi definido que o limite de erro percentual absoluto para o qual se considera a presença de falha é $EPA > 6\%$.

Os gráficos apresentados nas Figuras 5.12, 5.13 e 5.14 ilustram a quantidade de passos de tempo necessários para que cada tipo de falha fosse detectada, com os preditores 1, 2 e 3, respectivamente. Considera-se que o instante 1 é o instante de início da falha e que a análise é feita dentro de uma janela de 10 amostras, que corresponde ao tempo de duração da falha. Para fins de melhor representação a expressão T_Janela , identificada nas figuras e tabelas a seguir, corresponde ao tamanho da janela de tempo analisada.

Como se pode observar, as falhas abruptas são detectadas automaticamente no mesmo instante em que são geradas, pois o erro de predição aumenta bruscamente. Ao se observar a Figura 5.12 conclui-se que o mesmo não ocorre para as falhas incipientes, quando se aplica a predição de passo simples. Não foi possível detectar a falha 4 - **deriva de "offset"** nos sinais 1, 4, 5 e 9, e a barra correspondente não aparece no gráfico. Nos outros casos, a quantidade de passos de tempo necessária é muito alta. Como ilustrado nas Figuras 5.9(c) e 5.9(d), o comportamento do sinal na presença de falhas incipientes pode ser confundido com a ausência de falha.

Ao se utilizar o preditor de 2 passos, as falhas que antes não foram detectadas passaram a ser detectadas, porém em um passo de tempo ainda distante do instante inicial da falha. A falha 3 - **deriva de fator de escala** apresentou melhores resultados de detecção em relação ao preditor de passo simples. Com o preditor de 3 passos houve melhoria significativa no tempo de detecção da falha do tipo **deriva de "offset"** e todas as falhas são detectadas dentro de uma janela reduzida de 3 amostras.

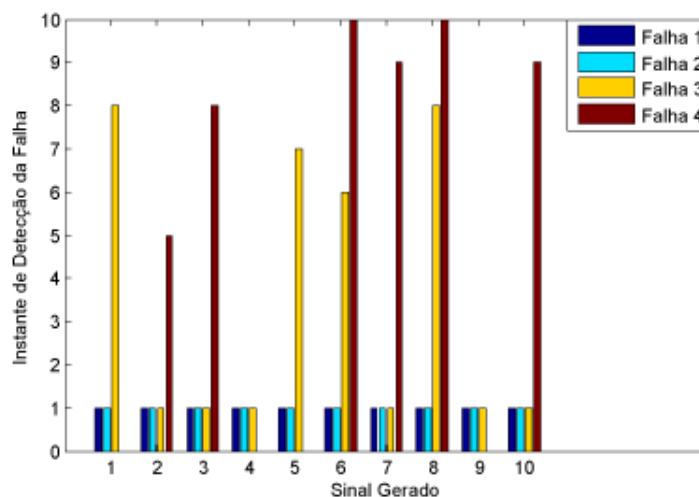


Figura 5.12: Tempo de detecção com preditor de passo simples ($T_Janela = 10$)

A análise anterior foi realizada dentro de uma janela com 10 períodos de amostragem,

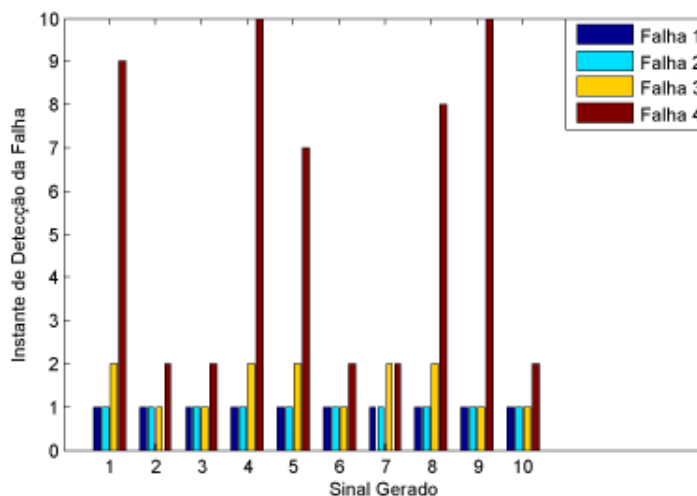


Figura 5.13: Tempo de detecção com preditor de passos múltiplos $h = 2$ ($T_{Janela} = 10$)

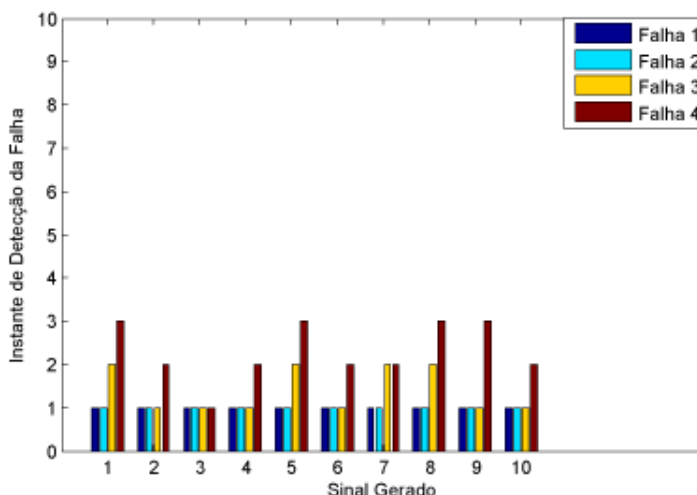


Figura 5.14: Tempo de detecção com preditor de passos múltiplos $h = 3$ ($T_{Janela} = 10$)

por ser considerado um tempo longo o suficiente para se detectar uma falha. Caso uma falha não seja detectada durante este intervalo de tempo, pode-se considerar que não será mais detectada nos instantes que seguem. Este intervalo, no entanto, é longo quando se analisa a detecção de falhas sob o ponto de vista da eficiência. Para avaliar a eficiência da atividade de detecção de falhas foi definida uma janela de tempo bem menor, de 3 amostras, na qual a primeira amostra corresponde ao instante de início da falha. A Tabela 5.2 quantifica o percentual de falhas que foram detectadas utilizando os três preditores,

dentro da janela de tempo. Como cada falha foi simulada em 10 sinais diferentes, este percentual representa em quantos sinais cada tipo de falha foi detectada.

Tabela 5.2: Percentual de Falhas Detectadas ($EPA > 6\%$, $T_{Janela} = 3$)

Tipo de Falha	Preditor 1	Preditor 2	Preditor 3
Falha 1	100%	100%	100%
Falha 2	100%	100%	100%
Falha 3	60%	100%	100%
Falha 4	0%	50%	100%

Como se pode observar, a falha **deriva de offset** não foi detectada nenhuma vez utilizando o preditor de passo simples, dentro de um intervalo de 3 amostras. Nesta situação e nas outras situações em que o percentual de detecção não atingiu 100% as falhas não seriam detectadas em tempo hábil para que uma ação corretiva ou um mecanismo de intertravamento, por exemplo, pudesse ser acionado. O preditor de 3 passos que apresentou 100% de acerto para todos os tipos de falha apresenta a vantagem de gerar sinais que, quando comparados ao sinal do sensor com falhas, permitem detectar falhas que não foram identificadas pelo preditor de passo simples e reduzir o tempo de detecção em relação ao preditor de 2 passos, além de manter o valor do erro predição dentro de um limite aceitável.

A melhoria na capacidade de detecção de falhas observada com o uso de preditores de passos múltiplos pode levar a uma maior incidência de alarmes falsos, devido ao aumento no erro de predição entre a saída real do sensor e o preditor de passos múltiplos. Foi observado durante os testes que a quantidade de alarmes gerados é pequena em relação à quantidade de amostras analisadas, como ilustra a Tabela 5.3. Além disso, observou-se que os alarmes gerados são pontuais, ou seja, diferente das falhas geradas, não apresentam duração maior que a de uma amostra.

Tabela 5.3: Percentual de alarmes falsos em relação ao conjunto de testes

Preditor	Incidência de alarmes falsos
Preditor 1	0%
Preditor 2	0,93%
Preditor 3	3,26%

5.2.2 Classificação de Falhas com Arquiteturas auto-organizadas

Da seção anterior os melhores resultados em relação à capacidade e eficiência em detecção de falhas foram obtidos a partir da aplicação do preditor de 3 passos, utilizado

para gerar o erro de predição. Após a detecção da falha, as saídas do sensor e do preditor que apresentou melhor resultado foram tratadas de forma a serem apresentadas às redes auto-organizadas para testar sua capacidade e eficiência na classificação dos tipos de falha.

Para que as redes auto-organizadas tenham a capacidade de reconhecer qualquer uma das falhas, todas as 4 falhas foram simuladas em instantes de tempo diferentes dentro do sinal de temperatura, gerando o sinal ilustrado na Figura 5.15. Este sinal é diferente dos outros 10 sinais gerados para teste e tem a finalidade de ser utilizado para treinar as redes SOM e *Fuzzy ART*.

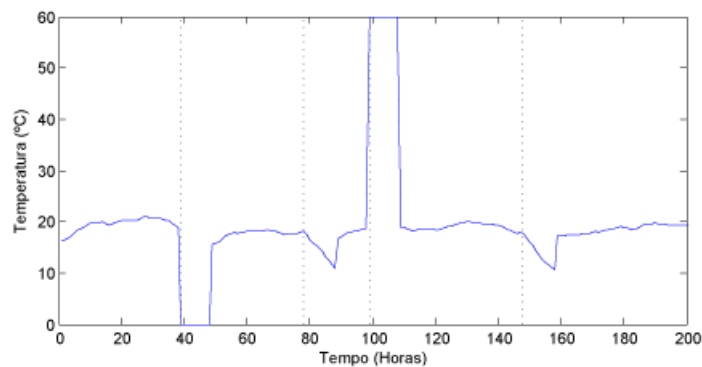


Figura 5.15: Sinal de temperatura com falhas simuladas

O sinal gerado é apresentado ao preditor de 3 passos e, então, o conjunto de entradas do classificador é criado da forma apresentada na Equação (5.4). À medida em que cada padrão de entrada é apresentado, a rede neural deve, ao longo do treinamento, aprender a associá-lo a um neurônio, ou conjunto de neurônios. Ao final do treinamento, os neurônios ativados nos instantes de tempo em que o sinal do sensor apresenta falha são detectados.

$$I = \begin{bmatrix} y(n) - y(n-1) \\ \hat{y}_3(n) - y(n-1) \end{bmatrix} = \begin{bmatrix} \Delta y(n) \\ \Delta \hat{y}_3(n) \end{bmatrix} \quad (5.4)$$

A arquitetura definida para a rede SOM é apresentada na Tabela 5.4, devido aos melhores resultados obtidos.

Como se trata de uma grade 3x3, a rede SOM pode gerar até 9 agrupamentos. Após o treinamento, foi observado o instante em que as falhas foram simuladas e o neurônio vencedor para cada um destes instantes. A Tabela 5.5 representa os neurônios predominantes para cada falha simulada. Como se pode observar, mais de um neurônio pode representar um mesmo sinal de falha, em virtude da característica dinâmica das entradas

Tabela 5.4: Parâmetros da arquitetura SOM

Topologia	Grade de neurônios 3x3
Função de distância	Distância Euclidiana
Taxa de aprendizagem	$\eta(n=0) = 0.9$
Épocas	1000

do SOM. A partir dos neurônios associados a cada tipo de falha, uma quantidade de *clusters* foi determinada e enumerada.

Tabela 5.5: *Clusters* formados pelo SOM

<i>Cluster</i>	Falha Correspondente	Neurônios
1	Falha 1 - valor zero	7 e 9
2	Falha 2 - valor fundo de escala	1 e 3
3	Falha 3 - deriva de fator de escala	8
3	Falha 4 - deriva de offset	8

Os vetores de entrada para a rede *Fuzzy ART* foram gerados da mesma forma que os vetores utilizados para o treinamento do SOM de forma a comparar o desempenho das duas arquiteturas. As entradas para esta rede devem ser reescaladas para se encontrarem no intervalo [0 1], conforme as regras do treinamento explanadas anteriormente. Além deste proceso, as entradas passam pelo processo de complemento de código, gerando o vetor de entradas a seguir:

$$I = \begin{bmatrix} \Delta y'(n) \\ \Delta \hat{y}'_3(n) \\ 1 - \Delta \hat{y}'_3(n) \\ 1 - \Delta y'(n) \end{bmatrix} \quad (5.5)$$

em que $\Delta y'(n)$ $\Delta \hat{y}'_3(n)$ correspondem às entradas reescaladas.

A arquitetura definida para o modelo *Fuzzy ART* que apresentou os melhores resultados possui os parâmetros apresentados na Tabela 5.6. A Tabela 5.7 apresenta os grupos dentro dos quais cada padrão de sinal foi enquadrado.

Tabela 5.6: Parâmetros da arquitetura *Fuzzy ART*

Parâmetro de escolha	$\alpha = 0.000001$
Parâmetro de Vigilância	$\rho = 0.98$
Taxa de aprendizagem	$\beta = 0.6$
Épocas	1000
Capacidade de Memória	9 neurônios

Tabela 5.7: *Clusters* formados pela rede *Fuzzy ART*

<i>Cluster</i>	Falha Correspondente	Neurônios
1	Falha 1 - Valor Constante zero	2 e 3
2	Falha 2 - Valor Constante fundo de escala	8 e 9
3	Falha 3 - Deriva de Fator de Escala	5
3	Falha 3 - Deriva de Offset	5

A Tabela 5.8, ilustra o percentual de cada falha que foi corretamente classificada, para a rede SOM e a rede *Fuzzy ART*. Ao se comparar os resultados, o mapa auto-organizável apresenta maior percentual de acerto em relação às falhas incipientes. Ainda assim, a rede *Fuzzy ART* tem capacidade de classificação com bons resultados.

Tabela 5.8: Percentual de Falhas Classificadas Corretamente ($T_Janela = 3$)

Tipo de Falha	SOM	<i>Fuzzy ART</i>
Falha 1	100%	100%
Falha 2	100%	100%
Falhas 3 e 4	100%	70%

As duas arquiteturas apresentadas acima se mostraram ferramentas que podem ser aplicadas à classificação de falhas no sensor. Observa-se que, para ambos os classificadores, não foi possível fazer uma distinção entre as duas falhas incipientes. Considera-se a falha simplesmente como uma falha de deriva de medição. No entanto, os resultados obtidos mostram o potencial que as arquiteturas não-supervisionadas apresentam ao distinguir padrões de comportamento de um sensor, aparentemente iguais, e classificá-los separadamente, auxiliadas pela predição de passos múltiplos fornecida através da rede recorrente. Graças ao poder de agregar dinâmica ao aprendizado, a rede recorrente é capaz de fornecer estimativas confiáveis em prazos de tempo mais longos, de tal forma que suas entradas não sejam inicialmente afetadas pela falha do sensor.

CAPÍTULO 6

Considerações Finais e Perspectivas Futuras

O gás natural é um produto que deve ser monitorado constantemente desde sua produção até o consumo, devido às diferentes responsabilidades por sua custódia, até chegar ao consumidor final. No processo de distribuição de gás natural é necessário que suas medições sejam confiáveis de forma que o produto seja aceito. Durante a atividade de transferência de custódia, as empresas devem garantir que a quantidade real de volume ou massa do gás entregue corresponda com o que é medido e compensado através dos sensores. Por esta razão, é indispensável que os sensores envolvidos no processo estejam em bom funcionamento.

Neste trabalho foram apresentados algoritmos inteligentes, investigados com o objetivo de validar localmente, no nível de campo, as medições de gás natural. A implementação destes algoritmos foi realizada considerando que cada sensor é monitorado individualmente, ou seja, a informação extraída do processo consiste apenas das medições coletadas pelos sensores durante o tempo. Para testar e validar os algoritmos propostos, um sinal de temperatura extraído de uma estação de medição de gás natural foi utilizado.

Implementou-se um preditor, baseado em rede neural recorrente, utilizado para reproduzir as saídas do sensor em bom funcionamento, recebendo como entradas suas medições presente e passadas e apresentando uma predição da próxima saída do dispositivo. Os resultados obtidos demonstraram a capacidade preditiva do algoritmo e se mostraram satisfatórios, mesmo quando o sinal analisado apresenta ruído. Com os resultados obtidos na predição de passos múltiplos pode-se concluir que o preditor neural tem capacidade de reproduzir o comportamento dinâmico do sensor ao longo do tempo. Sua aplicação ao monitoramento da qualidade das medições fornece uma base para criação de estratégias de detecção e diagnóstico de falhas.

A análise do erro de predição foi utilizada para detectar a ocorrência de uma falha no sensor. Os resultados atingidos ilustram que é possível detectar falhas abruptas de forma eficiente. No caso de falhas incipientes, ou seja, derivas de medição, concluiu-se que o uso da predição de passo múltiplos permite a detecção destas falhas com maior eficácia.

Como estratégia para classificação de falhas, foi investigada a capacidade das redes neurais não supervisionadas. Foi observado que os resultados obtidos a partir da rede *Fuzzy ART* e do mapa auto-organizável se mostraram satisfatórios e concluiu-se que estas arquiteturas têm potencial para atuar como classificadores de falhas. As duas falhas incipientes, por apresentarem características muito semelhantes, se mantiveram em um mesmo agrupamento para as duas redes neurais. Neste caso, outros mecanismos poderiam ser estudados para se classificar estas falhas com maior eficácia.

Os testes foram realizados, neste trabalho, com tipos básicos de falhas em sensores. Como sugestão para trabalhos futuros, propõe-se o estudo mais aprofundado e melhoria

dos algoritmos de classificação acima para diagnosticar outros tipos de falha.

Como citado neste trabalho, a implementação destes algoritmos de forma embarcada nos sensores permite que seu monitoramento seja efetivamente local. Como trabalho futuro, uma pesquisa voltada à implementação destes algoritmos de forma embarcada é sugerida.

O estabelecimento da arquitetura da rede, do método de treinamento, bem como do limite de erro para detecção de falhas foi realizado de forma heurística, através da observação dos resultados que melhor atendessem a aplicação para o qual este trabalho foi concebido, que é a detecção de falhas em sensores envolvidos na medição do gás natural. O uso de métodos formais para estabelecimento destes parâmetros pode prover melhores resultados em trabalhos futuros, bem como abranger esta proposta para outras aplicações.

Referências Bibliográficas

- Carpenter, G. A. & S. Grossberg (1987), 'A massively parallel architecture for a self-organizing neural pattern recognition machine', pp. 54–115.
- Carpenter, G. A., S. Grossberg & D. B. Rosen (1991), 'Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system', pp. 759–771.
- Carvalho, J. V. (2005), Modelagem temporal das medidas de vazão de drenos na Barragem de Funil (RJ) utilizando redes neurais e métodos estatísticos., Tese de doutorado, Pontífica Universidade Católica, PUC, Rio de Janeiro, RJ.
- Costa, I. M. (2006), Projeto e implementação em ambiente foundation fieldbus de filtragem estocástica baseada em análise de componentes independentes, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.
- CTGÁS (2007), '<http://www.ctgas.com.br/>'. Acessado em 21/11/2007.
- do Amaral, J. L. M. (2006), Sistemas Imunológicos Artificiais Aplicados à Detecção de Falhas, Tese de doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ.
- Fernandes, R. G. (2007), Detecção e isolamento de falhas em sistemas dinâmicos baseados em redes neurais, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.
- Garrett, A. (2003), 'Fuzzy art and fuzzy artmap neural networks'. <http://www.mathworks.com/matlabcentral/fileexchange/4306>.
- Gertler, J. J. (1998), *Fault Detection and Diagnosis in Engineering Systems*, 1ª edição, CRC Press, New York EUA.

- Gomes, D. T. (2005), Redes neurais recorrentes para previsão de séries temporais de memórias curta e longa, Dissertação de mestrado, Universidade Estadual de Campinas, UNICAMP, Campinas, SP.
- Grossberg, S. (1976), 'Adaptive pattern classifications and universal recording, I: Parallel development and coding of neural feature detectors', *Biological Cybernetics* **23**, 121–134.
- Haykin, S. (2001a), *Kalman Filtering and Neural Networks*, 1ª edição, John Wiley and Sons.
- Haykin, S. (2001b), *Redes neurais: princípios e Prática*, 2ª edição, Bookman.
- Hines, J. W., A. V. Gribok, I. Attieh & R. E. Uhrig (2000), 'Improved methods for on-line sensor calibration verification', *VIII International Conference on Nuclear Engineering*.
- Júnior, J. M. P. Menezes & G. A. Barreto (2006), A new look at nonlinear time series prediction with narx recurrent neural network, em 'Proceedings of SBRN'06', IEEE.
- Kalman, R. E. (1960), A new approach to linear filtering and prediction problems, em 'Transactions of the ASME, Journal of Basic Engineering', pp. 35–45.
- Kohonen, T. (1990), The self-organizing map, em 'Proceedings of the Institute of Electrical and Eletronics Engineers', pp. 1464–1480.
- Machado, V. P., L. Ramalho, A. D. Dória, J. D. de Melo, B. Costa & J. Medeiros (2008), A simulated environment for a multi-agent architecture based in a foundation fieldbus network function blocks, em 'IEEE International Conference on Distributed Human-Machine Systems'.
- Mandic, D. P. & J. A. Chambers (2001), *Recurrent Neural Networks For Prediction*, 1ª edição, John Wiley & Sons, San Francisco, EUA.
- Menezes, J. M. P. (2006), Redes neurais dinâmicas para predição e modelagem não-linear de séries temporais, Dissertação de mestrado, Universidade Federal do Ceará, Fortaleza, CE.
- Moustapha, A. I. & R. R. Selmic (2007), Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection, em 'Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control', pp. 313–318.

- Mueller, A. (1996), Uma aplicação de redes neurais artificiais na previsão do mercado acionário, Dissertação de mestrado, Universidade Federal de Santa Catarina, UFSC, Florianópolis, SC.
- Narendra, K. S. & K. Parthasarathy (1990), Identification and control of dynamical systems using neural networks, *em* 'IEEE Transactions on Neural Networks', pp. 4–27.
- Pereira, D. R. V. (2006), Algoritmos inteligentes para auto-calibração, auto-correção e auto-validação de medidas em sensores, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.
- Pereira, D. R. V., E. G. Júnior, R. Wendell, A. D. D. Neto & J. D. Melo (2005), 'Auto-compensação de sensores *foundation fieldbus* utilizando redes neurais recorrentes através de blocos funcionais', *VII Congresso Brasileiro de Redes Neurais* .
- Pinto, J. O. P., A. M. A. C. Pinto, G. S. Tatibana, R. O. Sanches, B. N. Carrasco, N. Flora & J. W. Hines (2005), 'Uma nova estratégia de modelagem para monitoramento de instrumentos e verificação de calibração usando redes neurais artificiais', *III Congresso Brasileiro de P&D em Petróleo e Gás* .
- Principe, J. C. & J. Kuo (1995), 'Dynamic modelling of chaotic time series with neural networks', *Advances in Neural Information Processing Systems 7* pp. 311–318.
- Sachenko, A., V. Kochan, V. Turchenko, V. Golovko, J. Savitsky, A. Dunets & T. Laopoulos (2000), 'Sensor errors prediction using neural networks'.
- Silva, D. R. C. (2005), Redes neurais artificiais no ambiente de redes industriais *foundation fieldbus* usando blocos funcionais padrões, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.
- Simani, S. & C. Fantuzzi (2000), 'Fault diagnosis in power plant using neural networks', *Information Sciences* pp. 125–136.
- Teixeira, A. J. (2005), Detecção identificação e reconfiguração de falhas múltiplas em sensores de sistemas lineares invariantes no tempo, Tese de doutorado, Instituto Nacional de Pesquisas Espaciais, INPE.
- Wrest, D. J., J. W. Hines & R. E. Uhrig (1997), Sensor calibration and monitoring using autoassociative neural networks, *em* 'proceedings of XI ENFIR International Conference on Nuclear Energy', Rio de Janeiro-RJ.

- Xu, X., J. W. Hines & R. E. Uhrig (1998), 'On-line sensor calibration monitoring and fault detection for chemical processes', *Maintenance and Reliability Conference (MARCON 98)* .
- Yung, S.K. (1993), 'Local validation of sensor signals', *Fault Diagnosis and Control System Reconfiguration, IEEE Colloquium on* pp. 7/1–7/8.
- Zhang, H. Q. & Y. Yan (2001), 'A wavelet-based approach to abrupt fault detection and diagnosis of sensors', *Instrumentation and Measurement, IEEE Transactions on* **50**(5), 1389–1396.

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](#)