



ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia Mecânica

**APLICAÇÃO DE REDES NEURONAIIS
ARTIFICIAIS À DETEÇÃO E ISOLAMENTO DE
FALHAS EM PROCESSOS INDUSTRIAIS**

VANUSA TAVARES CORTEZ

(Licenciada em Engenharia e Gestão Industrial)

Trabalho Final de Mestrado para obtenção do grau de Mestre
em Engenharia Mecânica

Orientador: Doutor João Manuel Ferreira Calado

Júri:

Presidente: Doutor Rui Pedro Chedas Sampaio

Vogais:

Doutor Pedro Miguel Abreu Silva

Doutor João Manuel Ferreira Calado

Janeiro 2015

RESUMO

Os processos industriais são cada vez mais tecnologicamente complexos. Os requisitos de competitividade impostos pelo mercado obrigam as condições de eficiência e operacionalidade próximas dos limites máximos da sua capacidade produtiva. Por outro lado, as paragens intempestivas dos mesmos resultam normalmente em perdas de produção, degradação da qualidade do produto final, danificação dos equipamentos, podendo dar origem a catástrofes ambientais com eventuais perdas de vidas humanas.

Estas são principais razões que motivaram a realização deste trabalho e conduziu a esta dissertação. Tem como objetivo, o estudo aplicabilidade de Redes Neurais Artificiais (RNAs), na deteção e isolamento de falhas que ocorrem em atuadores, utilizados no controlo automático de processos industriais, em tempo real.

Permitem que o processo continue a operar de forma controlada, mesmo depois de alguma falha se ter manifestado em algum dos seus atuadores, ou eventualmente, efetuar uma paragem controlada do processo se tal situação ocorrer e não houver capacidade de tolerância á falha em análise.

Neste estudo, foi utilizado um reator continuamente agitado (RCA), com nível e temperatura controlados no seu interior, através de duas malhas de controlo e reciclagem parcial do produto final é através de um permutador de calor. Foram simuladas falhas abruptas e incipientes em duas válvulas de regulação, sendo utilizada uma RNA do tipo feedforward, para deteção e isolamento, em tempo real, das falhas simuladas no processo acima mencionado.

A RNA foi treinada offline apenas com sintomas de falhas abruptas.

O RCA foi simulado com o seu modelo dinâmico utilizando a plataforma computacional MatLab/Simulink, sendo a RNA implementada e treinada com a toolbox de Redes Neurais do MatLab. O sucesso dos resultados obtidos com os trabalhos de simulação permite perspetivar a utilização de RNAs em situações reais, envolvendo processos industriais de elevada complexidade tecnológica sobretudo, quando é desejável uma atempada deteção e isolamento de falhas, antes de ocorrerem paragens não programadas do processo sobre supervisão.

Palavras-Chave: Deteção de falhas, Isolamento de falhas, Redes neuronais artificiais, Reator continuamente agitado.

ABSTRACT

Industrial processes are increasingly technologically complex. The competitiveness requirements imposed by market conditions require efficiency and operation close to the limits of its production capacity. On the other hand, unscheduled stoppages of them usually result in yield losses, degradation of quality of the final product, damage to equipment, may lead to environmental disasters with possible loss of life.

The above reasons motivated the work leading to the writing of this dissertation, which aimed to study of the applicability of Artificial Neural Networks (ANN) for detection and isolation of faults occurring in actuators used in automated industrial process control, in real-time, allowing the process to continue to operate in a controlled manner even after a failure has manifested itself in some of its actuators, or possibly make a controlled process shutdown if such a situation arises and there is no ability for fault tolerance.

In this study, a continuous stirred reactor (CSTR) was used, having controlled level and temperature inside through two control loops and partial recycling of the final product through a heat exchanger. Abrupt and incipient faults were considered in two control valves and a feedforward ANN has been used to detect and isolate faults in real time that have been simulated in the process mentioned above. The ANN was trained offline with only symptoms of abrupt failures.

The CSTR was simulated through its dynamic model using the computational platform MatLab / Simulink and the ANN was implemented and trained using the Neural Network Toolbox of MatLab. The successful results obtained with the simulation works performed allows to perspective the use of ANNs in real situations involving industrial processes of high technological complexity where early fault detection and isolation is desirable before unscheduled process shutdowns.

Keywords: Fault detection, Fault isolation, Artificial neural networks, Continuous stirred reactor.

*Quem morar no lugar secreto do Altíssimo
Procurará para si pouso sob a própria sombra
do Todo-Poderoso.*

(Salmos 91:1)

*Ao companheirismo e compreensão do
meu amado esposo, Faustino Cortez, e a
minha querida filha Katisa Tavares Cortez,
dedico com muito amor e carinho este
trabalho*

AGRADECIMENTOS

Agradeço a Jeová, pelo amor incondicional, proteção e providência em todos os momentos da minha vida em especial nos momentos difíceis.

Expresso minha imensa gratidão ao meu orientador, Doutor João Manuel Ferreira Calado, que incansavelmente, sempre teve disponibilidade e paciência, sempre me deu orientação necessária para o desenvolvimento deste trabalho.

Aos meus pais, Cesaltina Maria Tavares e Jesuíno Monteiro Tavares, que sempre lutaram com muito amor e dedicação para oferecer as condições necessárias para que os filhos tenham estudos.

Aos meus irmãos e sobrinhos pelos apoios, em especial a minha irmã Neusa Tavares que sempre esteve do meu lado ao longo do meu percurso acadêmico.

Aos professores Paulo Pinto, João Nogueira e Paulo Campos, pela forma explícita de ensinar.

Aos professores Pedro Fonte, Pedro Silva e Fenando Carreira pelas suas disponibilidades.

Aos meus colegas, em especial Renato Silva, José Reis e Adilson André pelo imenso apoio ao longo do percurso acadêmico.

A minha querida Prima e amiga, Jaqueline Rodrigues, pelo seu companheirismo e a família Mascarenhas pelas suas fraternidades.

NOMENCLATURA

<i>DIF</i>	Deteção e Isolamento de Falhas
<i>EMQ</i>	Erro Médio Quadrático
<i>FD</i>	Fault Detection
<i>FDI</i>	Fault Detection and Isolation
<i>FI</i>	Fault Isolation
<i>I. A</i>	Inteligência Artificial
<i>RNAs</i>	Redes Neurais Artificiais
<i>RCA</i>	Reator Continuamente Agitado
<i>PID</i>	Controlador Proporcional Integrativo Derivativo
<i>t</i>	Amostra no Instante do Tempo

ÍNDICE

1 INTRODUÇÃO	2
1.2 Motivação	5
1.3 Conceitos básicos de avarias e falhas	6
1.4 Detecção e isolamento de falhas	7
1.5 Arquitetura de um sistema de deteção e isolamento de falhas	8
1.6 Métodos de deteção e diagnóstico de falhas.....	11
1.7 Conclusão	14
2 REDES NEURONAIS ARTIFICIAIS	16
2.1 Motivação para RNAs: neurónio biológico.....	18
2.2 Modelo da RNA.....	21
2.3 Classificação das redes neuronais artificiais	23
2.4 Algoritmo retro propagação de erro (do inglês backpropagation)	29
2.5 Construção da rede neuronal	34
2.6 Conclusão	40
3 CASO DE ESTUDO.....	42
3.1 Reator continuamente agitado	42
3.2 Controlo das variáveis do processo	49
3.3 Conclusão	53
4. APLICAÇÃO DE RNA À DETEÇÃO E ISOLAMENTO DE FALHA.	55
4.1 Arquitetura da Rede Neuronal Artificial	55
4.2 Aprendizagem da Rede Neuronal Artificial	56
4.3 Detecção e isolamento de falhas	60
4.4 Conclusão	67
CONCLUSÃO E TRABALHO FUTURO	68
RERRENCIA BIBLIOGRÁFICA	71
ANEXO A.....	75
ANEXO B.....	77
ANEXO C.....	79
ANEXO D.....	83

ÍNDICE DE FIGURAS

Figura 1. 1 Central nuclear de Chernobil	5
Figura 1. 2 Tipos de falhas	6
Figura 1. 3 Sistema de deteção e diagnóstico de falhas	8
Figura 1. 4 Estrutura de isolamento de falhas baseada em modelo.....	9
Figura 1. 5 Estrutura de geração de resíduo	11
Figura 1. 6 Métodos de deteção de falhas	12
Figura 1. 7 Métodos de diagnóstico de falhas	13
Figura 2. 1 Estrutura do neurónio biológico	19
Figura 2. 2 Modelo do neurónio artificial	21
Figura 2. 3 Tipos de função de ativação.....	22
Figura 2. 4 RNA de camada única	23
Figura 2. 5 RNA Multicamadas	24
Figura 2. 6 RNAs recorrentes.....	25
Figura 2. 7 Aprendizagem supervisionada	27
Figura 2. 8 Aprendizagem não supervisionada	27
Figura 2. 9 Regras e paradigmas de aprendizagem de uma RNA.....	29
Figura 2. 10 Fluxo de processamento do algoritmo retro	30
Figura 2. 11 Exemplo dos conjuntos de padrões para aprendizagem	35
Figura 2. 12 Caracterização dos conjuntos de padrões a utilizar na aprendizagem e teste.....	36
Figura 2. 13 Característica teórica do erro médio quadrático (EMQ) do treino e validação	39
Figura 3. 1 Reator continuamente agitado	43
Figura 3. 2 Diagrama elementar do RCA.....	44
Figura 3. 3 Diagrama de bloco correspondente ao controlo do nível	49
Figura 3. 4 Diagrama de bloco correspondente ao controlo da temperatura.....	50
Figura 3. 5 Curva de resposta em forma de S	51
Figura 3. 6 Oscilação sustentada com período crítico P_{cr}	52
Figura 4. 1 Neural network fitting tool – definir dados	56
Figura 4. 2 Neural network fitting tool – validação e teste.....	57
Figura 4. 3 Neural network fitting tool – arquitetura da rede	57
Figura 4. 4 Neural network fitting tool — treino da rede	58
Figura 4. 5 Neural network fitting tool — característica da rede ao longo do treino.....	59
Figura 4. 6 Neural network fitting tool – desempenho da rede.....	60

Figura 4. 7 Dados de entrada na RNA	61
Figura 4. 8 Detecção e isolamento da falha/normal funcionamento do processo.....	62
Figura 4. 9 Detecção e isolamento da falha abrupta A.	62
Figura 4. 10 Detecção e isolamento da falha abrupta B.....	63
Figura 4. 11 Detecção e isolamento da falha abrupta C.....	63
Figura 4. 12 Detecção e isolamento da falha abrupta	64
Figura 4. 13 Detecção e isolamento da falha/Mudança de Referência.	64
Figura 4. 14 Detecção e isolamento da falha incipiente A.....	65
Figura 4. 15 Detecção e isolamento da falha incipiente B.....	66
Figura 4. 16 Detecção e isolamento da falha incipiente D.....	66
Figura 4. 17 Detecção e isolamento da falha incipiente C.....	67

ÍNDICE DE TABELAS

Tabela 3.1 Regra de sintonia de Ziegler-Nichos (primeiro método).....	51
Tabela 3.2 Regra de sintonia de Ziegler-Nichols baseada no ganho crítico e no período crítico	52
Tabela 3.3 Parâmetros do controlador PID	52
Tabela 4.1 Codificação da saída da RNA	61

Algumas definições

Reator – Entende-se por reator conjunto de equipamentos onde ocorrem reações químicas, isto é, onde um conjunto de compostos químicos é transformado, através de uma sequência mais ou menos complexas de reações, noutro conjunto de composto. O reator químico constitui o “coração” das unidades de fabricação de produtos químicos, encontrando-se quer em refinarias, complexos metalúrgicos ou nas mais variadas fábricas de produto químicos.

Nestas unidades industriais, o reator está geralmente rodeado por outros equipamentos colocados a montante e a jusante, destinados a tratamentos físicos das matérias-primas e dos produtos da reação.

Reator continuamente agitado – A mistura reacional é alimentada continuamente, e os produtos de reação são removidos continuamente.

Reator homogéneo – Dentro do reator encontra-se uma única fase.

Variável controlada – É a grandeza ou condição que é medida e controlada.

Variável manipulada – É a grandeza ou condição modificada pelo controlador de modo que afete a variável controlada.

Processo a controlar – Pode ser parte do equipamento ou apenas um conjunto de equipamentos que funcione de maneira integrada, com o objetivo de realizar determinada operação.

Processo – Operação contínua progressiva que consiste em uma série de ações controlados, sistematicamente destinados a atingir determinados fins ou resultados.

Perturbação – É um sinal que tende a afetar de maneira adversa o valor da variável de saída de um processo.

Controlo com realimentação – Controlo com realimentação ou de malha fechada refere-se a uma operação que, na presença de perturbações, tende a diminuir a diferença entre a saída de um processo e alguma entrada de referência e atua com base nessa diferença.

Estado – O estado de um processo dinâmico é o menor conjunto de variáveis (chamadas variáveis de estado) tais que o conhecimento dessas variáveis em $t = t_0$, juntamente com o conhecimento da entrada para $t \geq t_0$, determina completamente o comportamento do processo para qualquer instante $t \geq t_0$.

Espaço de estados – O espaço n-dimensional, cujos eixos coordenados são formados pelos eixos de $x_{e1}, x_{e2}, \dots, x_{en}$ onde $x_{e1}, x_{e2}, \dots, x_{en}$ são variáveis de estado, é chamado de espaço estados.

Diagrama de blocos – O diagrama de blocos de um processo é a representação gráfica das funções desempenhadas por cada componente e o fluxo de sinais entre eles. No diagrama de bloco, todas as variáveis do processo são ligadas umas as outras por meio de blocos funcionais.

Ponto de ramificação – É o ponto em que o sinal que vem de um bloco avança simultaneamente em direção a outros blocos ou somadores.

Controlador automático – Um controlador automático compara o valor real de saída da planta com entrada de referência, determina o desvio e produz o sinal de controle, que vai reduzir o desvio a zero ou um valor pequeno.

Ação de controle – É maneira pela qual o controlador automático produz o sinal de controle.

Resíduo – Também conhecido como desvio, corresponde a diferença entre o valor medido no processo real e o valor esperado no processo simulado.

Proteção do processo – Mecanismo que visa suprimir comportamento potencialmente perigoso do processo ou evitar consequências desse comportamento.

Falhas aditivas – São falhas que causam uma mudança na saída do processo através da adição das mesmas.

Falhas multiplicativas – São falhas que causam mudanças em parâmetros do processo. Essas falhas causam mudanças na saída que dependem também da magnitude de entradas conhecidas.

Isolabilidade – Capacidade em distinguir diferentes tipos de falhas.

Fiabilidade – Capacidade do processo, quando necessário, manter o correto funcionamento.

Disponibilidade – É a capacidade de um processo executar qualquer tarefa quando necessário.

Segurança – Característica que qualifica a capacidade de evitar acidente catastrófica que pode envolver perdas de vidas, danos ambientais ou custos excessivos de manutenção.

CAPÍTULO 1

SISTEMA DE DETECÇÃO E ISOLAMENTO DE FALHAS

Resumo

Neste capítulo é apresentada a definição de alguns tipos de falhas e efetuada uma breve descrição dos sistemas de deteção e isolamento de falhas, sendo apresentados alguns acontecimentos históricos e a importância da evolução dos sistemas de deteção e isolamento de falhas.

1 INTRODUÇÃO

Desde os anos 60 que a influência da automação industrial e controlo automático no funcionamento dos processos industriais tem vindo a aumentar progressivamente, dando origem a sistemas cada vez mais autónomas mas também mais sofisticados e complexos.

Este desenvolvimento na automação e controlo de processos industriais é resposta a crescente competitividade imposta pelo mercado, exigindo também um crescente desempenho dos processos com reduzidos tempos de paragem e uma produção muito próxima dos limites para que a instalação foi concebida e exigindo também uma elevada, qualidade dos produtos, que requer uma grande independência do desempenho dos sistemas relativamente a intervenção humana.

O grau de automação e controlo de processos industriais desenvolveu-se exponencialmente a partir de meados dos anos 70 com o aparecimento de computadores com elevada capacidade de cálculo e armazenamento de informação, para a época, capazes de implementar algoritmos de automação e controlo automático. Ao mesmo tempo, a par com os grandes desenvolvimentos no domínio do conhecimento da tecnologia eletrónica, houve grandes progressos tecnológicos nas áreas de sensores, atuadores e sistemas de comunicação.

Associada a um elevado aumento da produção industrial bem como maior competitividade mundial, é exigida maior segurança e fiabilidade nos sistemas dinâmicos que constituem os processos industriais.

Nos últimos anos os sistemas de deteção e isolamento de falhas têm sido aplicados com sucesso em muitas áreas tais como: indústria química, indústria petrolífera, robótica, aeronaves entre outros.

Os sistemas de deteção e isolamento de falhas, hoje em dia, são uma necessidade estratégica para o crescimento económico, e sobretudo para evitar acidentes que muitas vezes tem grandes impactos a nível ambiental económico e perdas de vidas humanas. Gertler [13] efetua alguns sumários das metodologias que têm vindo a ser aplicadas na implementação de sistemas de deteção e isolamento de falhas em processos industriais.

No mesmo artigo é efetuado uma abordagem às metodologias baseadas em modelos qualitativos originários do domínio do conhecimento designado por inteligência artificial (IA).

Os métodos de detecção e diagnósticos de falhas (identificação de falhas e isolamento de falhas) em sistemas dinâmicos podem ser basicamente divididos em métodos quantitativos e métodos qualitativos [28,29,30]. Os métodos quantitativos geralmente empregam observadores de estados ou estimadores de parâmetros para geração de resíduos, exigindo conhecimento do modelo matemático do sistema dinâmico [18].

Por outro lado, os métodos qualitativos geralmente empregam técnicas tais como redes neuronais artificiais ou lógica fuzzy que não necessitam do modelo matemático do sistema dinâmico [26].

As falhas podem causar deterioração da performance do sistema e levar o processo a uma situação de risco.

De acordo com [17] a falha é um desvio não permitido do valor de pelo menos um parâmetro do processo das condições aceitáveis padrão. São diversas as origens das falhas: humanas, sensores, atuadores etc. Podem resultar acidentes com consequências, graves ou muito graves, a curto prazo ou a médio e longo prazo, daí a importância de técnicas de detecção e isolamento de falhas, essas técnicas permitem maior segurança, proteção do ambiente e do equipamento.

Se as falhas forem detetadas na sua fase incipiente contribuirão na manutenção de unidades fabris, podendo originar melhoria de índice de segurança, diminuir os custos dos tempos de paragem etc. ou seja, quanto mais cedo as falhas forem identificadas, mais cedo será possível evitar qualquer consequência, alterando os parâmetros de operação, configurar os controladores ou programar as operações de manutenção adequadas.

A detecção e isolamento de falhas em processos dinâmicos têm sido tema de vários trabalhos nos últimos anos, dada a importância desses tópicos em aplicações reais, especialmente na área industrial.

Na indústria transformadora um sistema de detecção e isolamento de falhas eficaz, ou seja, que tenha a capacidade de detecção e isolamento de falhas numa fase incipiente do seu desenvolvimento, permite diminuir perdas económicas, aumentar a segurança e diminuir os impactos ambientais [18].

A proposta deste trabalho é desenvolver um sistema de detecção e isolamento de falhas baseado numa rede neuronal artificial (RNA) capaz de detetar e isolar as falhas em tempo real

num processo industrial, sendo usado como caso de estudo um reator continuamente agitado. (RCA)

A solução de problemas através de RNA é bastante atrativa, já que a forma como estes são representados internamente pela rede e o paralelismo natural inerente à arquitetura criam possibilidade de um desempenho superior ao dos modelos convencionais. Nas RNAs o procedimento usual na resolução de problemas passa inicialmente por uma fase de aprendizagem, em que o conjunto de exemplos é apresentado à rede, a qual extrai automaticamente as características necessárias para representar a informação fornecida [7].

No dia 25 de Abril de 1986 ocorreu a grande catástrofe que ficou na História da humanidade, conhecida como a explosão do reator nuclear de Chernobyl (ver figura 1.1), com consequências ainda bem patentes na atualidade. Enquanto alguns especialistas defendem a tese de que o motivo do acidente esteve relacionado com causas externas, nomeadamente, má condução do processo por parte dos operadores, outros defendem que o acidente ocorreu única e exclusivamente devido as causas internas, nomeadamente, ao nível do projeto. Por outro lado, outros especialistas defendem que o que aconteceu foi o resultado da conjugação das duas teses.

É de salientar que nos dias de hoje a gravidade de certos acidentes pode ser mitigada, ou certos acidentes podem mesmo ser impedidos de acontecer, devido à evolução dos sistemas de deteção e isolamento de falhas em processos industriais que permitem a deteção de falhas numa fase incipiente do seu desenvolvimento e o seu isolamento em tempo real.

O caso mais recente de acidente nuclear deu-se em Fukushima, no dia 11 de Março de 2011. Não foi possível evitar o acidente devido a ocorrência de um tsunami, mas a gravidade do acidente foi substancialmente reduzida devido ao funcionamento dos sistemas de controlo tolerante a falhas.

São estes tipos de situações e outras semelhantes que têm justificado os investimentos em investigação e desenvolvimento no domínio do conhecimento em apreço, visando o desenvolvimento de metodologias de deteção e isolamento de falhas e de metodologia de controlo tolerantes a falhas com capacidade de execução em tempo real e capazes de lidar com vários tipos de falhas que podem ocorrer num processo industrial, como por exemplo falhas em sensores, falhas em atuadores, falhas nas redes de comunicação, etc.



Figura 1. 1 Central nuclear de Chernobil [30]

1.2 Motivação

Na indústria química a qualidade do produto é assegurada através da monitorização das variáveis do processo e pelo controlo das suas variações no intervalo de tempo aceitável.

Quando as condições de operação excedem os limites estabelecidos, a qualidade do produto estará em causa e, se as correspondentes variações dos valores das variáveis de processo não forem atempadamente corrigidas causará eventos catastróficos tais como explosão, chamas ou emissão de gases tóxicos. A tarefa do operador em caso de falha ou falhas no processo é diagnosticar rapidamente a causa ou as causas para que se possa adotar com precisão medidas corretivas (a tempo).

Contudo, a análise do funcionamento incorreto do processo é uma tarefa difícil para os operadores. Em certas circunstâncias poderão ter dificuldade em lidar com eventos não previstos ou com baixa probabilidade de ocorrerem. Deste modo, surgiu a motivação em seleccionar um processo dinâmico, um reator continuamente agitado (RCA) como caso de estudo para testar a rede neuronal artificial na deteção e isolamento de falhas em tempo real.

1.3 Conceitos básicos de avarias e falhas

Em qualquer processo industrial real existem muitas razões para o aparecimento de avarias ou falhas, como por exemplo:

- Mau projeto, montagem incorreta dos equipamentos;
- Condução deficiente do processo e falta de manutenção;
- Envelhecimento, corrosão e desgaste durante o normal funcionamento.

Quando um processo se encontra em operação podem ocorrer repentinamente situações de funcionamento anormal implicando uma ligeira degradação do desempenho do processo, ou eventualmente causando uma elevada deterioração do comportamento do processo, podendo ainda o funcionamento anormal manifestar-se como uma evolução incremental da degradação de desempenho. Estas situações podem ser consequência da ocorrência de uma avaria ou falha.

O termo falha corresponde a uma situação em que se verifica um desvio nos valores de pelo menos um parâmetro ou variável do processo que não é aceitável, estando esse valor completamente fora do intervalo de valores aceitáveis para o parâmetro ou variável em apreço considerando-se as condições normais de funcionamento. As avarias correspondem a uma permanente interrupção da capacidade do processo em executar uma função de acordo com todos os parâmetros especificados, tratando-se de um evento que pode ser decorrente de uma ou mais falhas. As falhas podem ainda ser consideradas de dois tipos mencionadas seguidamente e representadas na figura 1.2.

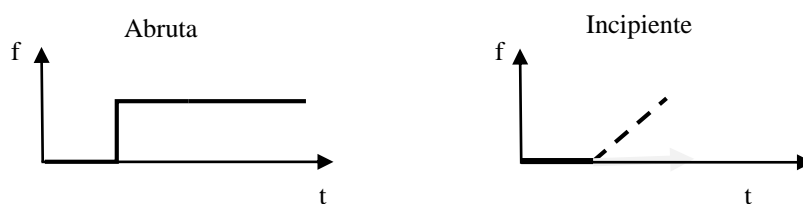


Figura 1. 2 (adaptado de [18]) Tipos de falhas

- **Abruta** - A falha ocorre de uma forma rápida – tipo degrau;
- **Incipiente** – Falha em que o componente diminui gradualmente o seu desempenho até chegar ao ponto de rutura., em que deixa de funcionar - tipo linear ou exponencial.

As falhas que ocorrem em qualquer processo fazem com que a fiabilidade, disponibilidade, desempenho e segurança dos processos industriais sejam colocadas em causa, daí a necessidade de desenvolvimento de sistemas eficazes na sua deteção e isolamento em tempo real.

1.4 Deteção e isolamento de falhas

Nenhum processo consegue trabalhar perfeitamente com desempenho constante ao longo dos tempos, ou seja, há fatores internos e externos que afetam “a vida” ou condições de funcionamento de qualquer processo real. Nas grandes indústrias, más ações de operação do processo, sensores ou atuadores defeituosos podem causar degradação de desempenho do processo ou acidentes fatais. A deteção e isolamento precoce de falhas são cruciais para ajudar a prevenir acidentes catastróficos em termos ambientais e humanos e consequentemente paragens intempestivas dos processos. O isolamento de falhas é uma das tarefas mais importante na segurança e controlo dos processos industriais.

Para maior eficácia, eficiência e segurança os processos industriais deverão integrar métodos avançados de deteção e diagnóstico de falhas e a sua tolerância de modo a cumprir os seguintes requisitos:

- Deteção precoce de falhas;
- Deteção de falhas em processos com anéis de controlo que podem mascarar o efeito de uma falha;
- Diagnóstico de avarias e falhas em sensores e atuadores ou outros componentes do sistema;
- Supervisão de processos em regime estacionário ou variações transientes decorrentes de alterações dos valores desejados para as variáveis controladas;
- Capacidades dos sistemas de controlo tolerarem falhas e correspondente capacidade de reconfiguração.

Um sistema de diagnóstico de falhas é um sistema de monitorização de processos utilizado para detetar e identificar falhas nos mesmos. Por esta razão muitas vezes o diagnóstico de falhas também é designado por sistemas de deteção e isolamento de falhas (DIF) – *Fault Detection and Isolation (FDI)*.

A deteção e diagnóstico de falhas incluem geralmente três funções:

- Detecção de falhas (DF). (*FD – Fault Detection*): determinação da existência de falhas presentes no processo;
- Isolamento de falhas (IF) (*Fault Isolation*): determinação da localização das falhas (nos sensores, nos atuadores no processo propriamente dito etc.) e dos instantes temporais em que ocorreram;
- Identificação de falhas (*Fault Identification*): tarefa efetuada após o isolamento das falhas, permitindo determinar a sua intensidade, ou seja, permitindo efetuar uma caracterização detalhada da falha. A figura 1.3 representa genericamente um sistema de deteção e diagnóstico de falhas.



Figura 1. 3 (adaptado de [13]) **Sistema de deteção e diagnóstico de falhas**

1.5 Arquitetura de um sistema de deteção e isolamento de falhas

A arquitetura de um sistema de deteção e isolamento de falhas num processo industrial, baseada em modelos de processo, pode ser observada na figura 1.4, integrando metodologias quantitativas na fase de deteção de falhas e metodologias qualitativas na fase de diagnóstico de falhas. Esta abordagem requer duas etapas: A geração e verificação dos resíduos, relação ente o comportamento real e o comportamento esperado, e a escolha de uma regra de decisão para isolamento das falhas. A figura 1.4 mostra a estrutura de um sistema de deteção e isolamento de falhas baseado em modelo.

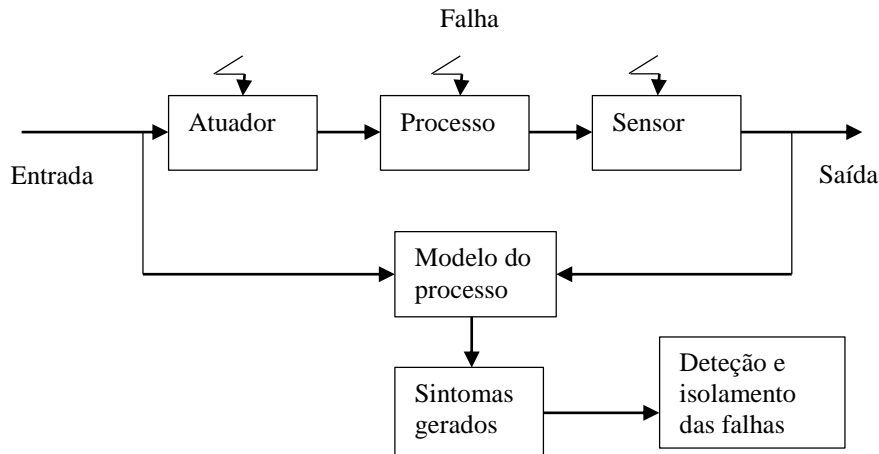


Figura 1. 4 (adaptado de [26]) **Estrutura de isolamento de falhas baseada em modelo**

Na fase de detecção de falhas, independentemente da metodologia utilizada, procura-se observar desvios nos valores dos parâmetros e/ou variáveis do processo para além dos valores que é suposto assumir em condições normais de operação. O resultado do conjunto dos desvios verificados constituem os sintomas que devem refletir uma falha específica. Os sintomas são introduzidos num sistema de decisão que no caso de estudo apresentado no capítulo seguinte, é uma rede neuronal artificial de forma a permitir realizar isolamento das falhas que, entretanto, ocorrem no processo.

Na literatura há diversos trabalhos relacionados com a detecção e isolamento de falhas em processos industriais, utilizando um conjunto de ferramentas diferentes, algumas baseadas em redundância física, outras utilizando redundância analítica [13,18]. Neste trabalho considera-se apenas o segundo caso.

Os métodos de detecção de falhas que baseiam em redundância física (ou de hardware) redundantes necessitam de equipamentos de hardware extras (como sensores e atuadores) para medir ou controlar uma determinada variável do processo. Desse modo, é possível verificar a consistência entre os elementos redundantes e com isso identificar a ocorrência de falhas nesses elementos. A grande desvantagem da redundância física é o elevado custo de aquisição e manutenção [13].

Os métodos de detecção e diagnósticos de falhas baseados em modelos (redundância analítica) descrevem o comportamento real do processo. Através da comparação entre o comportamento do processo e o comportamento esperado é possível detetar falhas no processo [13, 18, 26].

De acordo com [13,18] essa abordagem tem sido mais utilizada devido ao reduzido custo, pois não há necessidade de instalação de componentes de hardware extra, uma vez que a detecção e diagnóstico de falhas baseados em modelos são realizados via software existente no computador. Além disso, a detecção e diagnósticos de falhas baseados em modelos apresentam uma teoria bem consolidada, com muitos casos de estudos.

Métodos de detecção e geração de resíduos

Durante muito tempo, pesquisadores utilizaram a abordagem analítica, baseada em modelos quantitativos para detecção e isolamento de falhas. Em geral estes métodos baseiam-se na geração de resíduos que apresentam desvios entre o processo com falhas e o processo sem falhas. Existem vários métodos de geração de resíduos em sistemas de detecção e isolamento de falhas baseados em modelo, sendo os quatro principais indicados a seguir [13]:

- Filtro de Kalman- O erro de predição de kalman pode ser utilizado como um sistema de detecção residual, sua média é zero na ausência de falha e diferente de zero na ocorrência de falha;
- Observadores- Os observadores também são considerados como detetores de falhas por meio de geração de resíduos;
- Relação de paridade- As relações de paridade são estruturas diretamente das entradas e saídas do modelo, sujeitas a uma transformação linear. A transformada residual serve para detetar e isolar falhas;
- Estimador de Parâmetros- É uma abordagem de detecção e isolamento de falhas paramétricas (multiplicativas). O modelo do processo, primeiro, é identificado sem falha. Os desvios no modelo servem como base para detecção e isolamento de falhas.

A figura1.5 mostra a estrutura de geração de resíduo adaptado de Patton [26].

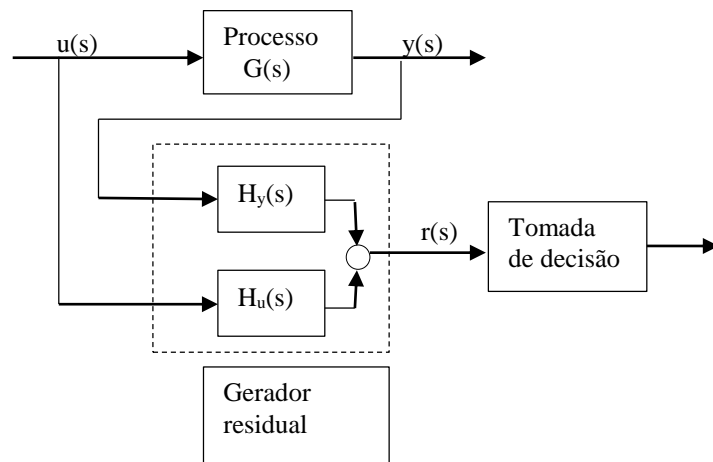


Figura 1. 5 (adaptado de [26]) **Estrutura de geração de resíduo**

Onde o vetor de sinais residuais $r(s)$ é dado por:

$$r(s) = H_u U(s) + H_y y(s) \quad (1.1)$$

em que :

$r(s) = 0 \rightarrow$ na ausência da falha

$r(s) \neq 0 \rightarrow$ quando ocorre falha

H_u e H_y representam as funções de transferência do processo e descreve o relacionamento entre as variáveis de entrada e saída em termos matemáticos.

$u(s)$ e $y(s)$ são as entradas e saídas do processo respectivamente.

As abordagens quantitativas baseadas em modelos matemáticos exigem grande precisão e são muito difíceis de usar em sistemas reais. Para superar tais dificuldades as redes neurais artificiais podem ser usadas tanto para gerar resíduos como para tomar decisões.

1.6 Métodos de detecção e diagnóstico de falhas

Em relação aos métodos de detecção, figura 1.6, e diagnóstico de falhas, figura 1.7, pode-se encontrar em [18] duas estruturas em árvores que classificam os diversos métodos.

A fase de detecção de falha tem como resultado a geração de sintomas. Os sintomas indicam o estado do processo (com falha ou assente de falha). Após a detecção de falhas, segue

o diagnóstico de falhas, que utiliza sintomas para o isolamento e a identificação das falhas através da aplicação de métodos de classificação ou de inferência. [18].

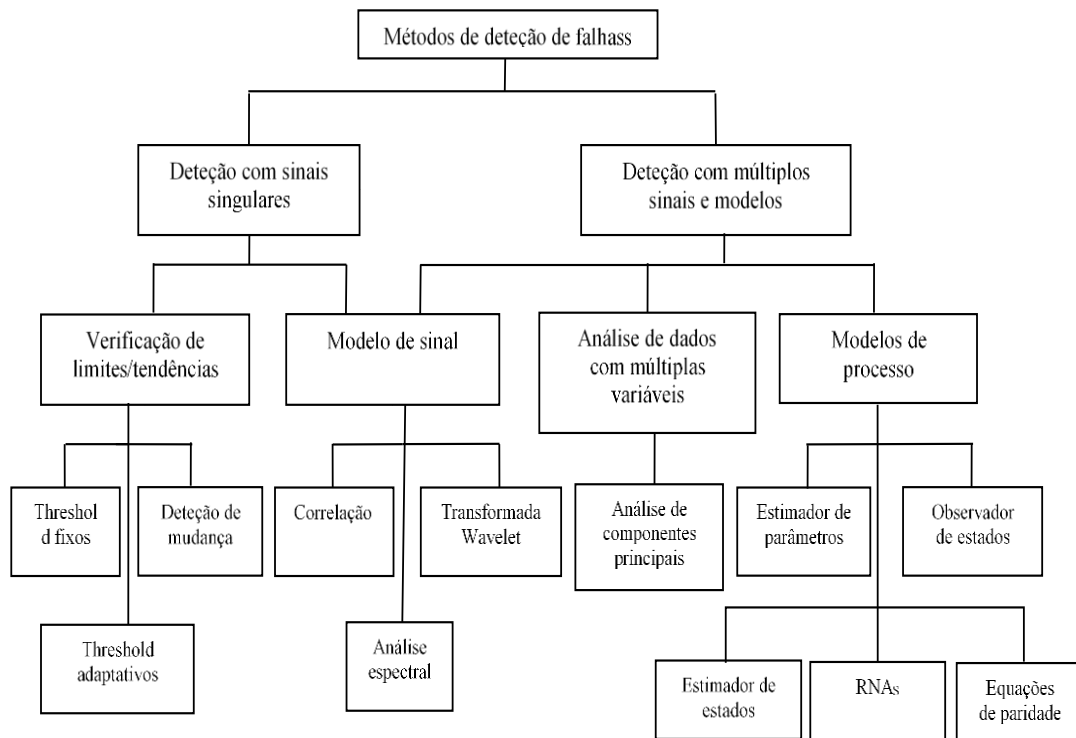


Figura 1. 6 (adaptado de [18]) **Métodos de detecção de falhas**

Um dos métodos mais tradicionais para detecção de falhas é a definição de limites máximos aceitáveis para variáveis do processo. Uma passagem dos limites definidos indica o aparecimento de falha. Contudo, este método não é eficaz quando se tem em conta variações das variáveis do processo para diferentes estados de operação. Em contrapartida, os métodos baseados em modelo de processo, que baseiam em geração de resíduos, comparam os valores reais do processo com os valores esperados para detetar eventual falha no processo.

Exemplos de alguns métodos baseados em modelo do processo são: redes neuronais artificiais, lógica fuzzy, estimativa de estados, estimativa de parâmetros, observadores de estados equações de paridade. Outros métodos baseiam em modelos de sinais, tais como, filtro passa baixa estimativa da entropia máxima e análise de espectro.

A figura 1.7 mostra os métodos utilizados no diagnóstico de falhas. Tais métodos são divididos em dois grupos: métodos de classificação e métodos de inferência. O primeiro é dividido em cinco subgrupos, o segundo em dois subgrupos.

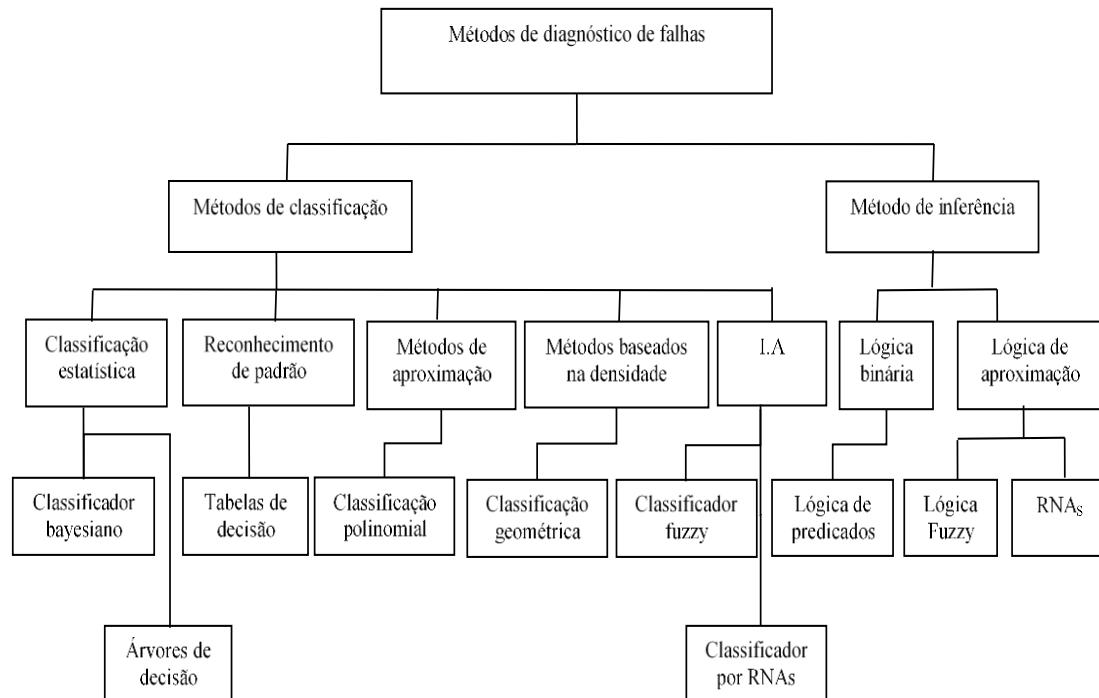


Figura 1. 7 (adaptado de [18]) **Métodos de diagnóstico de falhas**

Os métodos de classificação são usados quando não se tem informações sobre as relações entre sintomas e falhas. Dado um conjunto de sintomas não se sabe, a priori, quais falhas geram esses sintomas. Esses métodos utilizam um conjunto sintomas de referência para certas falhas no processo (obtido através de treino ou aprendizagem). A partir a comparação entre o conjunto de referência e o conjunto de sintomas observados é possível, através da classificação, determinar o conjunto de falhas no processo.

Se se tem mais informação sobre as relações das falhas e sintomas podem ser utilizados métodos de inferência. Essas informações podem ser expressas, por exemplo, através do tipo : **IF <sintomas> THEN<falhas>** Em [18] é apresentado um estudo sobre o uso de métodos de inferência.

Diversos métodos de detecção de falhas são apresentados nas literaturas [13,18,26].

De acordo com [27, 28, 29] nenhum método isolado possui todas as características desejáveis para um bom sistema de detecção e diagnóstico de falhas.

Alguns dos métodos podem ser complementares resultando em sistemas de diagnóstico mais eficiente quando utilizados em simultâneo.

1.7 Conclusão

Foi efetuada uma abordagem aos métodos de deteção e isolamento de falhas que tem vindo a ser utilizados em aplicações reportadas na literatura. Os métodos propostos podem ser divididos basicamente em métodos quantitativos, que precisam do modelo do processo, e métodos qualitativos, não precisam do modelo do processo. Os métodos qualitativos apesar de não necessitarem do modelo do processo necessitam de dados de treino. No âmbito deste trabalho será estudada a viabilidade de utilização técnicas de inteligência artificial, nomeadamente, redes neuronais artificiais, método qualitativo, para detetar e isolar falhas em tempo real nos processos industriais. Nesta conformidade, o próximo capítulo será dedicado ao estudo das RNAs

CAPITULO 2

REDES NEURONAIS ARTIFICIAIS

Resumo

Neste capítulo serão abordados os conceitos fundamentais relativos às redes neuronais artificiais.

É apresentada a arquitetura e estudada uma rede direta multicamada, apresentando-se também o algoritmo de treino

2 REDES NEURONAIS ARTIFICIAIS

As redes neuronais artificiais ou perceptrão surgiram em 1943 quando o neurobiologista Warren McCulloch e o matemático Walter Pitts fizeram uma analogia entre as células nervosas e o processo eletrónico no artigo publicado no *Bolletín of Mathematical Biophysics* com o título: *A Logical Calculus of the Ideas IMManent in Nervous Activity*. Em 1949 na história da inteligência artificial, nomeadamente a que se refere a implementação de redes neuronais artificiais, deu-se o grande acontecimento, que correspondeu a implementação da primeira regra de aprendizagem, proposta por Donald Hebb no seu livro *The organization of the behaviour* hoje conhecida como regra de Hebb (*Hebbian Rule*) Hebb mostrou como a aprendizagem das redes neuronais artificiais é conseguida através de pesos de entrada dos neurónios. Ele propôs uma teoria para explicar a aprendizagem em neurónios biológicos baseada no esforço das ligações sinápticas entre todos os neurónios excitados. Se dois neurónios estão ligados entre si e, o primeiro persiste repetidamente em ativar o segundo, então o valor sináptico da ligação entre ambos deve aumentar.

O neurobiologista Frank Rosenblat, em 1958, desenvolveu o modelo conhecido por perceptrão, como sendo uma máquina treinável capaz de aprender, classificar e determinar os padrões. O modelo perceptrão mais simples é constituído por uma rede de uma única camada, capaz de ajustar os seus pesos de modo a produzir uma resposta correta a um determinado padrão de entrada.

Em 1969, Minsky e Papet chamaram à atenção para alguns problemas que o perceptrão não era capaz de executar, já que este só resolve problemas linearmente separáveis, ou seja, problemas cuja solução pode ser obtida dividindo-se o espaço de entrada em duas regiões através de uma reta. O perceptrão, por exemplo, não consegue detetar paridade e simetria que são problemas não linearmente separáveis.

Nos anos 70 a abordagem sobre redes neuronais artificiais ficou adormecida (em grande parte devido a repercussão do trabalho de Minsky e Papet), apesar de alguns poucos pesquisadores continuarem a trabalhar na área. Entre eles são citados Igor Alesksander (redes sem pesos) na Inglaterra. Steven Gossberg (sistemas auto-adaptativos) nos Estados Unidos da América [7].

Bernard Widrow e Marcian Hoff desenvolveram um novo tipo de elemento de processamento de redes neuronais artificiais chamado Adaline (*Adaptive Linear Element*) em conjunto com algoritmo de treino que veio mais tarde a ser conhecido como *LMS (Least Mean Square)* que ainda possui aplicabilidade.

Nos anos 80 surgem novos modelos de redes neuronais artificiais propostos por John Hopfield e Teuvo Kohonen (memórias associativas e auto-organizadas), para mais detalhes ver bibliografia[6, 14, 15].

O ímpeto que se vive hoje na investigação das redes neuronais artificiais deve-se sobretudo ao artigo de John Hopfield apresentado à Academia Nacional de Ciências Americana em 1982. Neste artigo Hopfield chamou a atenção para as propriedades associativas das RNAs. Este artigo foi responsável por parte de retorno das pesquisas na área.

Em 1986 David E. Rumelhart e James L. McClelland, apresentaram um modelo matemático e computacional que proporciona supervisão dos neurónios artificiais, surgindo então o algoritmo retro propagação de erro.

Em 1987 ocorreu a primeira conferência de redes neuronais artificiais, também chamada de Sociedade Internacional de Redes Neuronais Artificiais (*International Neural Network Society*).

Uma definição explícita do que são as redes neuronais artificiais pode ser encontrada em [1,2, 5, 7,9]: “ A rede neuronal artificial é um processador paralelamente distribuído, constituído por unidades de processamento simples, que tem a capacidade de armazenar conhecimentos experimental e torná-lo disponível para a aplicação.

Uma área específica na qual muitas aplicações das redes neuronais artificiais podem ser desenvolvidas é reconhecimento automático de caracteres e padrões (dígitos ou letras). [1, 2, 7, 10].

Existem alguns paradigmas de aprendizagem aplicados às RNAs que podem ser vistos detalhadamente em. [1, 2, 5, 7, 14, 15].

Atualmente existem muitas tipificações de redes neuronais artificiais, sendo cada uma delas caracterizada por diversos fatores, tais como: a topologia, regras de aprendizagem, algoritmo de aprendizagem, o procedimento de treino, modo de funcionamento pós treino e as

suas principais aplicações. Pode encontrar-se informações mais detalhadas sobre os diferentes modelos existentes em [1, 2, 5, 7, 10, 14, 15].

Nos últimos anos, o estudo das redes neuronais artificiais e sua aplicação “aumentou exponencialmente”. Novas tecnologias em engenharias, física e medicina exigem métodos cada vez mais complexos de processamento de informação. Em [19] os autores mostram como as redes neuronais em tempo real podem ser implementadas para ampliar o leque de técnicas de processamento de sinais para ajudar a combater o problema de previsão.

Hirose [17] enfatiza as ideias básicas das redes neuronais que lidam com números complexos, as vantagens dessas redes e as áreas das suas aplicabilidades.

As redes neuronais artificiais têm sido utilizadas com sucesso na resolução de problemas complexos, em diferentes áreas de aplicação: medicina – umas das funcionalidades, monitorização de cirurgias e diagnósticos de doenças. Também é aplicada com sucesso na identificação de falhas nos circuitos telefónicos, bem como reconhecimento de padrões, por exemplo reconhecimento da escrita manual. As redes neuronais artificiais têm sido aplicadas com sucesso na previsão do estado do tempo, na área de investigação criminal [8], e na deteção e isolamento de falhas [18, 26] etc.

As redes neuronais artificiais com sua poderosa capacidade de aprendizagem através de exemplos e de generalizar a aprendizagem de forma a reconhecer elementos similares aos que foram incluídos no conjunto de treino, bem como a sua robustez a presença de informações falsas ou ausentes e por serem tolerantes a falha têm sido utilizadas com sucesso na deteção e isolamento de falhas.

As RNAs são inspiradas no cérebro biológico. Para melhor compreensão das RNAs nos próximos parágrafos dá-se a conhecer ao leitor um pouco de neurónio biológico.

2.1 Motivação para RNAs: neurónio biológico

O ser humano é constituído por um sistema nervoso que, entre outras funções, é responsável pelo equilíbrio do corpo e da mente relativamente ao ambiente que o envolve, sendo esta função realizada através da aprendizagem continua. O sistema nervoso controla as funções orgânicas do corpo e a interação deste com o ambiente.

O nosso encéfalo é uma estrutura extremamente complexa e não linear que processa paralelamente quantidades massivas de informações. O processamento é feito através de uma rede dinâmica de neurónios interligados que interagem entre si emitindo e recebendo informações de um e para outros neurónios.

O neurónio biológico é uma célula de sistema nervoso responsável pela condução do impulso nervoso. É constituído pelas seguintes partes:

- Um corpo celular – possui determinadas propriedades, essenciais ao funcionamento da célula. A partir do corpo celular projetam-se extensões filamentosas, as dendrites e os axónios;
- Numerosos conjuntos de dendrites – recebem sinais de outros neurónios através da sinapse;
- E por axónio – transmite estímulos ou respostas para outros neurónios.

Cada uma das partes do neurónio biológico, acima mencionadas, possui funções específicas contudo complementares.

Calcula-se que o cérebro humano tem cerca de 10^{11} neurónios, na sua célula fundamental, cada um com 10^4 sinapses. Cada um destes neurónios processa e comunica com milhares de outros, continuamente e em paralelo. A estrutura básica do neurónio biológico pode ser vista na figura 2.1.

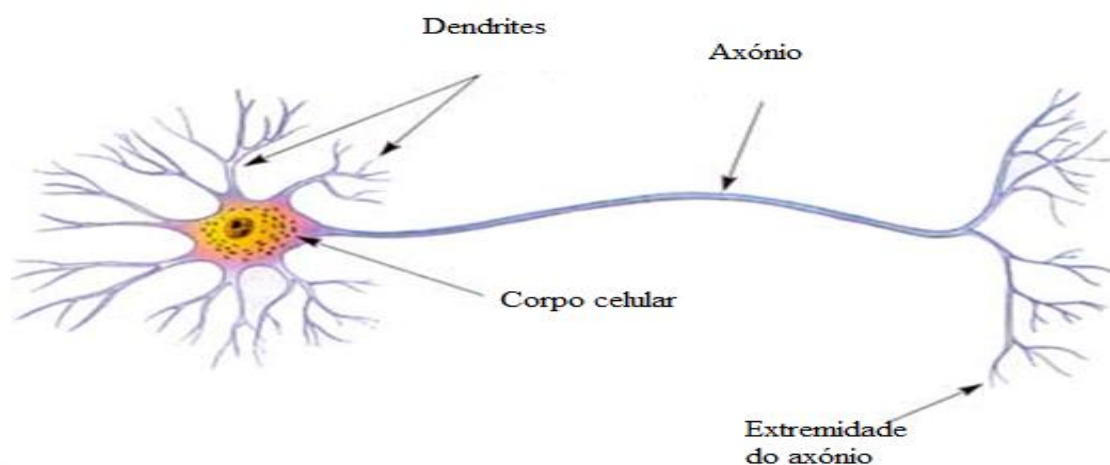


Figura 2. 1 Estrutura do neurónio biológico [33]

As dendrites recebem informações ou impulsos nervosos, vindas de outros neurónios e condu-las até ao corpo celular. Aqui as informações são processadas e novos impulsos são

gerados. Estes impulsos passam para outros neurónios através do axónio até às dendrites dos neurónios seguintes. O ponto de contato entre a extremidade do axónio de um neurónio e a dendrite de outro neurónio é designado de sinapse. É pela sinapse que os neurónios se unem funcionalmente, formando então redes neuronais.

Os seguintes tópicos refletem as mais importantes capacidades e características de um neurónio biológico que são essencialmente atrativas para serem simuladas nas RNAs:

- É uma estrutura complexa, não linear e paralela;
- Capacidade de aprender e generalizar – é uma das características mais importante das redes neuronais biológicas. Os humanos e os animais aprendem continuamente na sua interação com o meio envolvente. A informação disponível é automaticamente assimilada pelo cérebro que afeta o comportamento futuro;
- Facilidade de lidar com informação ruidosa;
- Capacidade de realizar tarefas complexas tal como, reconhecimento de padrões ou voz.

Muitas características funcionais proporcionadas pelos neurónios biológicos foram transpostas para os neurónios artificiais, tornando-se estas fundamentais no seu desempenho. As mais comuns são:

- Capacidade de aprender através de exemplos e de generalizar a aprendizagem de forma a reconhecer elementos similares que foram apresentados no conjunto de treino;
- A unidade de processamento pode receber muitos sinais de outros neurónios, em analogia com o corpo celular dos neurónios biológicos;
- A unidade de processamento manipula numericamente os sinais que recebe, sendo a soma a operação mais comum;
- Processamento paralelo de grandes quantidades de informação – permite que as tarefas complexas sejam realizadas num curto intervalo de tempo;
- A saída de um neurónio pode estar ligada a muitos outros neurónios;
- Tolerante a ruído – são capazes de ignorar o ruído, atributos irrelevantes e lidam com informação incompleta de forma eficiente;
- Organização própria – durante o treino a operação das redes neuronais artificiais organiza e cria as suas próprias representações da informação que recebe;
- As redes neuronais artificiais são modelos treinados e não programados;

- Flexibilidade – Grande domínio de aplicação.

No modelo da RNA apresentada na figura 2.2 é possível estabelecer a analogia com o neurónio biológico.

2.2 Modelo da RNA

Na figura 2.2 é mostrado o modelo de um neurónio artificial. Neste neurónio o processo é similar ao neurónio biológico. Os sinais X_1, X_2, \dots, X_m , provenientes de m neurónios são ponderados pelas sinapses $W_{k1}, W_{k2} \dots W_{km}$, para produzir sinais de entrada no neurónio v_k , o qual passa por uma função de ativação para produzir o sinal de saída do neurónio y_k .

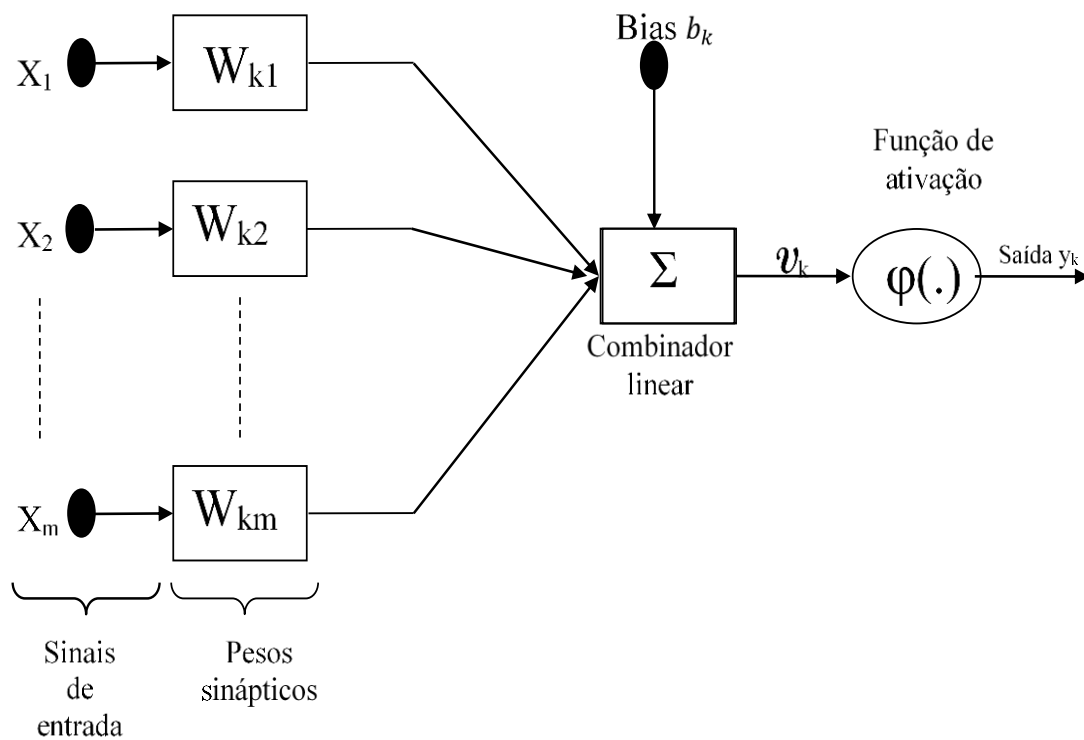


Figura 2. 2 (adaptado de [15]) **Modelo do neurónio artificial**

Analicamente a saída do neurónio pode ser determinada através da seguinte equação:

$$y_k = \varphi(v_k + b_k) \quad (2.1)$$

Onde v_k é dado por:

$$v_k = \sum_{j=1}^m W_{kj} X_j \quad (2.2)$$

De acordo com as equações 2.1 e 2.2 a saída y_k Será determinada de acordo com a expressão (2.3) abaixo dicada:

$$y_k = \varphi\left(\sum_{j=1}^m w_{kj}x_j + b_k\right) \quad (2.3)$$

Suponhamos agora que b_k é um peso correspondente a uma entrada constante (de valor 1). Simplificando a equação 2.3 será dada pela expressão abaixo indicada:

$$y_k = \varphi\left(\sum_{j=0}^m w_{kj}x_j\right) \quad (2.4)$$

Onde:

X_1, X_2, \dots, X_m – São sinais de entrada;

$W_{k1}, W_{k2}, \dots, W_{km}$ – Pesos sinápticos de cada ligação;

b_k – Bias;

$\varphi (v_k)$ – Função de ativação;

y_k – Valor da saída do neurónio;

k – Índice do neurónio na rede;

m – É o número de entradas no neurónio k ;

Σ - Somatório.

No modelo analítico do neurónio artificial acima apresentado constata-se:

- Um conjunto – de ligações sinápticas;
- Um somador – faz soma das entradas ponderadas;
- Uma entrada externa b_k – Este valor permite que a função tenha mais um grau de liberdade;
- Função de ativação – permite a passagem do sinal para a saída do neurónio, calculando o valor de saída do neurónio a partir do seu valor de ativação.

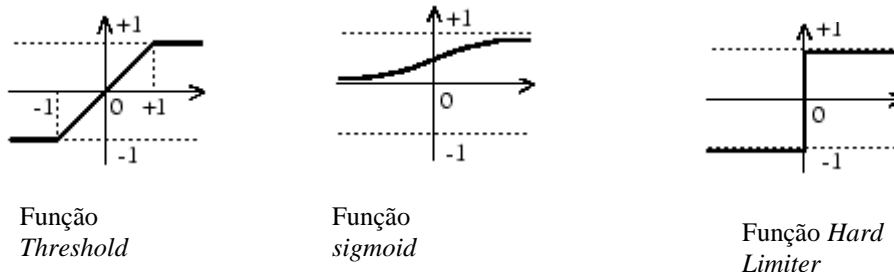


Figura 2. 3 (adaptado de [15]) **Tipos de função de ativação**

Uma das funções de ativação mais utilizada é a *sigmoid*, é uma função contínua limitada, normalmente fechada no intervalo entre [0,1] ou [-1,1] e monótona. Existem vários tipos de funções *sigmoid*. Na figura 2.3 estão representados três tipos de funções de ativação.

É de crucial importância a classificação das RNAs, uma vez que as características de cada grupo restringem o tipo de problemas que podem ser tratados pela rede. No capítulo seguinte são apresentados alguns tipos de RNAs.

2.3 Classificação das redes neuronais artificiais

Neste capítulo irá ser efetuado a classificação das redes neuronais artificiais sendo o primeiro critério utilizado o do número de camadas envolvidas na arquitetura da rede:

RNA mono camada – É composta por neurónios que recebem a informação diretamente das entradas e disponibilizam a informação de saída. A figura 2.4 representa uma hipotética RNA da camada única.

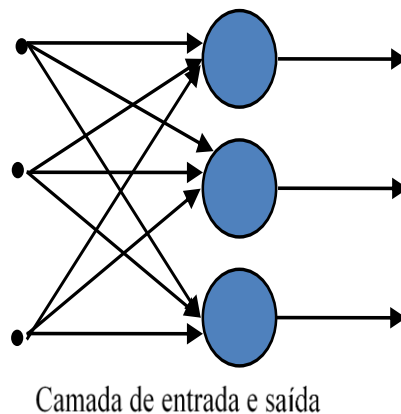


Figura 2. 4 RNA de camada única

RNA multicamadas – Denominada na literatura anglo-saxónica, Multilayer Perceptron (MLP). Entende-se por RNA multicamada uma rede que além de possuir camada de entrada e saída possui ainda pelo menos uma camada intermédia, conforme representado na figura 2.5.

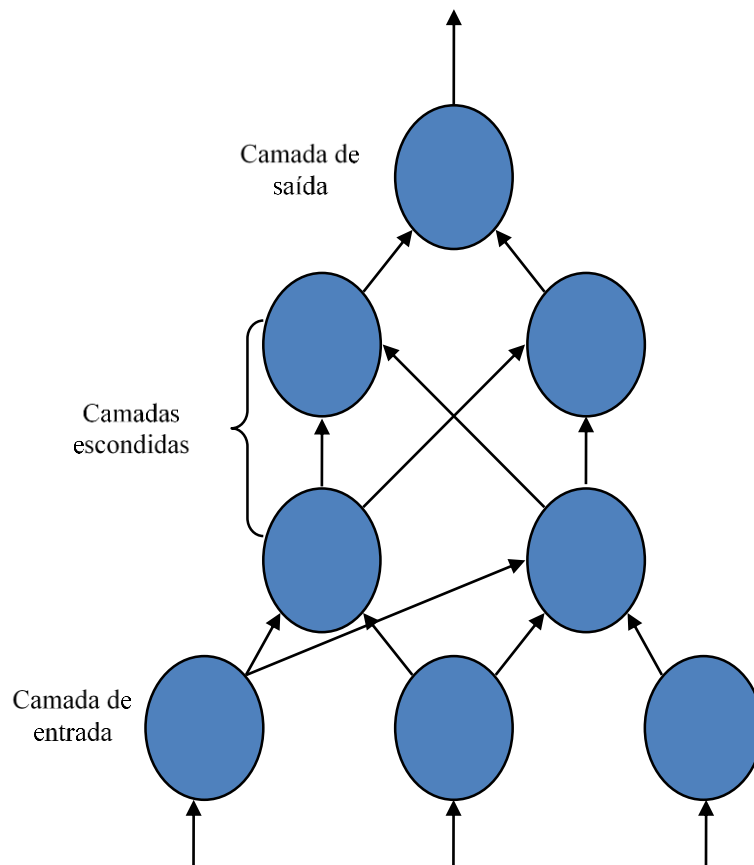


Figura 2. 5 RNA Multicamadas

A primeira camada é chamada de camada de entrada (*input layer*) e tem a função de armazenar os dados de entrada na rede. As camadas intermédias denominadas de camadas escondidas (*hidden layer*) são os “raciocínios” onde ocorrem os processamentos da rede. A última camada é definida como de saída (*output layer*) ou resposta.

Uma outra classificação possível é em função do tipo de ligações existentes entre camadas e/ou entre neurónios, distinguindo-se:

RNAs diretas (*feedeforward*) – As redes diretas são aquelas em que as saídas de todos os neurónios da RNA vão para as camadas posteriores, não havendo realimentação de informações, por outras palavras, o fluxo de informação é sempre de camada de entrada para camada de saída. O sentido do fluxo de informação de uma RNA direta pode ser visto nas figuras 2.4 e 2.5.

Redes neuronais artificiais recorrentes – são RNAs com realimentação de informação, ou seja, nestas RNAs existe pelo menos uma ligação, que liga a saída de um neurónio à sua própria entrada, ou então à entrada de um outro neurónio como ilustrado na figura 2.6.

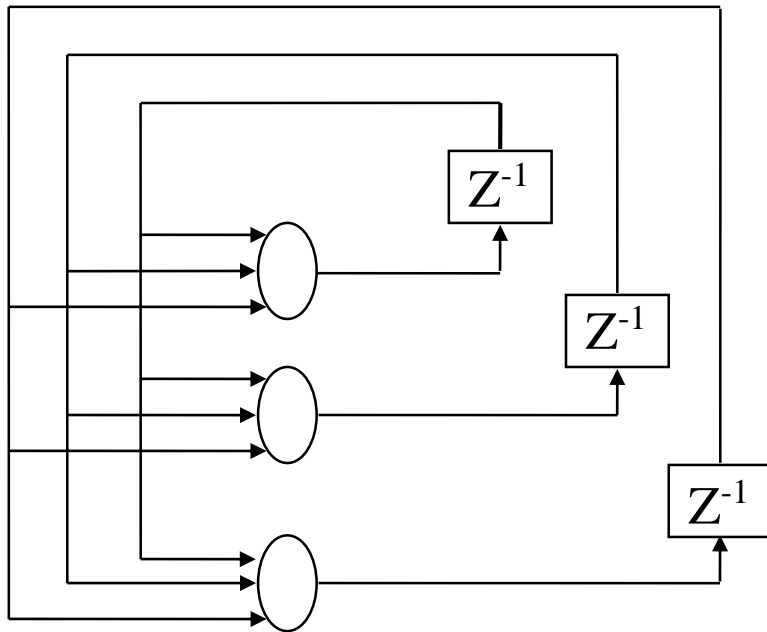


Figura 2. 6 (adaptado de [15]) RNAs recorrentes.

Como exemplo de redes recorrentes pode citar-se:

- **Rede de Hopfield** – Desenvolvida por *J J Hopfield*. Tem sido muito utilizada no reconhecimento de padrões. [6];
- **Redes de Kohonen** – Desenvolvida pelo *Teuvo Kohonen*, tem sido utilizada na classificação de padrões e otimização de problemas. [6, 14]
- **Redes ART** (*Adaptive Resonance Theory*) – tem sido de utilidade no âmbito do domínio do conhecimento do processamento de imagens. [6, 14];

Relativamente a forma como os neurónios se encontram ligados entre si poderá ser este o critério utilizado para efetuar a classificação das redes neuronais artificiais:

- Rede fracamente (ou parcialmente) ligada. Neste tipo de rede nem todas as saídas dos neurónios que constituem a camada anterior são entradas de todos os neurónios que constituem a camada seguinte;
- Rede completamente ligada – Neste tipo de rede todas as saídas dos neurónios de uma camada anterior são entradas de todos os neurónios da camada seguinte.

As redes neuronais artificiais possuem capacidade de aprender a partir de exemplos e fazer interpolações e extrapolações do que aprenderam. A utilização de uma RNA na resolução de problemas passa inicialmente por uma fase de aprendizagem. Uma definição do que será a

aprendizagem pode ser expressa da seguinte forma: a aprendizagem de uma rede neuronal artificial é a procura dos pesos associados às ligações existentes entre os neurónios constituintes da RNA de forma que esta esturra presente comportamento desejado, sendo o tipo de aprendizagem realizada definida pela maneira particular como ocorrem os ajustes realizados nos parâmetros. O processo de aprendizagem de uma rede neuronal artificial pode ser classificado em dois grupos: regras de aprendizagem e paradigmas de aprendizagem.

Regras e paradigmas de aprendizagem

Embora sejam bem conhecidos um número significativo de algoritmos de aprendizagem, estes têm sido desenvolvidos para aplicação em função do paradigma de aprendizagem que lhe estiver subjacente. São essencialmente três, os paradigmas de treino e respetiva aprendizagem das redes neuronais artificiais:

Aprendizagem supervisionada – este paradigma de aprendizagem é o mais comum no treino das RNAs. Tem essa designação porque a entrada e saída desejadas são fornecidas por um “professor” externo. O objetivo é ajustar os parâmetros da rede, de forma a encontrar uma função que efetue o mapeamento entre os pares, de entrada e saída, fornecidos.

A figura 2.7 ilustra o mecanismo de aprendizagem supervisionada. O “professor” indica explicitamente o valor desejado para a saída da rede quando se verifica determinada entrada, visando direcionar o processo de treino.

A rede tem a sua saída calculada e compara com a saída desejada. A cada padrão de entrada submetido à rede compara-se a resposta desejada com as respostas calculada e ajusta-se os pesos associados a cada uma das ligações entre neurónios para minimizar o erro. A soma dos erros quadráticos de todas as saídas é normalmente utilizada como medida de desempenho da rede e também como função de erro a ser minimizada pelo algoritmo de treino.

Este tipo de aprendizagem pode ser implementado basicamente de duas maneiras: *off-line* e *on-line*. Para treino *off-line* os dados de conjunto de treino não alteram, e uma vez obtida a solução para a rede, esta deve permanecer fixa. Caso novos dados sejam adicionados ao conjunto de treino, um novo treino envolvendo também os dados anteriores deve ser realizado para se evitar interferências no treino anterior.

Por outro lado, na aprendizagem *on-line* o conjunto de dados muda continuamente e a rede deve estar num contínuo processo de adaptação.

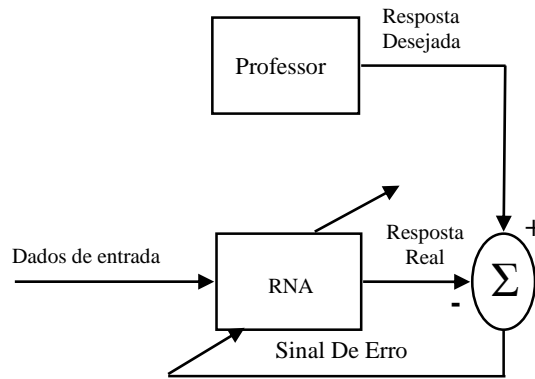


Figura 2. 7 (adaptado de [15]) **Aprendizagem supervisionada**

Aprendizagem não supervisionada – Neste tipo de aprendizagem, como o próprio nome indica, não há professor nem são conhecidas as respostas. A aprendizagem é feita através das descobertas dos padrões nos dados de entrada. Este paradigma está representado na figura 2.8.

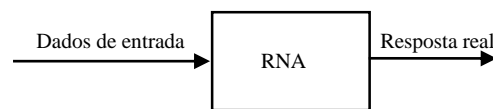


Figura 2. 8 (adaptado de [15]) **Aprendizagem não supervisionada**

Aprendizagem por reforço – A aprendizagem por reforço pode ser vista como um caso particular de aprendizagem supervisionada sendo também assumida a presença de um “professor” mas neste caso não são conhecidas as respostas. A principal diferença entre a aprendizagem supervisionada e aprendizagem por reforço é a medida de desempenho usada em cada um dos paradigmas de treino da rede. Na aprendizagem supervisionada a medida de desempenho é baseada num conjunto de respostas desejada através do uso do critério do erro conhecido, enquanto na aprendizagem por reforço, o desempenho é baseado em qualquer medida que possa ser fornecida ao algoritmo de aprendizagem permitindo apenas saber se a resposta é boa ou má.

Há pelo menos duas características comuns aos três paradigmas de aprendizagem, isto é:

Primeiro – Todos funcionam com base em exemplos dados pelos padrões de treino que são colocados na entrada da rede neuronal.

Segundo – Nem sempre, no final de um treino, a rede fica capaz de resolver os problemas corretamente. Nesta situação a aprendizagem não foi bem conseguida.

Por outro lado, visando a aprendizagem das RNAs os correspondentes algoritmos tendo sido desenvolvidos por base não só paradigma de aprendizagem que lhe está subjacente, mas também no conjunto de regras que estão devidamente descritas na literatura [15].

Aprendizagem com recurso a memória – Procura um padrão aproximado de entrada sendo que a seleção do padrão aproximado da região vizinha tem por base as experiências passadas. Quando é apresentado um novo exemplo a rede para o classificar recorre ao registo para saber a resposta mais adequada.

Se esse exemplo nunca foi registado são analisados exemplos que estejam numa vizinhança de RNA (do que foi apresentado), por fim, é gerada uma resposta aproximada para este caso. (os neurónios memorizam uma vasta gama de experiências passadas numa grande memória de pares entrada-saída. Quando é apresentado um novo padrão o algoritmo procura um padrão na região vizinha).

Aprendizagem de Hebb – Se dois neurónios estão ligados e ativos o peso da ligação entre eles deve ser reforçado, ou seja, à semelhança do que acontece no cérebro humano quando um neurónio da célula X está suficientemente perto para excitar uma célula Y ocorre um disparo e algum processo de mudança metabólica acontece numa ou ambas as células.

Aprendizagem competitiva – No início os neurónios da mesma camada tem pesos pequenos e iguais. Quando é fornecido um padrão à rede, o neurónio que responder melhor será “premiado” com o reforço dos pesos das suas ligações. Em alguns casos os pesos dos vizinhos também são reforçados. As saídas dos neurónios da mesma camada competem entre si para se tornarem ativas. O neurónio que apresentar um peso superior é ativado e envia um sinal unitário para a saída e, ao mesmo tempo, inibe os outros neurónios da camada de saída para não produzirem respostas.

Aprendizagem Boltzmann – Na rede neuronal artificial com aprendizagem de Boltzmann cada neurónio tem um valor binário (alto/baixo), com contributos gerais quantificados pelo somatório dos pesos de todas as ligações ativas. Segundo uma função de probabilidade, são seleccionados os neurónios aos quais se anulam o estado (alto/baixo), que mede o efeito produzido na rede. O ajuste dos pesos da ligação é realizado conforme a diferença entre a correlação demonstrada pelos neurónios envolvidos em cada ligação quando a rede se encontra inserida num determinado ambiente e a correlação dos mesmos quando não é aplicado a rede nenhum estímulo externo.

Aprendizagem por correção do erro ou regra delta – Através da aplicação de um estímulo a uma RNA registam-se as respostas. A diferença entre essas respostas e os valores que se pretendem obter é considerada erro. Conforme esse erro, os pesos da RNA são ajustados segundo um algoritmo de treino com o objetivo de o minimizar. O algoritmo de aprendizagem utilizado neste caso é designado por retro propagação do erro e será detalhadamente descrito no capítulo seguinte por ter sido o utilizado no âmbito do trabalho realizado conducente a escrita dessa dissertação. Mais detalhe sobre caracterização da RNA pode ser visto no anexo A.

A figura 2.9 ilustra as regras e paradigmas de aprendizagens citadas anteriormente.

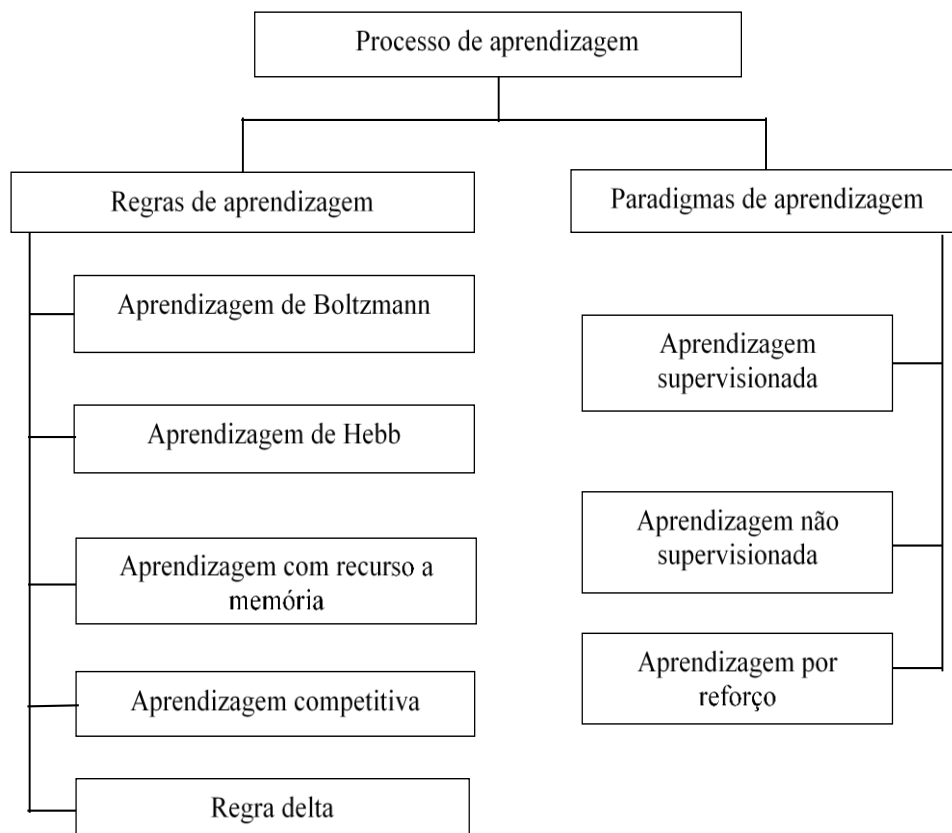


Figura 2. 9 (adaptado de [15]) **Regras e paradigmas de aprendizagem de uma RNA**

2.4 Algoritmo retro propagação de erro (do inglês backpropagation)

O algoritmo de aprendizagem é um procedimento cujo objetivo é o ajuste dos pesos e dos *bias* da rede, visando a otimização da aprendizagem. No caso de RNAs multicamadas, em que se considere o paradigma de aprendizagem supervisionada, o algoritmo utilizado para treino é o retro propagação de erro. Este algoritmo ou regra delta generalizada, é um método de gradiente descendente para minimizar o erro quadrático médio das saídas das RNAs, ao longo de todo o conjunto de treino.

Para funcionamento do algoritmo em apreço consideram-se duas fases, *forward* e *backward*, a figura 2.10 ilustra essas duas fases. Na fase *forward* aplica-se um estímulo à rede e regista-se o erro de resposta em relação ao valor de saída esperado. Na segunda fase esse erro é propagado pela rede, provocando um ajuste nos pesos das ligações entre os neurónios, contribuindo para a aprendizagem da rede. O algoritmo retro propagação de erro é um método de primeira ordem, ou seja, utiliza somente informação das derivadas de primeira ordem do erro quadrático do valor das saídas para correção dos pesos, visando a minimização desse erro.

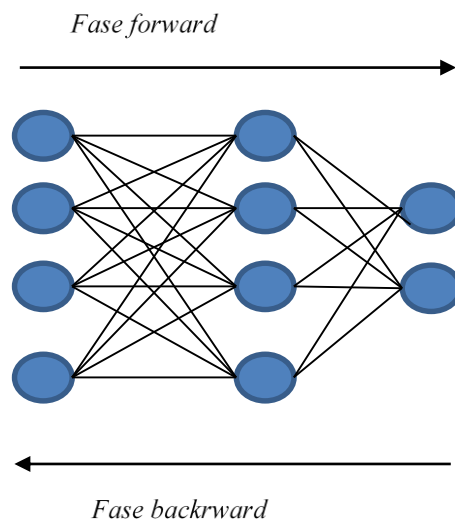


Figura 2. 10 Fluxo de processamento do algoritmo retro

Nos próximos parágrafos será descrita a regra delta ou gradiente descendente, no qual esse algoritmo se baseia. Para que a rede tenha o máximo desempenho, a função de erro E definida pela soma dos erros quadráticos e representada pela equação (2.5) deve assumir o menor valor possível:

$$E = \frac{1}{2} \sum_p \sum_{i=1}^k (d_i^p - y_i^p)^2 \quad (2.5)$$

Onde E é a medida de erro total, p é o número de padrões, k é o número de unidades de saída, d é a i -ésima saída desejada e y é a i -ésima saída gerada pela rede.

A regra delta generalizada requer que as funções de ativação utilizadas pelos neurónios sejam contínuas, diferenciáveis e monótonas.

A equação 2.6 ilustra o cálculo do valor de ativação.

$$y_j^p = f_j(\text{net}_j^p) \quad (2.6)$$

Onde:

$$\text{net}_j^p = \sum_{i=1}^n x_i^p w_{ji} \quad (2.7)$$

A constante n representa o número de ligações de entrada ao neurónio j e w_{ji} , o peso da conexão entre a entrada x_i^p e o neurónio j .

Embora o erro total E seja definido pela soma dos erros dos neurónios de saída para todos os padrões, será considerado, sem perda de generalidade, que a minimização do erro para cada padrão individualmente levará à minimização do erro total. Assim o erro passa a ser definido pela equação (2.8),

$$E = \frac{1}{2} \sum_{j=1}^k (d_j - y_j)^2 \quad (2.8)$$

A regra delta sugere que a variação dos pesos seja definida de acordo com o gradiente descendente do erro com relação ao peso, ou seja, que a variação do peso para um dado padrão seja definida pela equação (2.9),

$$\Delta w_{ji} \propto = \frac{-\partial E}{\partial w_{ji}} \quad (2.9)$$

Agora é necessário definir como cada um dos pesos de cada neurónio da rede deve ser ajustado de forma a diminuir o erro total gerado pela rede. Utilizando a regra da cadeia, tem-se que:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ji}} \quad (2.10)$$

Onde $\text{net}_j = \sum_{i=1}^n x_i w_{ji}$. A derivada do lado direito da igualdade, $\frac{\partial \text{net}_j}{\partial w_{ji}}$, é calculada de acordo com equação (2.11).

$$\frac{\partial \text{net}_j}{\partial w_{ji}} = \frac{\partial \sum_{i=1}^n x_i w_{ji}}{\partial w_{ji}} = x_i \quad (2.11)$$

A derivada do lado direito da igualdade da equação (2.10) mede o erro do neurónio j , sendo abreviada para δ_j , ou seja,.

$$\delta_j = \frac{\partial E}{\partial \text{net}_j} \quad (2.12)$$

O cálculo da derivada da equação (2.12) é definido pela regra da cadeia dando a equação (2.13),

$$\delta_j = \frac{\partial E}{\partial \text{net}_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \text{net}_j} \quad (2.13)$$

A segunda derivada da equação (2.13), $\frac{\partial y_j}{\partial \text{net}_j}$, é definida por:

$$\frac{\partial y_i}{\partial \text{net}_j} = \frac{\partial f(\text{net}_j)}{\partial \text{net}_j} = f'(\text{net}_j) \quad (2.14)$$

O cálculo da primeira derivada, $\frac{\partial E}{\partial y_j}$, que utiliza o erro, depende da posição onde se encontra o neurónio j . Se o neurónio estiver na última camada, o seu erro pode ser definido utilizando a equação 2.8. Assim:

$$\frac{\partial E}{\partial y_j} = \frac{\partial (\frac{1}{2}(\sum_{i=1}^k (d_i - y_i)^2))}{\partial y_j} = (d_j - y_j) \quad (2.15)$$

Substituindo os dois termos do lado direito da equação (2.13), obtém-se:

$$\delta_j = (d_j - y_j)f'(\text{net}_j) \quad (2.16)$$

Para um neurónio j que não seja um neurónio de saída, a regra da cadeia é utilizada

para escrever:

$$\frac{\partial E}{\partial y_j} = \sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} \frac{\partial \text{net}_l}{\partial y_j} = \sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} \frac{\partial \sum_{i=1}^n w_{il} y_i}{\partial y_j} = \sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} w_{jl} \quad (2.17)$$

Onde:

$$\sum_{l=1}^M \frac{\partial E}{\partial \text{net}_l} w_{jl} = \sum_{l=1}^M \delta_l w_{jl} \quad (2.18)$$

Substituindo mais uma vez os dois termos do lado direito da equação (2.13), obtém-se que, para os neurónios das camadas intermédias, o erro é definido por:

$$\delta_j = f(\text{net}_j) \sum_l \partial_l w_{lj} \quad (2.19)$$

A forma generalizada de ajuste dos pesos proposta na equação (2.9) fica para:

$$\Delta w_{ji} = \eta \delta_j x_i \quad (2.20)$$

ou:

$$w_{ji}(t + 1) = w_{ji}(t) + \eta \delta_j(t) x_i(t) \quad (2.21)$$

Se o neurónio for de saída, o erro δ_j , será definido pela equação (2.16), caso contrário, δ_j será dado pela equação (2.19).

O fator momentum

Existem certas zonas do espaço onde o contributo do erro para atualização dos pesos é muito fraco, pois nessas zonas de espaço dos pesos a curvatura da superfície de erro é pouco acentuada. Quanto menor for a taxa de aprendizagem menos mudanças os pesos das ligações sofrerão, de modo que a procura de mínimo global será favorecida pelo uso de saltos mais suaves. O problema é que desta forma tem-se uma aprendizagem mais lenta. Contudo se o aumento da taxa de aprendizagem for muito elevado, então os elevados saltos nas mudanças dos pesos poderão provocar instabilidade no treino.

Nestas zonas o fator *momentum* aumenta a taxa de aprendizagem acelerando o processo de convergência sem provocar instabilidade.

O termo *momentum* é representado pela equação (2.22),

$$\psi = \alpha(w_{ji}(t) - (w_{ji}(t - 1))) \quad (2.22)$$

Onde α representa a constante de *momentum*.

A fórmula completa de ajuste de pesos utilizando o termo *momentum* passa então a ser:

$$w_{ji}(t + 1) = w_{ji}(t) + \eta \delta_j(t) x_i(t) + \alpha(w_{ji}(t) - (w_{ji}(t - 1))) \quad (2.23)$$

A inclusão do termo *momentum*, consiste na introdução de um valor constante, normalmente, dado por um valor positivo no intervalo $[0, 1]$, funciona como memória que acelera o treino e evita oscilações dos valores dos pesos garantindo ao mesmo tempo a convergência do algoritmo. Os pesos são atualizados pela combinação do valor do gradiente no presente e no passado.

A introdução do fator *momentum* possibilita a convergência mais eficaz do algoritmo retro propagação de erros.

É de salientar que existem vários tipos de redes neuronais artificiais e algoritmos de treino que não foram citados, mas podem ser detalhados nas referências bibliográficas [1, 2, 5, 6, 7, 10, 14, 15, 20].

2.5 Construção da rede neuronal

Na construção de uma rede neuronal artificial existem três passos fundamentais a serem seguidos:

- Escolha e caracterização dos dados de entrada/saída, bem como, definição dos dados utilizados no processo de aprendizagem da rede;
- Definição da arquitetura da rede;
- Definição do critério de paragem do processo de treino da rede.

É crucial que os dados para a aprendizagem não sejam muito limitados nem muito extensos, pois um défice de informação pode conduzir a modelos poucos precisos, por outro lado quanto maior for a informação disponível mais recursos terá a rede de aprender. Se houver informação inútil a RNA ao não encontrar nenhuma relação entre os valores de entrada e de saída, limita-se a atribuir-lhes pesos nulos durante a aprendizagem devido a sua capacidade de filtrar informação que não seja importante para a aprendizagem.

A aprendizagem supervisionada é feita com base em exemplos, durante a qual se vão apresentando a rede pares de padrões, $[z^v \ w^v]$ (entradas-saídas desejadas) construídos a partir dos valores do conjunto total de dados.

É construída uma matriz linha com u valores, composta pelas z_i ($i = 1, 2, \dots, u-1$) observações do passado e pelo valor a prever w . Cada uma destas matrizes linha representa um padrão constituído por um par de vetores com os z_i valores a apresentar à entrada da rede e o valor w a apresentar na saída da rede.

Cada vez que é apresentado um padrão esta procura encontrar uma relação entre os valores da entrada e de saída.

Através de uma janela deslizante com comprimento u , a matriz coluna do conjunto total de dados com o comprimento n é transformada em v padrões com comprimento u , onde as entradas da rede são definidas por i valores consecutivos,

$$Z^v = (x_v, x_{v+1}, \dots, x_{v+i-1}) \text{ e a saída pelo valor a prever } w^v = x_{v+u-1}.$$

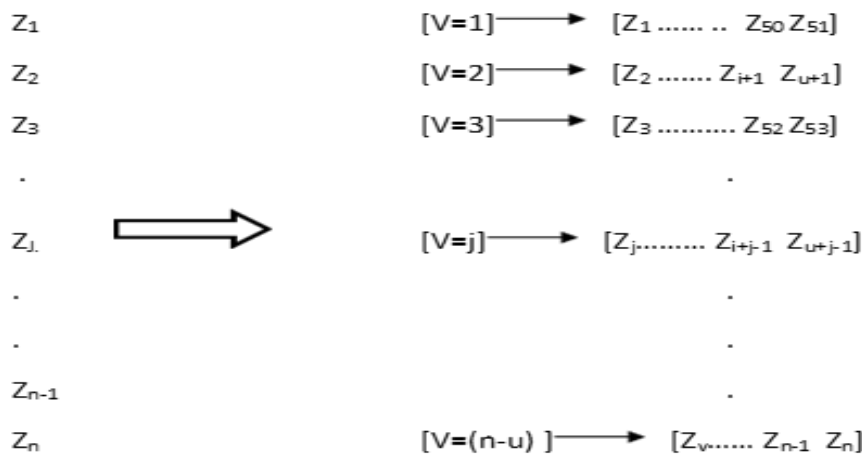


Figura 2. 11 (adaptado de [11]) **Exemplo dos conjuntos de padrões para aprendizagem**

A figura 2.11 mostra a construção da matriz do conjunto de padrões a utilizar na aprendizagem a partir da matriz do conjunto total de dados. Consequentemente o conjunto total com $(n-u)$ padrões será definido por uma matriz de $(n-u)$ linhas por u colunas. A matriz de padrões com dimensões $[(n - u) * u]$ será por sua vez dividida em dois conjuntos principais, um que será para a aprendizagem supervisionada da rede e outro de testes para simular uma aplicação real da mesma.

Simultaneamente o conjunto para aprendizagem será ainda dividido percentualmente em dois subconjuntos, um para o treino propriamente dito e outro para validação desse mesmo treino, sendo denominado por conjunto de validação [9,13].

Os três subconjuntos do conjunto total de padrões ficam, portanto, dispostos como indicado na figura 2.12

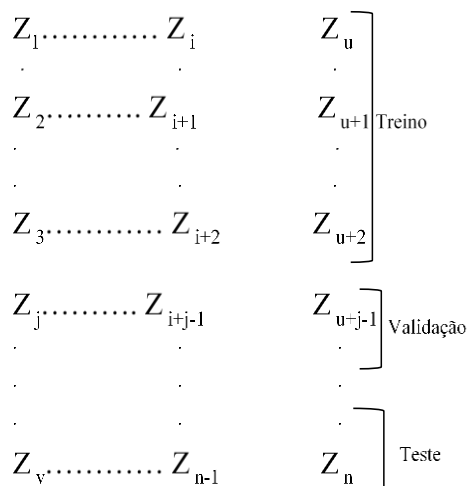


Figura 2. 12 (adaptado de [11]) **Caraterização dos conjuntos de padrões a utilizar na aprendizagem e teste**

Conjunto de treino

O conjunto de treino é constituído pelos padrões que serão apresentados à RNA durante o seu treino [11]. Os valores de entrada z^v ao serem aplicados na entrada da rede, irão produzir na saída um valor observado (o^v), mediante os pesos das ligações entre os neurónios. Simultaneamente, o valor de saída w^v desse mesmo padrão é comparado na saída com o correspondente valor esperado (d^v). Depois de calcular a diferença entre os valores observados e os esperados, o algoritmo retro propagação do erro tentará atualizar os pesos das ligações entre as unidades de modo a minimizar esta diferença.

A performance da rede neuronal MLP é fortemente dependente dos padrões. Ou conjunto de dados utilizado durante o treino, o qual deve possibilitar a melhor representação possível do domínio do problema. Contudo deve ter menor número de entradas possível, para que se possa trabalhar com a rede mais simples, além disso, é necessário que se tenha uma quantidade suficiente de padrões de treino.

Existem fórmulas para estimar o tamanho da amostra dos padrões de treino que seja suficiente bom para desempenho de generalização. Contudo verifica-se com frequência que existe uma diferença numérica bastante significativa entre o tamanho dos padrões realmente necessário e a prevista pelas fórmulas [15]. Das três entradas: nível, temperatura e valor da válvula 2.

Conjunto de validação

O conjunto de validação assume um papel preponderante a nível da aprendizagem. Os seus padrões são aplicados ciclicamente à rede durante a fase do treino para aferir a eficácia da aprendizagem e estimar a capacidade de generalização da rede. Não existe um valor ótimo para o mesmo, segundo os autores [5, 9] 10% dos valores na aprendizagem reúne o consenso.

Conjunto de teste

O conjunto de teste é constituído por padrões que nunca foram apresentados à rede durante o processo de aprendizagem. Diante de grandes números de redes neuronais artificiais a serem analisadas foi necessário estabelecer algum tipo de critério avaliativo para que a melhor rede neuronal artificial fosse selecionada. Através do cálculo do erro médio quadrático do teste é analisada a viabilidade da rede neuronal artificial para detetar e isolar falhas.

Após a escolha e a caracterização dos dados a utilizar na fase de aprendizagem, é necessário definir a estrutura da rede neuronal artificial.

Escolha da arquitetura da rede

Neste trabalho foi utilizado uma RNA direta com três camadas, a camada de entrada, a escondida e a camada de saída. Segundo [5,7,14,15] apenas uma camada escondida (desde que se possa variar infinitamente o número de neurónios) é suficiente para resolver qualquer problema de aproximação de funções apresentado a uma RNA direta, pelo que optou-se pela utilização de apenas uma camada escondida.

A utilização de um grande número de camadas intermédias não é recomendada, cada vez que o erro médio durante o treino é propagado para camada anterior, ele torna-se menos útil ou preciso. A única camada que tem uma noção precisa do erro cometido pela rede é a camada de saída [7].

Caraterização da camada escondida

Em relação ao número de neurónios nas camadas escondidas, este em geral definido empiricamente. Braga [7] afirma que:

- Uma camada intermédia é suficiente para aproximar qualquer função contínua;

- Duas camadas intermédias são suficientes para aproximar qualquer função matemática.

A camada escondida deve ter por volta de $2i+1$ neurónios, onde i é o número de variáveis de entrada [34].

Deve-se ter cuidado para não utilizar neurónios a mais na camada escondida, pois, pode levar a RNA a memorizar os padrões de treino, em vez de extrair características gerais que permitirão a generalização ou reconhecimento de padrões não vistos durante o treino (este problema é chamado de *overfitting*), por outro lado, défice de números de neurónios pode forçar a RNA a gastar tempo em excesso tentando encontrar uma representação ótima (as unidades utilizadas podem ficar sobrecarregadas por ter de lidar com elevado número de restrições) [7].

Uma vez definida a topologia de RNA é necessário utilizar um critério de paragem antes que haja perda de capacidade de generalização por parte da rede.

Critério de paragem

Um dos critérios de paragem geralmente atribuídos à convergência da rede, é o interromper o processo iterativo do treino, limitado por números de iterações pré-definidas.

Quando se limita o número de iterações a um número pré definido, não existe a garantia que os pesos atribuídos às ligações entre unidades sejam os que garantam a melhor generalização.

Outro critério de paragem do treino consiste na definição de um valor máximo de erro médio quadrático, embora não exista a garantia de que o sistema atinja o valor pré definido, levando a que o treino nunca termine.

Uma variação do método anterior consiste em deixar a rede neuronal aprender até que a diminuição do erro médio quadrático entre duas iterações seja menor que um certo valor pré definido. Este método tem como objetivo permitir ao sistema aprender até que não consiga extrair mais informação do conjunto de treino. Infelizmente este método apresenta problemas de paragens prematuras principalmente em regiões com superfícies de erro planas.

Todos os critérios apresentados anteriormente apesar de terem a sua validade não resolvem o problema da generalização, devendo por isso ser utilizado um método que pare a

aprendizagem no ponto em que a generalização é máxima. Este método é designado por validação cruzada [5].

Paralelamente ao processo de treino e com uma certa frequência, (entre 5 e 10 iterações) a capacidade de generalização da rede é testada, aplicando os padrões do conjunto de validação à rede até esse momento treinada [14]. Calculando o erro médio quadrático da validação, constata-se que este começa inicialmente a diminuir e volta a aumentar a partir de um certo número de iterações, como se constata nas curvas teóricas da figura 2.13 sendo esse sempre maior que o do treino [14]. A partir deste ponto diz-se que a rede está a sobre aprender ou em *overfittig* [7, 10].

Uma forma de evitar *overfittig* é estimar o erro de generalização durante o processo de treino. Para isso, o conjunto de dados é dividido em conjunto de treino e conjunto de validação. O conjunto de treino é utilizado na modificação dos pesos, e o conjunto de validação é utilizado para estimar a capacidade da generalização da rede durante o processo de aprendizagem.

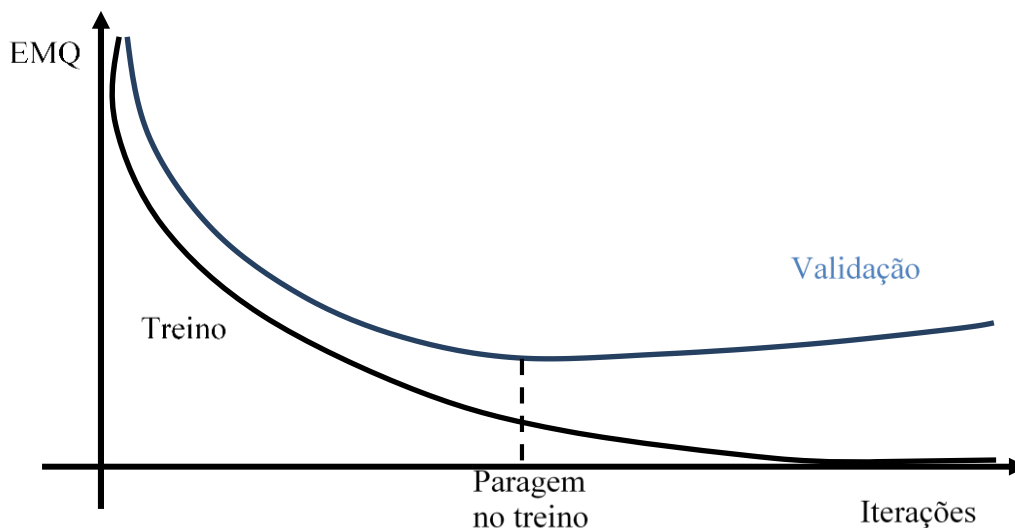


Figura 2. 13 (Adaptado de [14]) **Caraterística teórica do erro médio quadrático (EMQ) do treino e validação**

A melhor rede foi considerada após realização de vários testes com configurações diferentes, variando os números dos neurónios na camada intermédia, observou-se o número de iterações máxima para pequenas variações do erro quadrático médio, assim sendo, a topologia selecionada para discussão do resultado foi escolhida com base no erro quadrático médio.

Chegou-se a uma arquitetura com uma camada intermédia contendo 10 neurónios para o qual o treino e o teste da RNA foram satisfatórios, ver anexo D

2.6 Conclusão

Neste capítulo foram apresentados alguns conceitos básicos da rede neuronal biológica, nos quais a rede neuronal artificial foi inspirada. Seguidamente foram apresentados as redes neuronais artificiais. Foram apresentadas as arquiteturas das principais redes neuronais existentes, com ênfase nas redes diretas, que são redes utilizadas nesta dissertação. Foi apresentado de forma detalhada o algoritmo de treino de redes neuronais diretas, algoritmo retro propagação do erro, com uma análise de fatores que determinam seu desempenho.

CAPÍTULO 3

REATOR CONTINUAMENTE AGITADO

Resumo

Neste capítulo é efetuado uma descrição do processo utilizado como caso de estudo e apresentado o correspondente modelo matemático, usado para obter o processo em condições normais de operação e para obter o comportamento do processo durante simulação de falha, através de uma simulação implementada na plataforma computacional Matlab.

No controlo automático de processo foram implementados dois controladores PID.

3 CASO DE ESTUDO

O RCA é frequentemente escolhido quando o controlo da temperatura é um aspeto crítico, como na nitrção de hidrocarbonetos aromáticos ou glicerina, bem como na polimerização de estireno e butadieno ou quando reação entre duas fases deve ser realizada [12].

Em [3, 4] encontra-se exemplos da simulação dinâmica do RCA com controlo automático.

O processo (RCA) foi simulado através do seu modelo analítico apresentado no capítulo seguinte usando a plataforma computacional *Matlab*[®]/*Simiulink*, bem como, por forma a assegurar o seu controlo automático foram implementadas malhas fechadas de controlo utilizando controladores clássicos com ações de controlo proporcional, integral e derivativa (PID). O funcionamento deste tipo de controlador é descrito no subcapítulo 3.2, onde são também efetuadas algumas considerações relativamente ao seu desempenho. É também abordada a questão da sintonia dos controladores, a qual no âmbito deste trabalho foi efetuada recorrendo ao primeiro método de Ziegler-Nichols [24]. Na mencionada publicação o autor apresenta regras gerais para sintonia dos controladores, onde são detalhados alguns métodos tradicionais que podem ser usados para obter boas soluções para sintonia de controladores PID.

A implementação de um sistema de controlo automático tem como principal objetivo regular os valores das variáveis do processo, de forma a garantir a estabilidade da operação respeitando as condições operacionais.

3.1 Reator continuamente agitado

O reator continuamente agitado é um processo muito comum na indústria química embora haja inúmeras variações em relação a introdução extração da energia e de materiais [9].

Este processo apresenta diversas características que são de realçar, tais como múltiplas entradas e saídas e não linearidades, mas ao mesmo tempo, pode ser representado através de modelo matemático de dimensões relativamente reduzidas o que torna muito atraente a sua aplicação em estudos simulados. O *layout* do processo utilizado no âmbito deste trabalho é ilustrado na figura 3.1, assumindo-se que a reação ocorre no tanque do reator em condições

isotérmicas, que as propriedades do meio reacional são constantes, e para além disso, a reação é arrefecida por realimentação através de um permutador de calor.

O nível e a temperatura no reator são controlados por sistema de controlo em malha fechada. Foram usados controladores clássicos (PID) para realizar o controlo das variáveis do processo. A implementação de tais sistemas de controlo será discutida mais adiante.

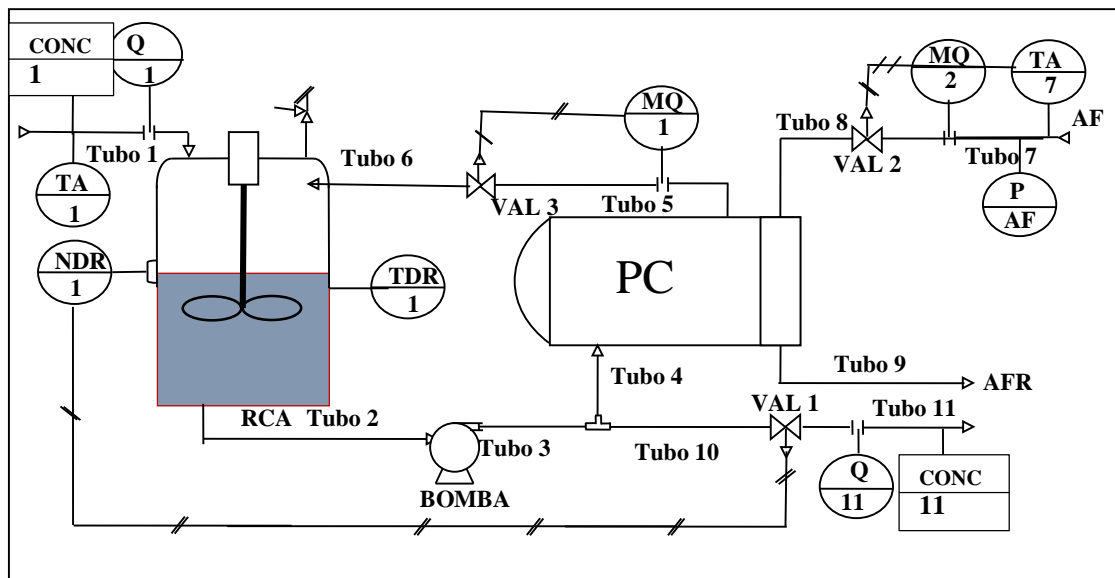


Figura 3. 1 (adaptado de [9]) Reator continuamente agitado

Em que:

CONC – Concentração;

MQ – Medidor do caudal;

NDR – Nível dentro do reator;

PAF – Pressão da água fria;

PC – Permutador de calor;

Q – Caudal;

TA – Temperatura da água;

TDR – Temperatura dentro do reator;

VAL – Válvula.

O modelo dinâmico do RCA apresentado neste capítulo é inspirado nos estudos realizados por Calado [9]. O modelo é usado para simular o processo em condições normais de operação e serve para obter o comportamento do processo durante uma situação de falhas simuladas utilizando o simulador desenvolvido no âmbito de estudos conducentes à escrita desta dissertação.

Na figura 3.2 é apresentado um diagrama elementar do reator com uma entrada/alimentação de fluido cujo caudal é Q_i (cm^3/s) e o caudal de saída Q_o (cm^3/s).

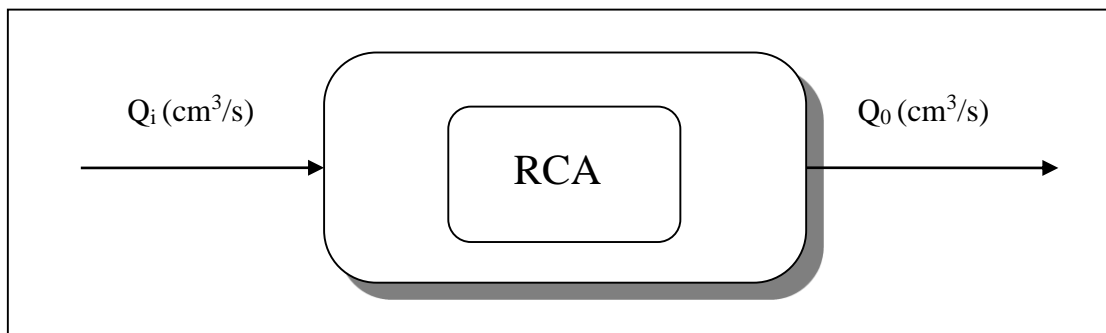


Figura 3. 2 (adaptado de [9]) **Diagrama elementar do RCA**

Assume-se que no RCA o caudal que sai tem a mesma temperatura e o composição que conteúdo do reator, sendo também assumido que a mistura perfeita ocorre no reator, bem como uma troca perfeita do calor ocorre no permutador de calor e a perda do calor para o ambiente é desprezável, assim como, o reagente e o produto final tem a mesma composição (densidade e calor específico).

O modelo é obtido a partir de balanços térmicos e de massa ao processo e é constituído pelas seguintes equações.

$$\frac{dL}{dt} = Q_1 + Q_2 - Q_3 \quad (3.1)$$

$$AL \frac{dC_a}{dt} = Q_1(C_{a0} - C_a) - r_a AL \quad (3.2)$$

$$AL \frac{dC_b}{dt} = r_a AL - C_b Q_1 \quad (3.3)$$

$$ALB_2 \frac{dT}{dt} = B_1 Q_1 (T_1 - T) - B_2 Q_2 (T - T_2) + H_r r_a \quad (3.4)$$

$$B_1 = C_{a0} \rho C + (1 - C_{a0}) \rho_0 C_0 \quad (3.5)$$

$$B_2 = \rho C(C_a + C_b) + (1 - C_a - C_b)\rho_0 C_0 \quad (3.6)$$

$$r_a = K_r C_a^n \quad (n > 0) \quad (3.7)$$

$$K_r = a_r e^{-b_r/T} \quad (3.8)$$

$$Q_2 = K_2 A_2 \sqrt{p} \quad (3.9)$$

$$Q_4 = K_4 A_4 \sqrt{p} \quad (3.10)$$

$$Q_5 = K_5 A_5 \sqrt{p_5} \quad (3.11)$$

$$Q_3 = Q_2 + Q_4 \quad (3.12)$$

$$p_f = p_0 + \Delta p \quad (3.13)$$

$$p_0 = L[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] \quad (3.14)$$

$$T_2 = \frac{C_0 \rho_0 Q_5 T_5 + Q_2 T [C\rho(C_a + C_b) + C_0 \rho_0 (1 - C_a - C_b)]}{C_0 \rho_0 Q_5 + Q_2 [C\rho(C_a + C_b) + C_0 \rho_0 (1 - C_a - C_b)]} \quad (3.15)$$

Relativamente ao modelo analítico acima apresentado foi utilizada a nomenclatura a seguir indicada,

L – Nível no reator (cm);

T – Temperatura no reator ($^{\circ}\text{C}$);

T_1 – Temperatura de entrada do reagente ($^{\circ}\text{C}$);

T_2 – Temperatura de retorno do reagente à saída do permutador de calor ($^{\circ}\text{C}$);

T_5 – Temperatura da água fria à entrada do permutador de calor ($^{\circ}\text{C}$);

A – Área transversal reator (cm^2);

Q_1 – Caudal de entrada do reagente (cm^3/s);

Q_2 – Caudal de retorno do reagente (cm^3/s);

Q_3 – Caudal do líquido à saída do reator (cm^3/s);

Q_4 – Caudal do produto (cm^3/s);

Q_5 – Caudal de água fria à entrada do permutador de calor (cm^3/s);

C_a – Concentração do reagente no reator;

C_b – Concentração do produto no reator;

C_{a0} – Concentração do reagente à entrada do reator;

r_a – Coeficiente de reação (g/s);

H_r – Constante de reação de calor (KJ/g);

ρ – Densidade do reagente (g/cm^3);

ρ_0 – Densidade do solvente (g/cm^3);

C – Calor específico do reagente ($\text{J/g}^\circ\text{C}$);

C_0 – Calor específico do solvente ($\text{J/g}^\circ\text{C}$);

K_r – Constante de reação (g/s);

a_r – Constante peculiar de reação (g/s);

b_r – Constante peculiar de reação ($^\circ\text{C}$);

K_2 – Restrição paramétrica da válvula 3 ($\text{cm}^4/\text{g}^{1/2}\text{s}$);

K_4 – Restrição paramétrica da válvula 1 ($\text{cm}^4/\text{g}^{1/2}\text{s}$);

K_5 – Restrição paramétrica da válvula 2 ($\text{cm}^4/\text{g}^{1/2}\text{s}$);

A_2 – Abertura da válvula 3;

A_4 – Abertura da válvula 1;

A_5 – Abertura da válvula 2;

p_f – Pressão do fluído à saída da bomba (g/cm^2);

p_0 – Pressão no fundo do reator (g/cm^2);

Δp – Diferença de pressão devido a bomba de circulação (g/cm^2);

p_5 – Pressão da água fria à entrada do permutador de calor (g/cm^2);

t – Tempo (s).

Os parâmetros do modelo e os valores nominais de certas variáveis do processo são dados no anexo B. Assim sendo, a partir das equações (3.9), (3.10), e (3.12) e usando os valores dos parâmetros de A , K_2 e K_4 , a equação (3.1) pode ser simplificada,

$$\frac{dL}{dt} = \frac{Q_1 - 43.4A_4\sqrt{P_f}}{300} \quad (3.16)$$

Seguindo um procedimento semelhante a partir das equações (3.7) e (3.8) e usando os valores dos parâmetros n , a_r , b_r e A , a equação (3.2) pode ser escrita na seguinte forma,

$$\frac{dC_a}{dt} = \frac{Q_1(C_{a0} - C_a) - 240C_aLe^{-66.9/T}}{300L} \quad (3.17)$$

A equação (3.3) pode ser reduzida para,

$$\frac{dC_b}{dt} = \frac{240C_aLe^{-66.9/T} - C_bQ_1}{300L} \quad (3.18)$$

A partir das equações (3.7), (3.8), e (3.9) e usando os valores dos parâmetros de A , K_2 , e H_r a equação (3.4) pode ser simplificada para,

$$\frac{dT}{dt} = \frac{B_1Q_1(T_1 - T) - 32.6A_2B_2\sqrt{P_f}T(T - T_2) + 344 \times 10^3 C_a e^{-66.9/T}}{300LB_2} \quad (3.19)$$

Substituindo a equação (3.9) e (3.11) na equação (3.15) e usando os valores de ρ , ρ_0 , C , C_0 , K_2 , K_5 , T_5 , e P_5 vem,

$$T_2 = \frac{1169.8375A_5 + 3.26A_2\sqrt{P_f}T [1.08(C_a + C_b) + 0.88(1 - C_a - C_b)]}{58.4919A_5 + 3.26A_2\sqrt{P_f} [1.08(C_a + C_b) + 0.88(1 - C_a - C_b)]} \quad (3.20)$$

Substituindo a equação (3.14) na equação (3.13) e usando os valores de ρ e ρ_0 vem,

$$p_f = L[1.2(C_a + C_b) + 1.1(1 - C_a - C_b) + \Delta p] \quad (3.21)$$

Substituindo os valores de ρ, ρ_0, C e C_0 , nas equações (3.5) e (5.6) resultam as duas equações respetivamente,

$$B_1 = 1.08C_{a0} + 0.88(1 - C_{a0}) \quad (3.22)$$

$$B_2 = 1.08(C_a + C_b) + 0.88(1 - C_a - C_b) \quad (3.23)$$

Resumidamente o modelo analítico do RCA pode ser representada pelas seguintes equações que foram usadas para simular o processo.

$$B_1 = 1.08C_{a0} + 0.88(1 - C_{a0}) \quad (3.24)$$

$$B_2 = 1.08(C_a + C_b) + 0.88(1 - C_a - C_b) \quad (3.25)$$

$$T_2 = \frac{1169.8375A_5 + 3.26A_2\sqrt{P_f}TB_2}{58.4919A_5 + 3.26A_2\sqrt{P_f}TB_2} \quad (3.26)$$

$$p_f = L[1.2(C_a + C_b) + 1.1(1 - C_a - C_b) + \Delta p] \quad (3.27)$$

$$\frac{dL}{dt} = \frac{Q_1 - 43.4A_4\sqrt{P_f}}{300} \quad (3.28)$$

$$\frac{dC_a}{dt} = \frac{Q_1(C_{a0} - C_a) - 240C_aLe^{-66.9/T}}{300L} \quad (3.29)$$

$$\frac{dC_b}{dt} = \frac{240C_aLe^{-66.9/T} - C_bQ_1}{300L} \quad (3.30)$$

$$\frac{dT}{dt} = \frac{B_1Q_1(T_1 - T) - 32.6A_2B_2\sqrt{P_f}(T - T_2) + 344 \times 10^3 C_a e^{-66.9/T}}{300LB_2} \quad (3.31)$$

O modelo do reator continuamente agitado foi desenvolvido no Matlab[®]/SIMULINK versão 2013a, mais pormenores podem ser vistos no anexo C. O próximo subcapítulo é dedicado ao controlo automático do nível e temperatura no reator, também implementado usando a mesma plataforma computacional.

3.2 Controlo das variáveis do processo

O controlo automático de sistemas tem desempenhado um papel importante no avanço da engenharia e da ciência. O controlo automático é essencial no controlo do projeto de sistemas de piloto automático aerospacial, no projeto de carros e camiões na indústria automóvel, também é essencial em operações industriais como: o controlo de pressão, da humidade, da viscosidade e do caudal nos processos industriais [24]. Como mencionado acima, a temperatura e o nível no reator são controlados por sistemas de controlo em malha fechada. No âmbito do trabalho realizado foi adotado o uso de controladores clássicos, PID. Mais de metade dos sistemas de controlo automático atualmente implementados na indústria utiliza controladores PID ou PID modificado [24]. Na mencionada publicação o autor apresenta alguns esquemas básicos dos controladores PID. Os parâmetros dos controladores tem que ser ajustados em função da aplicação específica a que vão estar associados, existindo muitas técnicas disponíveis para ajudar à sua sintonia [4, 24].

Um sistema de controlo automático em malha fechada compara o valor real da saída, ou seja, da variável controlada com o valor desejado para essa variável, determina o respetivo desvio e produz um sinal de controlo, em função da estratégia de controlo subjacente ao controlador utilizado, que tendencialmente reduzirá o desvio a zero ou a um valor pequeno [24]. As figuras 3.3 e 3.4 ilustram um sistema de controlo em malha fechada de temperatura e nível respetivamente com um controlador PID associado.

Como já anteriormente mencionado, no âmbito do caso de estudo em apreço os controladores utilizados foram controladores clássicos do tipo PID, aos quais estão subjacentes ações de controlo proporcional, integral e derivativa, aplicadas ao sinal de entrada do controlador, que no caso de um sistema em malha fechada será o erro existente entre o valor desejado para a variável controlada e o seu valor atual.

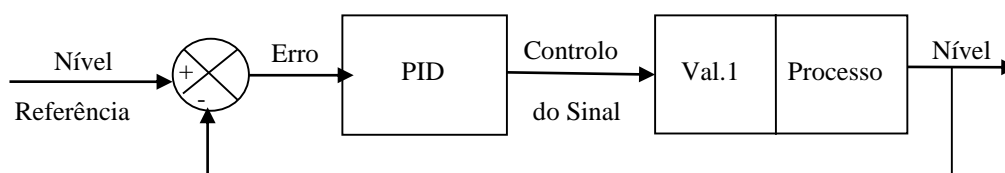


Figura 3. 3 Diagrama de bloco correspondente ao controlo do nível

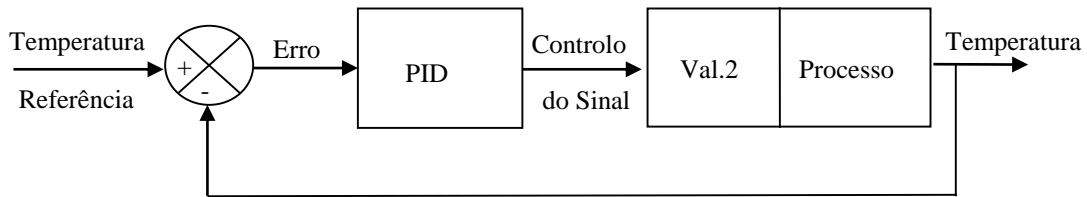


Figura 3. 4 Diagrama de bloco correspondente ao controlo da temperatura

Para regular o nível e a temperatura no reator o controlador PID é usado da seguinte maneira:

$$u(t) = K_p \left(\left(e(t) + \frac{\sum_{i=1}^t e(i)}{T_i} \right) + T_d [e(t) - e(t - 1)] \right) \quad (3.32)$$

Na lei de controlo definida pela equação (3.32) as seguintes nomenclatura são usadas:

$u(t)$ – Significa sinal de controlo no instante t ;

$e(t)$ – Significa o sinal de erro no instante t ;

K_p – É o ganho do controlador;

T_i – É o tempo de integração;

T_d – É o tempo derivativo;

t – É o instante do tempo.

Para que os controladores PID exibam um desempenho adequado à aplicação concreta a que estão associados será necessário atribuir valores aos seus parâmetros.

O processo de selecionar os parâmetros do controlador que garantam uma dada especificação de desempenho é conhecido como sintonia do controlador. Como já anteriormente mencionado, embora exista várias metodologias sistemáticas para ajudar nas sintonias dos controladores PID, no âmbito do presente trabalho foi adotado o primeiro método de Ziegler-Nichols. Ziegler e Nichols sugeriram que os ajustes dos parâmetros K_p , T_i e T_d seja baseado na resposta experimental do processo a um degrau ou no valor de K_p que resulta numa estabilidade marginal da malha de controlo, quando somente uma ação proporcional é utilizada.

Nesta conformidade, existem dois métodos denominados regras de sintonia de Ziegler-Nichols, o primeiro método e o segundo método, sendo apresentada de seguida uma breve descrição de cada um deles.

Primeiro método – no primeiro método obtém-se a resposta do processo a uma entrada em degrau unitário. Se o processo não possuir integradores nem polos complexos conjugados dominantes, então essa curva de resposta ao degrau unitário deve ter o aspeto de um S como representado na figura 3.5. É de realçar que esse método só se aplica caso a curva de resposta ao degrau unitário tiver aspeto de um S.

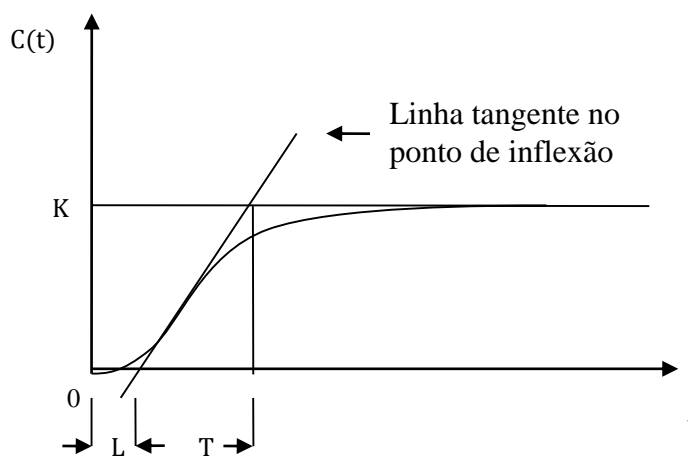


Figura 3. 5 Curva de resposta em forma de S

A curva com formato de S pode ser caracterizada por duas constantes, o atraso L e a constante do tempo T . O atraso e a constante de tempo são determinados desenhando-se uma linha tangente no ponto de inflexão da curva no formato S e determina-se a partir da interceção da linha tangente com o eixo dos tempos e com a linha $c(t) = k$ obtendo-se os parâmetros L e T como mostra a figura 3.5. Para determinar os ganhos K_p , T_i , e T_d dos controladores do tipo P, PI e PID, em função do atraso e da contante do tempo acima mencionado, Ziegler e Nichols sugeriram a utilização das fórmulas apresentadas na tabela 3.1 para efetuar a respetiva sintonia [24].

Tabela 3.1 Regra de sintonia de Ziegler-Nichos (primeiro método)

Tipo de Controlador	K_p	T_i	T_d
P	T/L	∞	0
PI	$0.9(T/L)$	$T/0.3$	0
PID	$1.2(T/L)$	$2L$	$0.5L$

Segundo método – este método pressupõe que as ações de controlo integral e derivativa são anuladas, o que significa que os correspondentes parâmetros do controlador são ajustados com os valores a seguir indicados: $T_i = \infty$ e $T_d = 0$.

Considerando-se somente a ação de controlo proporcional, aumenta-se o ganho proporcional do controlador, K de zero até ao valor crítico K_{cr} , que corresponde a que a variável controlada apresente uma oscilação sustentada pela primeira vez com o período crítico P_{cr} , conforme representado na figura 3.6. Portanto o valor do ganho crítico, K_{cr} , e o correspondente valor para o período crítico, P_{cr} , são determinados experimentalmente.

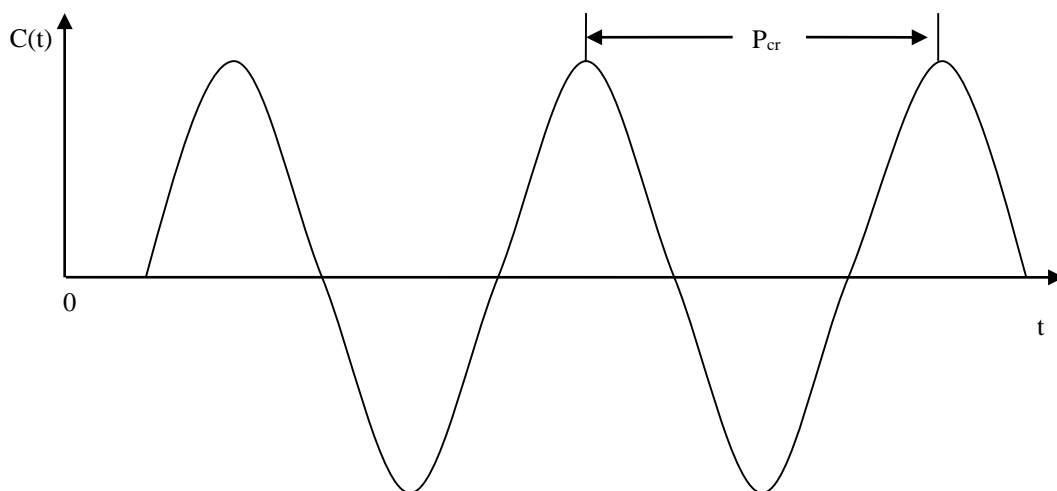


Figura 3. 6 Oscilação sustentada com período crítico P_{cr}

Com os valores de K_{cr} e P_{cr} a sintonia do controlador PID é efetuada com as fórmulas desenvolvidas por Ziegler e Nichols apresentadas na tabela 3.2 [24].

Tabela 3.2 Regra de sintonia de Ziegler-Nichols baseada no ganho crítico e no período crítico

Tipo de Controlador	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$(1/1.2) P_{cr}$	0
PID	$0.6K_{cr}$	$0.5K_{cr}$	$0.125 K_{cr}$

Os correspondentes valores dos parâmetros dos controladores são dados na tabela.3.3

Tabela 3.3 Parâmetros do controlador PID

Tabela de Parâmetros do Controlador			
Malha de Controlo	Parâmetros do controlador		
	K_p	T_i	T_d
(L)	3.6	0.1	0.025
(T)	8	0.04	0.025

3.3 Conclusão

Neste capítulo foi apresentado o modelo do reator continuamente agitado. Para tal processo, o modelo matemático está definido no subcapítulo 3.1, e foi implementado em Matlab[®]/simulink para realizar simulações do processo.

O modelo dinâmico do reator continuamente agitado é usado no próximo capítulo para simular falhas no processo em consideração.

O controlo de algumas variáveis de processo tem sido realizado por meio de controladores convencionais e durante estudos de simulação demonstraram um bom desempenho no sistema de controlo implementado

CAPÍTULO 4

APLICAÇÃO DE RNAs À DETECÇÃO E ISOLAMENTO DE FALHAS

Resumo

Neste capítulo é efetuado o estudo da aplicabilidade de redes neuronais artificiais á detecção e isolamento de falhas em processos industriais, sendo utilizado como caso de estudo o reator continuamente agitado anteriormente descrito.

4. APLICAÇÃO DE RNA À DETECÇÃO E ISOLAMENTO DE FALHA.

Neste capítulo será efetuado o estudo da aplicabilidade da RNAs, nomeadamente redes diretas com três camadas, à deteção e isolamento de falhas. Pretende-se que este objetivo seja alcançado com a utilização de uma única rede.

A rede foi treinada utilizando o algoritmo retro propagação do erro, em conformidade com o exposto anteriormente, sendo apenas consideradas sintomas de falhas abruptas para gerar os dados de treino a utilizar durante o processo de aprendizagem que é *off-line*. Todavia, sendo da maior importância na supervisão de processos industriais a deteção e isolamento de falhas o mais cedo possível relativamente ao instante de tempo em que elas se começam a manifestar no processo que estiver a ser monitorizado, como anteriormente mencionado, é desejável que através da utilização da capacidade de generalização das RNAs seja possível detetar e isolar falhas que se manifestam de forma incipiente.

Visando estudar a aplicabilidade das RNAs à deteção e isolamento de falhas, bem como, avaliar seu desempenho, foi utilizado o simulador do reator continuamente agitado, para simular o processo em condições normais de operação e em situações de falhas.

Nesta fase de estudo foram consideradas sintomas de falhas abruptas em alguns atuadores do processo em apreço. Este simulador foi desenvolvido utilizando a plataforma computacional MATLAB/SIMULINK.

Foi utilizado a *toolbox* de redes neuronais artificiais da mesma plataforma computacional, na qual foi criada a arquitetura da rede neuronal artificial utilizada e efetuado o respetivo treino *off-line*, por forma ser possível testar o desempenho, em tempo real, da metodologia proposta para detetar e isolar falhas em processos industriais.

4.1 Arquitetura da Rede Neuronal Artificial

Para a aplicação de redes neuronais artificiais ao problema proposto, foi utilizado o *toolbox* das RNAs do *Matlab*[®].

Foi implementada uma rede neuronal direta com três camadas: de entrada, saída e uma camada escondida. Para os neurónios da camada intermédia utilizou-se a função sigmoide e a função associada aos neurónios das camadas de entrada e saída é a identidade. Falta ainda definir o algoritmo de aprendizagem. Optou-se pelo uso do algoritmo retro propagação do erro.

Definida a arquitetura da rede, segue-se a fase de aprendizagem da rede a qual é tratada na secção seguinte.

4.2 Aprendizagem da Rede Neuronal Artificial

A interface gráfica *nftool* (*Neural Network Fitting Tool*) permite incorporar, criar, utilizar e exportar dados das RNAs. O acesso ao *NFTool* é conseguido digitando *nftool* na janela de comandos do *Matlab*[®]. Uma janela denominada *Neural Network Fitting Tool* (*nftool*) será aberta conforme a figura 4.1.

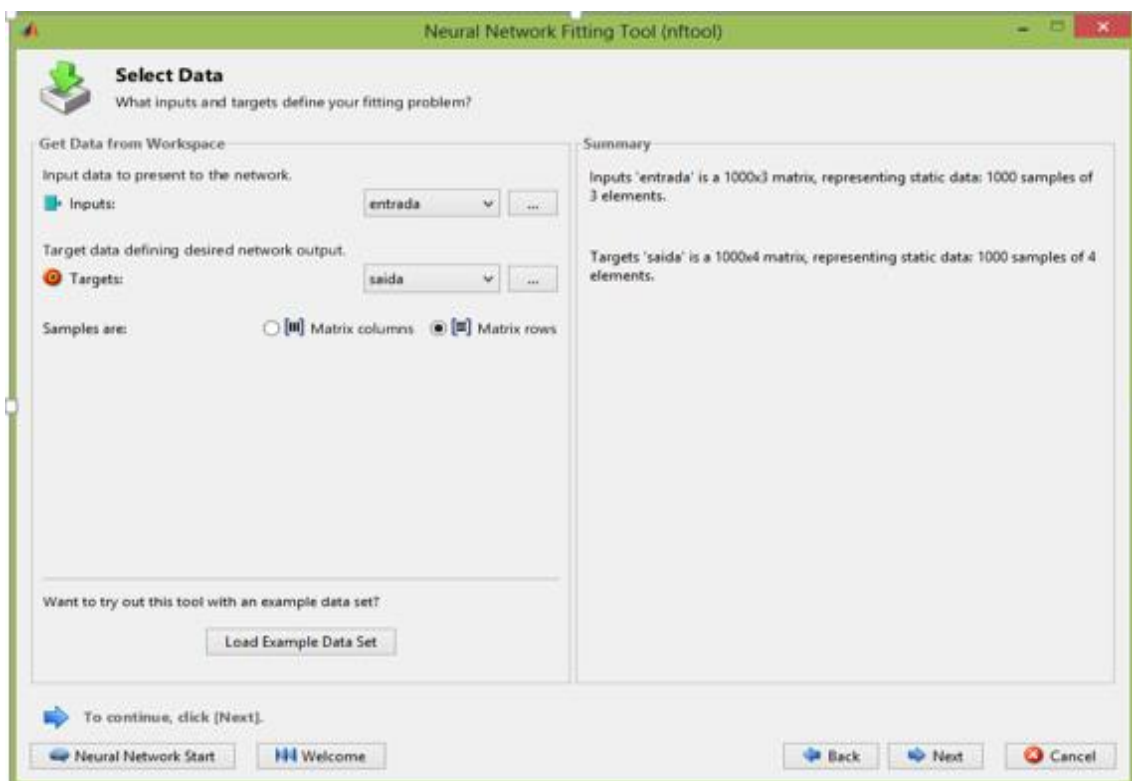


Figura 4. 1 Neural network fitting tool – definir dados

Os padrões (par entrada/saída) da RNA são definidos de acordo com a figura 4.1. Depois de selecionados os dados de treino, segue-se a fase de validação e teste. Nesta fase é definida a percentagem de dados para treino, validação e teste conforme mostrado na figura 4.2.

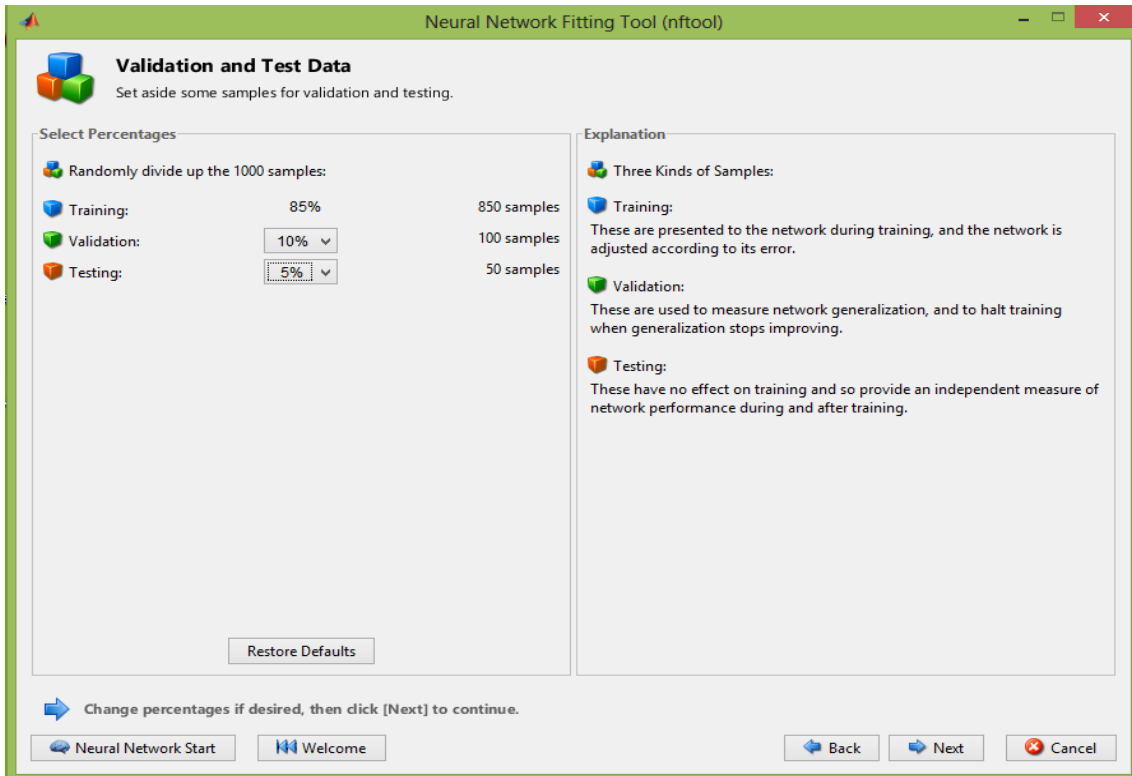


Figura 4. 2 Neural network fitting tool – validação e teste.

A próxima etapa corresponde à definição da arquitetura da rede, figura 4.3

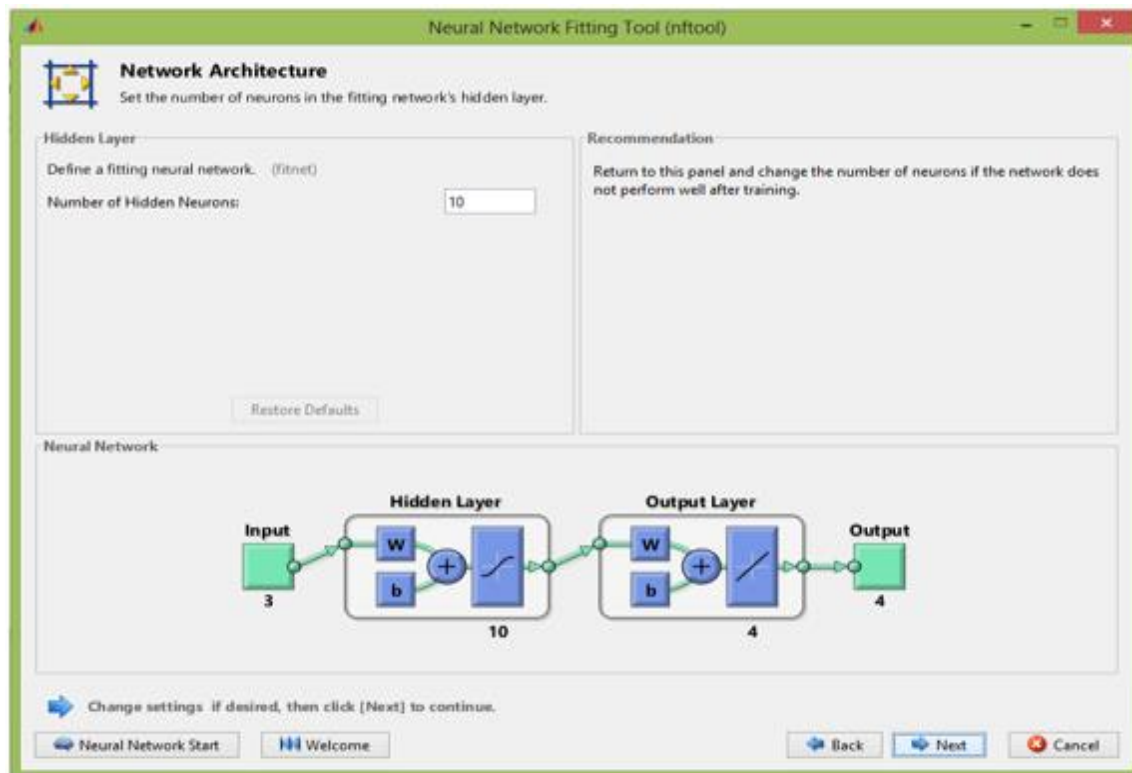


Figura 4. 3 Neural network fitting tool – arquitetura da rede

Definida a arquitetura da rede, inicia-se o processo de treino da mesma, figura 4.4.

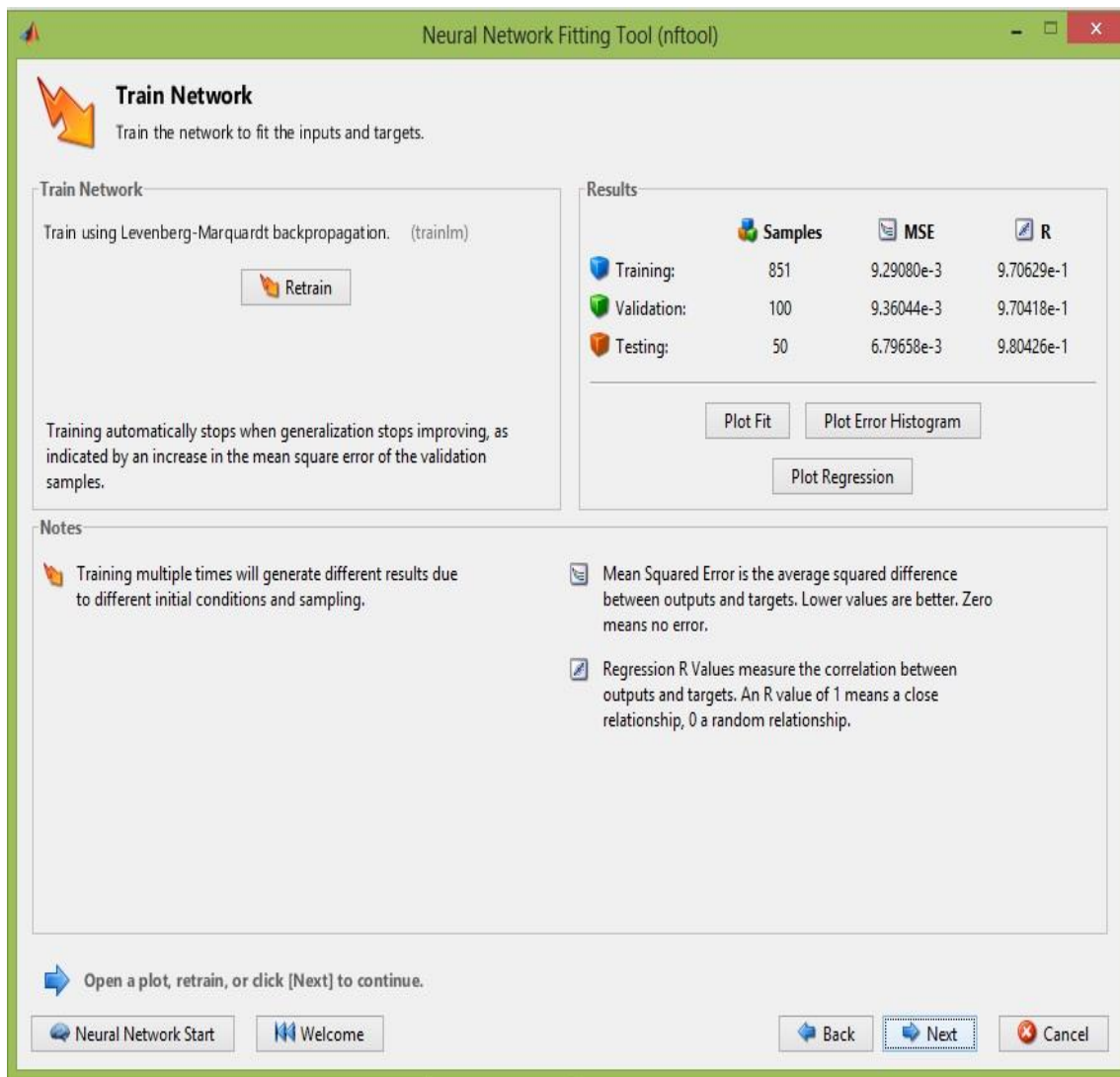


Figura 4. 4 Neural network fitting tool -- treino da rede

Uma janela será aberta, como a apresentada na figura 4.5, onde os parâmetros de treino da RNA são mostrados: números de camadas, algoritmo de treino, número de épocas, tempo de simulação entre outros. É possível representar graficamente algumas características da RNA, a citar uma delas, o desempenho do algoritmo de treino da RNA ilustrado na figura 4.5.

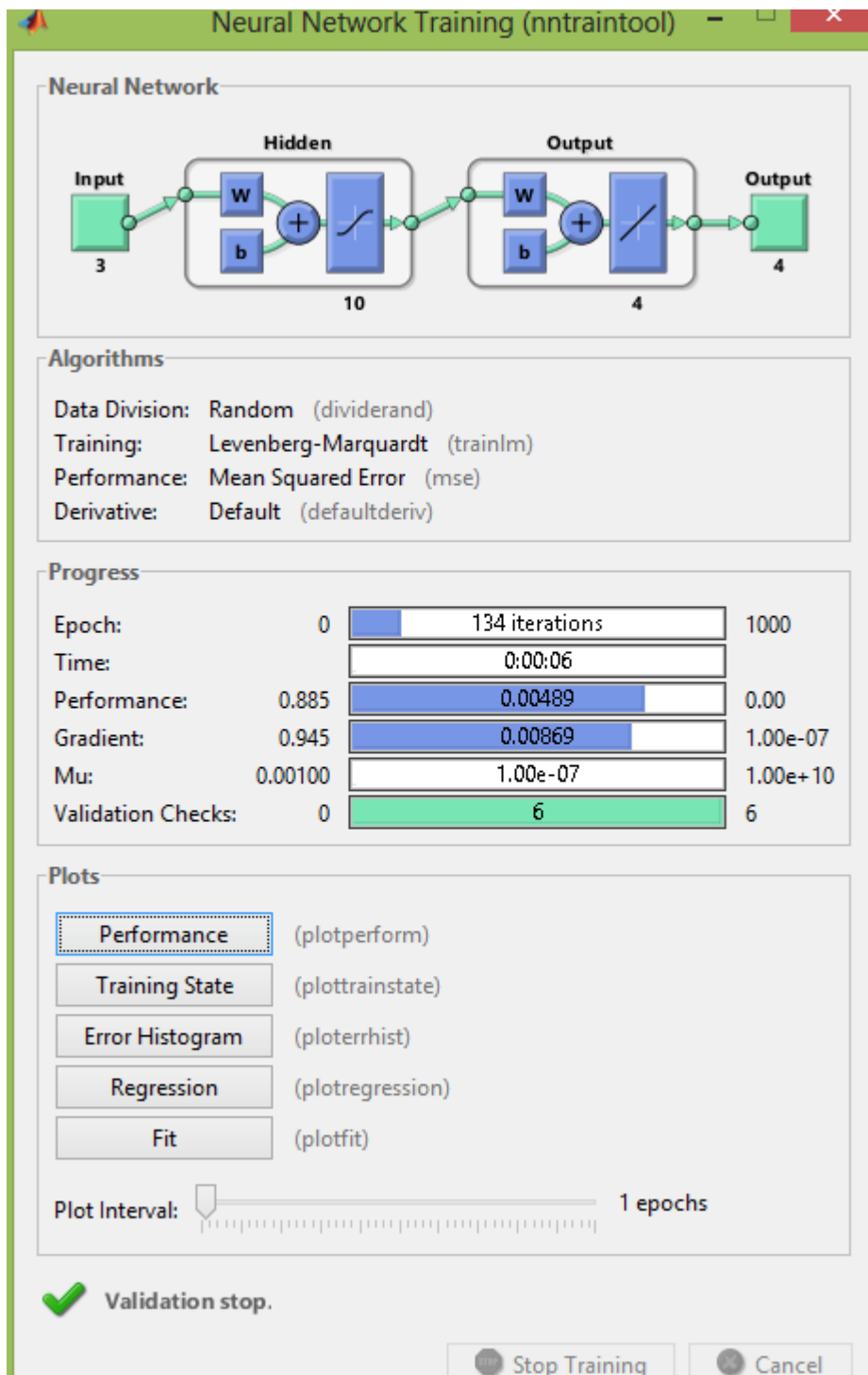


Figura 4. 5 Neural network fitting tool — característica da rede ao longo do treino

A comparação entre a saída da RNA e a saída desejada é mostrada na figura 4.6

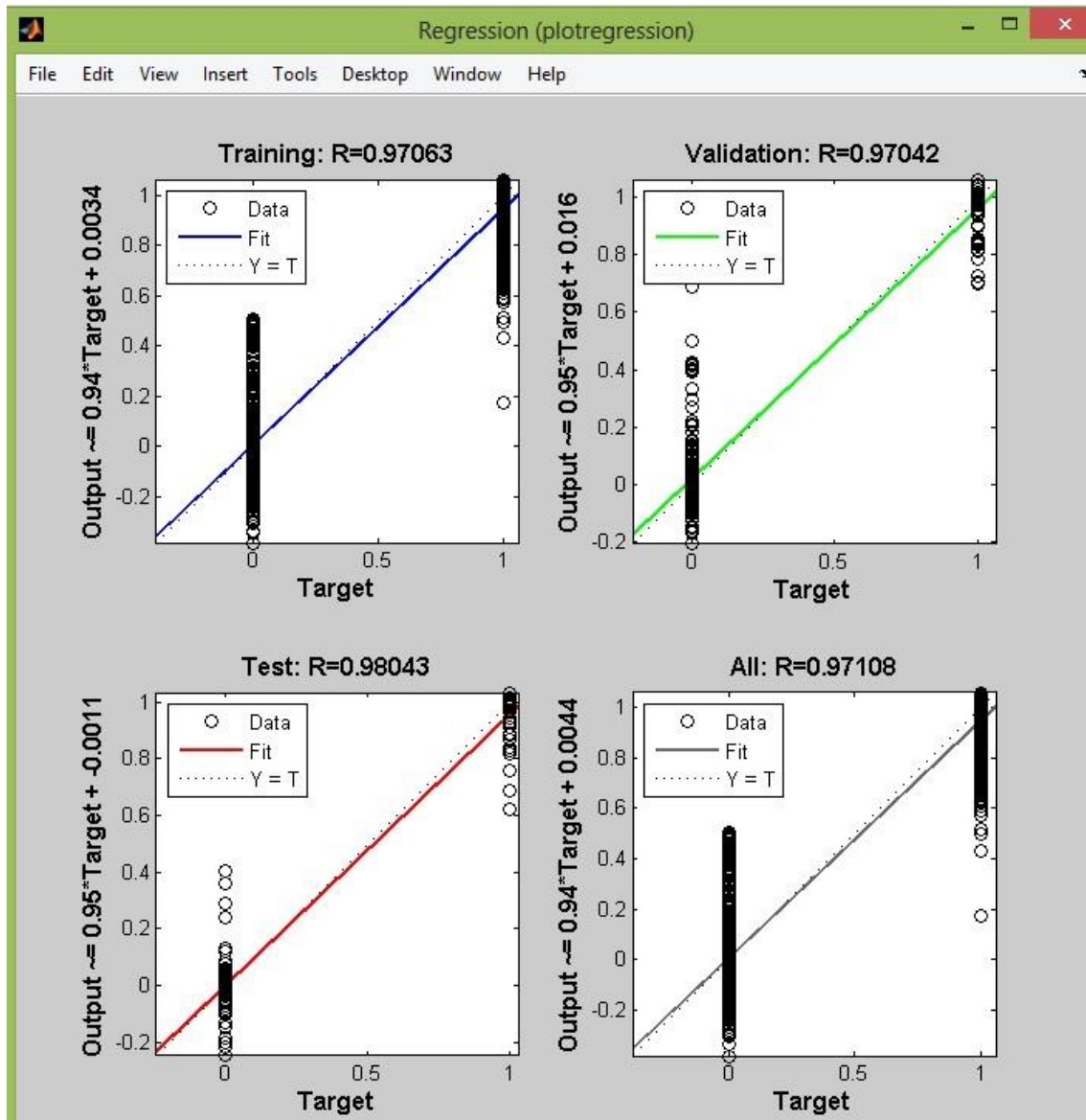


Figura 4. 6 Neural network fitting tool – desempenho da rede

Uma vez concluído o processo de aprendizagem, a RNA é ligada às variáveis de entradas provenientes do reator continuamente agitado para que se possa exercer a função de detectar e isolar falhas em tempo real quando essas ocorrerem.

Os próximos parágrafos são dedicados a detecção e isolamento de falhas em tempo real.

4.3 Detecção e isolamento de falhas

Nesta seção apresentar-se-ão os resultados obtidos utilizando uma rede neuronal artificial para detectar e isolar falhas simuladas no reator continuamente agitado

Os dados a serem processados na RNA devem conter a informação relevante da dinâmica do RCA

Os dados de entrada e saídas necessárias foram obtidos utilizando o modelo do RCA através de simulações de falhas abruptas, com o bloqueio das válvulas 1 e 2, e ainda com o processo no seu estado normal de funcionamento, ou seja, o valor da abertura das válvulas 1 e 2 é de 0.4. Na figura 4.7 estão ilustradas os dados de entrada e saída da RNA .

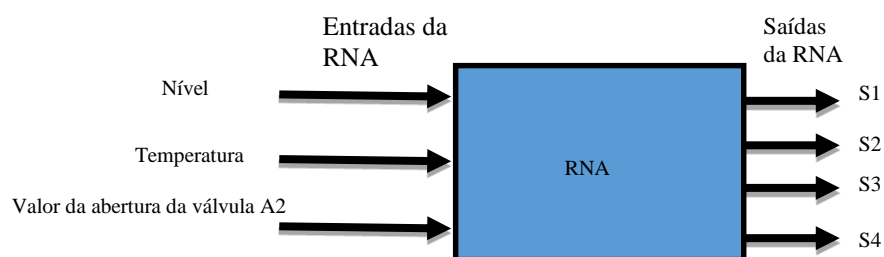


Figura 4. 7 Dados de entrada na RNA

Para avaliar a capacidade da rede detetar e isolar falhas foram realizadas algumas simulações, variando os parâmetros das possíveis falhas individualmente, geradas a partir do bloqueio das válvulas, a codificação da saída da RNA foi definida de acordo com o descrito na tabela abaixo.

Tabela 4.1 Codificação da saída da RNA

Descrição	Tipos de falha	S1	S2	S3	S4
FA	Válvula 2 bloqueada toda aberta	1	0	0	0
FB	Válvula 2 bloqueada toda fechada	0	1	0	0
FC	Válvula 1 bloqueada toda aberta	0	0	1	0
FD	Válvula 1 bloqueada toda fechada	0	0	0	1
AE	Ausência de Falha	0	0	0	0

Nas figuras abaixo são mostrados os resultados da aplicabilidade da RNA à detecção e isolamento de falhas no reator continuamente agitado. É de realçar que, deliberadamente, o estudo realizado não contempla o tratamento dos regimes transitórios porque tinha de se considerar um número infinito de situações e dificilmente conseguiria modelar todas as situações, nem foi implementado sistema de detecção de falso alarme.

Quando simuladas falha no RCA, no estado estacionário, espera-se que uma das saídas da RNA seja um ou valor próximo de um e as restantes saídas zero ou valor próximo deste.

A primeira simulação realizada foi com o processo no seu estado normal de funcionamento, ou seja as posições das válvulas 1 e 2 mantiveram a 0.4. Como tal é de esperar

que todas as saídas da RNA estejam a zero ou um valor próximo deste. Na figura 4.8 é mostrado o desempenho da RNA quando não ocorre falha no processo.

Como pode se verificar as quatro saídas da RNA ficaram a zero.

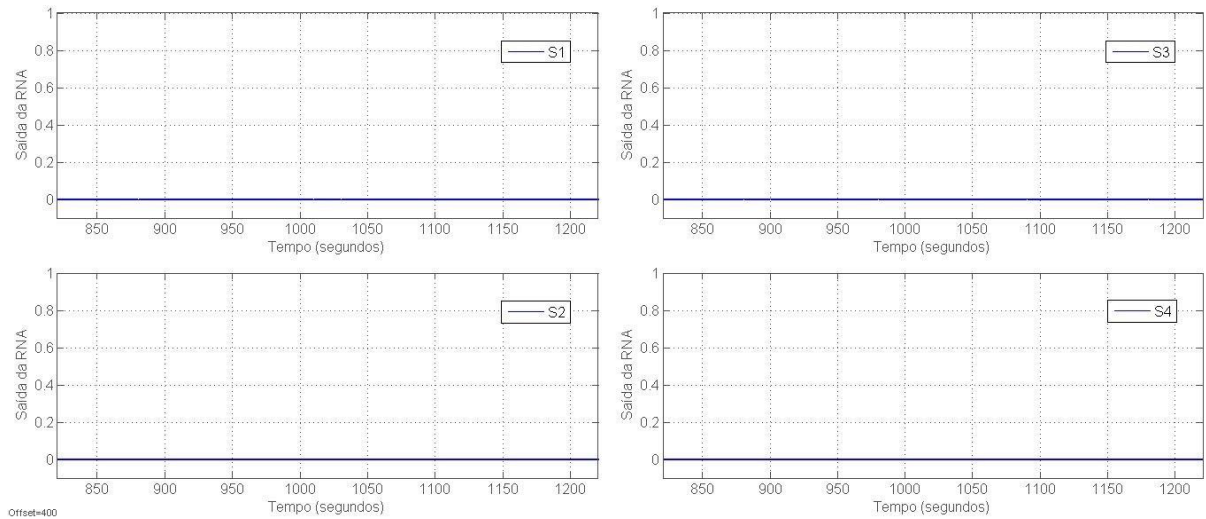


Figura 4. 8 Detecção e isolamento da falha/normal funcionamento do processo.

Existe agora, na figura 4.9, uma perturbação, simulação de falha abrupta, à partir de 1040 segundos até aos 1080 segundos. A válvula 2 foi bloqueada toda aberta, sendo assim é de esperar que entre 1040 segundos e 1080 segundos S1 seja igual a um e as outras saídas sejam zero. Como pode se verificar a saída S1 ficou a um e as outras saídas, S2,S3,e S4, ficaram a zero.

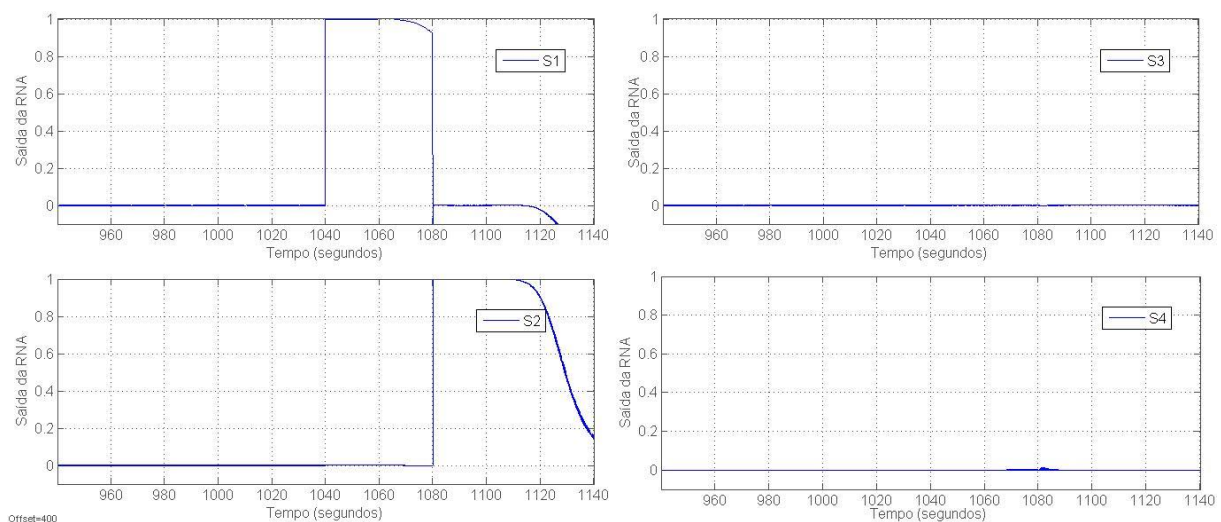


Figura 4. 9 Detecção e isolamento da falha abrupta A.

A figura 4.10 mostra o resultado da simulação da falha abrupta, a válvula 2 foi bloqueada toda fechada. Como pode ser visto na mesma, é detetada e isolada a falha no instante de 1030 segundos à 1060 segundos, a saída S2 é 1, correspondente a situação de falha B e as outras saídas S1, S3 e S4 são zero.

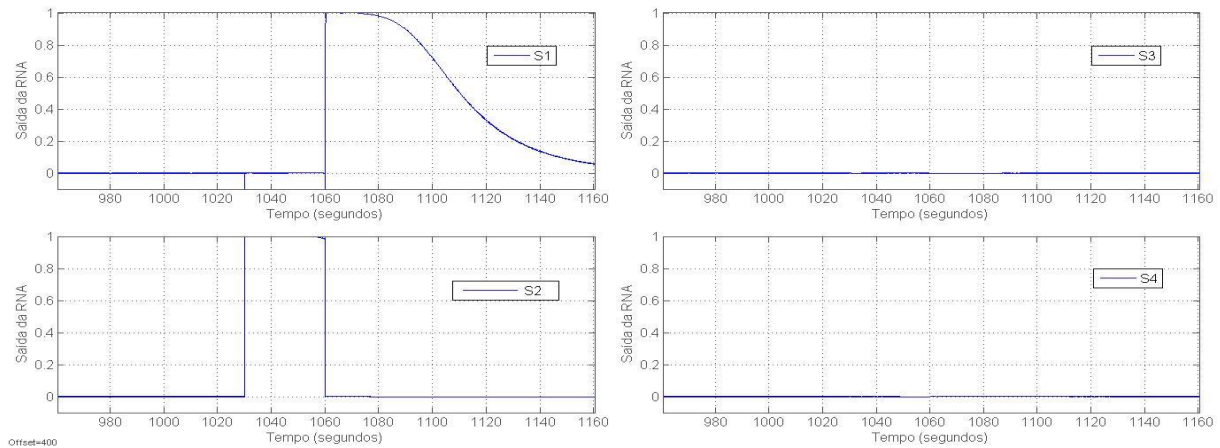


Figura 4. 10 Detecção e isolamento da falha abrupta B.

Como pode ser visto na figura 4.11 foi simulada falha abrupta na válvula 1. Para simular essa falha a válvula 1 foi bloqueada toda aberta entre 1020 segundos e 1040 segundos. A simulação da falha C faz com que a saída S3 seja um e as restantes saídas 0. A figura 4.11 mostra o desempenho da RNA na situação acima descrita.

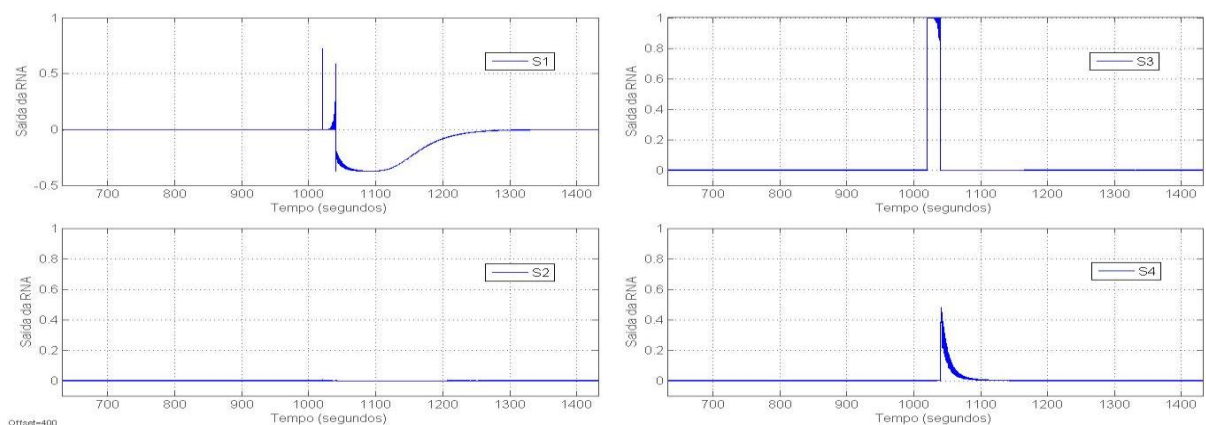


Figura 4. 11 Detecção e isolamento da falha abrupta C.

Pode-se perceber que até aos 1000 segundos o processo encontra-se numa situação normal de funcionamento, pois as quatro saídas da RNA são zero, até que nesse instante, 1000 até 1010 é provocado uma perturbação no processo, a válvula 1 é bloqueada toda fechada, com uma duração de 10 segundos.

A figura 4.12 mostra o desempenho da RNA na detecção e isolamento da falha D. Uma vez mais a RNA mostra-se apropriada para detecção e isolamento da falha. A saída S4 é um e as saídas S1,S2 e S3 são 0.

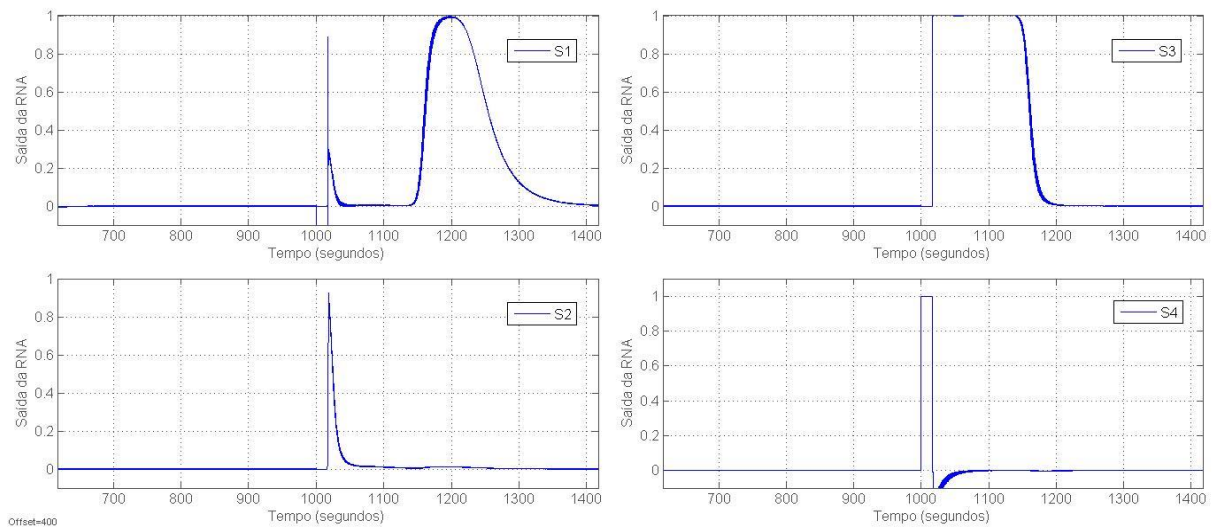


Figura 4. 12 Detecção e isolamento da falha abrupta D

A figura 4.13 representa situação de mudanças de referências entre 1200 segundos e 1800 segundos. A RNA mostrou-se incapaz de dar resposta as estas situações gerando assim falsos alarmes.

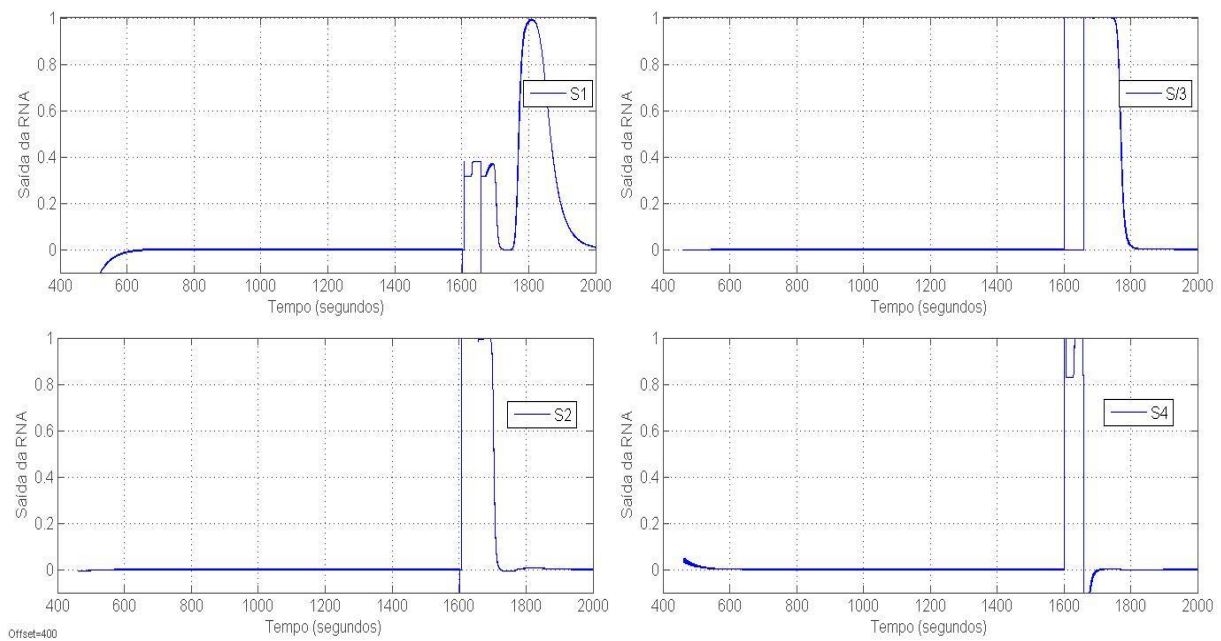


Figura 4. 13 Detecção e isolamento da falha/Mudança de Referência.

Normalmente, as redes neuronais artificiais tem a propriedade de generalizar os resultados para dados de uma gama um pouco mais ampla que o treino. Para verificar essas qualidades foram simuladas falhas com percentagem de abertura da válvula acima e abaixo da abertura da válvula para o normal funcionamento do processo e foi tido como limite máximo aceitável de afastamento para o normal funcionamento o valor 0.3 à saída da RNA.

Foi aplicado um salto de dez por cento na abertura da válvula 2 ente 1000 segundos e 1040 segundos, como pode ser visto na figura 4.14 a saída S1 é aproximadamente 0.4, valor superior ao limite para o normal funcionamento do processo, mas também, inferior ao valor considerado para falha abrupta. Quer dizer que ocorreu falha incipiente A. As saídas S2 S3 e S4 são zero.

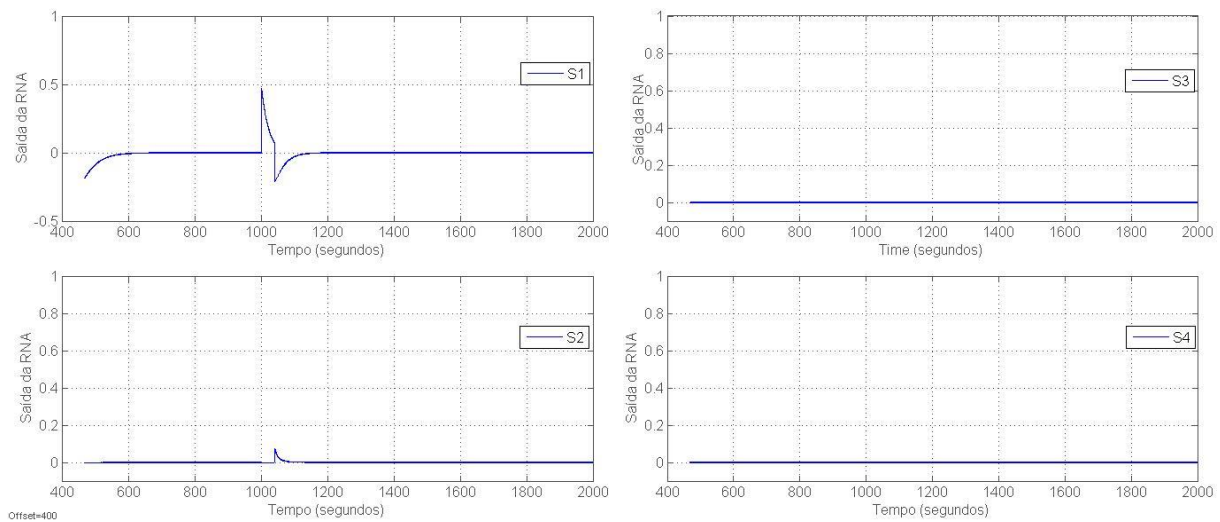


Figura 4. 14 Detecção e isolamento da falha incipiente A.

A segunda falha incipiente a ser simulada foi dado um salto de dez por cento na redução da abertura da válvula 2, cujo resultado da detecção e isolamento pode ser observado na figura 4.15. Observa-se nessa figura que à saída da RNA há afastamento do zero, entre 1000 e 1010 segundos, acima do limite considerado como normal funcionamento do processo. A saída S2 é aproximadamente 0.4 e as outras saídas são aproximadamente 0.

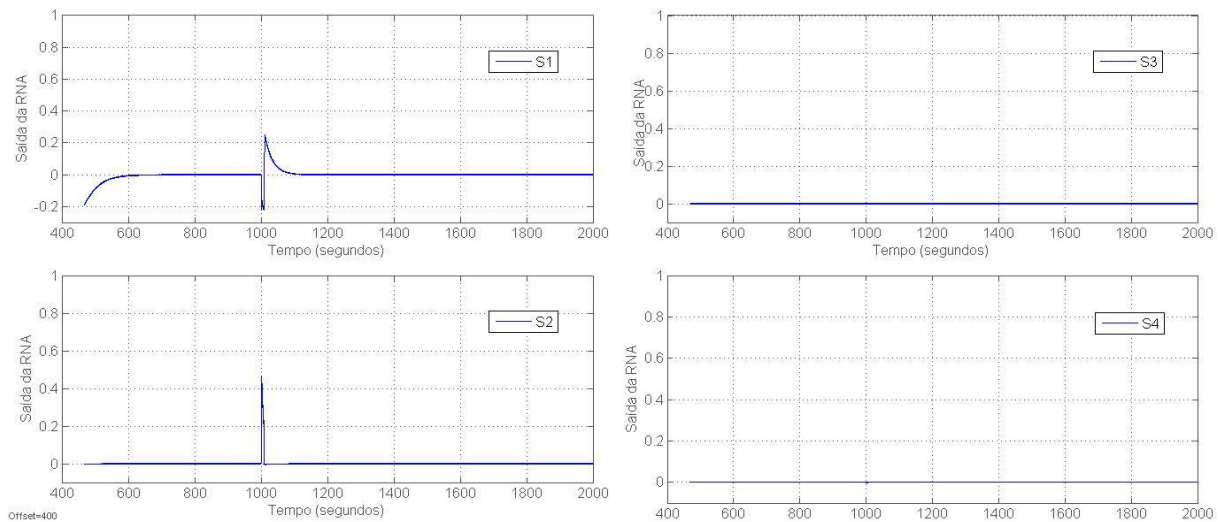


Figura 4. 15 Detecção e isolamento da falha incipiente B.

A figura 4.16 mostra o desempenho da RNA na detecção e isolamento da falha incipiente D, redução de 10% na abertura da válvula 1.

Pode-se perceber, claramente, que o processo encontra-se na situação normal de funcionamento até 1000 segundos, um salto de 10 por cento na redução da válvula 1 implica que a saída, S4, da RNA seja superior ao limite que é considerado como normal funcionamento no intervalo de tempo entre 1000 e 1010 segundos. A saída S1 é aproximadamente -0.3, a saída S2 e S3 são zero e a saída S4 é aproximadamente 0.5.

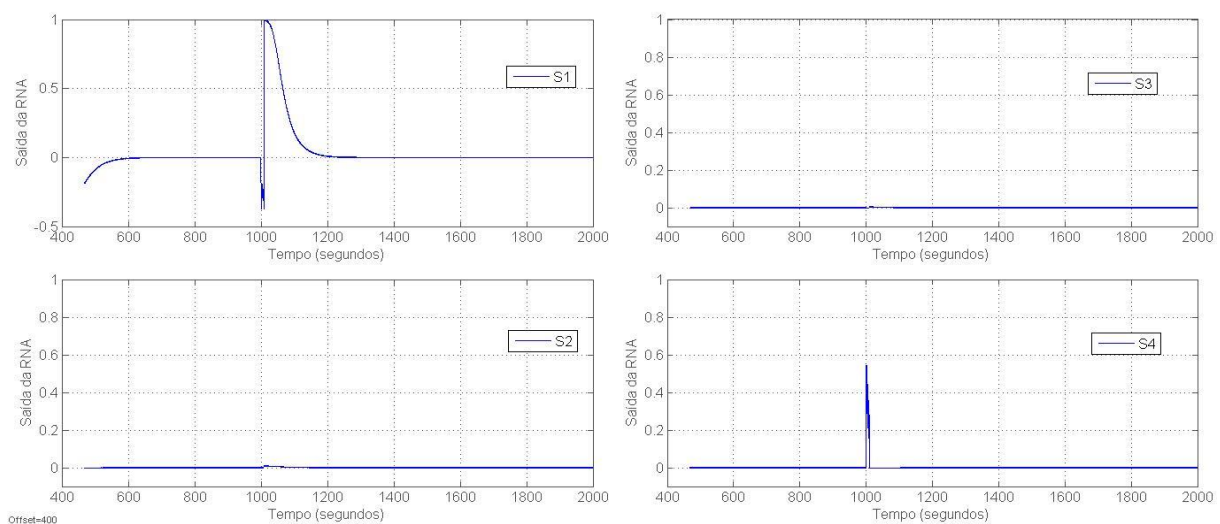


Figura 4. 16 Detecção e isolamento da falha incipiente D.

A figura 4.17 mostra momento em que ocorre a falha incipiente C, ou seja, um salto de dez por cento no aumento da válvula 1 entre 1000 segundos e 1010 segundos. As consequências desta falha são os desvios afastados do zero nas saídas S1 e S3, as outras restantes saídas permaneceram a zero. O resultado obtido para simulação dessa falha não está condizente com a tabela 4.1, pois entre 1000 segundos e 1010 segundos a saída S1 é aproximadamente 0.8 e a saída S3 é aproximadamente 0.6.

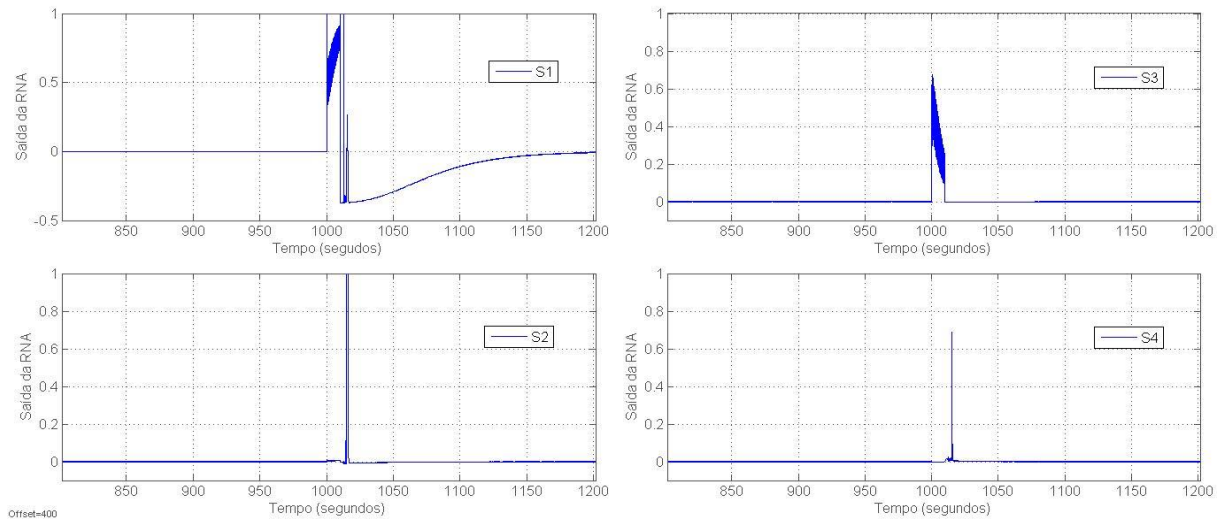


Figura 4. 17 Detecção e isolamento da falha incipiente C.

4.4 Conclusão

Neste capítulo foi apresentado estudo de simulações de falhas no reator continuamente agitado. A rede mostrou ser capaz de detetar e isolar falhas abruptas e incipientes, apesar de ser treinada apenas com sintomas de falhas abruptas. Por outro lado, A RNA não distingue mudanças de referências.

Já que foi possível detetar e isolar falhas na sua fase incipiente possibilita a intervenção do operador para evitar a propagação da mesma.

CONCLUSÃO E TRABALHO FUTURO

Esta dissertação teve como objetivo aplicação da rede neuronal artificial à detecção e isolamento de falhas em tempo real num processo dinâmico. Primeiramente foram apresentados os conceitos básicos de RNA, bem como a modelagem e algoritmo de treino, conceitos esses que serviram de suporte na aplicação da rede neuronal artificial neste trabalho.

Foi implementada RNA MLP, com o algoritmo retro propagação do erro para o cumprimento do objetivo

Os resultados obtidos confirmam a viabilidade da rede neuronal na detecção e isolamento de falhas. A rede neuronal artificial apresenta resultado satisfatório para simulações de falhas abruptas e incipientes apesar de ter sido treinada com sintomas de falhas abruptas. Uma das vantagens da RNA é a aprendizagem por um conjunto de treino, junto com um caráter universal da arquitetura da rede e o fato de não ter-se de preocupar com os parâmetros internos da rede.

Uma das tarefas da engenharia é manter os processos com maior índice de disponibilidade. A detecção e isolamento de falha incipiente é fundamental para evitar paragens intempestivas do equipamento e melhoria do índice de segurança.

Tentou-se esclarecer as capacidades e limitações de uma rede neuronal artificial. A RNA mostrou-se incapaz de lidar com situações de mudanças de referências, ou seja sempre que há mudança de referência as saídas da RNA são aleatórias.

Na ocorrência da falha incipiente C a rede não soube detetar e isolar de forma clara essa falha, na simulação dessa falha é patente que existe duas saídas, S1 e S3, que indicam ocorrência de falha no mesmo instante de tempo, o que não está condicente com o descrito na tabela 4.1.

Também não foi possível detetar e isolar a falha, incipiente e abrupta, C com duração superior a 10 segundos. Pelo que simulação dessa falha só foi durante 10 segundos.

O desempenho da rede neuronal é influenciado pelas escolhas dos parâmetros usados, entre eles estão o número de camadas ocultas, o número de neurónios em cada camada, o algoritmo de aprendizagem, função de ativação e os critérios de paragem. É aconselhável no trabalho futuro analisar o desempenho da RNA através da alteração dos vários parâmetros comparar os resultados e escolher a melhor rede de todas.

As principais contribuições deste trabalho foram análise sobre a necessidade de detecção e isolamento de falhas em engenharia de processos. A importância deste assunto cresce a medida que se confirma a tendência do aumento da complexidade nos processos mais

modernos, deve-se ter em conta as vantagens e desvantagens de diversos métodos para deteção e isolamento de falhas nos processos dinâmicos

Face aos resultados obtidos nesta dissertação sugere a aplicação de outro tipo de controlador para o controlo do nível e análise de outros métodos de deteção e isolamento de falhas como lógica de fuzzy ou neuro-fuzzy para superar algumas inconveniências apresentadas pela rede tais como incapacidade de detetar de forma sucinta a falha incipiente C e identificar as mudanças de referências.

REFERÊNCIAS BIBLIOGRÁFICAS

REFERENCIA BIBLIOGRÁFICA

- [1] Anderson, J. A. (1996) - *An Introduction to Neural Networks*. Massachusetts.
- [2] Arbib, Michael. A. (2003) - *The Handbook of Brain Theory and Neural Networks*. Massachusetts. Institute of Technology. 2ª edição.
- [3] Bequette, B Wayne. (1998) *Process Dynamics, Modeling, Analysis, and Simulation*. Prentice Hall.
- [4] Bequette, B Wayne. (2003) *Process Controls, Modeling, Design, and Simulation*. Prentice Hall.
- [5] Bishop, Christopher. M. (1995) - *Neural Networks for Pattern Recognition*. Oxford University Press, London.
- [6] Bose, N. K. Liang, P. L. (1996) - *Neural Network Fundamentals with Graphs, Algorithms and Applications*. McGraw-Hill.
- [7] Braga, A. P. Carvalho, A. P. L. F. Ludemir, B. T. (200) - *Redes Neurais Artificiais, Teoria e Aplicações*, LTC. Rio de Janeiro.
- [8] Buscema, Massimo. Tastle, William J. T. (2013) - *Intelligent Data Mining in Law Enforcement Analytics*. New Neural Networks Applied to Real Problems. Springer.
- [9] Calado, J. M. F. (1996), *On-line Fault Diagnosis of Industria Process Based on Artificial Intelligence techniques*, PhD Thesis, Control Engineering Research Centre, City University, London, May.
- [10] Fausett, L. (1994) - *Fundamentals of Neural Networks. Architectures, Algorithms, and Applications*. Prentice Hall, New Jersey, EUA.
- [11] Fonte, P. M.N. (2006), *Previsão de Potência em Geradores Eólicos*, Dissertação para obtenção do Grau de Mestre em engenharia Eletrónica e de Computadores, Instituto superior técnico, Lisboa, Março, Lisboa.
- [12] Froment, Gilbert. F, Bischoff. (1990), *Chemical Reactor Analysis and Design*. Second Edition, Wiley & Sons.
- [13] Gertler J.J (1998) - *Fault Detection and Diagnosis in Engineering Systems*. Marcel.
- [14] Hassoun, M .H. (1995) - *Fundamentals of Artificial Neural Networks*, Massachusetts Institute of Technology, The MIT Press, Cambridge, Massachusetts, London, England.
- [15] Haykin, Simon. (1999) - *Neural Networks. A Comprehensive Foundation*. 2ª Edição.

Prentice Hall, New Jersey, EUA.

[16] Hirose, Akira (2012) - *Complex - Valued Neural Networks*. 2ª edição, Springer.

[17] Hirose, Akira (2012) - *Complex - Valued Neural Networks*, advanced and applications. Wiley.

[18] Isermann, R. (2006) - *Fault Diagnosis Systems: An Introduction From Fault Detection to Fault Tolerance*. Springer.

[19] Landau, L. J and Taylor, J G (1998) - *Concepts for Neural Networks*. A Survey. Springer.

[20] Liang, N. k. B.P. (1996) - *Neural Network Fundamentals*. with Graphs, Algorithms, and Applications. McGRAW-HILL.

[21] Mendes, Mário. J. G. C. (2001) – *Diagnóstico de Avarias Baseado em Técnicas Hierárquicas Fuzzy*, Dissertação para obtenção do Grau de Mestre em engenharia Mecânica, Instituto superior técnico, Lisboa, Março.

[22] Mendes, Mário. J. G. C. (2008) – *Multi-Agente Approach to Fault Tolerante Control Systems*, Dissertação para obtenção do Grau de Doutor em Engenharia Mecânica, Instituto superior técnico, Lisboa.

Tese, Dezembro.

[23] Mondic, Danilo. Chambers, Jonathom. (2001) - *Recurement Neural Networks for Prediction: Laning Algoritms, Architetctures and Stability*. Adaptative and Learning Systems for signal Processing, Communications and Control Series. Wiley.

[24] Ogata, Katsuhiko. (2007) *Engenharia De Conrtrole Moderno*

[25] Pincipe, José C (2000). Euliano, Neil R, Lefebyre. W Curt, *Neural and Adaptative Systems- Fundamentals Trough Simulation*, John Wiley & Sons, inc, N.Y, EUA.

[26] Patton, R. J. Chen, J. (1997) - *Fault Detection, Supervision and Safety for Techical Process*. A Proceedings volume from the IFAC Symposium Kingston Upon Hull. 26-28 August .

[27] Rao, A. Ravishankar. e Cecchi, Guillermo A. (2012) - *The Relevance of Time Domain to Neural Network Models*. Springer.

[28] Venkatasubramanian, V. e S, N. Kavuri. (2003) - *A review of process fault detection and diagnosis part I: Qualitative model based methods*. Computer and Chemical Engineering,

vol. 27, pp. 293-311.

[29] Venkatasubramanian, V. e S, N Kavuri (2003) - *A review of process fault detection and diagnosis part II: Qualitative model and sear chstrategies*, Computer and Chemical Engineering, Vol. 27,pp. 313-326.

[30] Venkatasubramanian, V. e S, N. Karvuri (2003) - *A review of process fault detection and diagnosis part III: Process history based methods*. Computer and Chemical Engineering, vol. 27, pp. 327-346.

[31] http://pt.wikipedia.org/wiki/Acidente_nuclear_de_Chernobil, acesso a 23/01/2014

[32] http://pt.wikipedia.org/wiki/Acidente_nuclear_de_Fukushima, acesso a 23/01/2014

[33] http://www.hermano.com.br/cgi-bin/index.pl?ls=/estudos/redes_neurais/Perceptron&cat=00-leia.txt

[34] [http:// www.din.uem.br/ia/neurais](http://www.din.uem.br/ia/neurais), acesso a 22/03/2015

ANEXO A

TABELA COM CARATERIZAÇÃO DAS RNAS

Resumo

Nesse anexo é apresentado tabela com os paradigmas de aprendizagem, regras de aprendizagem e respectivos arquiteturas e algoritmos de aprendizagem

ANEXO A

Paradigmas de aprendizagem	Regra de Aprendizagem	Arquitetura da Rede	Algoritmo de aprendizagem
Supervisionada	Correção do Erro	Perceptrão com uma Camada	Algoritmo de aprendizagem do Perceptrão
		Perceptrão com muitas Camadas	Retro propagação Adaline -Madaline
	Boltzman	Recorrente	Algoritmo de aprendizagem de Boltzman
	Hebb	Multicamada em Avanço	Análise discriminante linear
	Competitiva	Competitiva	Quantificação do do vetor de Aprendizagem
		Rede ART	ARTMAP
Não supervisionada	Correção de Erro	Multicamada em Avanço	Projeção de Sammon
	Hebb	Multicamada em avanço ou competitiva	Análise da Componente Principal
		Rede Hopfield	Aprendizagem por memória associativa
	Competitiva	Competitiva	Quantificação de vetores
		SOM (Kohonen)	SOM (Kohonen)
		Rede ART	ART1 ART2
Aprendizagem Por reforço	Correção de erros e competitiva	Rede RBF	Algoritmo de aprendizagem RBF

Anexo A - Caracterização das RNAs

ANEXO B

PARÂMETROS E VALORES NOMINAIS DE CERTAS VARIÁVEIS DO PROCESSO

Resumo

Neste anexo é apresentado os valores e parâmetros nominais das variáveis do RCA

ANEXO B

Parâmetros	Valores
ρ	1.2 g/cm ³
ρ_0	1.1 g/cm ³
Δp	200 g/cm ²
A	300cm ²
a_r	0.8 g/sec
b_r	66.9 °C
C	0.9 J/g °C
C_0	0.8 J/g °C
C_{a0}	0.8
H_r	430 KJ/g
K_2	32.6 cm ⁴ /g ^{1/2} sec
K_4	43.4 cm ⁴ /g ^{1/2} sec
K_5	47 cm ⁴ /g ^{1/2} sec
N	1
P_5	200 g/cm ²
Q_1	300 cm ³ /sec
T_1	20 °C
T_5	20 °C

Anexo B - Parâmetros e valores nominais de certas variáveis do processo

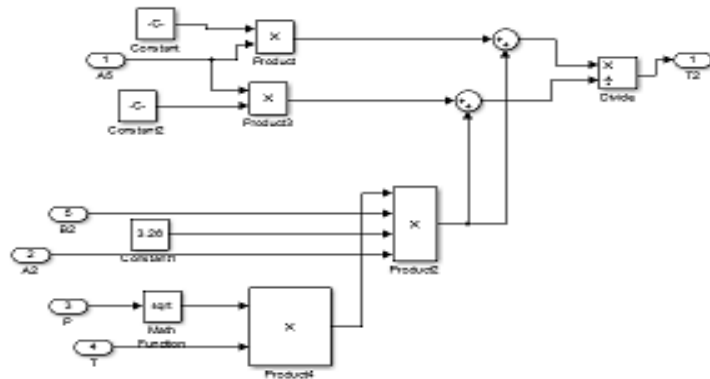
ANEXO C

DIAGRAMA DE BLOCO DO REATOR CONTINUAMENTE AGITADO

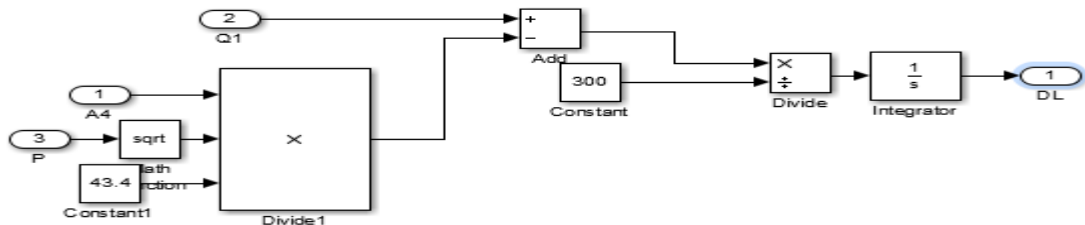
Resumo

Neste anexo são apresentados os diversos subsistemas das equações algébricas do modelo de processo.

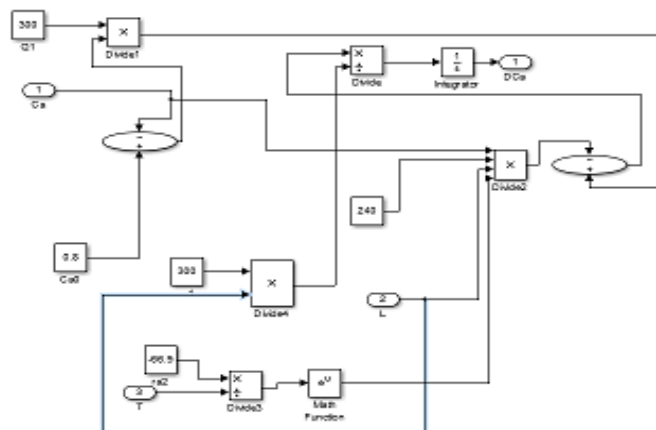
ANEXO C



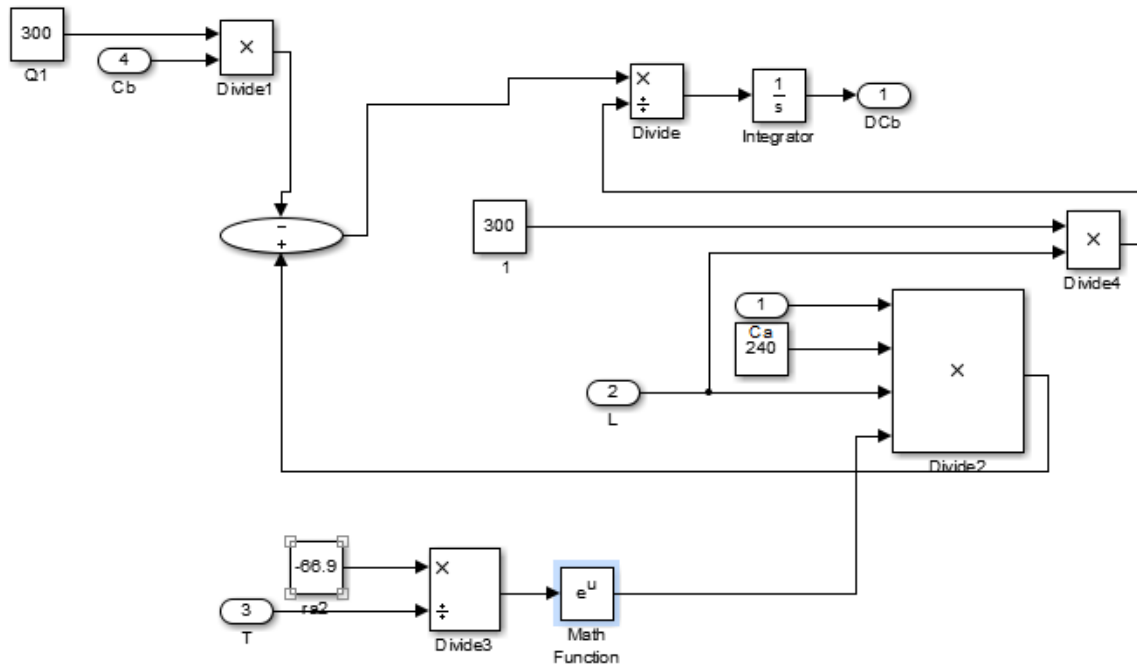
Anexo C.1 - Subsistema da temperatura2



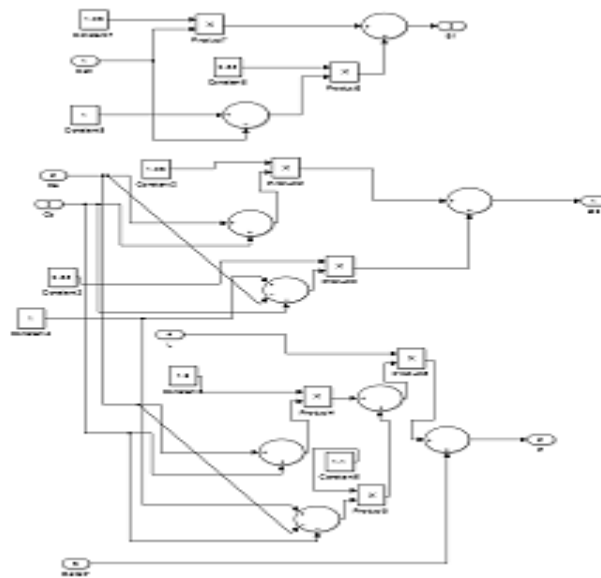
Anexo C.2 - Subsistema do nível



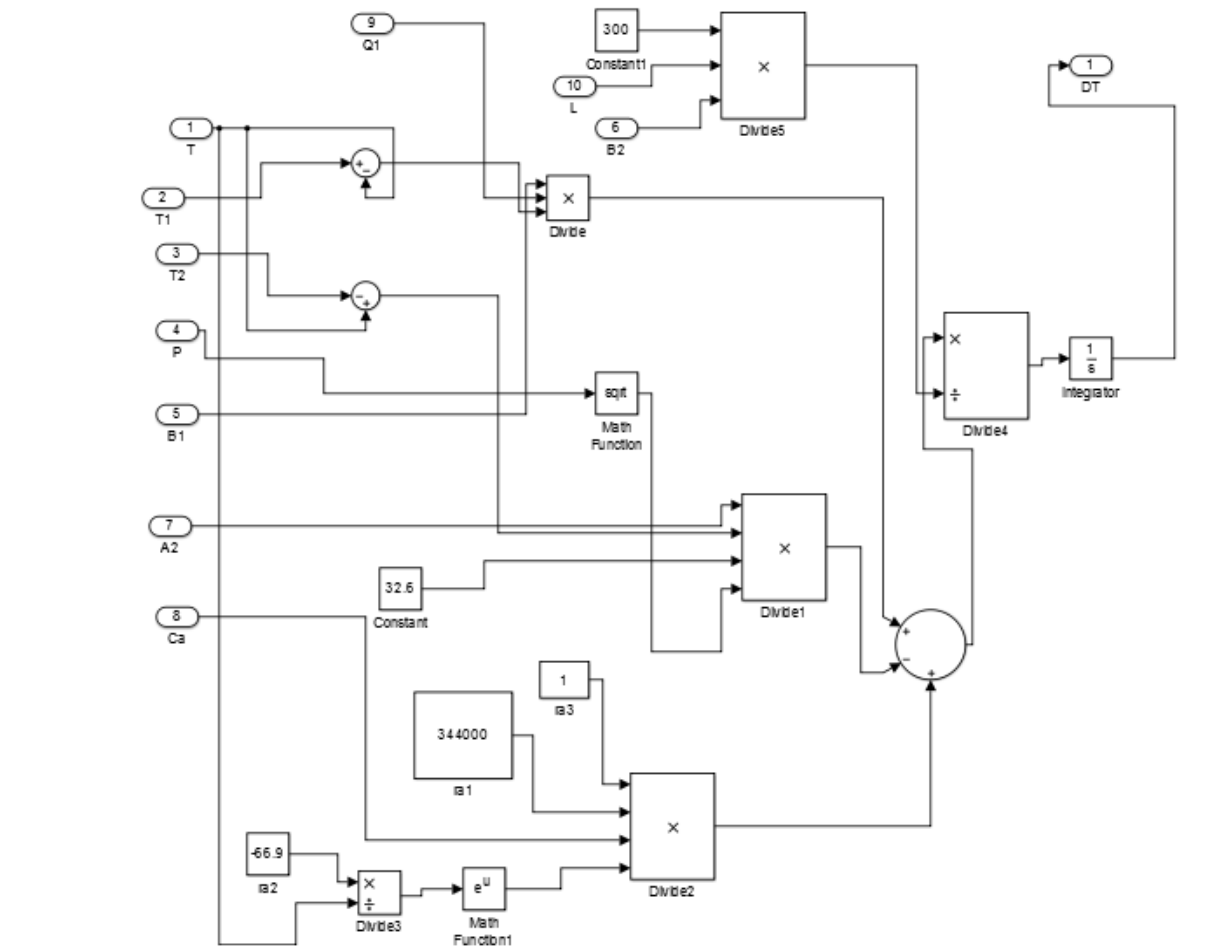
Anexo C.3 – subsistema da concentração a



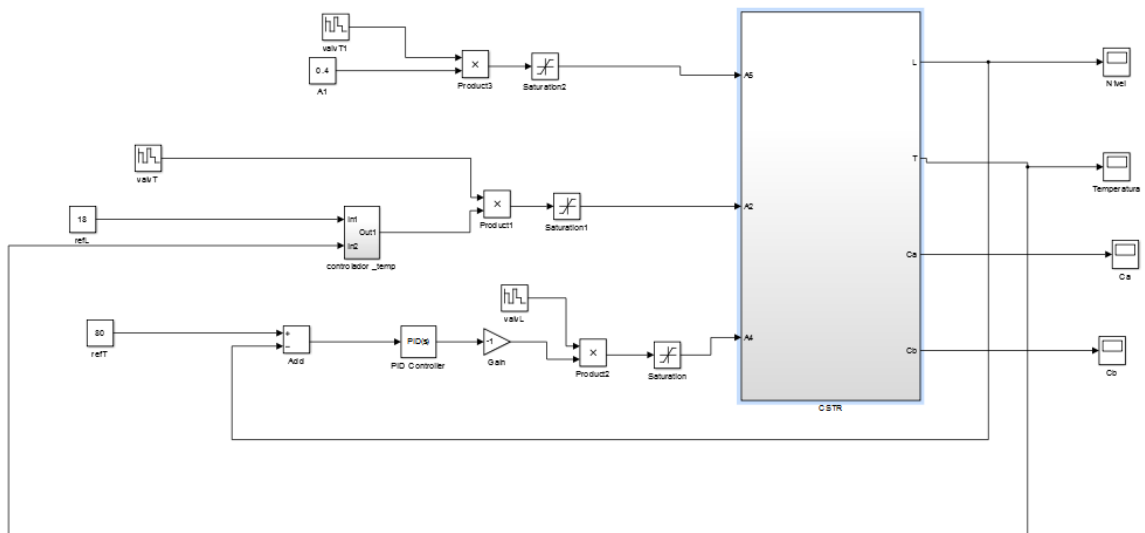
Anexo C.3 – subsistema da concentração b



Anexo C.4 – Subsistema de B1, B2 e P



Anexo C.5 – Subsistema da temperatura



Anexo C.6 – Sistema do RCA com malhas de controlo

ANEXO D

RESULTADOS COMPLEMENTARES

Resumo

Neste anexo são apresentados as tabelas para os diversos casos estudados, com erros médios quadrático do treino validação e teste das melhores redes para a arquitetura e conjunto de dados

ANEXO D

Erro médio quadrático			Erro médio quadrático			Erro médio quadrático		
1			2			3		
Treino	Validação	Teste	Treino	Validação	Teste	Treino	Validação	Teste
1.13071*10 ⁻¹	1.07085*10 ⁻¹	1.07085*10 ⁻¹	9.04088*10 ⁻²	1.00161*10 ⁻¹	6.74562*10 ⁻¹	4.94362*10 ⁻²	5.48056*10 ⁻²	5.70841*10 ⁻²
1.07592*10 ⁻¹	1.12272*10 ⁻¹	1.21824*10 ⁻¹	6.11058*10 ⁻²	6.89092*10 ⁻²	5.08110*10 ⁻²	8.76000*10 ⁻²	6.57712*10 ⁻²	8.56753*10 ⁻²
1.10744*10 ⁻¹	1.05531*10 ⁻¹	1.13397*10 ⁻¹	6.20689*10 ⁻²	5.75239*10 ⁻²	5.85554*10 ⁻²	1.77448*10 ⁻²	2.45838*10 ⁻²	2.20048*10 ⁻²
1.10529*10 ⁻¹	1.08740*10 ⁻¹	1.10988*10 ⁻¹	9.40580*10 ⁻²	8.20262*10 ⁻²	7.86318*10 ⁻²	6.11144*10 ⁻²	5.50574*10 ⁻²	5.79314*10 ⁻²
1.10619*10 ⁻¹	1.06298*10 ⁻¹	1.16959*10 ⁻¹	6.27191*10⁻²	5.25007*10⁻²	4.93355*10⁻²	5.95789*10 ⁻²	5.98594*10 ⁻²	5.76326*10 ⁻²
1.39529*10 ⁻¹	1.36915*10 ⁻¹	1.38605*10 ⁻¹	1.07682*10 ⁻¹	1.17516*10 ⁻¹	9.54216*10 ⁻²	6.52579*10 ⁻²	5.66410*10 ⁻²	6.08744*10 ⁻²
1.11621*10 ⁻¹	1.08421*10 ⁻¹	1.09567*10 ⁻¹	6.33156*10 ⁻²	6.09519*10 ⁻²	5.22230*10 ⁻²	2.09224*10⁻²	2.23358*10⁻²	2.30693*10⁻²
1.10431*10⁻¹	1.05145*10⁻¹	1.16080*10⁻¹	6.14457*10 ⁻²	5.69054*10 ⁻²	6.92042*10 ⁻²	8.72663*10 ⁻²	8.82576*10 ⁻²	8.40741*10 ⁻²
1.10065*10 ⁻¹	1.10946*10 ⁻¹	1.13040*10 ⁻¹	1.0922*10 ⁻¹	1.14651*10 ⁻¹	1.22351*10 ⁻¹	5.10041*10 ⁻²	5.08147*10 ⁻²	4.69297*10 ⁻²
1.13021*10 ⁻¹	1.10739*10 ⁻¹	1.03148*10 ⁻¹	90.2828*10 ⁻²	8.91645*10 ⁻²	9.90092*10 ⁻²	5.63267*10 ⁻²	5.92795*10 ⁻²	6.39011*10 ⁻²
1.10482*10 ⁻¹	1.15589*10 ⁻¹	1.09400*10 ⁻¹	1.0886*10 ⁻¹	1.26095*10 ⁻¹	1.13430*10 ⁻¹	5.89321*10 ⁻²	6.52187*10 ⁻²	7.6360*10 ⁻²

Anexo D.1 - Evolução do erro médio quadrático do treino, validação e teste.

Erro médio quadrático			Erro médio quadrático			Erro médio quadrático		
4			5			6		
Treino	Validação	Teste	Treino	Validação	Teste	Treino	Validação	Teste
2.01636*10 ⁻²	1.99698*10 ⁻²	2.02748*10 ⁻²	2.21372*10 ⁻²	1.76518*10 ⁻²	1.84187*10 ⁻²	7.40308*10 ⁻³	9.66071*10 ⁻³	3.79481*10 ⁻³
4.5295*10 ⁻²	4.58565*10 ⁻²	5.51038*10 ⁻²	2.10641*10 ⁻²	2.36896*10 ⁻²	2.22871*10 ⁻²	6.85523*10 ⁻³	6.04197*10 ⁻³	1.10244*10 ⁻²
1.71614*10 ⁻²	1.97784*10 ⁻²	1.87466*10 ⁻²	1.88776*10 ⁻²	1.97913*10 ⁻²	1.49471*10 ⁻²	1.52990*10 ⁻²	1.61558*10 ⁻²	1.32194*10 ⁻²
6.52938*10 ⁻²	7.78347*10 ⁻²	7.55572*10 ⁻²	1.98882*10 ⁻²	1.74863*10 ⁻²	2.18447*10 ⁻²	8.69657*10 ⁻³	9.37139*10 ⁻³	1.04720*10 ⁻²
3.84849*10 ⁻²	3.85756*10 ⁻²	3.31025*10 ⁻²	1.33877*10 ⁻²	1.61565*10 ⁻²	1.38137*10 ⁻²	2.01489*10 ⁻²	2.18238*10 ⁻²	1.64504*10 ⁻²
2.153115*10 ⁻²	2.24364*10 ⁻²	1.84265*10 ⁻²	2.31029*10 ⁻²	2.48406*10 ⁻²	2.27343*10 ⁻²	2.35366*10 ⁻²	2.49155*10 ⁻²	2.31536*10 ⁻²
1.97459*10 ⁻²	2.14203*10 ⁻²	1.65111*10 ⁻²	2.15243*10 ⁻²	2.41613*10 ⁻²	1.93912*10 ⁻²	1.43359*10 ⁻²	1.42408*10 ⁻²	1.49870*10 ⁻²
1.70791*10⁻²	1.86686*10⁻²	1.37184*10⁻²	4.21780*10 ⁻²	4.92070*10 ⁻²	3.83204*10 ⁻²	2.23805*10 ⁻²	2.44310*10 ⁻²	2.32746*10 ⁻²
2.10103*10 ⁻²	2.25363*10 ⁻²	2.16485*10 ⁻²	1.05902*10 ⁻²	8.70061*10 ⁻²	1.01084*10 ⁻²	1.21441*10⁻³	1.15262*10⁻³	6.80138*10⁻⁴
3.98128*10 ⁻²	3.56518*10 ⁻²	4.35518*10 ⁻²	1.05902*10 ⁻²	2.62163*10 ⁻²	2.61453*10 ⁻²	1.38127*10 ⁻²	1.60816*10 ⁻²	1.77448*10 ⁻²
4.56505*10 ⁻²	4.36785*10 ⁻²	4.62099*10 ⁻²	9.65317*10⁻³	7.73447*10⁻³	8.04094*10⁻²	2.00736*10 ⁻²	1.46876*10 ⁻²	3.10212*10 ⁻²

Anexo D.2 - Evolução do erro médio quadrático do treino, validação e teste.

Erro médio quadrático			Erro médio quadrático			Erro médio quadrático		
7			8			9		
Treino	Validação	Teste	Treino	Validação	Teste	Treino	Validação	Teste
4.89449*10 ⁻³	6.08847*10 ⁻³	1.75651*10 ⁻²	5.92798*10 ⁻³	4.90026*10 ⁻³	5.47597*10 ⁻³	1.81549*10 ⁻²	2.01794*10 ⁻²	1.75613*10 ⁻²
1.63428*10 ⁻²	1.83731*10 ⁻²	1.59436*10 ⁻²	3.89910*10 ⁻³	3.98304*10 ⁻³	7.91444*10 ⁻²	8.50445*10 ⁻⁴	3.61115*10 ⁻⁴	1.83293*10 ⁻⁴
2.93417*10 ⁻³	5.12524*10 ⁻³	6.40312*10 ⁻³	7.91946*10⁻³	1.02546*10⁻³	4.45147*10⁻²	5.99238*10 ⁻³	4.2593*10 ⁻³	6.03313*10 ⁻³
2.16936*10 ⁻²	1.67175*10 ⁻²	2.27200*10 ⁻²	1.66260*10 ⁻²	1.50106*10 ⁻²	1.49913*10 ⁻²	3.82433*10 ⁻³	4.81854*10 ⁻³	7.04269*10 ⁻³
4.38453*10⁻³	2.74202*10⁻³	3.34252*10⁻³	2.13564*10 ⁻²	2.52082*10 ⁻²	2.25963*10 ⁻²	3.24261*10 ⁻³	4.2593*10 ⁻³	2.37832*10 ⁻³
1.75360*10 ⁻²	1.86330*10 ⁻²	1.31355*10 ⁻²	4.21002*10 ⁻³	5.43613*10 ⁻²	6.38687*10 ⁻³	8.37113*10⁻⁴	1.49873*10⁻⁴	4.69744*10⁻⁴
1.48513*10 ⁻²	1.52943*10 ⁻²	1.61171*10 ⁻²	2.02012*10 ⁻²	1.93253*10 ⁻²	2.85956*10 ⁻²	2.63957*10 ⁻⁹	1.66667*10 ⁻³	1.24141*10 ⁻⁶
1.48163*10 ⁻²	1.76718*10 ⁻²	1.81301*10 ⁻²	2.03824*10 ⁻²	2.20991*10 ⁻²	1.92518*10 ⁻²	1.78900*10 ⁻³	4.54238*10 ⁻³	1.60588*10 ⁻³
2.10878*10 ⁻²	1.70353*10 ⁻²	2.49747*10 ⁻²	2.11271*10 ⁻³	1.28429*10 ⁻³	1.58567*10 ⁻³	6.39058*10 ⁻³	7.53028*10 ⁻³	7.00559*10 ⁻³
1.93506*10 ⁻²	1.64934*10 ⁻²	1.75753*10 ⁻²	2.57382*10 ⁻³	2.72704*10 ⁻³	4.60211*10 ⁻³	1.43175*10 ⁻³	9.75535*10 ⁻⁴	7.87410*10 ⁻⁴
4.94925*10 ⁻²	5.71728*10 ⁻²	5.38016*10 ⁻²	3.96753*10 ⁻³	2.85980*10 ⁻³	3.80936*10 ⁻³	1.20099*10 ⁻²	1.15406*10 ⁻²	1.15000*10 ⁻²
Erro médio quadrático			10					
Treino	Validação	Teste						
1.63334*10 ⁻²	1.70370*10 ⁻²	2.19531*10 ⁻²						
2.54706*10 ⁻²	2.86283*10 ⁻²	3.06141*10 ⁻²						
7.88389*10 ⁻⁴	3.58618*10 ⁻³	4.34034*10 ⁻³						
4.48352*10 ⁻³	9.72221*10 ⁻⁴	1.26254*10 ⁻⁴						
4.48352*10 ⁻³	6.17938*10 ⁻³	4.07538*10 ⁻³						
5.62900*10 ⁻³	5.03451*10 ⁻³	4.64735*10 ⁻³						
1.07179*10 ⁻²	1.61249*10 ⁻²	1.14604*10 ⁻²						
2.80745*10 ⁻²	2.66163*10 ⁻²	2.72485*10 ⁻²						
8.53792*10 ⁻⁴	4.31767*10 ⁻⁴	7.45979*10 ⁻⁴						
1.86622*10 ⁻²	1.80403*10 ⁻²	1.74120*10 ⁻²						
5.00449*10⁻⁴	1.19031*10⁻⁴	1.36771*10⁻⁴						

Anexo D.3 - Evolução do erro médio quadrático do treino, validação e teste.