

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

**IMPLEMENTAÇÃO DE UM CONTROLADOR PREDITIVO BASEADO EM
UM MODELO NEURAL ASSOCIADO A UM SISTEMA ESPECIALISTA**

Dissertação submetida à Universidade Federal de Santa Catarina
para obtenção do grau de
Mestre em Engenharia Química

Marcos Marcelino Mazzucco

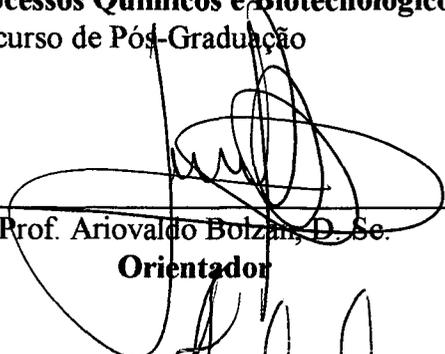
Florianópolis (SC), março de 1997.

**IMPLEMENTAÇÃO DE UM CONTROLADOR PREDITIVO BASEADO EM
UM MODELO NEURAL ASSOCIADO A UM SISTEMA ESPECIALISTA**

por

Marcos Marcelino Mazzucco

Esta dissertação foi julgada para a obtenção do título de
Mestre em Engenharia
Especialidade **Engenharia Química**
Área de Concentração **Desenvolvimento de Processos Químicos e Biotecnológicos**
e aprovada em sua forma final pelo curso de Pós-Graduação

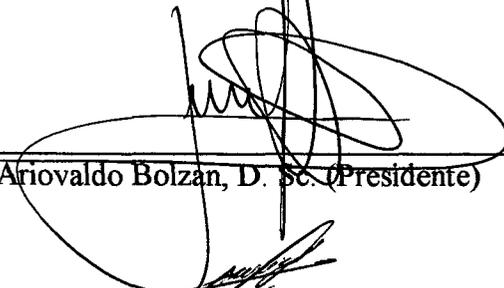


Prof. Ariovaldo Bolzan, D. Sc.
Orientador



Prof. Leonel Teixeira Pinto, D. Sc.
Coordenador do curso de Pós-Graduação e Eng. Química

Banca Examinadora :



Ariovaldo Bolzan, D. Sc. (Presidente)



Agenor Furigo Jr., D. Sc.



Ricardo Miranda Barcia, Ph. D.



Luiz Valcov Loureiro, Ph.D.

Florianópolis (SC), março de 1997.

Agradecimentos

Ao professor Ariovaldo Bolzan, pela orientação, e principalmente, pela amizade.

Ao departamento de Engenharia Química pela oportunidade concedida.

Ao CNPq pelo apoio financeiro.

Aos colegas do Departamento de Engenharia Química, pelo imenso auxílio e paciência.

Aos amigos do laboratório pelo auxílio e ensinamentos.

Ao servidor Edevilson pela paciência e préstimos.

Aos meus pais, Tito e Vivaldina pelo incentivo e confiança.

RESUMO

A evolução dos equipamentos das indústrias químicas e as restrições impostas pelos órgãos ambientais têm levado à criação de processos industriais complexos. Com esta evolução, a modelagem matemática destes processos, baseada nos princípios de conservação, tem se tornado bastante complicada, obtendo-se, na maioria das vezes, uma representação através de sistemas de equações diferenciais parciais não lineares. O tratamento destes modelos através de técnicas clássicas de projeto de controladores, baseadas na transformada de Laplace, tem apresentado problemas, principalmente em sistemas com tempo morto e fortes não linearidades. Recentemente, técnicas de modelagem matemática não fenomenológicas tem sido adotadas. A utilização de redes neurais, sistemas *fuzzy* e sistemas especialistas tem propiciado o desenvolvimento de modelos possíveis de serem utilizados na implementação de sistemas de controle para processos químicos.

Neste trabalho, implementou-se um controlador preditivo baseado em redes neurais artificiais, cuja ação é calculada através da minimização de um critério de desempenho quadrático e o modelo interno de predição é representado por uma rede neural. O controlador testado possui um caráter adaptativo. A utilização de um sistema especialista, onde vários conjuntos de pesos determinam etapas do processo permite acompanhar as principais alterações dinâmicas que ocorrem no tempo.

Para testar experimentalmente o controlador proposto, utilizou-se duas operações bastante comuns na engenharia química, o controle de nível de um tanque e o controle de temperatura de um trocador de calor. No controle de nível foi utilizado um tanque cônico em que há uma grande variação da constante de tempo do sistema em função da altura do líquido no interior do tanque. No caso do trocador de calor, em função do sistema de aquecimento disponível, tem-se dinâmicas bastante diferentes nas etapas de aquecimento e resfriamento.

Os resultados obtidos mostraram que o desempenho do controlador com sistema especialista foi superior ao mesmo controlador sem sistema especialista e bem melhor que o controlador clássico PI, tanto para mudanças no *setpoint* como para perturbações do tipo carga.

ABSTRACT

The chemical industries equipment evolution and the restrictions imposed by environmental agencies has led to an increasing complexity of industrial processes. Due to it, their mathematical modeling, which is based on conservation principles, has become very complicated and most of the time one obtains their representation by non-linear partial differential equation systems. The treatment of these models through control design classical techniques, such as the Laplace's transform, has shown problems, mainly in systems with time delaying or/and strong non-linearity. Recently, non-phenomenological mathematical modeling techniques have been used in such processes. The use of neural networks, fuzzy systems and expert systems has allowed model's development that can be used for implementing chemical process control systems.

This work presents the implementation of a predictive controller based on artificial neural networks. Its action is calculated through minimizing a quadratic performance criterion and its internal prediction model is represented by a neural network. The tested controller has an adaptive character. The use of an expert system in which various sets of weights, determining process stages, allows to follow the main process alterations over time.

In order to test the controller experimentally, two common chemical engineering operations were chosen: a) control over a liquid level in a reservoir and b) temperature control in a heat exchanger. In the first operation, a conical reservoir was used. This was subject to a major variation of the system time constant in function of the liquid height (level) within the reservoir. For the second one, due to the heat exchanger available, there were rather different dynamics in the heating and cooling stages.

The results have shown that the performance of the controller with the expert system was superior than without it. It was even more superior than the classical PI controller for both set point and step input changes.

NOMENCLATURA

| | |
|---------------|--|
| $At_{L,i,p}$ | Ativação do i -ésimo neurônio na camada L correspondente ao p -ésimo padrão. |
| $At_{L,j,p}$ | Ativação do j -ésimo neurônio na camada L correspondente ao p -ésimo padrão apresentado. |
| $At_{3,k,p}$ | Ativação do k -ésimo neurônio na camada 3 correspondente ao p -ésimo padrão apresentado. Saída da rede neural. |
| E | Erro global. Somatório dos erros entre as saídas da rede e as saídas reais. |
| E_p | Erro entre o p -ésimo padrão apresentado à rede e a saída real. |
| H | Horizonte de predição. |
| i | Neurônio na camada de entrada. |
| j | Neurônio na camada intermediária. |
| k | Neurônio na camada de saída. |
| K_c | Modo proporcional. |
| N_{NE} | Número de neurônios na camada de entrada. |
| N_{NI} | Número de neurônios na camada intermediária. |
| N_{NS} | Número de neurônios na camada de saída. |
| N_p | Número de padrões. |
| NU | Horizonte de controle. |
| p | Padrão. |
| T_i | Modo derivativo. |
| T_i | Modo integral. |
| TM | tempo morto. |
| U_{direto} | Ação de controle correspondente à válvula de vapor. |
| U_k | Ação de controle. Variável manipulada. |
| U_{max} | Valor máximo da variável manipulada. |
| U_{min} | Valor mínimo da variável manipulada. |
| $U_{reverso}$ | Ação de controle correspondente à válvula d'água fria. |

| | |
|-----------------------|--|
| $W_{(L-1),ij}$ | Peso do i -ésimo neurônio da camada (L-1) para o j -ésimo neurônio na camada L. |
| $\Delta W_{(L-1),ij}$ | Varição do peso entre o neurônio i na camada (L-1) e o neurônio j na camada L. |
| y | Variável controlada. |
| $Y_{k,p}$ | Saída real do processo. |
| y_{max} | Valor máximo da variável controlada. |
| y_{min} | Valor mínimo da variável controlada. |
| y_{ref} | Trajectoria a ser seguida para atingir o <i>setpoint</i> . Trajetória de referência. |
| y_{pred} | Predição da rede neural para o instante (k+1). |
| y_{SP} | <i>Setpoint</i> . |

Letras Gregas:

| | |
|-----------|--|
| α | Fator de ajuste da trajetória de referência. |
| η | Taxa de aprendizagem |
| λ | Fator de penalização da ação de controle. |

SUMÁRIO

| | |
|--|------|
| RESUMO | iv |
| ABSTRACT | vi |
| NOMENCLATURA | viii |
| Letras Gregas: | ix |
| INTRODUÇÃO | 1 |
| REVISÃO BIBLIOGRÁFICA | 4 |
| II.1. Inteligência Artificial - O paradigma | 4 |
| II.1.1. Sistemas Especialistas | 6 |
| II.2. Redes Neurais Artificiais | 7 |
| II.2.1. A Base Biológica | 7 |
| II.2.2. A Base Histórica | 8 |
| II.2.3. Alguns Tipos de ANN's | 10 |
| II.2.3. Aplicações | 15 |
| FUNDAMENTOS E APLICAÇÕES | 20 |
| III.1. Definições | 20 |
| III.2. Arquiteturas das Redes Neurais | 21 |
| III.2.1. Uma rede <i>Feedforward</i> multicamadas | 22 |
| III.2.2. Funções de Ativação | 24 |
| III.3. Definição Matemática de uma Rede Neural <i>Feedforward</i> | 25 |
| III.3. O Método <i>Backpropagation</i> | 28 |
| III.3.1. Equacionamento do Algoritmo <i>Backpropagation</i> | 30 |
| III.4. Aplicações das FANN | 35 |
| III.4.1. Aplicando uma FANN como modelo para o controle de nível de um tanque | 35 |
| III.4.2. Aplicando uma FANN como modelo para o controle de temperatura de um trocador de calor | 38 |

| | |
|---|------------|
| III.5. Um Controlador Genérico Baseado em um Modelo Neural..... | 39 |
| III.5.1. Esquema de Controle Utilizado..... | 45 |
| MATERIAIS E MÉTODOS | 50 |
| IV.1. O Laboratório de Controle de Processos..... | 50 |
| IV.1.1. Sistema Tanque Cônico-Cilíndrico..... | 51 |
| IV.1.2. Trocador de Calor..... | 54 |
| IV.3. O Controle..... | 57 |
| IV.4. O Treinamento..... | 61 |
| RESULTADOS E DISCUSSÕES..... | 69 |
| V.1 Resultados para o Tanque Cônico-Cilíndrico..... | 69 |
| V.1.1 Treinamento..... | 70 |
| V.1.2 Controle..... | 76 |
| V.2. Resultados para o Trocador de Calor..... | 84 |
| CONCLUSÕES E SUGESTÕES..... | 94 |
| VI.1. Conclusões..... | 94 |
| VI.2. Sugestões..... | 95 |
| APÊNDICES | 97 |
| VII.1. Especificações dos equipamentos utilizados..... | 98 |
| VII.2. Calibração para o termopar e sensor de pressão..... | 103 |
| VII.3. Exemplo da magnitude dos pesos nos modelos neurais utilizados..... | 105 |
| BIBLIOGRAFIA | 107 |

CAPÍTULO I

INTRODUÇÃO

O objetivo de se controlar um processo é encontrar uma solução que conduza um dado sistema a um estado que é o alvo do controle, denominado “referência”. Para isso, um controlador deve encontrar um grupo de ações que possa conduzir um sistema para o estado desejado, ou de um estado a outro. O problema realmente reside em determinar este grupo de ações. A teoria clássica de controle fornece ferramentas baseadas na resposta transiente dos sistemas em estudo às variações típicas do tipo degrau, rampa, senóide. De fato, estas técnicas são a base da teoria de controle de processos, tanto que na grande maioria dos trabalhos apresentados a performance das estratégias de controle propostas é comparada ao mais clássico dos controladores, o PID (Proporcional Integral Derivativo). O PID, por sua simplicidade e robustez, embora não apresente uma boa performance em sistemas com fortes não linearidades, é sem dúvida o controlador mais adotado industrialmente. Favorável a isso, há de se considerar questões históricas e evolutivas, como por exemplo a dificuldade, a algum tempo atrás, de se implantar funções matemáticas como exponenciais em *hardware*. Com a evolução dos sistemas computacionais, dos protocolos de transmissão de dados e o surgimento de poderosas estruturas de *hardware* dedicadas a funções específicas, surgiu a possibilidade do uso de técnicas mais complexas e elaboradas de controle. Com isso, técnicas como o STC (*Self-Tuning Control*) Clarke e Gawthrop (1975), o DMC (*Dynamic Matrix Control*) de Cutler e Ramaker (1980) e o GPC (*Generalized Predictive Control*) de Clarke *et alli* (1987) evoluíram rapidamente e substituíram o PID em grande parte das situações onde este era ineficiente.

Em especial, duas técnicas de controle têm recebido a atenção dos pesquisadores há algum tempo, o controle preditivo e o adaptativo. Cabe também destacar uma classe específica, os controladores preditivos baseados em modelos dos processos. Nestes, existe uma visível dificuldade de se obter modelos que representem fielmente processos complexos, o que acaba

forçando a necessidade do uso de técnicas linearizantes que podem restringir a performance de um controlador, embora em muitas situações isto seja suficiente para produzir um bom controle do processo. Enquanto o uso de modelos lineares não atende completamente as exigências de um modelo para um controlador, modelos completos podem representar um problema se sua solução não puder ser obtida em tempo real. O controle preditivo não linear têm grande potencial de aplicação industrial por apresentar sensíveis mudanças aos chamados métodos lineares. Assim, um controle preditivo baseado em uma estratégia não linear do tipo NMPC (*Non linear Model based Predictive Control*) pode ser obtido de uma Linear como o MPC (*Model based Predictive Control*) pela adoção de um modelo não linear antes que um modelo de convolução linear. É claro que isso não representa um situação tão simples de se implementar, mas o fato é que muitos estudos tem provado a viabilidade desta técnica, justificando o esforço adicional requerido. Contudo, pesa ainda uma questão importantíssima para qualquer algoritmo de controle, a já citada simplicidade. Para ser viável, uma técnica de controle deve manter um formato que permita sua implementação em tempo real. Neste ponto, grandes restrições inviabilizam o uso de modelos físicos do processo diretamente como modelo para o controlador.

Na área de processos químicos, uma nova técnica tem sido utilizada na obtenção de modelos não lineares. Como uma ramificação da ciência denominada **Inteligência Artificial** os modelos baseados em **Redes Neurais Artificiais** ou ANN's (*Artificial Neural Networks*) apresentam-se como uma possibilidade de modelar a dinâmica de um processo com os tradicionais ruídos e oscilações naturais a que um processo pode estar sujeito. Aplicações em tempo real são uma das mais significantes contribuições que as Redes Neurais Artificiais têm fornecido. A simplicidade da forma final e a capacidade de aprendizado tornaram-na uma ferramenta poderosa para a implementação de estratégias de controle que necessitem de modelagem. Tipicamente, o algoritmo de aprendizagem utilizado para o treinamento de uma ANN é o *backpropagation*, com *steepest descent* como algoritmo de otimização. Esta combinação, apesar de suficiente para a maioria dos casos, é um tanto lenta e apresenta problemas de convergência por mínimos locais. Isto pode acarretar um problema de “aprendizado” deficiente em algumas regiões dos dados de treinamento fornecidos à rede. Com isso, o que se propõe é o uso de mais que uma ANN para o mesmo processo. Isso pode ser feito mediante o acoplamento de duas redes neurais à um sistema especialista. Desta forma uma melhor aproximação do processo pode ser obtida.

Os processos químicos são particularmente difíceis de serem controlados pelo forte acoplamento de fenômenos físicos e químicos. Este acoplamento gera situações de difícil modelagem e controle. Aliado a isso, não linearidades e histerese dos equipamentos de controle acentuam o grau de dificuldade do controle destes processos. Equipamentos comuns à indústria química como reatores químicos, trocadores de calor, reservatórios, misturadores, são controlados tipicamente pela manipulação de correntes de fluidos, sejam elas refrigerantes ou componentes do processo. De qualquer maneira, as formas mais comuns são o controle de vazão através do nível e o controle da temperatura através de correntes de aquecimento ou resfriamento. Para o propósito deste trabalho, as duas situações são representadas por um reator/trocador de calor e um tanque cônico-cilíndrico.

Para a implementação da estratégia proposta o trabalho foi dividido em três fases distintas, o desenvolvimento de um *software* para implementar, em tempo real, a estratégia de controle, o desenvolvimento de um *software* para o treinamento da rede neural e o teste em dois sistemas, um tanque com dinâmica variante e um trocador de calor, onde o problema do tempo morto é acentuado.

Desta forma, este documento está dividido em seis capítulos, sendo o capítulo I representado por esta introdução. O capítulo II descreve brevemente tópicos como inteligência artificial, sistemas especialistas e redes neurais artificiais no contexto do controle de processos em engenharia química. No capítulo III, é estabelecida a fundamentação necessária para a compreensão do trabalho. A fundamentação é objetiva e situa o leitor no método utilizado. No capítulo IV, laboratório, equipamentos e *softwares* são detalhados, mostrando os aparatos e técnicas utilizados, visto esta dissertação estar baseada em resultados experimentais. No capítulo V, os resultados são discutidos e as técnicas propostas comparadas entre si e com um controlador PI clássico. Finalmente, no capítulo VI são apresentadas as conclusões e sugestões para a continuidade deste e trabalhos futuros.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

Apesar das Redes Neurais Artificiais já estarem bem difundidas em muitas áreas de estudo, inclusive na Engenharia Química, este capítulo lista alguns dos principais artigos e títulos que servem como base para o estudo das aplicações das Redes Neurais. Alguns artigos mais recentes servem como referência do que está sendo produzido e dos rumos que as aplicações neurais estão seguindo. Além disso uma breve descrição sobre Inteligência artificial e Sistemas Especialistas será tecida, sem pretender fixar atenção nestes tópicos.

II.1. Inteligência Artificial - O paradigma.¹

Inicialmente, a Inteligência Artificial (IA) se restringia aos estudiosos das ciências da computação com o uso de sistemas especialistas. O fato é que a IA têm conquistado espaço e confiança de muitos cientistas nas mais diversas áreas. A IA teve ascensão por volta de 1960 com a criação do LISP que, realmente, foi a primeira linguagem voltada para estudos em nesta área. Em 1961 Marvin Minsky com o artigo *Steps Towards Artificial Intelligence (The Institute of Radio Engineers Proceedings, 1961)* definitivamente introduziu o conceito de Inteligência Artificial inaugurando a era dos estudos de IA. Em 1964 o famoso programa ELIZA, escrito por Joseph Weizenbaum, que imitava o comportamento de um psicanalista teve um forte impacto, dando a impressão de que estava próxima a época das “máquinas inteligentes”. Já em 1970, com

¹ Informações extraídas de:

SCHILD, Herbert (Autor), SILVEIRA, Cláudio G (Tradutor), **Inteligência Artificial Utilizando Linguagem C**, McGraw-Hill Ltda, São Paulo, 1989.

SOUCEK, Branko, **Neural and Intelligent Systems Integration**, 1ª ed., Wiley Interscience, Canada, 1989.

um bom crescimento da capacidade computacional, ao contrário do que se esperava, a produção das “máquinas inteligentes” parecia estar mais distante. O problema agora residia no desenvolvimento de algoritmos mais velozes. Um exemplo típico era o simples problema da ordenação de um grupo de números. No uso do método “Bolha” o tempo de processamento é proporcional a N^2 , o que significa que independentemente do poder computacional a lentidão do método condena sua utilização. Por exemplo, se a ordenação de um grupo de 10 elementos gasta 1 segundo, uma lista de 100 elementos gastaria 100 segundos. Um método alternativo desenvolvido é o “QuickSort” onde o tempo de processamento é proporcional a $N^{1.2}$, o que representa uma melhora significativa. Paralelo aos trabalhos com LISP (EUA), na Europa outra linguagem, criada por Alain Colmerauer (França, 1972), o PROLOG, guiou os estudos em IA. Com o passar do tempo e a disseminação das técnicas de IA para várias áreas de conhecimento, linguagens com C, FORTRAN, BASIC, PASCAL também foram utilizadas largamente. Com a crescente necessidade de capacidade computacional, com o desenvolvimento de técnicas como *Fuzzy*, Algoritmos Genéticos, Redes Neurais Artificiais e com o inerente paralelismo do atual acoplamento de várias técnicas, linguagens específicas para o tratamento de processamento paralelo foram desenvolvidas e povoam as literaturas específicas. Hoje, pode-se citar como alguns dos ramos da IA:

- Sistemas Especialistas
- Redes Neurais
- Algoritmos Genéticos
- Algoritmos Inteligentes
- Incerteza e Lógica Difusa
- Processamento de Linguagem Natural

Neste trabalho, o enfoque está em Redes Neurais, com Sistemas Especialistas em segundo plano, portanto informações sobre as demais ramificações não são pertinentes.

II.1.1. Sistemas Especialistas²

Sistemas especialistas são programas que através de uma base de conhecimentos específica procuram imitar o comportamento de especialistas humanos. Mais especificamente, são sistemas projetados para atuar com alta eficiência em situações específicas.

Atualmente, muitos produtos de *software* e *hardware* são sistemas especialistas. Enquanto um *software* clássico é baseado em computação sequencial e determinística, um sistema especialista é baseado em computação paralela. Assim uma dada condição ambiente ativa um módulo para realizar uma tarefa específica. O número de módulos é definido em tempo de escrita pelo programador especialista no assunto tratado. Em medicina muitos sistemas especialistas foram projetados para auxiliar o diagnóstico de doenças. O exemplo mais clássico de um sistema especialista de sucesso é o TEIRESIAS/MYCIN, que foi projetado para o diagnóstico de doenças para infecção bacterial. O MYCIM executa o diagnóstico, enquanto o TEIRESIAS faz a interface entre o médico especialista e o MYCIN. A vantagem é que um sistema especialista como este não sofre de *stress*, problemas de ambientação e independentemente da capacidade do especialista humano sempre age da mesma forma.

Segundo Seeborg *et alli* (1989) a estrutura de um sistema especialista pode ser dada conforme a figura II.1.

² Informações extraídas de :

SEEBORG, D. E., EDGAR, T. F., MELLICHAMP, D. A., Process Dynamics and Control, John Wiley & Sons, Canada, 1989.

SCHILD, Herbert (Autor), SILVEIRA, Cláudio G (Tradutor), **Inteligência Artificial Utilizando Linguagem C**, McGraw-Hill Ltda, São Paulo, 1989.

SOUCEK, Branko, **Neural and Intelligent Systems Integration**, 1ª ed., Wiley Interscience, Canada, 1989.

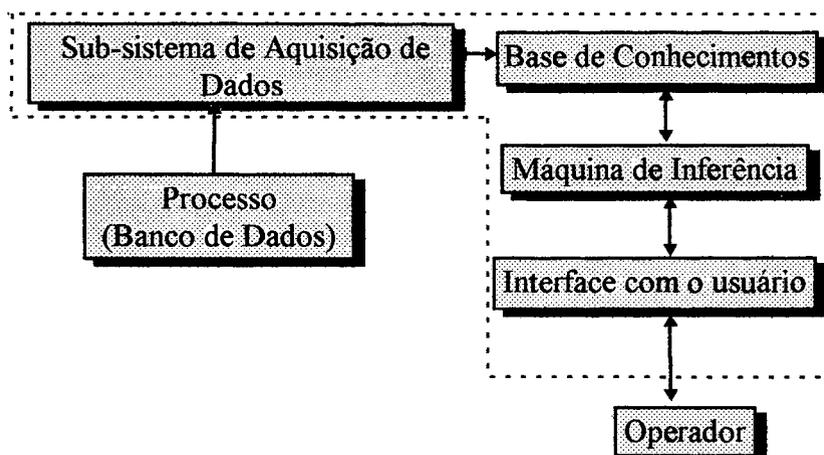


Figura II.1 - Um exemplo de estrutura de um sistema especialista.

Desta forma, um sistema especialista pode auxiliar um operador a tomar uma decisão no controle de um processo, ou simplesmente tomar a ação que for julgada correta. Um exemplo simples do uso de sistemas especialistas em controle de processos é a detecção e o diagnóstico de falhas.

II.2. Redes Neurais Artificiais.

Segundo Maren *et al.* (1990) REDES NEURAIIS são sistemas computacionais, de *software* ou *hardware*, que imitam as habilidades computacionais dos sistemas biológicos humanos usando um grande número de neurônios artificiais interconectados. Este, apesar de bem simplificado, é um conceito que traduz a principal característica das ANN's, sua base em modelos biológicos.

II.2.1. A Base Biológica.

O sistema nervoso recebe milhares de sinais de diferentes receptores do corpo. Estes sinais são processados e comparados com as informações armazenadas por hereditariedade ou

por experiências vividas, gerando uma resposta. Dos milhares de sinais recebidos, nem todos podem ser relevantes para uma determinada ação. Desta forma, a lógica sugere que exista um processamento dos sinais, a partir dos quais um novo grupo de sinais é gerado, sendo então enviados (agora em número reduzido) para regiões específicas do cérebro onde a ação apropriada será gerada. O cérebro é a parte do sistema nervoso que fornece a maior quantidade de ações necessárias para o funcionamento do corpo. Ele contém dois tipos especiais de células: as *glias* que são responsáveis pelo suporte e nutrição do sistema nervoso, e os *neurônios* que são dedicados à atividade nervosa. O cérebro humano é composto de aproximadamente 10^{11} neurônios. Cada neurônio é composto de um corpo dotado de ramificações denominadas *dendritos*. Os dendritos conduzem os sinais de uma extremidade para o corpo da célula. Uma outra ramificação denominada axônio transmite um sinal do corpo para a outra extremidade da célula. Desta forma, um axônio conduz um sinal de um neurônio, para os dendritos de outro neurônio, gerando as denominadas *sinapses*. Um axônio pode ainda estar ligado a um outro axônio ou ao corpo de um neurônio. Um neurônio comunica-se com um outro pela transferência de energia elétrica através dos *neurotransmissores*. Os neurotransmissores são produtos químicos que ocupam o *gap sináptico*, que é o espaço compreendido entre duas células. Através de mecanismos de polarização-despolarização sobre a membrana dendrítica a possibilidade de um estímulo ser transmitido pode aumentar ou diminuir. Já as sinapses tem importância fundamental na magnitude do sinal a ser transmitido. O mecanismo desta transmissão é denominado *facilitação*. Este mecanismo é baseado na lei de Hebb onde *a força de uma conexão sináptica é incrementada se esta estiver entre dois neurônios excitados simultaneamente*. A idéia é que as sinapses têm a capacidade de estocar informações. Portanto, cada vez que um sinal passa através de uma sequência de sinapses, cada sinapse torna-se mais capaz de conduzir um sinal da próxima vez que este surgir, daí o nome *facilitação*.

Acredita-se que estas informações sejam suficientes para explicar a plausibilidade biológica das redes Neurais, sem muito aprofundamento nas teorias e mecanismos cerebrais. Estas informações foram extraídas de Soucek (1989) e Sari (1994).

II.2.2. A Base Histórica.

Em 1943, (Carvalho filho *et alli*,1993) o primeiro modelo matemático de um neurônio foi proposto por McCulloch e Pitts. Neste modelo o neurônio era suposto ativado ou desativado

correspondendo, na álgebra booleana, a 1 e 0, respectivamente. Assim, um neurônio produz uma saída 1 somente se a soma de suas entradas for maior que um limiar. De outra forma a saída é 0.

Em 1949, Hebb enunciou uma regra para o aprendizado em redes neurais. A regra diz que se um estímulo de entrada influencia o estímulo de saída, então o peso da conexão entre os neurônios deve ser incrementado (De Souza Jr., 1993).

Em 1951, D. Edmonds e M. Misky implementaram um máquina denominada *Snark* com muitos motores, garras elétricas automáticas, tubos e conexões. O sucesso desta implementação culminou em 1954 com a primeira simulação de redes neurais em computadores digitais (De Souza Jr., 1993).

Em 1960 B. Widrow e M. E. Hoff desenvolveram a ADALINE (*ADaptive Linerar Element*) e posteriormente MADALINE (*Multiple ADaptive Linerar Element*) com aplicações em filtragem e equalização de sinal e reconhecimento de padrões.

Em 1962, F. Rosemblat publica seu livro *Principles of Neurodinamics* sugerindo um novo conceito para a modelagem do cérebro e da Inteligência Artificial (IA), o *percertron*. *Perceptrons* são redes de neurônios de limiar baseados no modelo de McCulloch e Pitts (Rita, 1995).

Em 1969 S. Papert e M. Misk, que defendiam o processamento simbólico, contrapondo-se ao *perceptron*, para a modelagem em IA, demonstraram em seu livro *Perceptrons* que os *perceptrons* de uma camada não podiam separar padrões de entrada em duas classes, a menos que as duas classes fossem linearmente separáveis (um hiperplano pudesse dividir o espaço dos padrões de entrada (Simpson 1990, segundo Carvalho Filho *et alli*, 1993)). Isso reduziu a velocidade dos estudos em ANN por alguns anos.

Em 1974, P. J. Werbos estabeleceu o algoritmo *backpropagation* introduzindo a mais usada ferramenta de toda a história da IA.

Em 1979, S. Grossberg propõe a Teoria da Ressonância Adaptativa (ART- *Adaptive Resonance Theory*), caracterizando as redes como sistemas constituídos de uma camada de unidades adaptativas competitivas com função de ativação sigmoideal (Hietch-Nielsen, 1990 segundo Carvalho Filho, 1993).

Em 1982, John Hopfield em seu trabalho *Neural Networks and Physical Systems with Emergent Collective Computational Abilities* apresentou um modelo neural que viria a ser adotado em muitos trabalhos com redes neurais. A estrutura proposta por Hopfield foi batizada de Rede de Hopfield.

Em 1986, David Rumelhart e colaboradores publicaram o livro *Backpropagation*, O algoritmo *Backpropagation* será discutido em detalhes no próximo capítulo

Em 1987 os laboratórios AT & Bell anunciaram o desenvolvimento de *chips* neurais baseados nas redes de Hopfield (Anderson e Rosenfield, 1988 segundo Carvalho Filho, 1993).

II.2.3. Alguns Tipos de ANN's.

Os estudos básicos das teorias sobre redes neurais tiveram um grande evolução na década de 80 após a adesão do físico John Hopfield. Segundo Hietch-Nielsen (1990), em 1986 aproximadamente um terço das pessoas envolvidas em trabalhos com ANN's foram atraídos pela fama e confiança de Hopfield. Como uma evidência destes avanços temos uma lista alfabética de tipos de Redes Neurais desenvolvidos até a década de 80 com algumas características. A lista a seguir foi extraída de Maren *et al*, 1990 (conforme Sari, 1994).

Rede: *ADALINE/MADALINE*.

Ano: 1960.

Autor: B. Widrow.

Primeiras Aplicações: Filtragem de sinal adaptativo; equalização adaptativa.

Vantagens: Rápida, fácil de implementar, pode ser feito utilizando controle analógico ou VLSI (*Very Large Scale Integration*).

Desvantagens: É necessária uma relação linear entre a entrada e a saída. Somente é possível classificar espaços linearmente separáveis.

Rede: *Backpropagation (BP) perceptron, Basic.*

Ano: 1974/1986.

Autor: P. J. Werbos, D. Parker, D. Rumelhart.

Primeiras Aplicações: Reconhecimento de padrões, filtragem de sinais, remoção de ruído, segmentação sinal/imagem, classificação, mapeamento, controle robótico adaptativo, compressão de dados.

Vantagens: Rápida operação. Boa representação dos dados entrada, classificação e muitas outras tarefas. Muito estudada, com muitas aplicações de sucesso.

Desvantagens: Tempo de treinamento longo.

Rede: *Adaptative Resonance Theory (ART).*

Ano: 1983.

Autor: G. Carpenter & S. Grossberg.

Primeiras Aplicações: Reconhecimento de Padrões.

Vantagens: Capaz de aprender novos padrões de novas categorias e guardar as categorias aprendidas.

Desvantagens: Natureza dos exemplares categóricos deve mudar com o tempo.

Rede: *Recurrent.*

Ano: 1987.

Autor: Almeida, Pineda.

Primeiras Aplicações: Controle robótico, reconhecimento de fala, previsão de elemento sequencial.

Vantagens: Excelente rede para classificação, mapeamento de informações variando no tempo.

Desvantagens: Rede complexa, pode ser difícil treinar e otimizar.

Rede: *Time-delay.*

Ano: 1987.

Autor: D. W. Tank & J. J. Hopfield.

Primeiras Aplicações: Reconhecimento de fala.

Vantagens: Performance equivalente aos melhores métodos convencionais, rápida operação.

Desvantagens: Janela fixa para a atividade temporal representada, responde desastrosamente para diferentes escalas de entrada.

Rede: *Functional-Link Network*.

Ano: 1988.

Autor: Y. H. Pao.

Primeiras Aplicações: Classificação e mapeamento.

Vantagens: Somente duas camadas (entrada e saída) são necessárias, rápidas para treinar.

Desvantagens: Não é claro o modo de identificar funções “functional links”.

Rede: *Radial Basis Function Network (RBF)*.

Ano: 1987-1988.

Autor: Vários grupos de pesquisadores

Primeiras Aplicações: Classificação, mapeamento.

Vantagens: Rede com uma única camada oculta de neurônios RBF, é equivalente a rede BP básico com duas camadas ocultas.

Desvantagens: Alguns problemas de estabilidade. Performance equivalente a *Backpropagation*

Rede: *Backpropagation of Utility Function Through Time*.

Ano: 1974.

Autor: P. J. Werbos.

Primeiras Aplicações: Maximiza a performance da indexação ou utiliza a função sobre o tempo, controle neural (robótica).

Vantagens: A maior aproximação neural extensiva para controle e/ou baseado em previsão.

Desvantagens: Pode ser usado somente depois do modelo diferenciável identificado, deve adaptar-se *off line* se o modelo é dinâmico, e assumindo que o modelo é exato.

Rede: Bidirecional Associative Memory (BAM).**Ano:** 1987.**Autor:** B. Kosko**Primeiras Aplicações:** Heteroassociativa (memória endereçada).**Vantagens:** Simples, regra de aprendizado , arquitetura e dinâmica raras. Prova clara da estabilidade dinâmica.**Desvantagens:** Capacidade de armazenamento pobre, recuperação exata pobre.**Rede: Boltzmann Machine, Cauchy Machine.****Ano:** 1984, 1986.**Autor:** G. Hinton, T. Sejnowski, D. Ackley, H. Szu.**Primeiras Aplicações:** Reconhecimento de padrões (imagem, sonar, radar), otimização.**Vantagens:** Capaz de formar ótima representação da característica do padrão. Segue superfície de energia para otimização mínima. A Máquina de Cauchy apresenta aprendizado rápido.**Desvantagens:** A máquina de Boltzmann apresenta aprendizado lento.**Rede: Boundary Countor System.****Ano:** 1985.**Autor:** S. Grossberg, E. Mingolla.**Primeiras Aplicações:** Processamento de imagem de nível baixo.**Vantagens:** Aproximação baseada biologicamente para segmentação excelente.**Desvantagens:** Complexa, arquitetura multicamadas.**Rede: Brain-State-in-a-Box.****Ano:** 1977.**Autor:** J. Anderson.**Primeiras Aplicações:** Revocação associativa (complementação do padrão dado um padrão parcial).**Vantagens:** Possivelmente melhor performance que a rede de Hopfield.**Desvantagens:** Incompleta exploração em termos de performance e potencial de aplicação.

Rede: Hopfield.

Ano: 1982.

Autor: J. Hopfield.

Primeiras Aplicações: Revocação associativa (complementação do padrão dado um padrão parcial), otimização.

Vantagens: Conceito simples, produz estabilidade dinâmica, fácil implementação em VLSI.

Desvantagens: Incapaz de aprender novos estados, armazenamento de memória pobre.

Rede: Learning Vector Quantization.

Ano: 1981.

Autor: T. Kohonen.

Primeiras Aplicações: Revocação associativa, compressão de dados.

Vantagens: Capaz de auto-organizar a representação de um vetor para distribuição probabilística num dado. Execução rápida depois que o treinamento é completado.

Desvantagens: Decisão não resolvida na seleção do número de vetores usados para o tempo de treinamento apropriado. Treinamento lento.

Rede: Neocognitron.

Ano: 1975-1982.

Autor: K. Fukushima.

Primeiras Aplicações: Reconhecimento de caracteres manuscritos e reconhecimento de padrões invariantes a rotação.

Vantagens: Capaz de reconhecer padrões independente da escala, translação e rotação.

Desvantagens: Requer muitos elementos de processamento e camadas, estrutura complexa.

Rede: Self-Organizing Topology-Preserving Map.

Ano: 1981.

Autor: T. Kohonen.

Primeiras Aplicações: Mapeamento complexo (envolvendo relações de vizinhança), compressão de dados, otimização.

Vantagens: Conceito simples, produz estabilidade dinâmica, fácil implementação em VLSI.

Desvantagens: Incapaz de aprender novos estados, armazenamento de memória pobre.

II.2.3. Aplicações.

As redes neurais têm recebido uma grande atenção do meio científico nos últimos 20 anos, surgindo muitas aplicações práticas e teóricas com tendência a crescer com o aperfeiçoamento dos métodos de treinamento e desenvolvimento de estruturas de *hardware* dedicadas (VLSI). Soucek (1989) em seu livro *Neural and Concurrent Real-Time Systems* lista um grupo de fabricantes de *hardware* e *software* específicos para processamentos com redes neurais.

O mais relevante neste contexto, são as aplicações em engenharia, mais especificamente para o controle e modelagem de processos em engenharia química, portanto a base de pesquisa deste trabalho concentra-se nesta área. É interessante observar que, apesar dos grandes desenvolvimentos com redes neurais acontecerem na década de 80, a introdução das técnicas neurais em aplicações na engenharia química se concentraram no final da década de 80. Ainda hoje o uso destas técnicas é vista com reserva por parte de muitos profissionais.

Muitas novas estruturas de redes neurais foram desenvolvidas nos últimos anos. Várias apenas estendendo a clássica *Backpropagation* e outras introduzindo novos conceitos. Isto pode ser visto em alguns dos trabalhos a seguir.

Segundo De Souza Jr. (1993), os primeiros trabalhos aplicando redes neurais em engenharia química foram os de G. Birky, T. J. McAvoy, N. Bhagat, J. C. Hoskims, D. M. Himmelblau, V. Venkatasubramanian, R. Vaidyanathan. Os três primeiros aplicaram redes neurais na identificação e controle de processos químicos. Os demais no diagnóstico e detecção de falhas.

Em 1989, Venkatasubramanian & Chan estudaram uma metodologia para a detecção de falhas, comparando-a com métodos estatísticos.

Watanabe *et alli* (1989), aplicaram redes neurais no diagnóstico de falhas em um reator químico, onde a estratégia usada era composta de dois estágios, um para discriminar as falhas, e outro para determinar o grau de severidade das mesmas.

Em 1990, Nguyen e Widrow testaram redes neurais na solução de problemas de controle com altas não linearidades. Em seus testes aplicaram uma rede neural multicamadas como modelo dinâmico do processo produzindo então um simulador. O simulador foi utilizado para treinar uma outra rede neural como controlador e este aplicado a um processo em tempo real.

Em 1990 Bath *et alli* modelaram, utilizando uma rede *Backpropagation*, três processos não lineares, um reator em estado estacionário, um tanque agitado com pH dinâmico, e um biosensor.

Em 1990, Chu *et alli* apresentaram duas técnicas para a utilização de redes neurais artificiais na identificação da dinâmica de sistemas. Na primeira, uma rede de Hopfield é usada para uma estimação por mínimos quadrados para sistemas variantes e invariantes no tempo. A segunda técnica (no domínio da frequência) utiliza um grupo de funções de base ortogonal e análise de Fourier para determinar um sistema em termos dos coeficientes de Fourier.

Em 1990, Chen aplicou uma rede *Backpropagation* na produção de um esquema de controle adaptativo, auto-ajustável para tratar processos não lineares arbitrários.

Em 1992, Lee & Park aplicaram uma rede neural conectada em paralelo a um controlador MPC, em um esquema *feedforward*, para o controle simulado de um reator não linear e uma coluna de destilação. Neste esquema uma rede neural treinada *on line* foi utilizada como um compensador dos efeitos não modelados.

Megan & Cooper (1992), utilizaram duas redes neurais, uma aprendendo a relação entre um dado padrão de erro e a diferença entre o ganho do processo modelado e o ganho atual do processo, e a outra no mesmo esquema com a constante de tempo. Esta estratégia foi utilizada,

via simulação e testes em escala piloto, no ajuste dos parâmetros de um controlador PI atuando no controle do nível de dois tanques em série.

Em 1992, Lima Verde *et alli* utilizaram redes neurais na previsão da velocidade de escoamento de sólidos em qualquer ponto em um reator de leito fluidizado recirculante.

Em 1992, Hunt *et alli* apresentaram uma revisão sobre modelagem, identificação e controle de sistemas não lineares utilizando redes neurais, com uma linguagem dedicada a engenheiros de controle.

Em 1993, Nikolau & White apresentaram uma maneira de abordar o diagnóstico de falhas em processos químicos em estado estacionário através de redes neurais. Um algoritmo *backpropagation* convencional com alterações na camada de entrada foi utilizado na simulação de um processo de produção de tolueno a partir de heptano.

Em 1993, segundo De Souza Jr., Borges e Castier utilizaram redes neurais para avaliar o desempenho de modelos termodinâmicos como UNIQUAC e UNIFAC no cálculo do equilíbrio líquido-vapor de sistemas binários isotérmicos.

Em 1993, De Souza Jr., utilizou redes neurais na predição da área metálica de um catalisador a partir das condições de manufatura e na modelagem de um reator não linear isotérmico, formando esquemas de controle preditivo.

Em 1993, Gupta & Narasimhan aplicaram redes neurais para diagnosticar erros e falhas em sensores através de padrões de entrada/saída para treinamento da rede, e comparação com métodos estatísticos tradicionais.

Em 1992, Pollard *et alli*, descreveram uma série de testes com um algoritmo *backpropagation* para a identificação de processos. Modificações no grupo de treinamento, na função objetivo e o uso da técnica da validação cruzada foram testadas usando dados simulados e dados reais de processos.

Em 1995, Rita utilizou um algoritmo *Backpropagation* para modelagem e controle do nível de um tanque simples e de dois tanques acoplados. O algoritmo compunha um esquema de predição com Horizonte de Controle e Horizonte de Predição iguais a um. O Trabalho de Rita constituiu-se a base do presente trabalho.

Em 1995, Van Can *et alli* apresentou o projeto e teste de um controlador preditivo baseado em uma rede neural para um sistema SISO (*Single Input Single Output*) não linear. Técnicas heurísticas para a identificação de sistemas não lineares foram propostas. A estratégia foi testada em vaso sob pressão.

Em 1995, Chen e Bruns utilizaram uma nova arquitetura de rede denominada *WaveARX Neural Network*. Enquanto a estrutura de uma rede neural convencional é derivada de procedimentos de otimização de tentativas e erro, uma *WaveARX* (o termo *ARX* refere-se a *AutoRegressive eXternal input model*) integra análise de multiresolução, conceitos da transformada *wavelet* e o modelo externo auto-regressivo (*ARX*) em uma rede *feedforward* de três camadas.

Em 1995, Chen *et alli* propuseram um esquema adaptativo para o controle, através de redes neurais, de uma classe de sistemas não lineares discretos no tempo. Um grupo de funções não lineares é definido como representante de um processo desconhecido. Assim uma rede neural multicamadas é usada para modelar o sistema desconhecido e gerar um controle *feedback*.

Tsen *et alli* (1996) utilizaram um modelo neural híbrido para a predição do controle de qualidade em um reator de polimerização em batelada de vinil acetato. Uma rede neural foi utilizada como modelo para prever a qualidade como uma função das variáveis manipuladas e de medidas *on line*. O modelo é construído usando um grupo de dados aumentado obtidos a partir de um modelo de tendências do processo.

Em 1996, Chen e Bruns testaram *on line* uma rede neural *WaveARX* na identificação do modelo de um CSTR com pH variante.

Em 1996, Rossiter *et alli* apresentaram uma metodologia para análise das condições necessárias para a estabilidade de uma trajetória predita. Um método para calcular o horizonte de controle infinito para o GPC, sem resolver a equação de Liapunov, também é apresentado.

Em 1996, Dong *et alli*, enfatizando a importância dos processos em batelada, utilizaram redes neurais para a otimização destes processos usando uma rede neural como modelo de processo.

Em 1996, Lewis & Liu testaram um controlador denominado FLC (*Fuzzy Logic Control*). Neste artigo dificuldades nos parâmetros de projeto do FLC são apontadas e a teoria matemática de controladores lógicos *fuzzy* é introduzida.

Em 1996, Jagannathan *et alli* aplicaram um controlador baseado em uma rede neural num esquema adaptativo para o controle de uma classe de sistemas dinâmicos não lineares MIMO (*Multi Input Multi Output*). Foi explicitada a relação entre o parâmetro de velocidade de aprendizagem e o número de neurônios na camada escondida.

Em 1996, Napolitano *et alli* relataram seus testes com redes neurais para detecção de falhas utilizando um algoritmo *Backpropagation* estendido. Os testes foram validados em testes com um atuador e um sensor, entre outros.

É evidente que, a literatura a cerca de redes neurais é bastante abrangente. Por isto, procurou-se neste capítulo, enfatizar suas aplicações em processos químicos e com isso tornar esta revisão adequada a este escopo.

CAPÍTULO III

FUNDAMENTOS E APLICAÇÕES.

Este capítulo dedica-se às definições básicas de uma rede neural, como: *backpropagation*, aprendizagem, padrões, topologias, pesos, camadas, algoritmos. Como o objetivo desta dissertação é aplicar redes neurais artificiais ao controle de sistemas com variações dinâmicas e não o estudo específico de estruturas neurais, os conceitos serão tecidos de forma sucinta e sob a visão do controle de processos.

III.1. Definições.

A primeira consideração a respeito de redes neurais, certamente, é sua fundamentação biológica. Isto porque as redes neurais artificiais são realmente sistemas, que imitam as habilidades computacionais dos sistemas biológicos usando um grande número de neurônios artificiais interconectados. Como o análogo biológico as redes neurais são constituídas de neurônios, axônios e dendritos. Os axônios e os dendritos podem ser representados juntamente com as sinapses pelos pesos.

Os pesos são os detentores da maior parte das informações que uma rede neural pode acumular. Isto implica em que a determinação dos pesos é a fase mais delicada do trabalho com redes neurais. Os pesos são obtidos na fase denominada treinamento.

Os dados que uma rede neural deve “aprender” são denominados padrões. por exemplo, considerando um padrão A_1 onde uma rede neural recebe três informações A, B, C, o padrão A_1 será :

$$A_1 = \{A, B, C\}$$

A fase de treinamento ou aprendizagem compreende a apresentação de um grupo de padrões à rede neural com o acoplamento de algum algoritmo de otimização que minimize o erro entre a saída da rede e saída real. Naturalmente, a primeira vez que o grupo de padrões é apresentado à rede (que ainda não “aprendeu” nada) o erro é grande. Contudo, o erro diminui ao repetir-se a operação, pela ação do algoritmo de otimização que minimize este erro.

Os neurônios artificiais são unidades que processam as informações pelo uso de uma função denominada **Ativação**. As funções de ativação mais comuns serão explicitadas a *posteriori*. Os neurônios artificiais estão dispostos em camadas. Cada neurônio de uma camada liga-se à cada um dos neurônios na camada seguinte. Assim um neurônio de uma camada (L+1) recebe as informações de todos os neurônios na camada L.

Podemos, portanto, estabelecer a seguinte relação entre neurônios biológicos e artificiais:

Tabela III.1 - Analogia Neurônio Biológico / Artificial.

| Neurônios | |
|------------|--|
| Biológicos | Artificiais |
| Dendritos | Canais de entrada: $\text{Peso} \times \text{Ativação do Neurônio anterior}$ |
| Corpo | Função de Ativação do Somatório dos canais de entrada |
| Axônio | Ativação |
| Sinapses | Pesos |

III.2. Arquiteturas das Redes Neurais.

Existem várias estruturas de redes neurais artificiais, no entanto, duas merecem destaque a *feedforward* multicamadas (FANN- *Feedforward Artificial Neural Network*) e a rede de Hopfield. A rede de Hopfield é uma rede com realimentação de informações, o que a torna um

pouco mais complexa que a FANN. Como a rede *feedforward* multicamadas é o alvo deste estudo, apenas esta rede será detalhada.

III.2.1. Uma rede *Feedforward* multicamadas.

Redes do tipo *feedforward* são o tipo mais estudado de ANN's. Esta classe de redes é de fácil compreensão e sua definição matemática é razoavelmente simples.

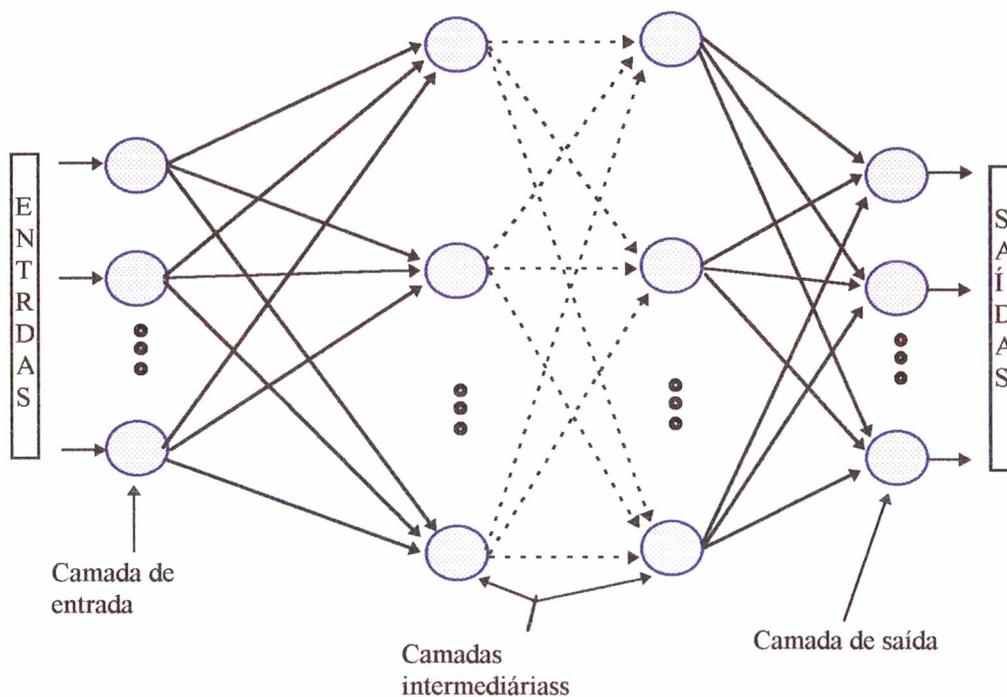


Figura III.1 - Uma FANN com 4 camadas (2 intermediárias).

Uma rede *feedforward* (Figura III.1) é composta por camadas, designadas por:

- **Camada de Entrada:** responsável apenas por receber os padrões.
- **Camada(s) intermediária(s), escondida(s) ou oculta(s):** responsáveis por processar as informações da camada anterior. No caso de uma FANN com três camadas, a camada intermediária processa as informações da camada de entrada diretamente.
- **Camada de Saída:** na camada de saída são realizados os últimos processamentos e o(s) resultado(s) é obtido.

É importante citar, que não existem técnicas formais para se determinar o número de camadas de uma rede neural, nem o número de neurônios que cada camada intermediária deve conter. Contudo, Hiecht-Nielsen (1989) provaram que uma FANN com três camadas pode aproximar qualquer função contínua com qualquer grau de precisão, desde que existam neurônios suficientes na camada intermediária.

Suponha-se uma FANN (previamente treinada) com 3 camadas, 2 neurônios na camada de entrada, 2 na camada intermediária e 1 na camada de saída podemos, resumidamente, assumir o fluxo de informações conforme a figura III.2:

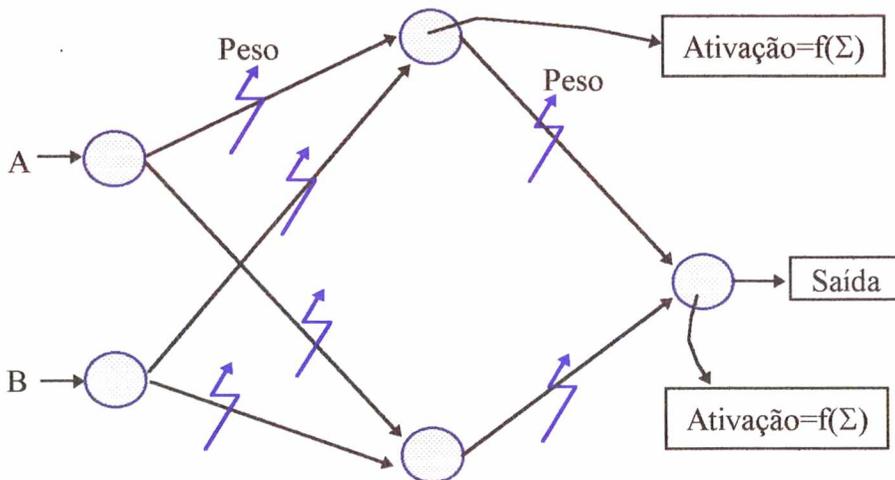


Figura III.2 - Uma FANN com três camadas.

- Um **padrão** ($A_1=\{A,B\}$) é apresentado à **primeira camada** da FANN. Cada **Neurônio** na primeira camada recebe uma informação (A e B). Os neurônios na primeira camada não realizam nenhum processamento, apenas recebem os dados e se encarregam de transferir as informações para os neurônios na segunda camada.
- Cada saída da primeira camada é multiplicada por um peso.
- Cada neurônio na segunda camada recebe as saídas da primeira camada multiplicadas pelos pesos, procede a somatória e em seguida o valor obtido é passado pela função de ativação.
- As saídas da segunda camada são então novamente ponderadas pelos pesos.

- Finalmente, na terceira camada, uma nova somatória sucedida por uma nova ativação encerra o processo e a saída desejada é obtida.

III.2.2. Funções de Ativação.

Funções de Ativação são funções matemáticas que atuam sobre a somatória de todas as entradas de um neurônio artificial. A função de ativação é a essência do corpo de um neurônio artificial, ela é a responsável pela união das informações armazenadas nos pesos e nos neurônios das camadas anteriores (se existirem). Dois tipos de função de ativação podem ser citados, segundo as definições de Bakshi e Stephanopoulos, 1993 (conforme De Souza Jr., 1993):

- Função Local: apresenta resposta ativa para valores de entrada na vizinhança imediata do seu centro de ativação. As funções locais mais comuns são as chamadas Funções de Base Radial (RBF). Um exemplo RBF é a gaussiana:
- Função Global: produz uma saída ativa para um largo intervalo de entradas. Uma função global (a Sigmoidal) foi adotada neste trabalho.

As funções de ativação globais mais comuns, segundo De Souza Jr (1993) são:

Função Linear:

$$f(x) = x$$

Função Degrau Limiar:

$$f(x) = \begin{cases} +\xi & \text{se } x > 0 \\ -\zeta & \text{se } x \leq 0 \end{cases}$$

Função Tangente Hiperbólica:

$$f(x) = \tanh(x)$$

Função Sigmoidal:

$$f(x) = \frac{1}{1 + \exp(-x)}$$

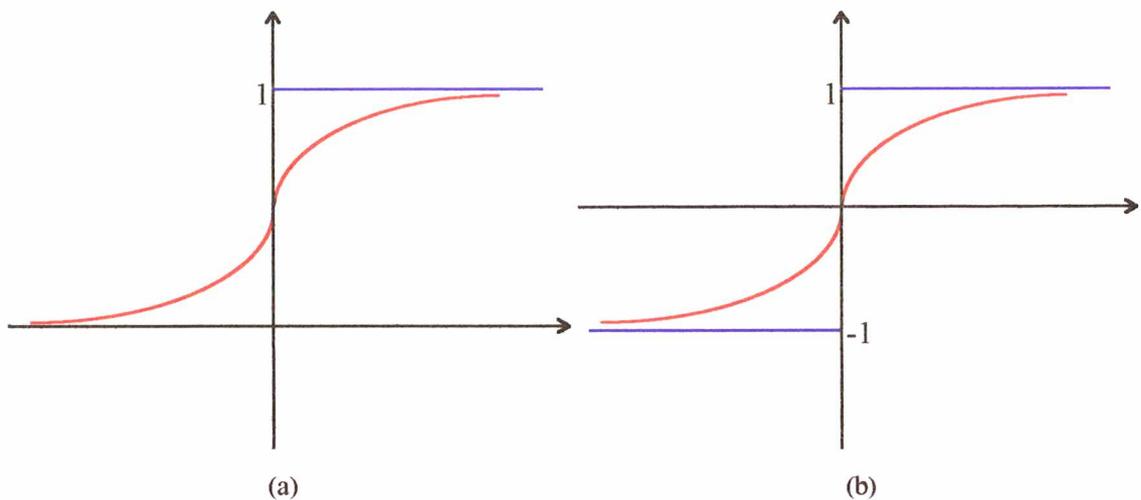


Figura III.3 - (a) Função Sigmoidal. (b) Função Tangente Hiperbólica.

As duas funções mais comumente adotadas para o algoritmo *Backpropagation* são a tangente hiperbólica e sigmoidal (figura III.3). Alguns autores usam funções de base radial, contudo estas não são aconselháveis para estudos iniciais por estarem sujeitas a problemas de estabilidade e convergência. Apesar de muitos trabalhos de sucesso as utilizarem, seu uso deve ser cauteloso. As funções tangente hiperbólica e sigmoidal foram exaustivamente testadas apresentando propriedades semelhantes. As principais diferenças entre estas funções podem ser observadas na figura III.3. Uma característica importante das funções sigmoidal e tangente hiperbólica são seus intervalos limitados, o que vem de encontro com o sistemas físicos (De Souza Jr, 1993).

III.3. Definição Matemática de uma Rede Neural *Feedforward*.

Matematicamente, cada neurônio é um elemento processador simples. A união de neurônios em uma rede compõe uma estrutura complexa com uma alta capacidade de

processamento. Esta capacidade depende da disposição dos neurônios e da quantidade de neurônios na rede. Em uma rede *feedforward* os neurônios estão dispostos em camadas onde as informações são propagadas apenas para frente. Neste trabalho, uma rede neural *feedforward* com três camadas e função de ativação sigmoïdal com variações no número de neurônios na camada intermediária foi empregada.

Um neurônio i para a camada de entrada de uma rede FANN, submetido à um padrão p , pode ser descrito como mostrado na figura III.4:

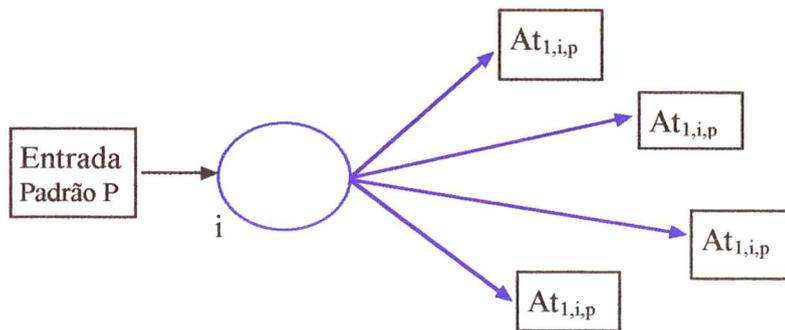


Figura III.4 - Neurônio i na camada de entrada.

$$At_{1,i,p} = f(x_{p,i}) \quad \text{com } i = 1, \dots, N_{NE}$$

Onde :

$At_{1,i,p}$ = Ativação do i -ésimo neurônio na camada 1 correspondente ao p -ésimo padrão.

N_{NE} = Número de neurônios na camada de entrada.

Os neurônios na primeira camada, por apenas distribuir informações, são denominados neurônios inativos.

Em uma FANN, exceto os neurônios na camada de entrada, todos os demais são ativos. Desta forma, um neurônio j com uma função de ativação sigmoideal numa camada L pode ser descrito por:

$$At_{L,j,p} = f(x_j) \quad (\text{III.1})$$

$$f(x_j) = \frac{1}{1 + \exp(-x_j)} \quad (\text{Função Ativação Sigmoideal para o neurônio } j) \quad (\text{III.2})$$

$$x_j = \sum_i W_{(L-1),i,j} At_{(L-1),i,p} \quad (\text{III.3})$$

A figura III.5, mostra a posição de um neurônio ativo numa camada L em relação a suas entradas e saídas. Os pesos devem ser considerados como parte do próprio neurônio e não são entradas deste.

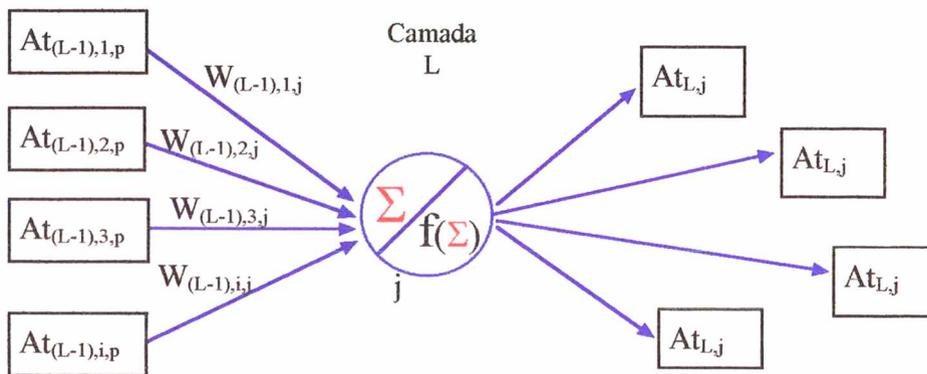


Figura III.5 - Neurônio j na camada L

Onde:

$At_{L,i,p}$ = Ativação do i -ésimo neurônio na camada L correspondente ao p -ésimo padrão apresentado.

$W_{(L-1),i,j}$ = Peso do i -ésimo neurônio da camada $(L-1)$ para o j -ésimo neurônio na camada L .

Definidas, a função de ativação (sigmoïdal) e a estrutura da rede (*feedforward*), a rede neural está completa, resta portanto definir o método segundo o qual serão determinados os pesos. Os pesos são a memória da rede neural, ou seja, onde as informações realmente estão armazenadas. A fase onde os pesos são determinados é denominada **treinamento ou aprendizagem**. O método comumente empregado para o treinamento de uma FANN, e adotado neste trabalho, é denominado *Backpropagation*.

III.3. O Método *Backpropagation*.

O método *Backpropagation*, ou Regra Delta Generalizada (Rumelhart et alli, 1986) é o método mais adotado na solução de problemas com redes neurais em engenharia química. Embora possua algumas limitações, é um método de fácil implementação e dedução e, geralmente, produz bons resultados.

O algoritmo do método *Backpropagation* segue o Fluxograma da figura III.6 e os seguintes passos:

- **Passo 1:** Gerar um grupo aleatório de Pesos.
- **Passo 2:** Apresentar os padrões à rede neural. Os padrões devem estar normalizados entre 0 e 1 para o caso da função de ativação sigmoïdal.
- **Passo 3:** Submeter os padrões à rede. Inicialmente os pesos utilizados pela rede neural são aleatórios, portanto as saídas estarão distantes da saída real. Cada neurônio ativo pode receber um valor constante denominado Bias. De acordo com a complexidade da rede utilizada, os biases podem contribuir para a estabilidade do treinamento. Neste trabalho biases nulos foram utilizados, e por simplicidade não aparecem nas figuras e equacionamentos.
- **Passo 4:** Comparar as saídas da rede neural com as saídas reais, gerando um erro. O erro é comparado com um valor tolerado indicando o final ou prosseguimento do processo. Se o erro for maior que o tolerado o Passo 4 será ativado, caso contrário o processo será encerrado.

- **Passo 5:** Otimizar os pesos. Nesta fase é determinada a direção seguida por cada um dos pesos, ou seja, se sofrerão acréscimo ou decréscimo.
- **Passo 6:** Ajustar os valores dos pesos conforme o algoritmo de otimização e reiniciar o processo com os novos pesos calculados.

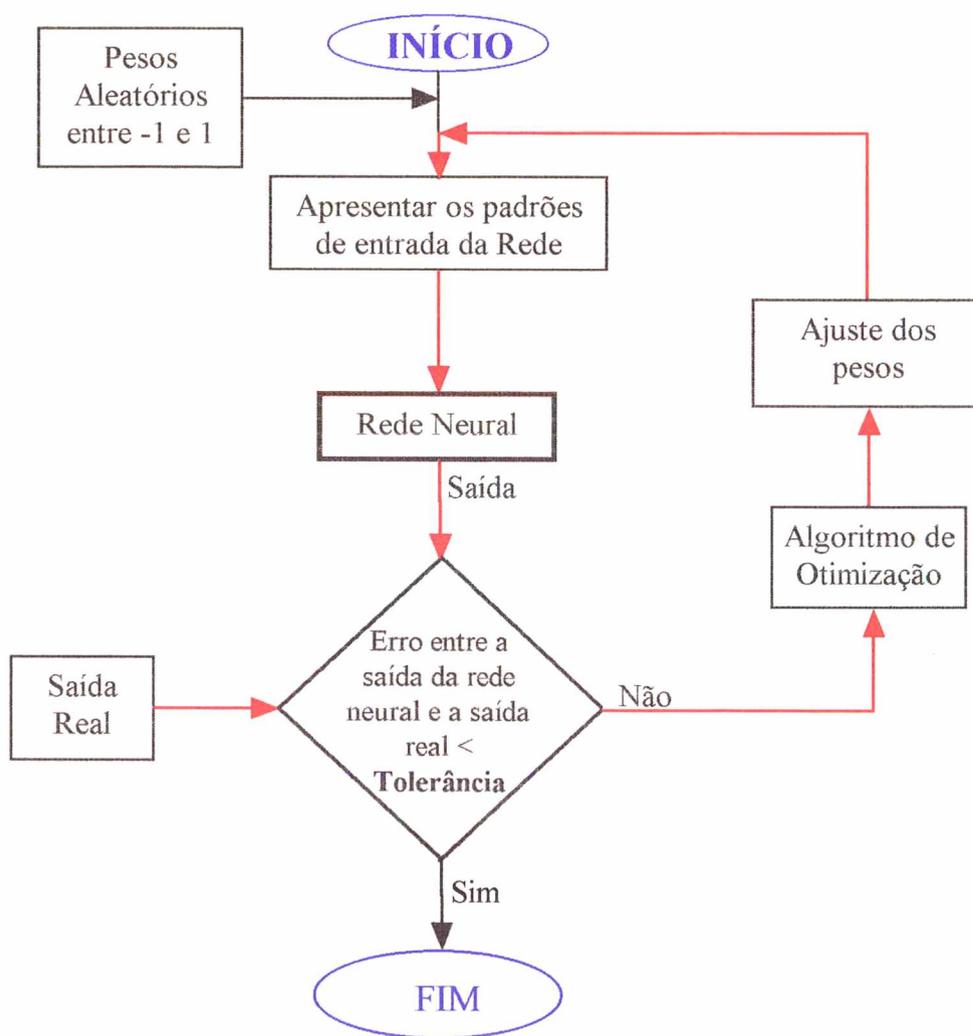


Figura III.6 - Fluxograma do Algoritmo *Backpropagation*.

III.3.1. Equacionamento do Algoritmo *Backpropagation*.

Como o objetivo da fase de aprendizagem é gerar pesos que permitam à rede neural caracterizar um sistema deve-se estabelecer uma função objetivo (também chamada função custo) que determinará a direção na qual os pesos devem ser alterados. A função aqui adotada é uma função custo quadrática, baseada no erro médio entre as saídas da rede e as saídas reais.

Considerando que o erro global gerado pela rede neural é a soma de todos erros em relação a cada padrão, pode-se escrever:

$$E = \sum_{p=1}^{N_p} (E_p)^2 \quad (\text{III.4})$$

Portanto, o erro por padrão será dado por:

$$E_p = \frac{1}{2} \sum_{k=1}^{N_{NS}} (At_{3,k,p} - Y_{k,p}) \quad (\text{III.5})$$

E o erro global será:

$$E = \sum_{p=1}^{N_p} \frac{1}{2} \sum_{k=1}^{N_{NS}} (At_{3,k,p} - Y_{k,p})^2 \quad (\text{III.6})$$

onde:

N_p = Número de padrões

N_{NS} = Número de neurônios na camada de saída

$At_{3,k,p}$ = Saída desejada (na camada 3 para o caso de uma rede com 3 camadas)

$Y_{k,p}$ = Saída real que é o alvo da saída desejada.

p = padrão

k = neurônio na camada de saída

Definida a função custo como o erro global, deve-se definir o método segundo o qual este erro será minimizado. O método adotado foi o método do passo descendente (*Steepest Descent*) onde a direção de busca é o negativo do gradiente (Himmelblau, 1988).

Genericamente o ajuste dos pesos entre um neurônio i de uma camada $(L-1)$ e o neurônio j de uma camada L após $(it+1)$ iterações fica :

$$W^{it+1}_{(L-1),i,j} = W^{it}_{(L-1),i,j} + \eta \Delta W^{it}_{(L-1),i,j} \quad (\text{III.7})$$

Com η , sendo o fator de aprendizagem que define a razão em que os pesos são modificados, η varia entre 0 e 1. Os valores de η devem ser ajustados de forma a evitar a divergência do algoritmo.

Como ΔW é a direção da alteração dos pesos, que segundo Himmelblau (1988), é o negativo do gradiente, têm-se :

$$\Delta W_{(L-1),i,j} = - \frac{\partial \mathcal{E}}{\partial W_{(L-1),i,j}} \quad (\text{III.8})$$

Como a rede neural utilizada neste trabalho é composta de 3 camadas, sendo a primeira camada composta de 3 neurônio e a terceira camada composta de apenas 1 neurônio, as seguintes expressões resultam:

Para o ajuste dos pesos entre as camadas de entrada e intermediária:

$$\Delta W_{1,i,j} = - \frac{\partial \mathcal{E}}{\partial W_{1,i,j}} \quad (\text{III.9})$$

$$\Delta W_{1,i,j} = - \frac{1}{2} \frac{\partial}{\partial W_{1,i,j}} \left[\sum_{n=1}^{N_p} \sum_{k=1}^{N_{NS}} (At_{3,k,p} - Y_{k,p})^2 \right] \quad (\text{III.10})$$

$$\Delta W_{1,i,j} = -1 \sum_{n=1}^{N_p} \sum_{k=1}^{N_{NS}} (At_{3,k,p} - Y_{k,p})^2 \frac{\partial At_{3,k,p}}{\partial W_{1,i,j}} \quad (\text{III.11})$$

onde :

$$At_{3,k,p} = \frac{1}{1 + \exp\left(-\sum_{j=1}^{N_{NI}} (W_{2,i,k} At_{2,i,p})\right)} = \left[1 + \exp\left(-\sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p})\right)\right]^{-1} \quad (\text{III.12})$$

$$\frac{\partial At_{3,k,p}}{\partial W_{1,i,j}} = \frac{\partial}{\partial W_{1,i,j}} \left[1 + \exp\left(-\sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p})\right)\right]^{-1} \quad (\text{III.13})$$

$$\frac{\partial At_{3,k,p}}{\partial W_{1,i,j}} = (At_{3,k,p})^2 W_{2,j,k} \exp\left(-\sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p})\right) \frac{\partial At_{2,j,p}}{\partial W_{1,i,j}} \quad (\text{III.14})$$

A partir de III.12 :

$$\exp\left(-\sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p})\right) = \frac{1 - At_{3,k,p}}{At_{3,k,p}} \quad (\text{III.15})$$

Substituindo III.15 em III.14:

$$\frac{\partial At_{3,k,p}}{\partial W_{1,i,j}} = (At_{3,k,p}) (1 - At_{3,k,p}) W_{2,j,k} \frac{\partial At_{2,j,p}}{\partial W_{1,i,j}} \quad (\text{III.16})$$

onde :

$$\frac{\partial At_{2,j,p}}{\partial W_{1,i,j}} = \frac{\partial}{\partial W_{1,i,j}} \left[1 + \exp \left(- \sum_{i=1}^{N_{NI}} (W_{1,i,j} At_{1,i,p}) \right) \right]^{-1} \quad (\text{III.17})$$

$$\frac{\partial At_{2,j,p}}{\partial W_{1,i,j}} = \left[1 + \exp \left(- \sum_{i=1}^{N_{NI}} (W_{1,i,j} At_{1,i,p}) \right) \right]^{-2} \exp \left(- \sum_{i=1}^{N_{NI}} (W_{1,i,j} At_{1,i,p}) \right) At_{1,i,p} \quad (\text{III.18})$$

Analogamente a III.15:

$$\frac{\partial At_{2,j,p}}{\partial W_{1,i,j}} = At_{2,j,p} (1 - At_{2,j,p}) At_{1,i,p} \quad (\text{III.19})$$

Finalmente substituindo III.19 em III.16, e III.16 em III.11 obtemos a forma para o ajuste dos pesos entre a primeira e a segunda camadas:

$$\Delta w_{1,i,j} = \sum_{p=1}^{N_p} \left[\sum_{k=1}^{N_{NS}} (Y_{k,p} - At_{3,k,p}) At_{3,k,p} (1 - At_{3,k,p}) \right] At_{2,i,p} (1 - At_{2,i,p}) At_{1,i,p} \quad (\text{III.20})$$

Para o ajuste dos pesos entre as camadas intermediária e de saída:

$$\Delta W_{2,j,k} = - \frac{\partial \mathcal{E}}{\partial W_{2,j,k}} \quad (\text{III.21})$$

$$\Delta W_{2,j,k} = - \frac{1}{2} \frac{\partial}{\partial W_{2,j,k}} \left[\sum_{n=1}^{N_p} \sum_{k=1}^{N_{NS}} (At_{3,k,p} - Y_{k,p})^2 \right] \quad (\text{III.22})$$

$$\Delta W_{2,j,k} = \sum_{p=1}^{N_p} \sum_{k=1}^{N_{NX}} (At_{3,k,p} - Y_{k,p}) \frac{\partial At_{3,k,p}}{\partial W_{2,j,k}} \quad (\text{III.23})$$

onde:

$$\frac{\partial At_{3,k,p}}{\partial W_{2,j,k}} = \frac{\partial}{\partial W_{2,j,k}} \left[1 + \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \right]^{-1} \quad (\text{III.24})$$

$$\frac{\partial At_{3,k,p}}{\partial W_{2,j,k}} = \left[1 + \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \right]^{-2} \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) At_{2,j,p} \quad (\text{III.25})$$

Substituindo III.15 e III.12 em III.25:

$$\frac{\partial At_{3,k,p}}{\partial W_{2,j,k}} = At_{3,k,p} (1 - At_{3,k,p}) At_{2,j,p} \quad (\text{III.26})$$

Finalmente substituindo III.26 em III.23 :

$$\Delta W_{2,j,k} = \sum_{p=1}^{N_p} \left[\sum_{k=1}^{N_{NX}} (Y_{k,p} - At_{3,k,p}) At_{3,k,p} (1 - At_{3,k,p}) At_{2,j,p} \right] \quad (\text{III.27})$$

A associação do algoritmo *Backpropagation* ao *Steepest descent* têm uma aplicabilidade extensa, desde modelagem de processos até reconhecimento de padrões visuais e sinais de sonar. Podemos indicar como qualidades deste algoritmo: simplicidade, velocidade da forma final, o paralelismo inerente das redes neurais, boa capacidade de modelagem e a popularidade de ambos os métodos. Contra este agrupamento pesa: lentidão do treinamento, problemas de convergência

por- mínimos locais e uma certa dificuldade em se estabelecer o número de neurônios que cada camada intermediária deve conter.

III.4. Aplicações das FANN.

As inúmeras aplicações do algoritmo *Backpropagation* lhe garantiram a maior popularidade dentre todos os algoritmos de redes neurais. Em engenharia química é o algoritmo mais comum com aplicações em modelagem de processos, controle de processos, previsão de propriedades físicas, detecção de falhas, etc. Com certeza as maiores aplicações concentram-se em na modelagem e controle de processos. Neste trabalho uma rede neural *feedforward* foi aplicada como modelo para dois processos, um tanque com dinâmica variante e um trocador de calor.

III.4.1. Aplicando uma FANN como modelo para o controle de nível de um tanque.

Um dos sistemas mais simples e fáceis de montar é o sistema de controle de nível de um tanque. Para aumentar o grau de dificuldade e testar a capacidade de modelagem de uma rede neural *feedforward*, foi utilizado um tanque com duas seções, uma cônica e uma cilíndrica. Além disso, uma estratégia de acoplamento duas redes neurais, uma treinada em cada seção do sistema, foi testada associando-as a um sistema especialista. Para diferenciar as duas estratégias, a composta por uma única rede neural foi denominada RNG (estratégia baseada em uma Rede Neural Generalista), enquanto a composta por duas redes neurais foi denominada RNE (Estratégia baseada em uma rede neural Especialista).

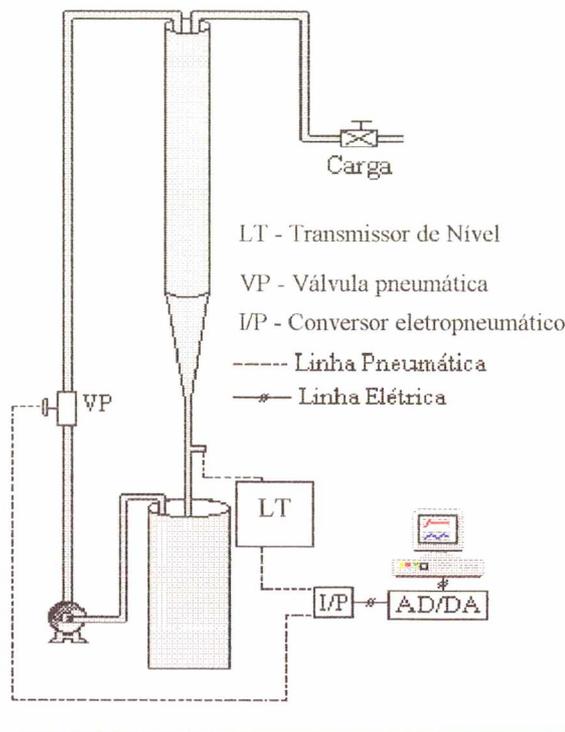


Figura III.7 - Sistema de controle de nível simplificado.

Para o caso do controle de nível do tanque (figura III.7) com a estratégia RNG uma rede neural *feedforward* foi treinada, através do algoritmo *Backpropagation*, com dados do nível do tanque (altura) e valores da abertura da válvula (volt). Cada padrão apresentado à rede estava disposto de forma que uma saída atual do processo (Y_k), uma saída passada (Y_{k-1}) e um valor atual de abertura da válvula (U_k) eram utilizados.

No caso do uso da estratégia RNE, os padrões estavam dispostos da mesma forma que os da estratégia RNG, contudo dois grupos de dados, coletados nas duas seções do tanque, funcionavam como modelo do processo. O sistema especialista realiza a seleção do grupo de pesos que atua em cada instante de amostragem. Assim, as saídas da rede neural são função das variáveis mencionadas, conforme as expressões abaixo:

$$Y_{k+1} = f(Y_{k-1}, Y_k, U_k) \quad (\text{III.28})$$

Sujeito a:

$$1 \leq U_k \leq 5 \text{ - limites de ação da válvula.}$$

$$0.7 \leq Y_k \leq 1.5 \text{ - altura do tanque.}$$

Para o sistema especialista:

$$Y_{k+1} = f(Y_{k-1}, Y_k, U_k) \quad 0.7 \leq Y \leq 1.15 \quad (\text{III.29})$$

$$Y_{k+1} = f(Y_{k-1}, Y_k, U_k) \quad 1.15 \leq Y \leq 1.5 \quad (\text{III.30})$$

De uma forma genérica:

$$Y_{(k+1)} = f(Y_k, Y_{(k-1)}, Y_{(k-2)}, Y_{(k-3)}, \dots, Y_{(k-n+1)}, U_k, U_{(k-1)}, Y_{(k-2)}, \dots, U_{(k-m+1)}) \quad (\text{III.31})$$

Assim, o esquema final da rede neural utilizada segue o fluxograma da figura III.8. Os neurônios na camada intermediária foram omitidos por questão de simplicidade.

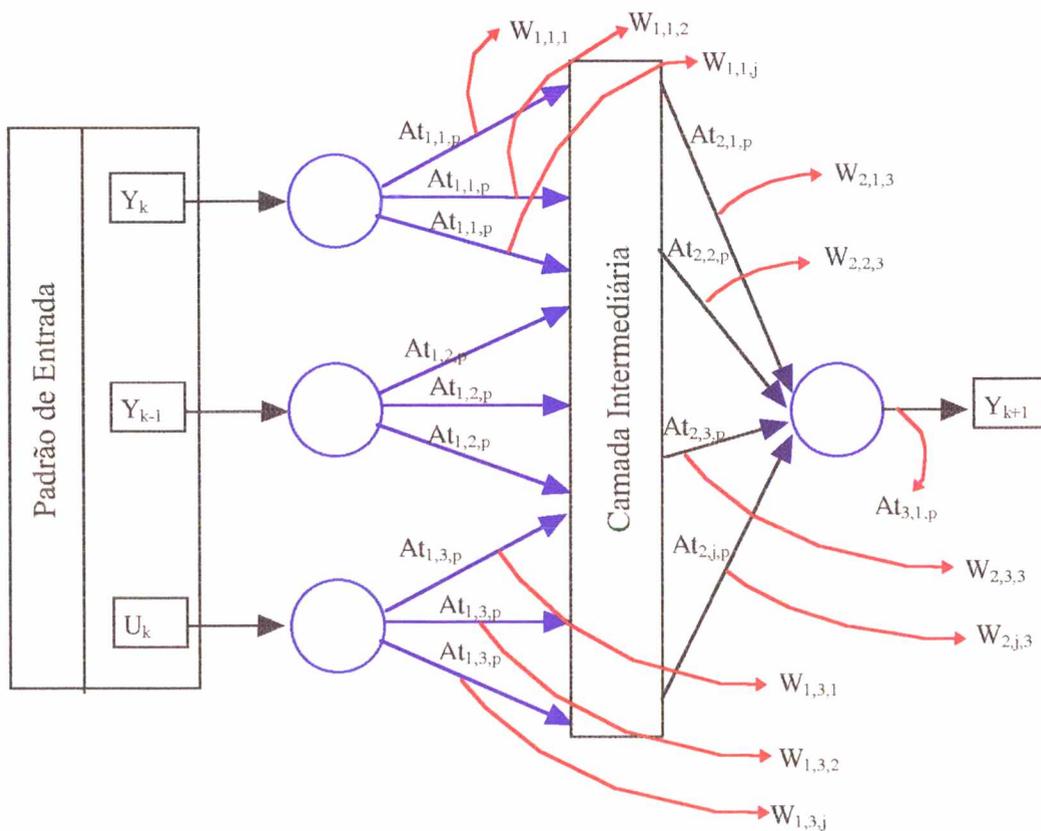


Figura III.8 - Estrutura utilizada neste trabalho.

III.4.2. Aplicando uma FANN como modelo para o controle de temperatura de um trocador de calor.

O estudo de sistemas que envolvem transferência de calor deparam-se frequentemente com problemas de atraso em sua resposta em relação a uma dada perturbação ou ação de controle. Este atraso é conhecido como tempo morto. Para o estudo do comportamento de um controlador baseado em um modelo neural frente a este tipo de problema foi utilizado um reator (este sistema será detalhado posteriormente). Como nenhuma reação química foi testada, o reator operou como um trocador de calor, ou seja, apenas as dinâmicas de aquecimento e resfriamento foram modeladas.

A metodologia para a aplicação de uma rede *feedforward* para o controle da temperatura de um trocador de calor é basicamente igual a do controle de nível.

A opção adotada para contornar o problema da presença de tempo morto foi o uso de padrões deslocados T_M unidades de tempo no passado, como mostra o esquema da figura III.9, onde T_M equivale ao tempo morto do sistema.

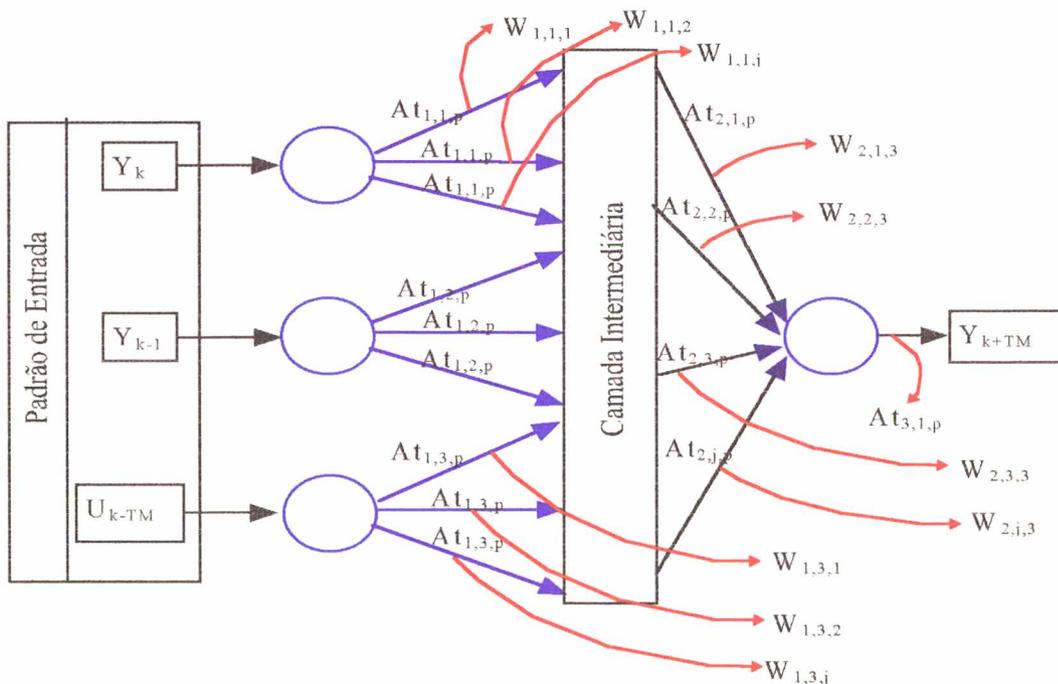


Figura III.9 - Estrutura utilizada considerando tempo morto.

Assim, temos que :

$$Y_{k+TM} = f(Y_{k-1}, Y_k, U_{k-TM}) \quad (\text{III.32})$$

Sujeito a:

$$1 \leq U_k \leq 5$$

$$20 \leq Y_k \leq 90$$

Pode-se definir, portanto, de uma forma genérica:

$$Y_{(k+TM)} = f(Y_k, Y_{(k-1)}, Y_{(k-2)}, Y_{(k-3)}, \dots, Y_{(k-n+1)}, U_{k-TM}, U_{(k-2*TM)}, \dots, Y_{(k-m*TM)}) \quad (\text{III.33})$$

III.5. Um Controlador Genérico Baseado em um Modelo Neural.

A definição de uma estratégia de controle baseada em um modelo de um processo pode caracterizar-se por minimizar o erro entre uma saída do processo e uma saída do modelo utilizado, independentemente da origem deste. O erro gerado é comparado ao erro da saída do processo em relação a uma referência, gerando uma ação de controle. As estratégias de controle mais antigas levavam em consideração apenas o erro em relação à referência, o que trazia consigo vantagens e desvantagens. Como vantagem, pode-se considerar a generalidade do método pela independência de modelos, e como desvantagem a necessidade de ajustes frequentes dos parâmetros do controlador. Um exemplo típico de simplicidade é o clássico PID (Proporcional Integral Derivativo), representado na equação III.34 (Seeborg *et alli*, 1989):

$$U = U_0 + K_c \left(e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{de}{dt} \right) \quad (\text{III.34})$$

Onde:

U= Ação de controle a ser implementada

U₀= Ação de controle implementada no intervalo de tempo passado

K_c = Parâmetro responsável pela ação proporcional

τ_i - Parâmetro responsável pela ação integral

τ_d - Parâmetro responsável pela ação derivativa

$e(t)$ = Erro no tempo, relativo a uma referência

Para exemplificar a facilidade desta implementação, a equação III.35 mostra o algoritmo digital do PID na chamada forma da velocidade (Seeborg *et alli*, 1989):

$$U = U_0 + K_c \left((e_n - e_{n-1}) + \frac{\Delta t}{\tau_i} e_n + \frac{\tau_d}{\Delta t} (e_n - 2e_{n-1} + e_{n-2}) \right) \quad (\text{III.35})$$

onde, Δt é o intervalo entre cada amostragem.

As deficiências do PID em lidar com processos fortemente não lineares ou com tempo morto acentuado reforçou o interesse no estudo de técnicas mais eficientes, mesmo com maior esforço humano e computacional. Com isso, os trabalhos de Clarke e Gawthrop com o STC (*Self-Tuning Control*), Cutler e Ramaker com o DMC (*Dynamic Matrix Control*) e Clarke *et alli* com o GPC (*Generalized Predictive Control*) tornaram-se populares e servem como base para a produção de controladores não lineares ou linearizados. Trabalhos com controle Adaptativo e Auto-ajustável têm produzido excelentes resultados. Uma outra corrente de trabalhos têm investido nas denominadas técnicas de controle preditivo (utilizadas neste trabalho). Estas técnicas utilizam um modelo do processo para prever saídas futuras do processo permitindo que as ações necessárias sejam implementadas. Neste ponto, a necessidade de um modelo pode ser suprida com o uso de um modelo neural. O modelo neural é obtido através do treinamento de uma rede neural, neste caso uma FANN, com dados representativos do processo. O treinamento pode ser realizado *on line*, isto é os padrões são apresentados a cada instante de amostragem, ou *off line*, onde os dados são previamente coletados da planta e submetidos à rede neural. O treinamento *off line* foi adotado neste trabalho, por ser mais simples de se implementar.

O uso de um modelo neural na elaboração de um controlador não linear têm a vantagem da facilidade da obtenção do modelo e a não linearidade inerente das redes neurais artificiais, o que reduz os ajustes no controlador. Uma outra opção para a produção de um controlador não linear é o uso de uma rede neural diretamente como controlador, contudo esta prática é menos

comum. Considerando resolvido o problema da obtenção do modelo não linear do processo deve-se, ainda, estabelecer uma estratégia que torne possível o acesso à referência sem desestabilizar o processo. Uma técnica comum é minimizar uma função objetivo quadrática (Bath *et alli*, 1990) do tipo:

$$J = \sum_{i=1}^H \left[y_{pred(k+i)} - y_{ref(k+i)} \right]^2 + \lambda \sum_{j=1}^{NU} \left[\Delta u_{(k+j-1)} \right]^2 \quad (III.36)$$

Com a trajetória de referência definida por:

$$y_{ref(k+i)} = ay_{ref(k+i-1)} + (1-a)y_{SP} \quad i = 1 \dots H \quad (III.37)$$

Assim, considerando a predição de um passo futuro temos pela derivação de J em relação a U:

$$\frac{\partial J_k}{\partial U_k} = 2 \left[y_{pred} - y_{ref(k+1)} \right] \frac{\partial y_{pred}}{\partial U_k} + 2\lambda \left[U_k - U_{(k-1)} \right] \quad (III.38)$$

Por simplicidade $y_{pred(k+1)} = y_{pred}$.

Como o objetivo é a minimização:

$$\frac{\partial J_k}{\partial U_k} = 0 \quad (III.39)$$

Assim:

$$U_k = U_{(k-1)} - \frac{1}{\lambda} \left[y_{pred} - y_{ref(k+1)} \right] \frac{\partial y_{pred}}{\partial U_k} \quad (III.40)$$

Onde:

H= Horizonte de predição

NU= Horizonte de controle

λ = Fator de penalização da ação de controle

α = Fator de ajuste da trajetória de referência

y= Variável controlada

y_{SP} = Setpoint - Ponto ao qual o sistema deve ser conduzido

y_{pred} = Valor de Y predito pelo modelo para o instante (k+1)

y_{ref} = Trajetória a ser seguida para atingir o *Setpoint*

U_k = Valor da variável manipulada a ser implementado

$U_{(k-1)}$ = Valor da variável manipulada implementado no intervalo anterior

Resta ainda o termo da derivada de Y_{pred} em relação a U_k a ser determinado. Técnicas numéricas ou analíticas podem ser adotadas para explicitar este termo. No uso desta forma um controlador não linear será obtido. A performance deste controlador já está aprovada e tem sido amplamente adotada. Como está evidente, o problema é a determinação do termo da derivada. O uso de um modelo neural para a predição implica em que a saída da rede equivale a um valor futuro do processo, ou seja:

$$At_{3,k,p} = y_{pred} \quad (III.41)$$

$$\frac{\partial y_{pred}}{\partial U_k} = \frac{\partial At_{3,k,p}}{\partial U_k} \quad (III.42)$$

$$\frac{\partial y_{pred}}{\partial U_k} = \frac{\partial}{\partial U_k} \left[1 + \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \right]^{-1} \quad (III.43)$$

$$\frac{\partial y_{pred}}{\partial U_k} = \left[1 + \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \right]^{-2} \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} \frac{\partial At_{2,j,p}}{\partial U_k} \right) \quad (III.44)$$

Conforme III.15 e III.12:

$$\frac{\partial y_{pred}}{\partial U_k} = At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_M} \left(W_{2,j,k} \frac{\partial At_{2,j,p}}{\partial U_k} \right) \quad (III.45)$$

Onde :

$$\frac{\partial At_{2,j,p}}{\partial U_k} = \left[1 + \exp \left(- \sum_{i=1}^{N_{NE}} (W_{1,i,j} At_{1,i,p}) \right) \right]^{-2} \exp \left(- \sum_{i=1}^{N_{NE}} (W_{2,j,k} At_{2,j,p}) \right) \sum_{i=1}^{N_{NE}} \left(W_{1,i,j} \frac{\partial At_{1,i,p}}{\partial U_k} \right) \quad (III.46)$$

Aplicando as definições analogamente a III.45, e considerando que apenas o terceiro neurônio da primeira camada recebe U_k :

$$\frac{\partial At_{2,j,p}}{\partial U_k} = At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j} \frac{\partial At_{1,3,p}}{\partial U_k} \quad (III.47)$$

Como $At_{1,3,p}$ é a própria entrada U_k , então:

$$\frac{\partial At_{1,3,p}}{\partial U_k} = 1 \quad (III.48)$$

Assim:

$$\frac{\partial At_{2,j,p}}{\partial U_k} = At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j} \quad (III.49)$$

Substituindo III.49 em III.45 :

$$\frac{\partial y_{pred}}{\partial U_k} = At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_M} (W_{2,j,k} At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j}) \quad (III.50)$$

Finalmente, substituindo III.50 em III.40 obtém-se a expressão da lei de controle:

$$U_k = U_{(k-1)} - \frac{1}{\lambda} [y_{pred} - y_{ref(k+1)}] At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_M} (W_{2,j,k} At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j}) \quad (III.51)$$

Sabendo que, $At_{3,k,p} = y_{pred}$ e que $y_{ref(k+1)}$ é dado pela expressão III.37 com $i=1$, ou seja, $y_{ref_k} = y_k$ a lei de controle fica:

$$U_k = U_{(k-1)} - \frac{1}{\lambda} [y_{pred} - \alpha y_k - y_{SP} + \alpha y_{SP}] y_{pred} (1 - y_{pred}) \sum_{j=1}^{N_M} (W_{2,j,k} At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j}) \quad (III.52)$$

Na lei de controle expressa na equação III.52 duas características interessantes podem ser observadas. A primeira é que apenas os valores dos pesos ($W_{2,j,k}$) são fixos, as demais variáveis são calculadas em tempo de execução, o que significa que a lei de controle é dinâmica e fortemente dependente da situação em que o controlador se encontra. Assim, um modelo neural com um aprendizado pobre pode degenerar o controle em algumas situações, o que conduziu ao desenvolvimento de um sistema especialista englobando duas redes neurais ao invés de uma. Outra característica a ser observada na lei de controle são os parâmetros α e λ . Enquanto λ atua como penalizador da ação de controle, reduzindo os incrementos na ação de controle, α atua como um suavizador da trajetória de controle, daí seu nome trajetória de referência. Estes dois parâmetros de ajuste, α e λ , atuam sobre a performance do controlador. A carência de regras explícitas para o referido ajuste, força o uso de métodos de tentativa baseados na resposta do sistema ao controlador, conforme a influência ditada pelas equações III.36, III.37 e III.40.

III.5.1. Esquema de Controle Utilizado.

O esquema de controle utilizado baseia-se no treinamento *off line* de uma rede neural (ou duas no caso RNE), que funciona como modelo do processo e parte integrante da lei de controle conforme visto na equação III.51. A implementação desta estratégia de controle apresenta, ainda, um problema de disposição temporal dos elementos dos padrões. Cada padrão apresentado a rede para o cálculo da saída futura do processo é composto por Y_k , $Y_{(k-1)}$ e U_k , sendo que U_k é o valor de U (ação de controle) a ser implementado no instante k , ou seja, que ainda não está determinado. A saída utilizada foi adotar um esquema iterativo para a determinação de U_k . Neste esquema, assume-se um valor de U para o instante k , submete-se juntamente com os valores de Y_k , $Y_{(k-1)}$ à rede neural, obtendo-se um valor de y_{pred} . O valor de y_{pred} obtido é utilizado na lei de controle (equação III.52) onde um novo valor de U_k é encontrado. Este valor pode estar distante do valor correto a ser implementado, assim, o este novo valor de U_k é novamente submetido à rede neural, e o processo se repete até que a diferença entre os valores de U_k obtidos seja suficientemente pequena, conforme a equação III.53.

$$\Delta U = abs(U_{k_{assumido}} - U_{k_{Lei\ de\ Controle}}) \quad (III.53)$$

Desta forma, se ΔU for suficientemente pequeno, o valor de y_{pred} obtido é assumido como $y_{(k+1)}$ e o valor de U_k calculado é implementado.

A implementação das estratégias de controle (RNE e RNG) em dois sistemas diferentes (tanque e trocador de calor) introduz leves modificações na forma com que os padrões são apresentados à rede, como exemplificado na figuras III.7 e III.8.

Tanque:

Padrão:

$$Y_{(k+1)} = f(Y_k, Y_{(k-1)}, U_k) \quad \text{III.28}$$

Lei de Controle:

$$U_k = U_{(k-1)} - \frac{1}{\lambda} \left[y_{pred} - \alpha y_k - y_{SP} + \alpha y_{SP} \right] y_{pred} (1 - y_{pred}) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} At_{2,j,P} (1 - At_{2,j,P}) W_{1,3,j} \right) \quad \text{(III.52)}$$

Reator:

Padrão

$$Y_{(k+TM)} = f(Y_k, Y_{(k-1)}, U_{k-TM}) \quad \text{III.32}$$

Lei de Controle:

$$U_k = U_{(k-1)} - \frac{1}{\lambda} \left[y_{pred} - \alpha y_k - y_{SP} + \alpha y_{SP} \right] y_{pred} (1 - y_{pred}) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} At_{2,j,P} (1 - At_{2,j,P}) W_{1,3,j} \right) \quad \text{(III.52)}$$

Como pode-se observar, a diferença básica entre a implementação nestes sistemas é o aparecimento do termo de U deslocado T_M (Tempo Morto) intervalos de tempo. Isto implica, a nível de implementação, na manutenção de um vetor de U para uso como entrada da rede neural. Uma observação importante é que a Lei de Controle não sofre nenhuma modificação.

Do ponto de vista de estratégia implementada, uma modificação no algoritmo de controle é introduzida. Na estratégia RNE, dois grupos de pesos são apresentados a uma estrutura seletora (sistema especialista) havendo uma avaliação a cada instante de amostragem do domínio de atuação de cada grupo de pesos. O teste é realizado de forma diferente para cada sistema. Como citado anteriormente, o critério adotado é a variação dinâmica mais acentuada contida pelo sistema.

No controle de nível do tanque a altura é avaliada a cada instante, sendo que alturas inferiores ao limite seção cônica/seção cilíndrica ativam o uso do Grupo 1 de pesos (grupo modelo da seção cônica), enquanto alturas superiores a este limite citado ativam o Grupo 2 de pesos.

No controle da temperatura do trocador de calor, a temperatura obtida a cada instante de amostragem é comparada com o valor da temperatura anterior. Assim, a estratégia interpreta que se a temperatura atual é maior que a anterior a temperatura no reator é crescente e o Grupo 1 de pesos (grupo modelo da faixa crescente de temperaturas) é ativado, caso contrário o grupo 2 (grupo modelo da faixa decrescente de temperaturas) é ativado. Isto permite robustez suficiente para a filtragem dos ruídos do processo. A figura III.10 apresenta um diagrama da estratégia de controle RNG enfatizando a dependência da Lei de Controle dos processamentos intermediários na rede neural, $At_{2,j,p}$ e $W_{1,3,j}$.

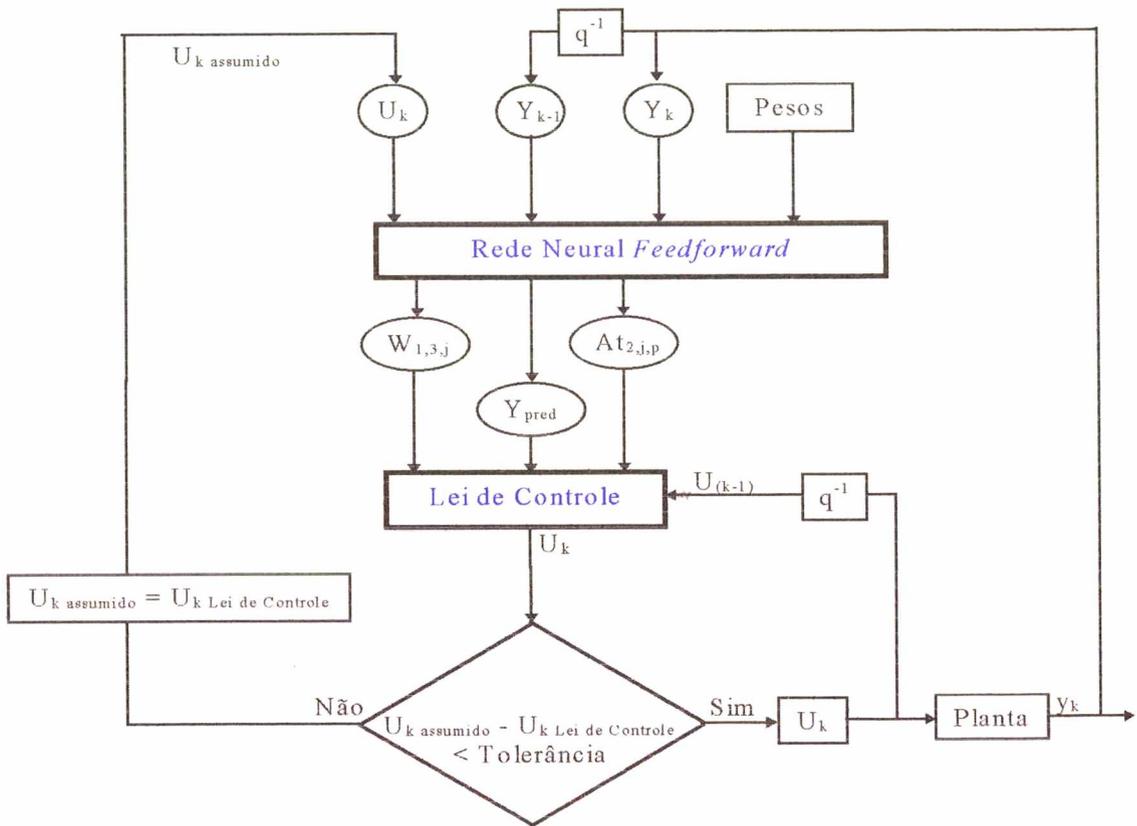


Figura III.10 - Esquema da estratégia de Controle RNC

Na figura III.11, está representado o esquema da estratégia RNE, onde observa-se uma pequena mudança em relação à RNC, a inclusão de mais um grupo de pesos. O critério de decisão denominado “Definir Grupo de Pesos” é fixo para o controle de nível (limite com cilindro), e móvel, y_{k-1} , para o controle de temperatura. Conforme citado anteriormente, a definição destes limites segue a dinâmica do sistema.

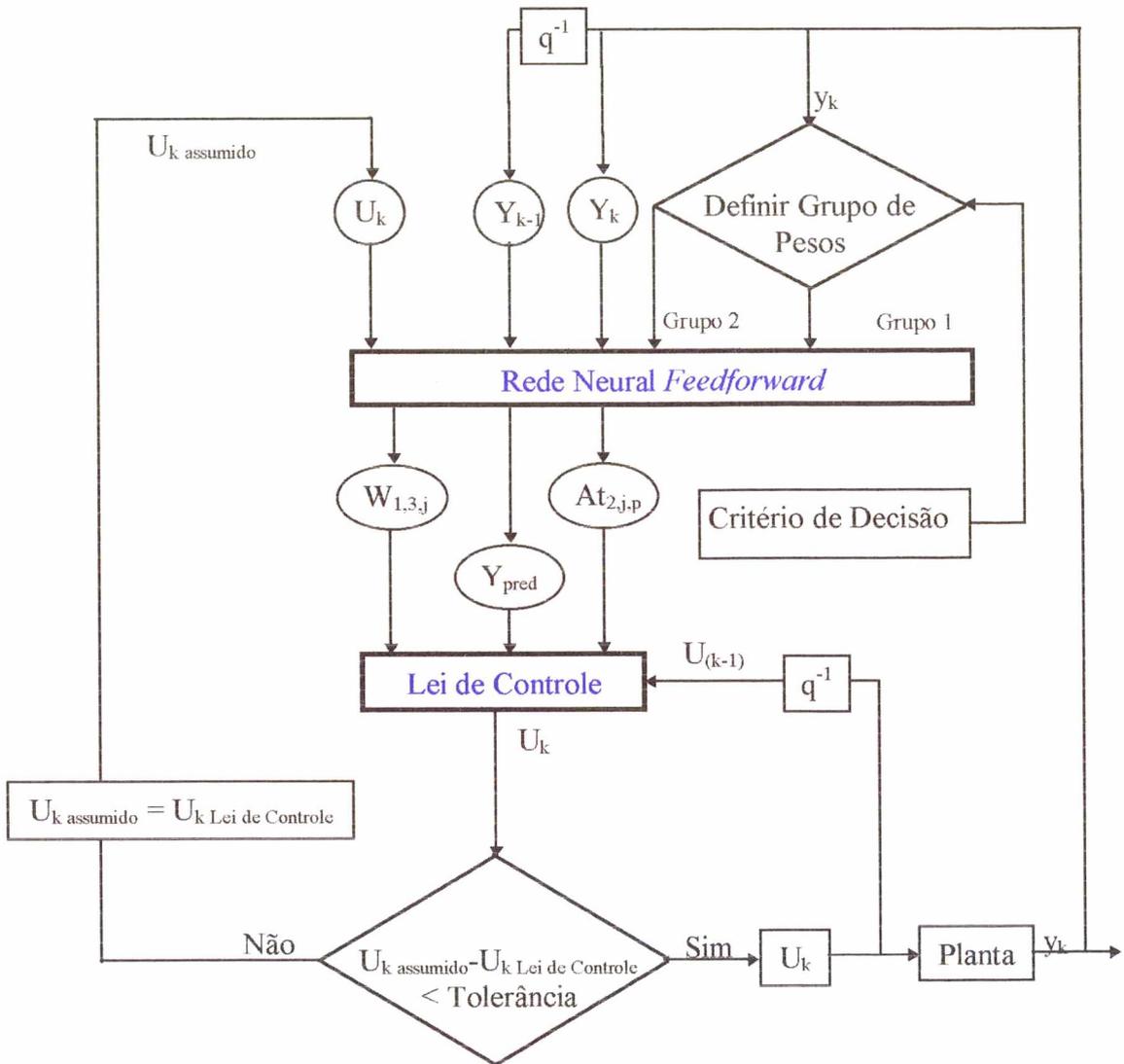


Figura III.11 - Esquema da estratégia de Controle RNE

Como foi visto neste capítulo, o algoritmo *Backpropagation* é fácil de ser deduzido e simples de ser implementado. Variações deste algoritmo podem ser encontradas, utilizando algoritmos de otimização mais potentes e funções de ativação de base radial (RBF) ou tangente hiperbólica. De qualquer forma é um bom algoritmo, bem estudado e com muitas aplicações.

CAPÍTULO IV

MATERIAIS E MÉTODOS

Neste capítulo, um panorama geral do Laboratório de Controle de Processos com um detalhamento dos sistemas, *softwares* e metodologias utilizados nos testes é mostrado.

IV.1. O Laboratório de Controle de Processos.

O Laboratório de Controle de Processos (LCP) está instalado no departamento de Engenharia Química (ENQ) do Centro Tecnológico da Universidade Federal de Santa Catarina (UFSC). O laboratório serve tanto ao desenvolvimento de estratégias de controle em tempo real no curso de Pós-Graduação em Engenharia Química (CPG-ENQ) como ao ensino dos cursos de graduação em Engenharia Química, Engenharia de Alimentos e Engenharia de Controle e Automação da UFSC.

As pesquisas desenvolvidas no LCP são testadas em unidades piloto do laboratório, as quais concentram-se em testes com controle de nível em tanques simples, acoplados, cônico-cilíndrico e controle de temperatura em reatores e trocadores de calor. Neste trabalho, os sistemas utilizados para teste das estratégias de controle foram um tanque cônico-cilíndrico e um reator atuando como trocador de calor.



Figura IV.1 - Vista parcial do Laboratório de Controle de Processos.

IV.1.1. Sistema Tanque Cônico-Cilíndrico.

O sistema para controle de nível do tanque cônico-cilíndrico faz parte de um grupo de sistemas do LCP. Por questões ergonômicas e econômicas este sistema compartilha equipamentos com outros sistemas do laboratório. Este sistema é constituído de um tanque de aço inoxidável com duas seções de comprimento aproximadamente equivalente, e alimentado pela rede hidráulica do laboratório, constituída de tubulações em PVC de 12,7 mm de diâmetro, uma bomba centrífuga de 0,25 cv de potência, uma válvula de controle com acionamento pneumático para tubulações 19,05 mm de diâmetro, e válvulas manuais. O sistema está detalhado nas figuras IV.2 e IV.3.

A parte inferior da seção cônica do tanque, mostrado na figura IV.2, está ligada ao sensor de pressão situado a 0,7 m do tanque, isto significa que o sistema deve ser inicializado manualmente em 0,7 m para o início dos testes.

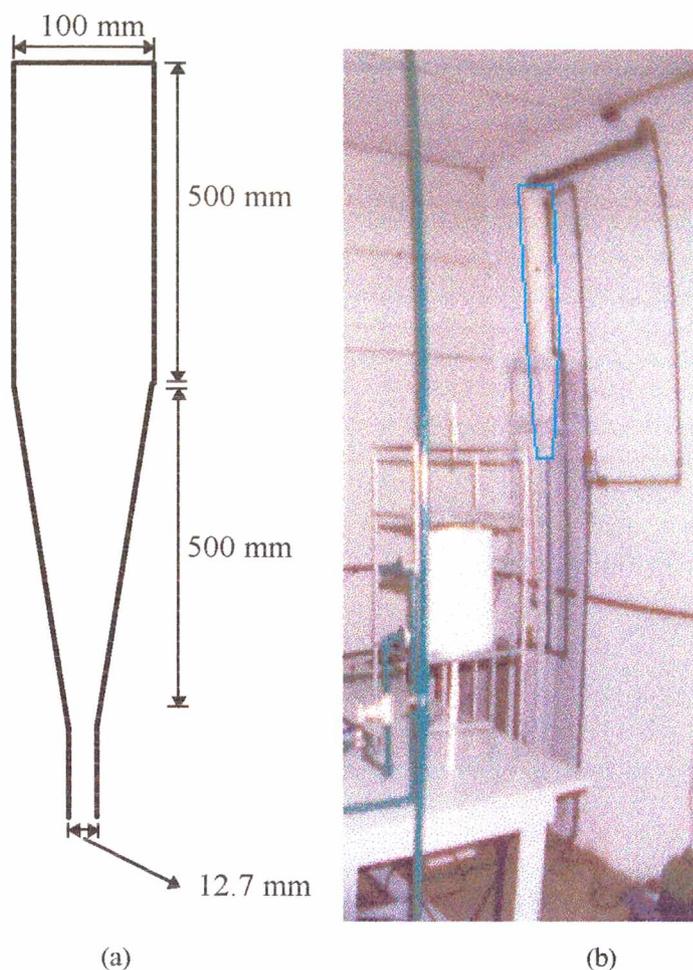


Figura IV.2 - Tanque Cônico-Cilíndrico.

(a)Dimensões. (b) Vista, em detalhe, do equipamento no LCP.

O sistema de controle de nível é muito simples e flexível, alterações simples podem ser realizadas em sua configuração, de forma a incluir atraso no tempo e variações dinâmicas bruscas sem grande esforço. A flexibilidade, facilidade de manuseio e baixo custo de operação deste sistema permite a realização de muitos testes, caracterizando um sistema de alta produtividade para o estudo de técnicas de controle.

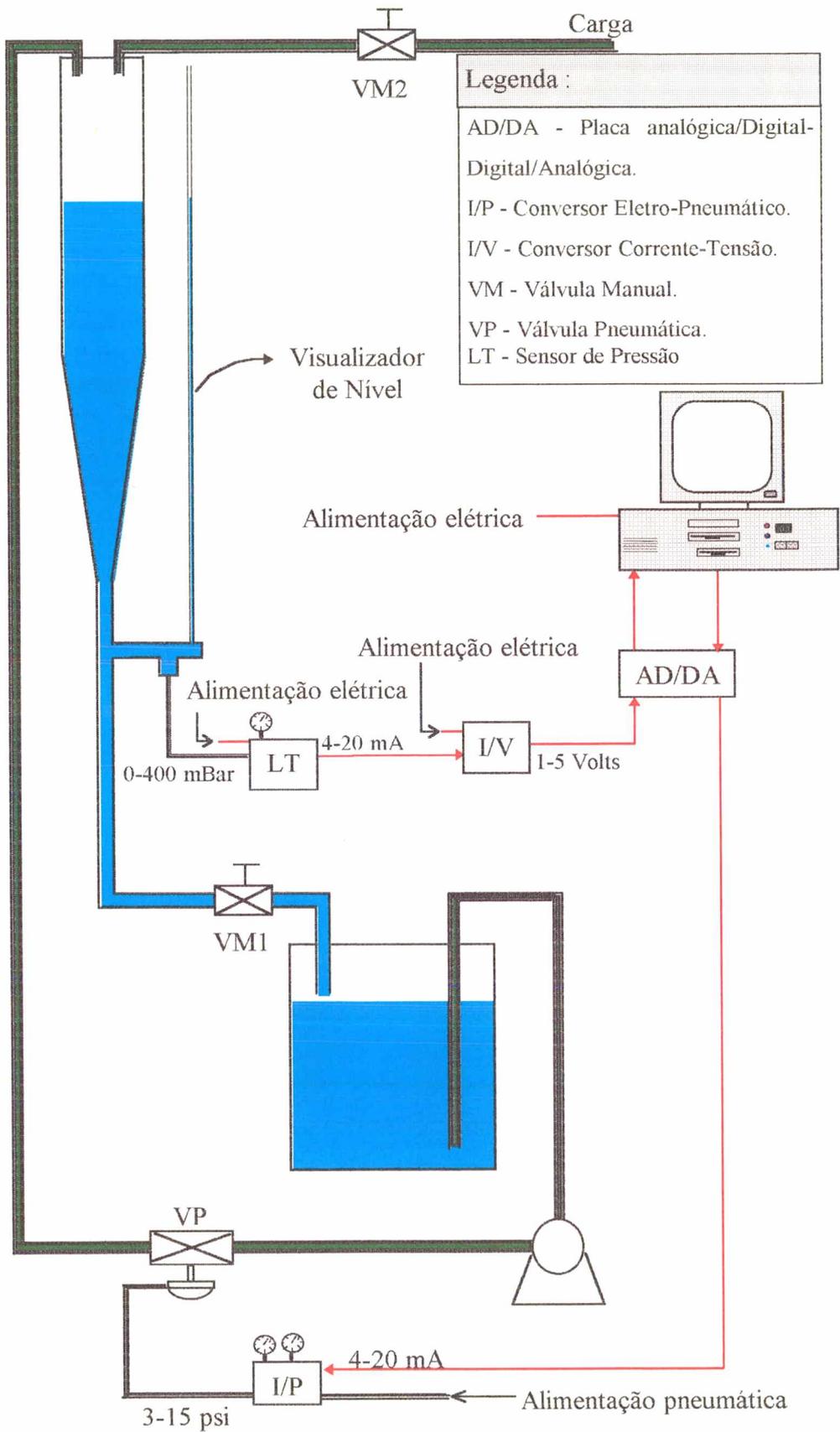


Figura IV.3 - Sistema tanque cônico cilíndrico.

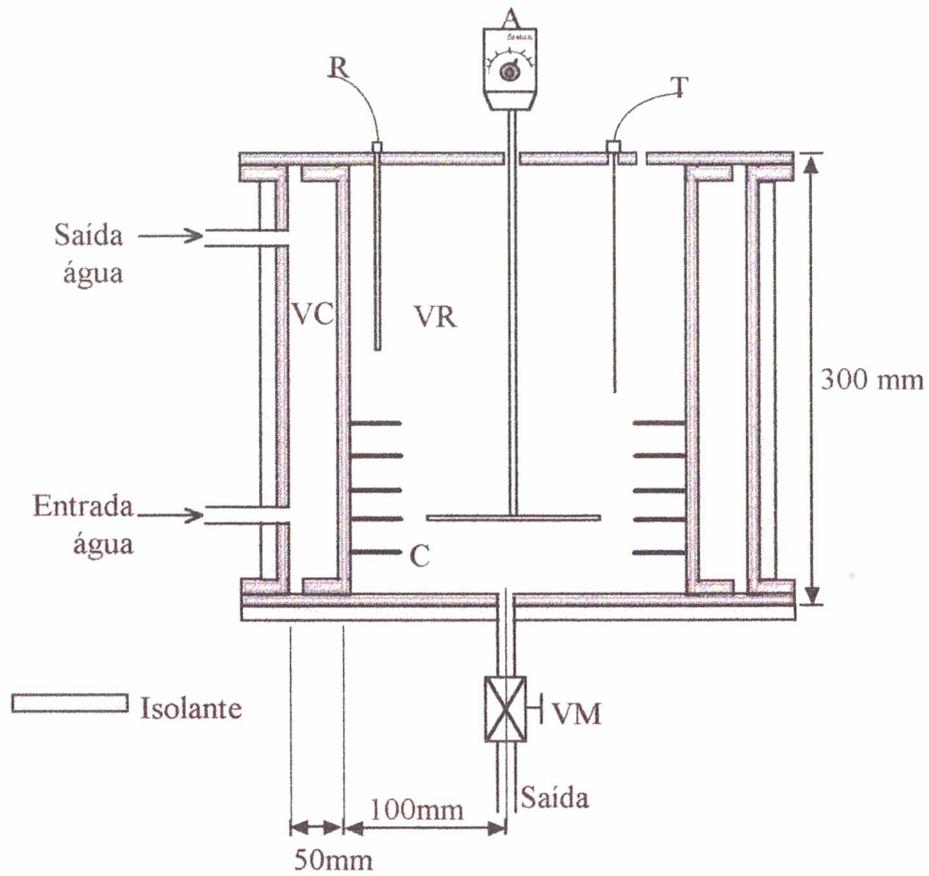
IV.1.2. Trocador de Calor.

Como no reator utilizado nos testes apenas as dinâmicas de aquecimento e resfriamento, forma modeladas, atuando como um trocador de calor, o sistema será denominado apenas trocador de calor (TC). O uso de um reator, e não de um trocador de calor diretamente, foi intencional, com o objetivo de realizar trabalhos futuros com reação química.

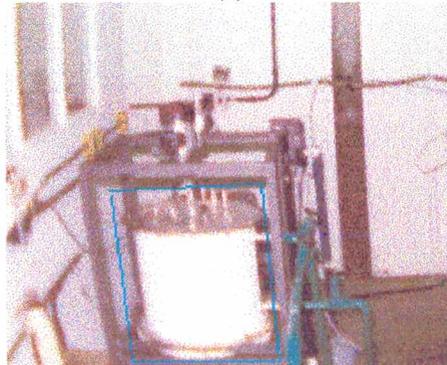
Um TC constituído de um tanque encamisado com 9,3 litros de volume interno foi utilizado para o teste das estratégias de controle propostas no controle da temperatura pela manipulação de uma corrente de água fria e uma corrente de vapor. O sistema foi projetado de forma que o vapor não é alimentado diretamente na camisa do TC. A água na saída do TC é conduzida a um reservatório, de onde é bombeada para um trocador de calor a placas (em contra corrente) onde troca calor com a corrente de vapor. Saída do trocador de calor a placas, a corrente de água mistura-se, em uma conexão do tipo T, à uma corrente fria originária da rede hidráulica do laboratório. A corrente de vapor, na entrada do trocador de calor a placas é manipulada, juntamente à corrente de água fria para aquecer ou resfriar o TC. A estratégia adotada para a manipulação das duas válvulas é conhecida como controle dividido (*split-range control*). A estratégia *split-range* equivale a impor a ação de controle reversa sobre uma das variáveis manipuladas e ação direta para a restante. Neste caso, foi adotada ação direta para a válvula de vapor e ação reversa para a válvula de água fria. Quando a ação de controle direta (U_{direto}) é implementada em 2 volts, por exemplo, a ação reversa (U_{reverso}) equivale a 4 volts, visto que a ação de controle é limitada entre 1 e 5 volts, e dada pela expressão IV.1:

$$U_{\text{reverso}} = 6 - U_{\text{direto}} \quad (\text{IV.1})$$

Na figura IV.4 pode-se observar melhor a estrutura do equipamento utilizado, com detalhes como agitador, chicanas, e resistência para aquecimento elétrico. A resistência elétrica, mostrada na figura IV.4, não foi utilizada neste trabalho, mas aparece para demonstrar a flexibilidade do sistema. Na figura V.5 o sistema completo da malha de controle está detalhado. Nela pode-se observar a incidência de tempo morto no reservatório, no trocador de calor a placas e no próprio TC, pelo seu volume interno e de sua camisa de resfriamento. A soma destes fenômenos é agrupada em um único modelo neural.



(a)



(b)

Legenda :

A - Agitador.

R - Resistência elétrica.

T - Termopar.

VC - Volume da Camisa = 16,5 l.

VR - Volume do Reator = 9,3 l.

VM - Válvula Manual.

C - Chicanas.

Figura - IV.4 - Trocador de Calor

(a) Dimensões. (b) Vista, em detalhe, do equipamento no LCP.

IV.3. O Controle.

A implementação computacional de uma estratégia de controle depende de duas ferramentas, *hardware* e *software* especiais dedicados. O *hardware* utilizado é um conjunto de duas placas, uma de terminais e uma de aquisição de dados (AD/DA). A placa AD/DA utilizada é da marca DataTranslation e possui uma resolução de 12 bits, 24 canais de entrada, 16 digitais e 8 *single ended* e 2 canais de saída analógicos, com faixa de entrada e saída de 1-5 volts. Suas funções são completamente programáveis. A placa possui ainda documentação completa para a implementação em *software*.

Atendendo a sugestão de outros trabalhos, foi desenvolvido um *software* de controle para ambiente Windows™. O *software*, denominado *ArtificialSINAPSER Módulo de Controle*, é um *software* de controle em tempo real e foi desenvolvido com técnicas de orientação a objeto, de forma a permitir o manuseio por pessoas sem o conhecimento de sua estrutura completa. Sua interface gráfica é intuitiva e de fácil manejo. Foi escrito em *Object PASCAL* e as rotinas mais básicas como a de aquisição de dados e as estratégias de controle (PID, RNE e RNG) foram escritas em PASCAL. O *software* conta com três opções de temporizadores com diferentes graus de precisão, configuração de *hardware* em tempo de execução (canais de entrada e saída), permutação entre controle PID, ANN (RNE, RNG) e malha aberta, configuração do controlador neural e pesos da rede neural, etc..

Basicamente, o *software* de controle opera pela ação de um temporizador. Assim, inicialmente procede-se a configuração dos dados de *hardware*, do controlador neural (se desejado), seguida à configuração do tipo de sistema (reator, tanque cônico, tanque simples, tanques acoplados) e da curva de calibração. A curva de calibração é obtida em malha aberta e representa a relação linear entre as medidas realizadas em volts e as medidas reais do sistema (m, °C). Isto é representado em um gráfico do tipo Volt x Metro ou Volt x Grau Celsius e os valores dos coeficientes da equação da reta obtidos são inseridos no *software* através da caixa de diálogo **Configurações** na página **Calibração**.

O botão **Iniciar**, na tela do **Painel de Controle**, inicia a leitura do sistema escolhido se o botão **Malha Aberta/Fechada** estiver pressionado, caso contrário, o processo de controle será

iniciado e os procedimentos de leitura do sistema e manipulação da(s) válvula(s) de controle serão acionados, estando a operação, portanto, em malha fechada.

A figura IV.6 exibe o diagrama de blocos sobre o qual está fundamentado o *software* de controle.

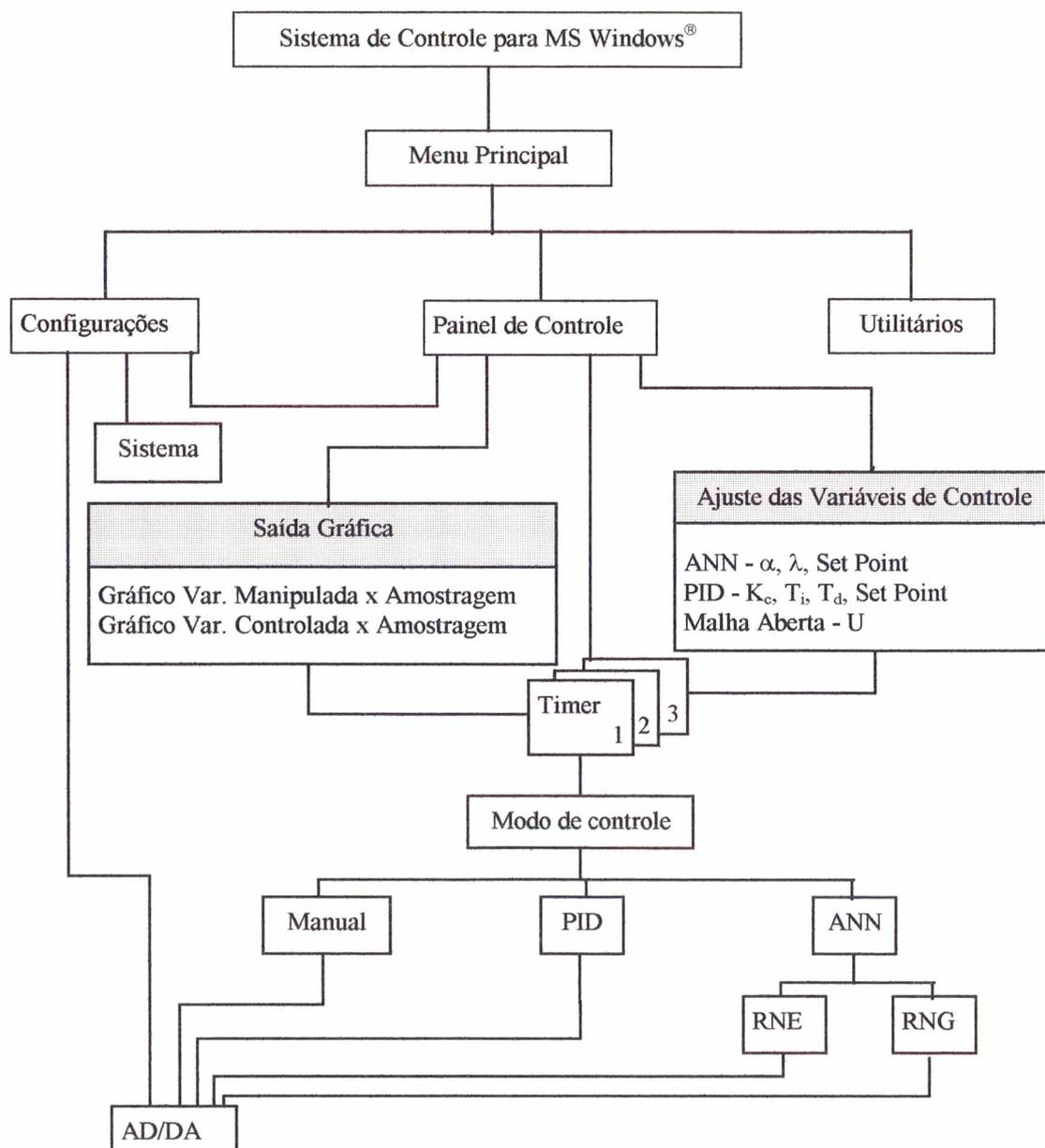


Figura IV.6 - Diagrama de blocos do *software* de controle.

No diagrama de blocos do *software* de controle, destacam-se as rotinas de aquisição de dados e dos controladores. A rotina de aquisição de dados por se caracterizar de uma programação de baixo nível é a mais difícil de ser implementada.

Para a implementação das estratégias de controle propostas, um microcomputador 386 DX 40 MHz com 4 MBytes de memória RAM e 230 MBytes de disco rígido operando com sistema operacional MS DOS 6.0 e MS Windows[®] 3.11 foi utilizado. Com isso, pode-se observar a baixa exigência computacional do *software*.

Nas figuras IV.7 e IV.8, as telas principal e de configuração de *hardware* podem ser vistas. Uma característica interessante da tela principal (Painel de Controle) é que os gráficos de visualização da ação de controle e variável manipulada podem armazenar 5000 amostras, o que equivale a aproximadamente 7 horas com intervalo de amostragem igual a 5 segundos. A visualização dos dados armazenados é feita simplesmente pela movimentação da barra de rolagem sob o gráfico.

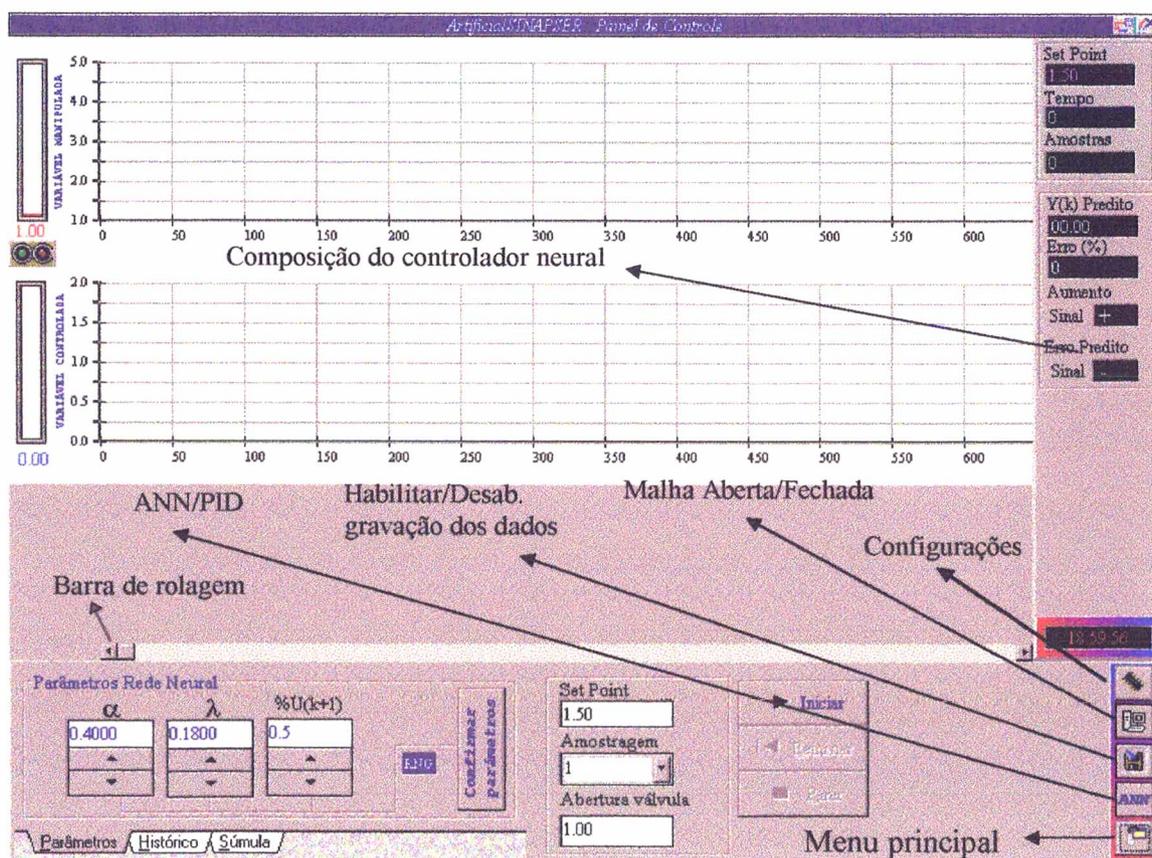


Figura IV.7 - Tela principal do *software* de controle utilizado.

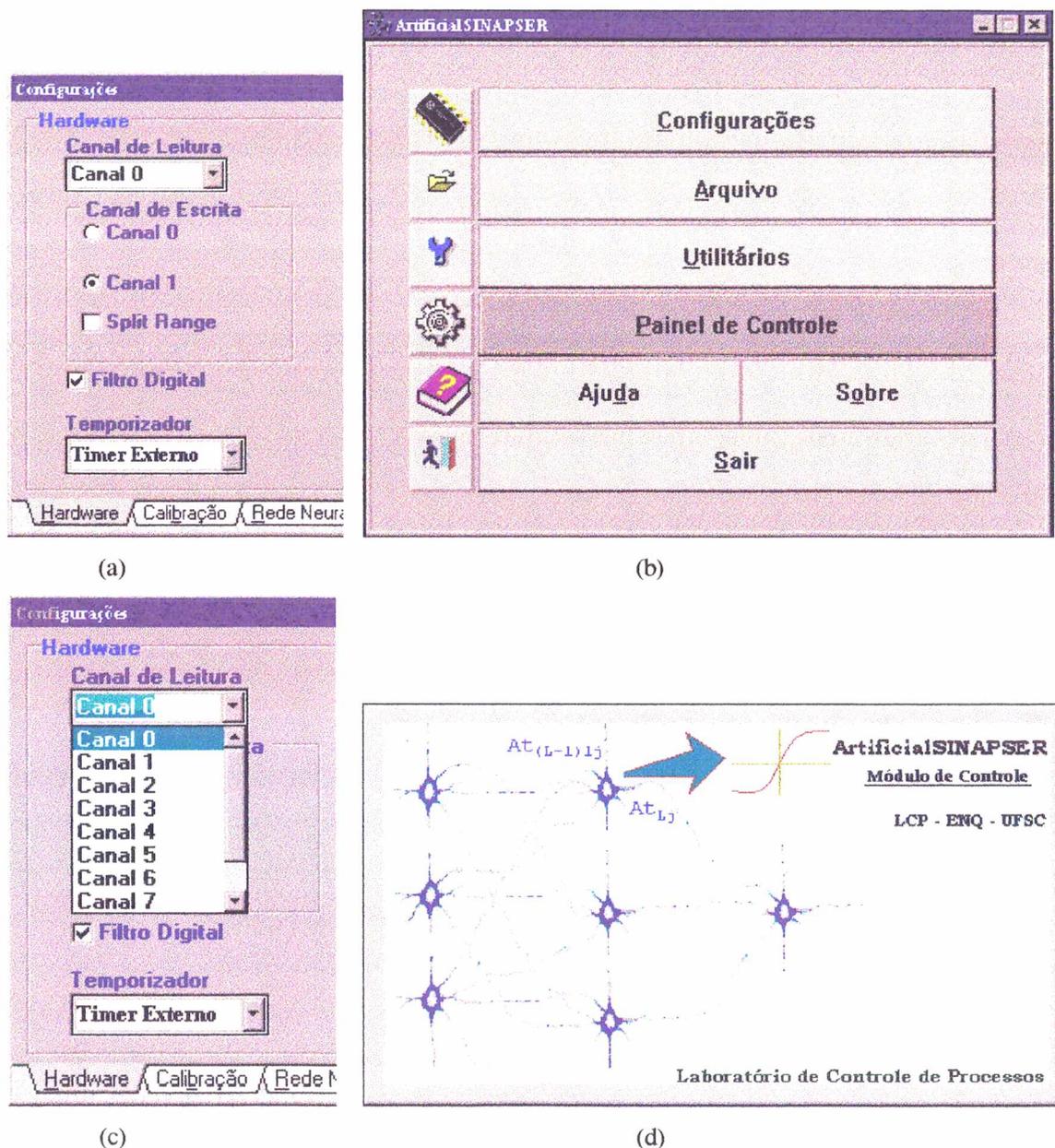


Figura IV.8 - (a) e (c) Tela de Configuração de *Hardware*. (b) Menu principal. (d) Tela de abertura.

A inicialização da malha de controle de temperatura deve ser feita enchendo-se o reator com 8 litros d'água, acionando o sistema de agitação e inserindo o termopar no Trocador de Calor (figura IV.4). Feito isso, estando acionado o amplificador e selecionado o canal de leitura correto, o processo pode ser posto em marcha.

As especificações de todos os equipamentos utilizados encontram no Apêndice I.

IV.4. O Treinamento.

O treinamento é uma das partes mais importante de um processo de implementação de um controlador preditivo baseado em um modelo neural. Além disso, ele requer um significativo tempo computacional. O tempo consumido no treinamento de uma rede neural *Backpropagation* depende da quantidade de padrões e, é claro, da velocidade do computador utilizado. Para o treinamento das redes neurais usadas, um microcomputador *pentium*[®] 75 MHz com 16 MBytes de memória RAM e 840 MBytes de disco rígido e um microcomputador 486 DX4 100 MHz com 8 MBytes de memória RAM e 540 MBytes de disco rígido foram utilizados.

O *software* utilizado foi produzido para operar em sistema operacional MS Windows[®] 95. Uma versão (escrita em FORTRAN) para DOS desenvolvida por Rita (1995) serviu como partida para o desenvolvimento deste *software*. O *software*, denominado *ArtificialSINAPSER Módulo de Treinamento*, apresenta uma interface de fácil adaptação para o treinamento de redes neurais no uso como modelos para controle. O *software* não objetiva uso com propósito geral e, portanto, limita-se ao uso nos casos de controle preditivo com e sem tempo morto. Esta limitação pode ser eliminada pela inclusão de novas rotinas para a de padrões. Um acompanhamento gráfico da evolução do treinamento facilita a avaliação da qualidade do modelo e do erro da rede relativo às saídas reais. O *software* permite a intervenção do usuário durante o processo, permitindo, por exemplo, a alteração da taxa de aprendizagem e a parada do mesmo. Após parado, este pode ser reiniciado exatamente no ponto que a intervenção foi efetuada, permitindo maleabilidade e segurança nos ensaios.

O diagrama de blocos do *software* de treinamento é apresentado na figura 4.1. As telas não apresentadas a tela principal e uma das telas de opções do programa para treinamento.

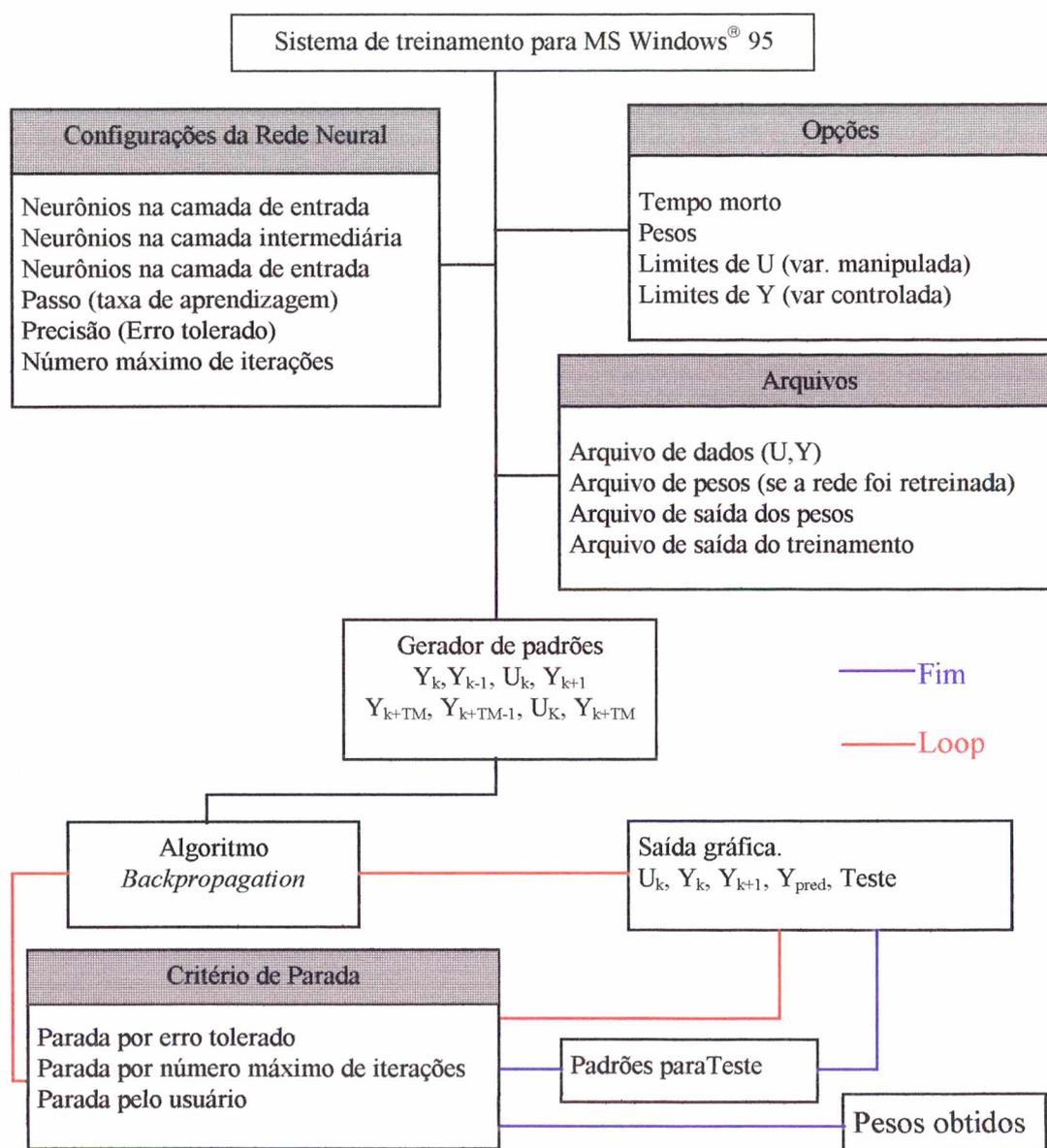
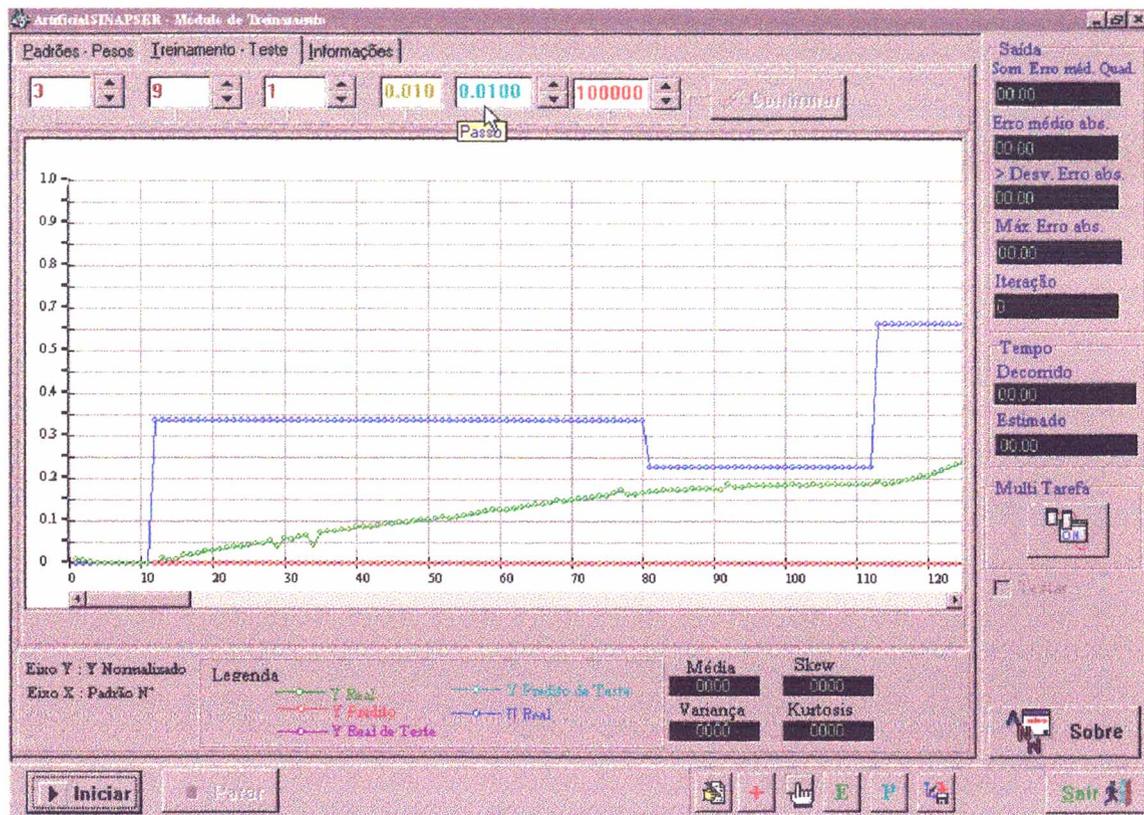
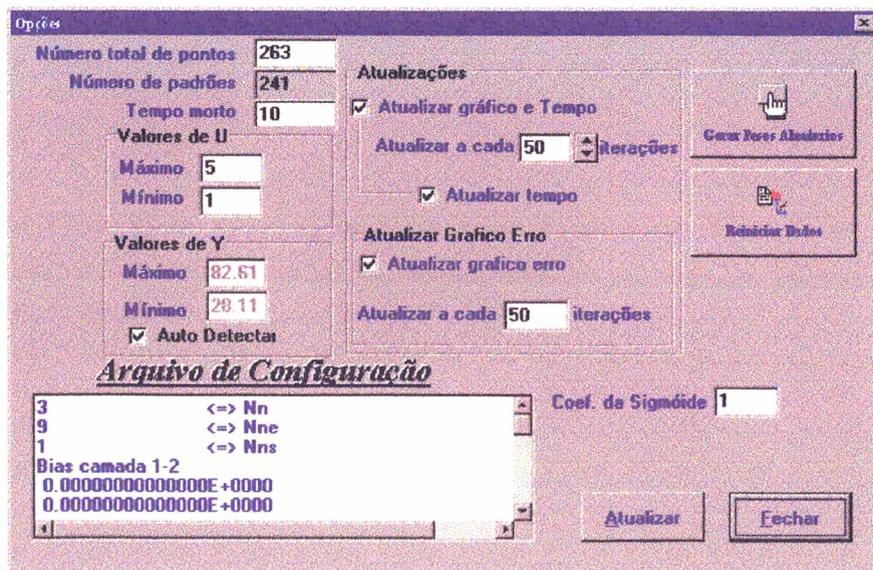


Figura IV.9 - Diagrama de blocos do *software* de treinamento.



(a)



(b)

Figura IV.10 - (a) Tela principal do *software* de treinamento. (b) Tela com algumas opções do *software*.

Um ponto importantíssimo a ser considerado é a definição do tamanho da rede neural. O tamanho é tecnicamente denominado arquitetura ou topologia. A arquitetura de uma rede neural

compreende o número de camadas e o número de neurônios que cada camada deve conter. O número de neurônios na camada de entrada é função das variáveis que se deseja infundir à rede. Neste caso o número de neurônios na primeira camada é 3, Y_k , Y_{k-1} , U_k , ou Y_{k+TM} , Y_{k+TM-1} , U_k . O número de neurônios na camada de saída é dependente do número de saídas desejado. Como neste caso, o que se deseja é obter a predição de uma saída futura (Y_{k+1} ou Y_{k+TM}) o número de neurônios na camada de saída é 1. Como o número de camadas adotado é 3, resta portanto, o número de neurônios na camada intermediária a ser definido. Este número não pode ser pequeno demais para haver uma aproximação pobre e nem grande demais que gere perda da generalidade da rede. Como não existem técnicas formais para a definição do número de neurônios na camada intermediária, técnicas de tentativa e erro são comumente adotadas. Após testes variando o número de neurônios na camada intermediária, obteve-se 9 como o número de neurônios necessários para uma aproximação satisfatória.

Além da determinação da arquitetura da rede, a obtenção dos padrões é outro ponto importante para a qualidade final do modelo neural. Os padrões são obtidos a partir de dados coletados da planta em malha aberta. Os dados devem ser representativos do processo, portanto devem ser coletados em toda a faixa de trabalho do sistema que se deseja modelar. Quanto da existência de atrasos como tempo morto a coleta deve ser procedida de forma à permitir a manifestação do fenômeno sobre a variável a controlar. Uma maneira prática de se observar a formação dos padrões a partir de um grupo de dados é a disposição em “janelas”, como mostram as tabelas IV.1, IV.2 e IV.3

Tabela IV.1 - Primeiro padrão de treinamento

| k | Y_k | U_k |
|---|-------|-------|
| 1 | Y_1 | U_1 |
| 2 | Y_2 | U_2 |
| 3 | Y_3 | U_3 |
| 4 | Y_4 | U_4 |
| 5 | Y_5 | U_5 |

Tabela IV.2 - Segundo padrão de treinamento

| k | Y_k | U_k |
|---|-------|-------|
| 1 | Y_1 | U_1 |
| 2 | Y_2 | U_2 |
| 3 | Y_3 | U_3 |
| 4 | Y_4 | U_4 |
| 5 | Y_5 | U_5 |

Tabela IV.3 - Terceiro padrão de treinamento

| k | Y_k | U_k |
|---|-------|-------|
| 1 | Y_1 | U_1 |
| 2 | Y_2 | U_2 |
| 3 | Y_3 | U_3 |
| 4 | Y_4 | U_4 |
| 5 | Y_5 | U_5 |

Como é necessário o valor de Y_{k+1} para avaliar a qualidade dos pesos obtidos no treinamento, o valor de Y_{k+1} pode ser incluído na formação dos padrões. A não inclusão de Y_{k+1} na formação dos padrões pode ser feita mediante a consciência de que o último dado do grupo utilizado é necessário para a formação do padrão $k-2$.

A organização dos padrões quando um processo sofre a influência de tempo morto é semelhante á descrita anteriormente, considerando o tempo morto. Considerando-se um tempo morto igual a 3 intervalos de amostragem, o deslocamento da janela de padrões resulta nas tabelas IV.4, IV.5 e IV.6.

Tabela IV.4 - Primeiro Padrão com Tempo Morto igual a 3 intervalos de amostragem.

| k | Y_k | U_k |
|-----|-------|-------|
| 1 | Y_1 | U_1 |
| 2 | Y_2 | U_2 |
| 3 | Y_3 | U_3 |
| 4 | Y_4 | U_4 |
| 5 | Y_5 | U_5 |
| 6 | Y_6 | U_6 |
| 7 | Y_7 | U_7 |
| 8 | Y_8 | U_8 |
| 9 | Y_9 | U_9 |

Tabela IV.5 - Segundo Padrão com Tempo Morto igual a 3 intervalos de amostragem.

| k | Y_k | U_k |
|-----|-------|-------|
| 1 | Y_1 | U_1 |
| 2 | Y_2 | U_2 |
| 3 | Y_3 | U_3 |
| 4 | Y_4 | U_4 |
| 5 | Y_5 | U_5 |
| 6 | Y_6 | U_6 |
| 7 | Y_7 | U_7 |
| 8 | Y_8 | U_8 |
| 9 | Y_9 | U_9 |

Tabela IV.6 - Terceiro Padrão com Tempo Morto igual a 3 intervalos de amostragem.

| k | Y _k | U _k |
|---|----------------|----------------|
| 1 | Y ₁ | U ₁ |
| 2 | Y ₂ | U ₂ |
| 3 | Y ₃ | U ₃ |
| 4 | Y ₄ | U ₄ |
| 5 | Y ₅ | U ₅ |
| 6 | Y ₆ | U ₆ |
| 7 | Y ₇ | U ₇ |
| 8 | Y ₈ | U ₈ |
| 9 | Y ₉ | U ₉ |

As tabelas IV.4 IV.5 e IV.6 ilustram o deslocamento temporal dos componentes de um padrão onde um tempo morto igual a três intervalos de amostragem é considerado. A predição de T_M intervalos de tempo futuros elimina (parcialmente) a influência do tempo morto sobre o sistema. Como o tempo morto é difícil de ser determinado exatamente, e em certos casos é variável, seu efeito não pode ser eliminado completamente. Desta forma, a composição de um padrão com $T_M = 3$ exige 7 elementos, ou seja, 6 intervalos de amostragem. Assim, um grupo de 9 elementos produz (como visto nas tabelas IV.4 IV.5 e IV.6) 3 padrões. Genericamente, um processo com um tempo morto (T_M) apresenta $K-2*T_M$ padrões.

Uma consideração que dever ser observada é relativa ao intervalo de atuação da função de ativação sigmoideal. Como este intervalo é limitado entre 0 e 1 os padrões também devem estar limitados nesta faixa. Para tanto, procede-se a normalização dos dados conforme as equações IV.2 e IV.3:

$$y = \frac{y - y_{min}}{y_{max} - y_{min}} \quad (IV.2)$$

$$U = \frac{U - U_{\min}}{U_{\max} - U_{\min}} 0.6 + 0.2 \quad (\text{IV.3})$$

Como os valores de U_{\max} e U_{\min} são constantes e iguais a 5 e 1 (volts):

$$U = \frac{U - 1}{5 - 1} 0.6 + 0.2 = \frac{U - 1}{4} 0.6 + 0.2 \quad (\text{IV.4})$$

Neste capítulo, uma noção dos métodos utilizados para a implementação das estratégias de controle, foi apresentada, com enfoque especial aos sistemas e *softwares* de controle utilizados. A implementação, em tempo real, de uma estratégia de controle é uma tarefa gratificante nos estudos com controle de processos, e é seguramente a tarefa mais agradável. A integração direta com um sistema real traz uma quantidade enorme de informações estimulando novos estudos.

CAPÍTULO V

RESULTADOS E DISCUSSÕES

O desenvolvimento de um *software*, apesar das modernas ferramentas de programação, sempre consome um bom tempo de trabalho. A necessidade do desenvolvimento de um aplicativo para controle de processos é maior por sua característica e especificidade, o que restringe bastante o uso dos chamados pacotes comerciais. Além do que, o uso de compiladores, ao invés do uso de pacotes, permite uma maior flexibilidade e velocidade. Neste trabalho, dois *softwares* também podem ser apresentados como resultados, sem contudo ser o objetivo desta dissertação.

Duas estratégias de controle baseadas em redes neurais artificiais foram testadas e comparadas a um controlador do tipo PID. Cada uma destas estratégias foi aplicada a dois sistemas, um tanque com dinâmica variante no tempo e um trocador de calor. Estes processos, de uma forma geral englobam típicos problemas de controle, como tempo morto, dinâmica variante no tempo e histerese. Desta forma os resultados serão apresentados conforme estes dois sistemas e na seção final alguns comentários comparativos serão tecidos.

V.1 Resultados para o Tanque Cônico-Cilíndrico.

Além das não linearidades inerentes dos sistemas tanque e trocador de calor (TC), outra consideração importante é a não linearidade das próprias válvulas de controle. Por uma característica própria, igual percentagem, a válvula representa uma não linearidade adicional. Além desta não linearidade, outro fator é a histerese, característica das válvulas deste modelo. Estas características podem ser observadas na figura V.1.

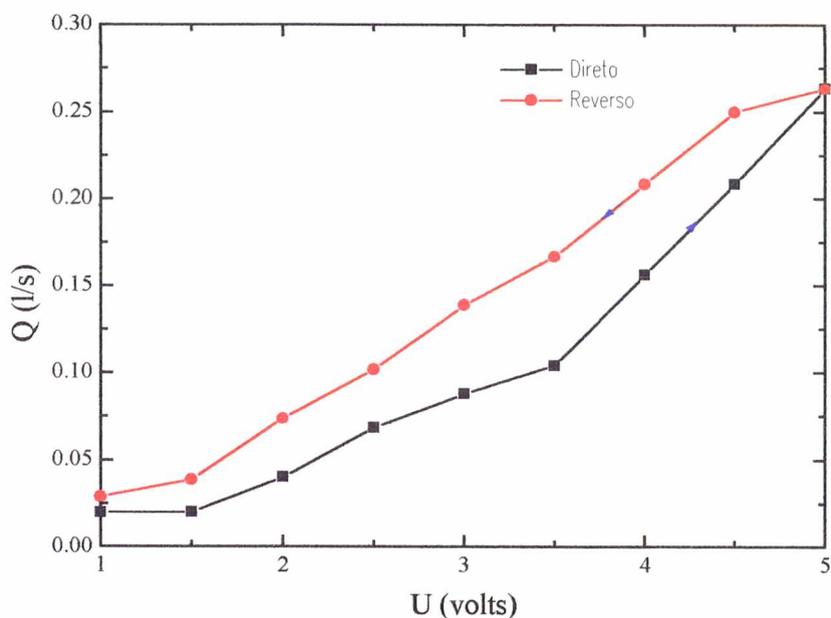
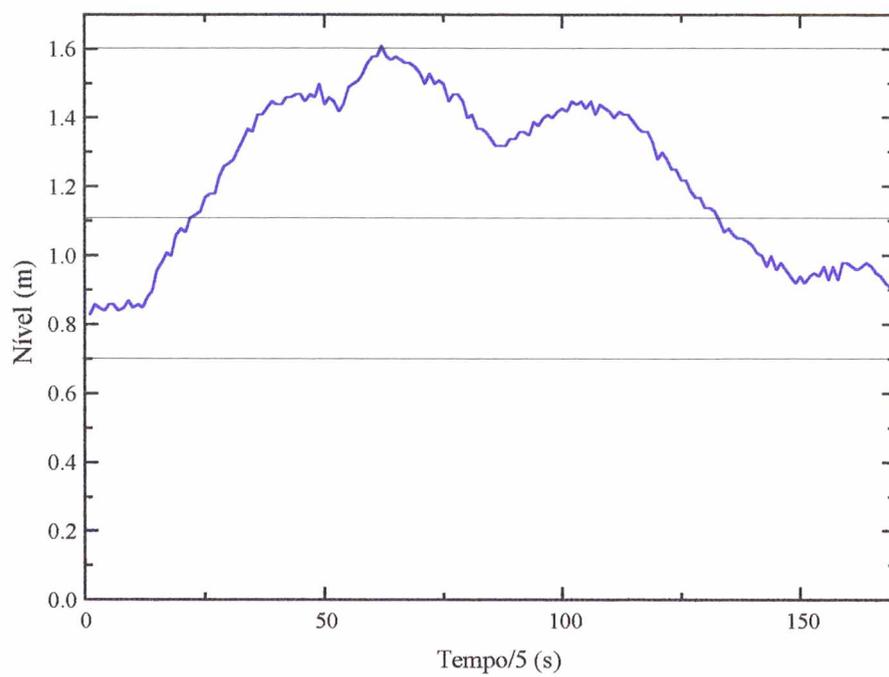


Figura V.1 - Dinâmica e histerese da válvula controle de água fria.

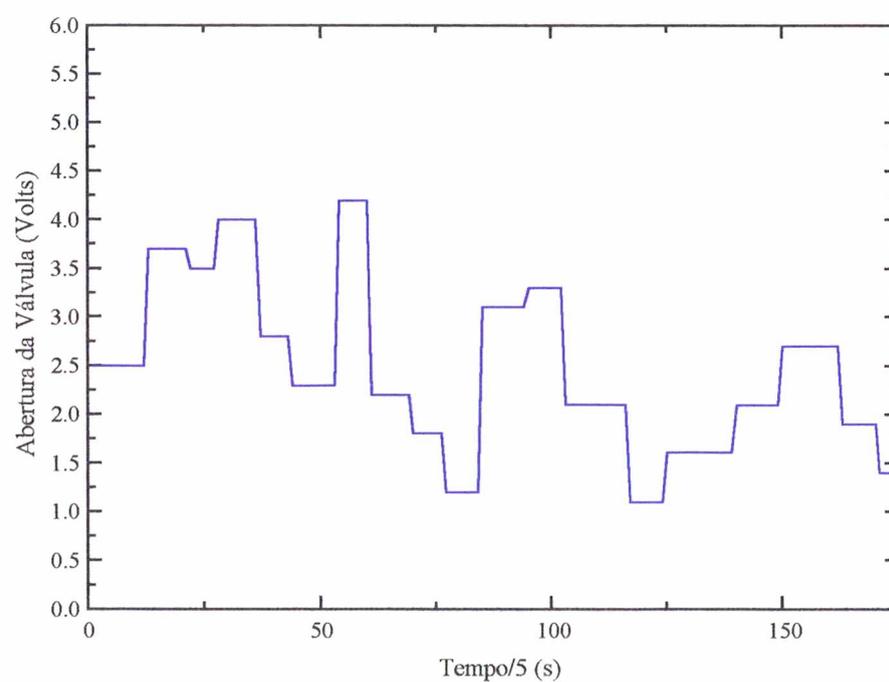
V.1.1 Treinamento.

Para o treinamento das redes neurais utilizadas como modelo para as estratégias de controle RNE e RNG três grupos de dados foram necessários. Os grupos de dados foram coletados *off line*. Para a estratégia RNE os dados foram coletados respeitando o limite seção cônica-seção cilíndrica (1,10m). Os dados utilizados para o treinamento dos modelos RNE e RNG foram coletados em intervalos de 5 segundos e estão nas figuras V.2, V.3 e V.4. Entenda-se por modelos RNE e RNG os modelos neurais para as estratégias de controle RNE e RNG, respectivamente.

A figura V.2 mostra o grupo de dados utilizado para a obtenção do modelo RNG. Variações entre 1 e 4.5 volts (0 e 87,5 % da abertura) na válvula de controle geraram variações no nível do tanque entre 0.8 e 1.6 metros. Isto garantiu uma boa caracterização do sistema, considerando que estas variações foram obtidas por diferentes posições na válvula de controle.



(a)



(b)

Figura V.2 - Grupo de dados de treinamento para o modelo RNG.

(a) Nível. (b) Abertura da válvula.

As figuras V.3 e V.4 representam os grupos de dados utilizados no treinamento das duas redes neurais para o modelo RNE. Na figura V.3 estão os dados coletados na seção cônica do tanque e são correspondentes ao primeiro grupo de dados do modelo RNE. Na figura V.4 estão os dados correspondentes ao segundo grupo de dados do modelo RNE, representando a seção cilíndrica.

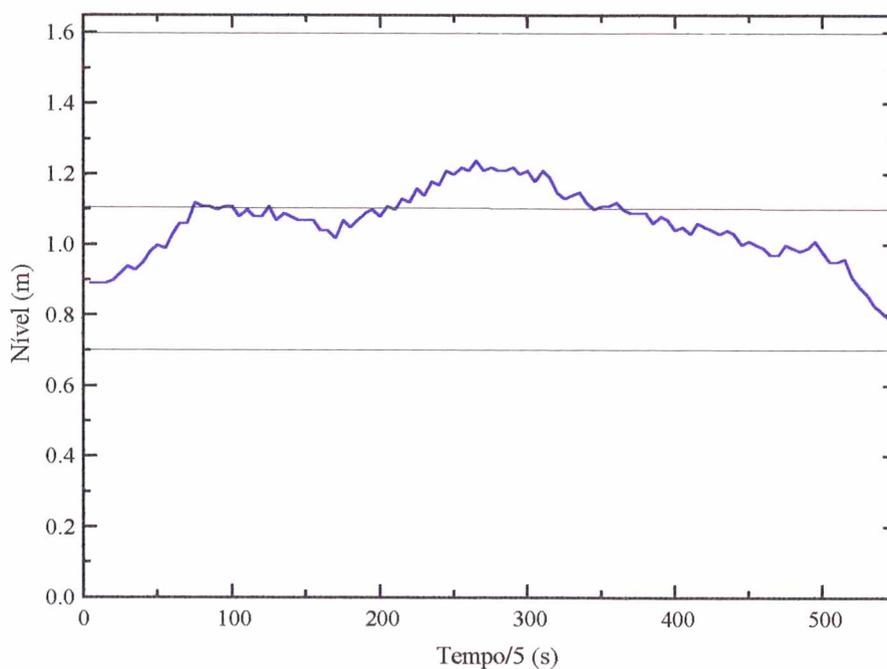


Figura V.3 (a) - Grupo dados de treinamento para o modelo RN.

Nível seção cônica

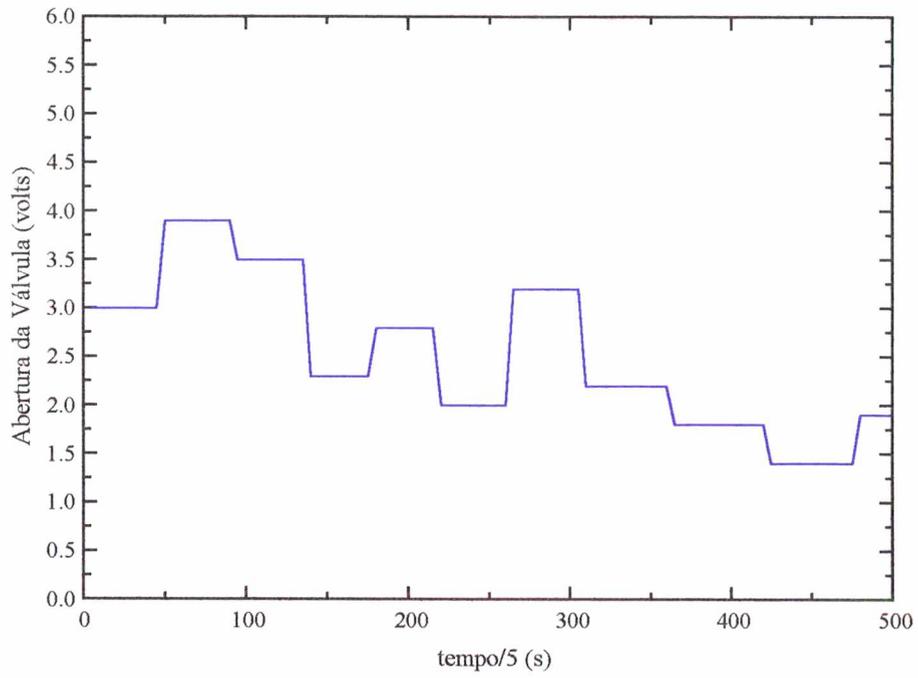


Figura V.3 (b) - Grupo dados de treinamento para o modelo RNE.

Abertura da válvula para a seção cônica.

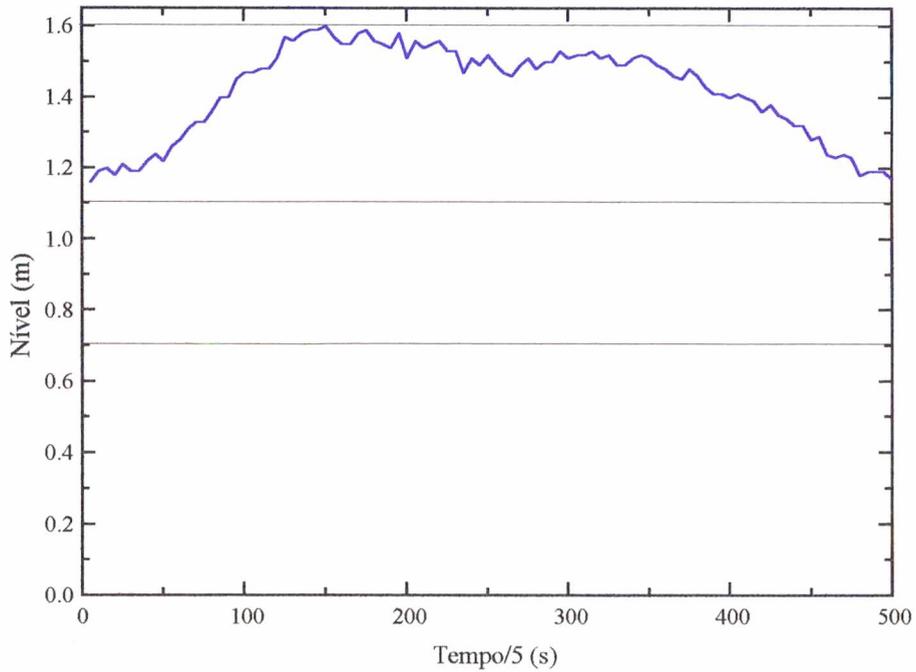


Figura V.4 (a) - Grupo de dados de treinamento para o modelo RNE.

Nível seção cilíndrica.

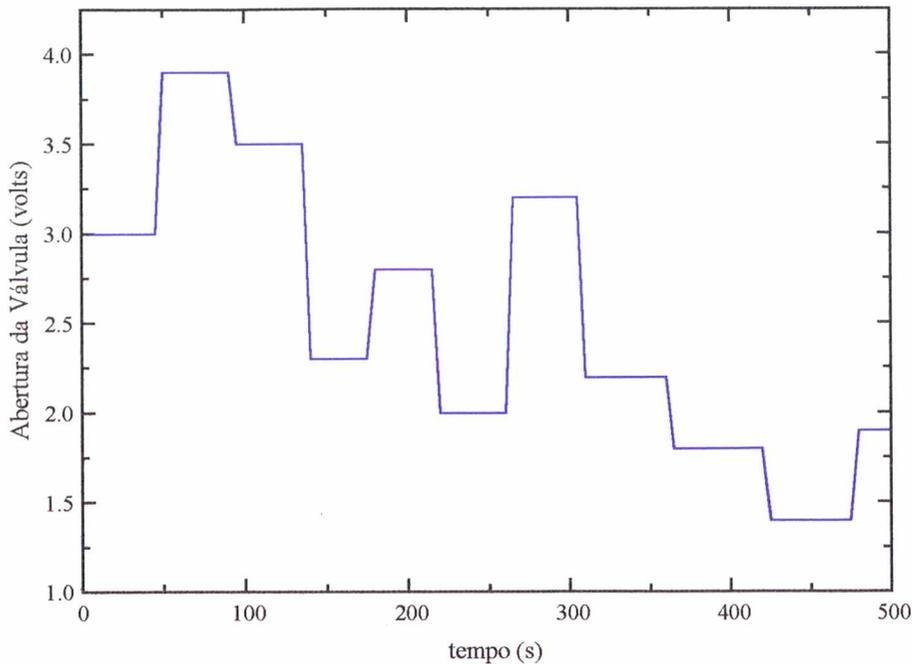


Figura V.4 (b) - Grupo de dados de treinamento para o modelo RNE.

Abertura da válvula para a seção cilíndrica.

Para testar a validade do modelo obtido *off line* o modelo pode ser submetido a um grupo extra de dados. Este é um grupo diferente do utilizado para o treinamento e deve ser obtido da planta real da mesma forma que o grupo padrão. O *software* de treinamento foi projetado para suprir esta necessidade, assim simplesmente deve ser fornecido um arquivo com os dados para o teste e quando o treinamento é findo o teste é realizado automaticamente. A qualidade dos modelos obtidos pode ser visualizada nas figuras V.5 e V.6.

Os pesos utilizados como modelo para as estratégias utilizadas não serão apresentados neste trabalho por serem de pouca representação visual e particulares da malha de controle adotada, mudanças na malha forçam um novo treinamento e a geração de novos pesos. Apenas como uma idéia da magnitude dos pesos utilizados, o Apêndice III lista os pesos usados como modelo para o controle de nível.

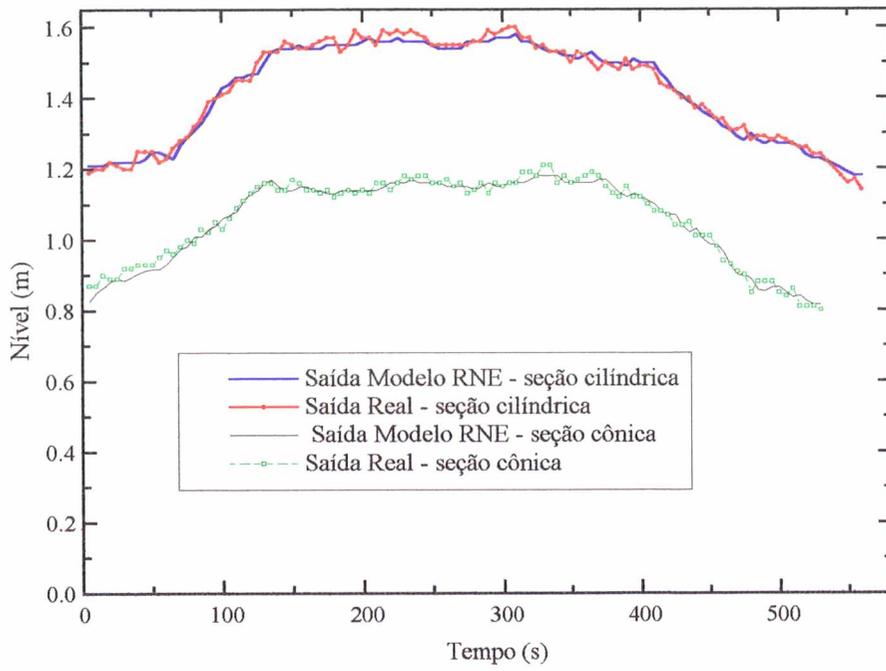


Figura V.5 Teste do modelo RNE.

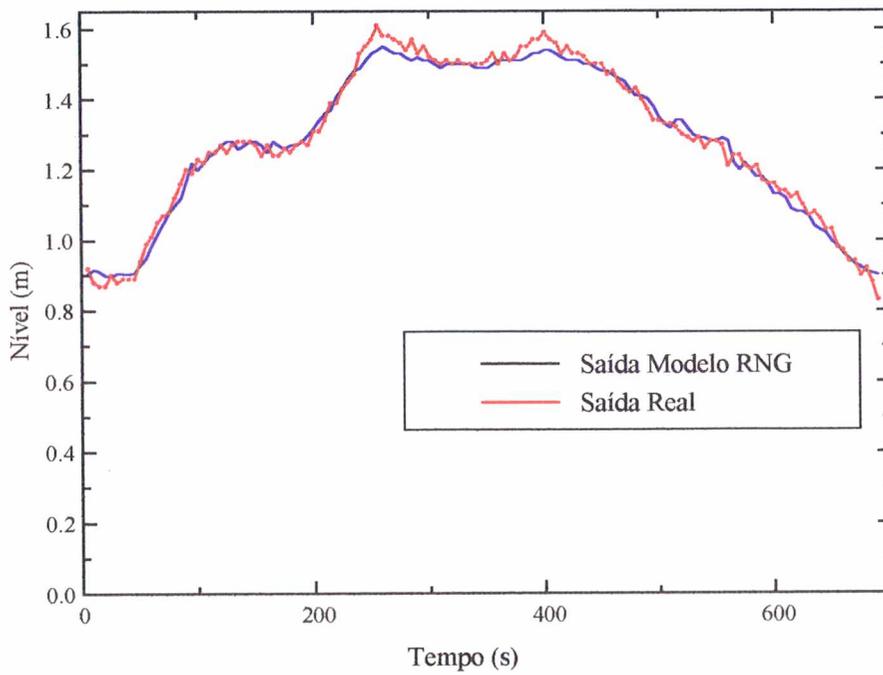


Figura V.6 - Teste do modelo RNG.

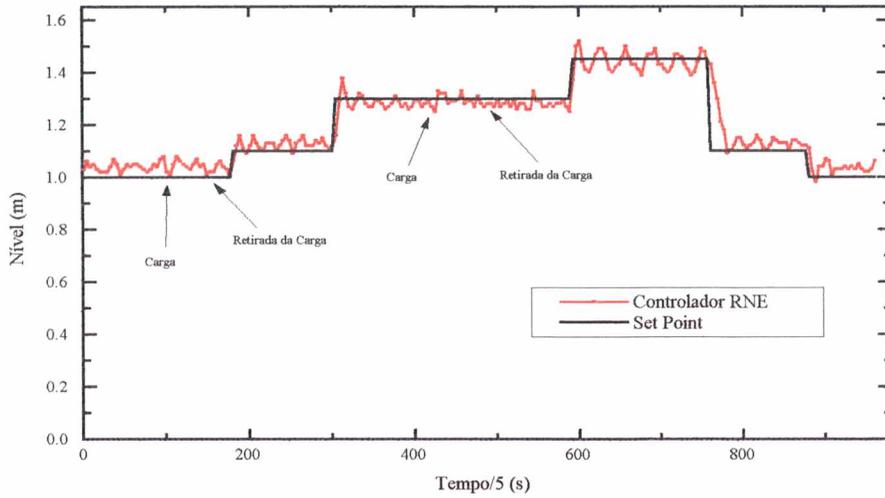
Pode ser facilmente observado, pela comparação das figuras V.5 e V.6, que a capacidade de interpolação do modelo RNE é bastante superior à do modelo RNG. Os dois grupos de pesos do modelo RNE forneceram melhores resultados, quando submetidos a grupos de dados diferentes dos quais foram treinados, que os obtidos com o modelo RNG. Isto é esperado, considerando que o modelo RNE tem informações mais específicas que o modelo RNG. Esta vantagem confere ao modelo RNE maior robustez e capacidade de rejeição a ruídos.

V.1.2 Controle.

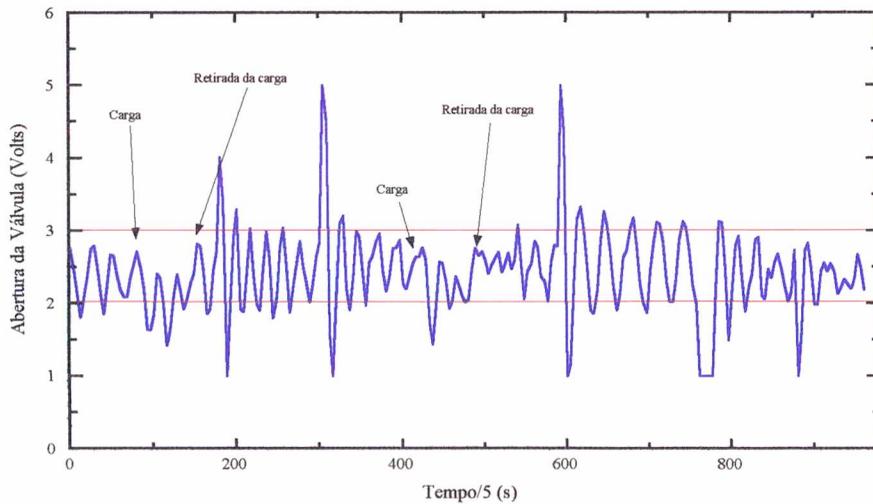
Para o controle do sistema tanque cônico-cilíndrico, este deve ser inicializado com o nível em 0.7 m, isto em função do sensor de pressão estar posicionado a 0.7 m do tanque (ver figura IV.2). Antes da inicialização os pesos do modelo neural (obtidos durante o treinamento) devem ser inicializados no *software* de controle através da página **Rede Neural** na caixa de diálogo **Configurações**. Inicializados os pesos, o processo pode ser posto em malha fechada e submetido aos testes desejados.

Uma prática comum para o teste de uma estratégia de controle é submetê-la a perturbações do tipo carga e mudanças no *setpoint*. Isto pode ser observado na figura V.7. Neste sistema, as cargas foram aplicadas através da válvula manual VM2 (figura IV.3), representando 30 % da vazão de entrada imposta pela válvula de controle VP (figura IV.2).

A figura V.7 também apresenta a robustez do modelo neural. Nesta figura, um modelo (RNE) com apenas 10000 (dez mil) iterações foi submetido a um teste rígido. Mesmo com pouco tempo de treinamento, a rede neural foi capaz de mapear bem o processo produzindo um controle bom, sem ações de controle exageradas.



(a)



(b)

Figura V.7 - Performance do Controlador RNE (10000 iterações) à mudanças do tipo carga e Setpoint sucessivas. (a) Nível. (b) Ação de Controle.

Na figura V.8, as estratégias de controle RNE e RNG (submetidas a 50000 (cinquenta mil) iterações) são comparadas, revelando uma performance superior para a estratégia RNE, com *offset* e sobrelevação (*overshoot*) menores. Com relação às ações de controle, a figura V.8b mostra, que os dois controladores se equivalem utilizando aproximadamente 25 % da faixa de ação da válvula. A estratégia RNE revelou-se significativamente mais capaz de aproximar o

processo, que a estratégia RNG. Vale lembrar que a estratégia RNE é composta de duas redes neurais, enquanto a estratégia RNG é composta de apenas uma rede neural. No que se refere ao ajuste do controlador, quanto melhor for o modelo, menos suscetível à variações bruscas (por modelagem deficiente) ele será, e mais fácil será o ajuste do controlador.

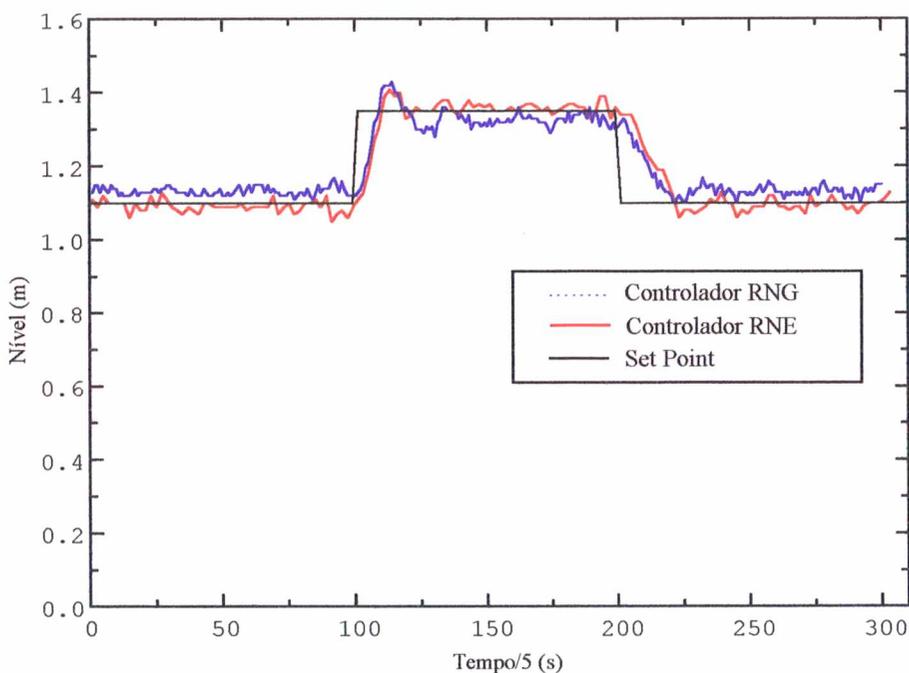


Figura V.8 (a) - Comparação da performance dos controladores RNE e RNG (50000 iterações) à um degrau no *Setpoint*. Nível do tanque.

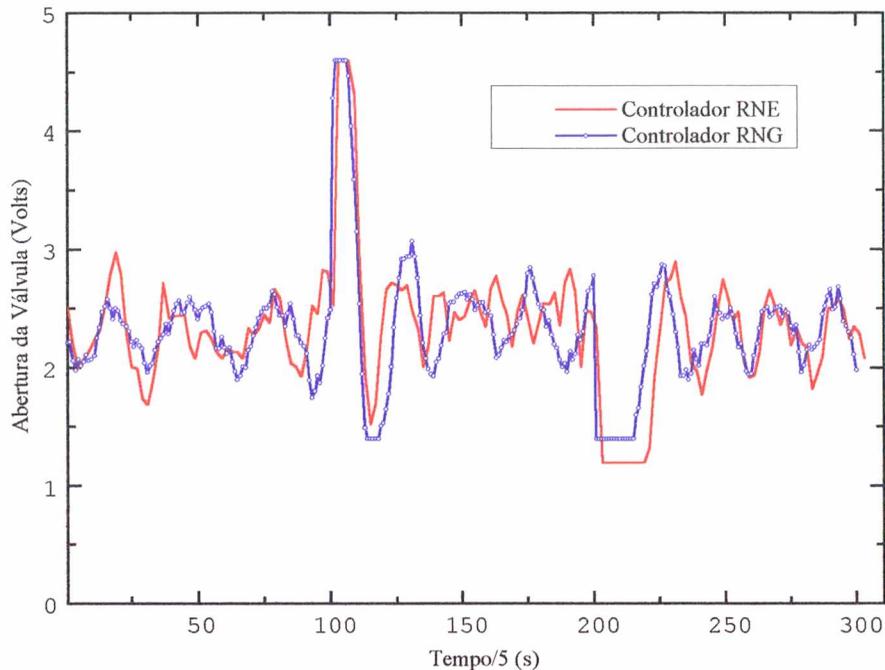
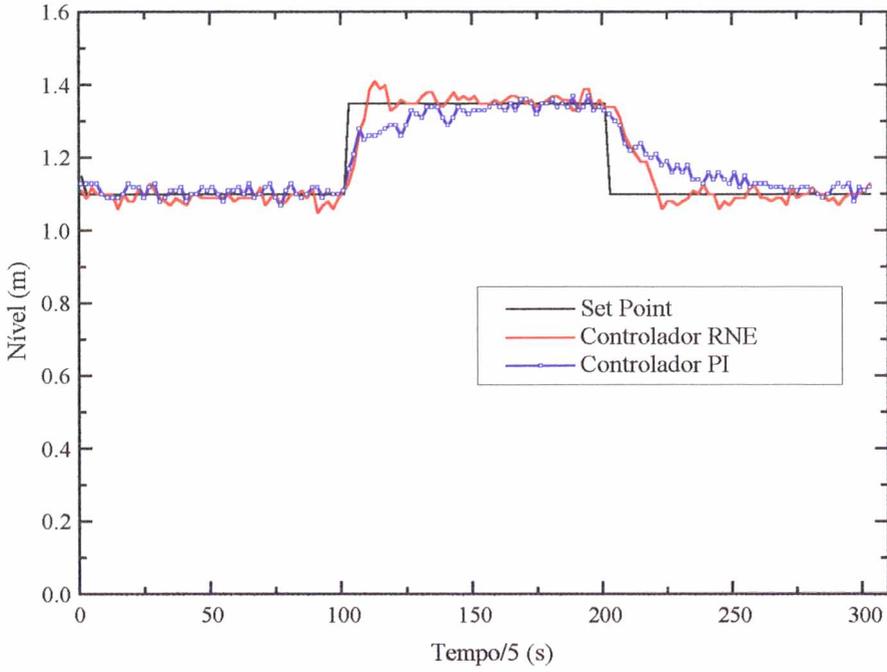
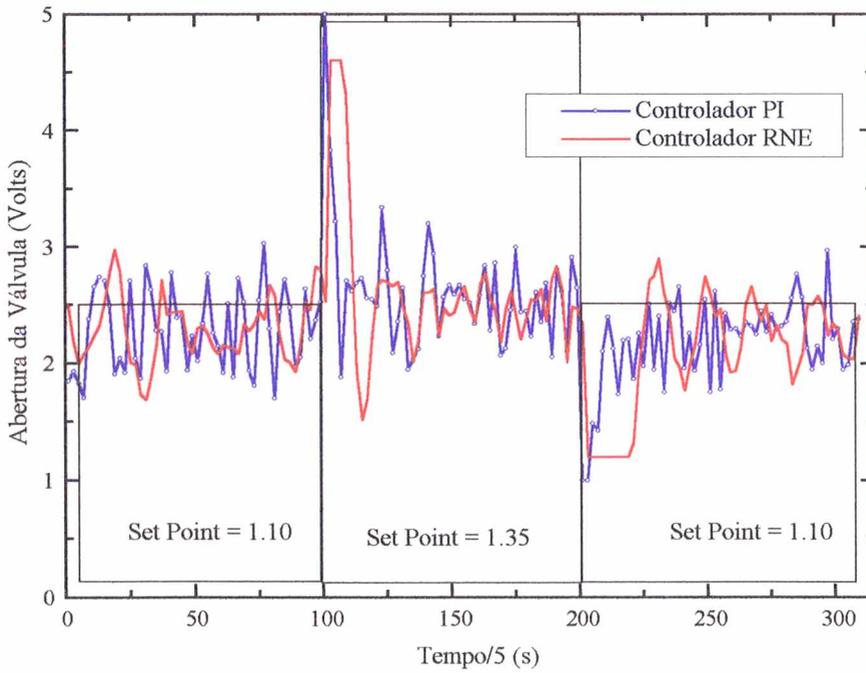


Figura V.8 (b) - Comparação da performance dos controladores RNE e RNG (50000 iterações) à um degrau no Setpoint. Ação de controle.

Outra prática comum na avaliação de uma estratégia de controle, é comparar sua performance com a de um controlador como o PI (Proporcional Integral). Na figura V.9 a estratégia RNE é comparada a um controlador PI clássico bem ajustado, onde observa-se principalmente uma velocidade superior nas transições de *setpoint*. Enquanto o Controlador RNE gastou 10 intervalos de amostragem para conduzir o sistema ao novo *setpoint*, o controlador PI gastou 16, ou seja, 60 % mais tempo. Comparativamente, o controlador RNE apresentou ações de controle mais brandas que o controlador PI (figura V.9b). A estratégia RNE apresentou uma sobre-elevação de 5 cm, ou seja, na transição para *setpoint* 1.35 m o nível subiu para 1.40 m antes de estabilizar-se. Embora pequeno, este problema pode ser amenizado, estendendo-se o horizonte de predição para mais que um passo futuro.



(a)



(b)

Figura V.9 - Comparação da performance de um controlador PI e de um controlador RNE.

(a) Variável Manipulada. (b) Variável Controlada.

Uma dificuldade constante para qualquer técnica de controle é o ajuste do controlador. Particularmente, o controlador utilizado carece de técnicas de ajuste. Para demonstrar a influência do ajuste do controlador sobre a ação de controle e sobre as oscilações no nível controlado, o sistema foi posto em estado estacionário, e o valor de λ foi fixado em 0.065, assim o valor de α foi variado de 0.4 a 0.8 (figura V.10). Isto determinou a presença de *offset* em algumas circunstâncias, ou ações de controle exageradas em outras. O valor adotado foi 0.6, representando um estado intermediário nas situações citadas. A influência de α sobre a ação de controle é forte, concentrando as ações de controle desde 12 % (0.5 volt) da faixa de abertura da válvula com $\alpha=0.6$, até 50% da faixa (2 volts) com $\alpha=0.4$

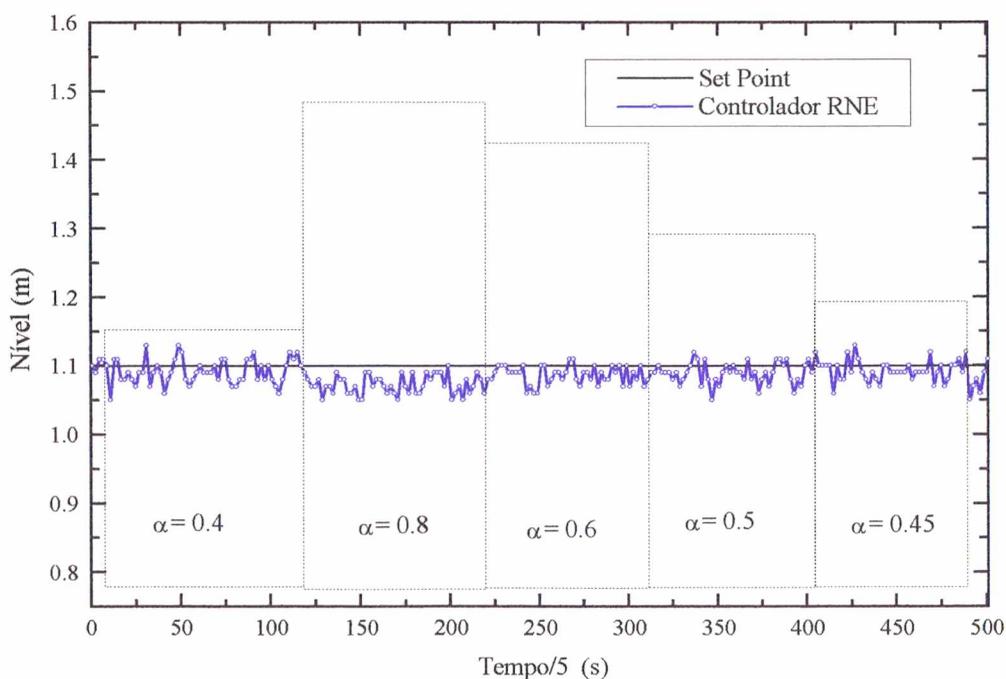


Figura V.10 (a) - Influência do ajuste de α sobre a oscilação no nível.

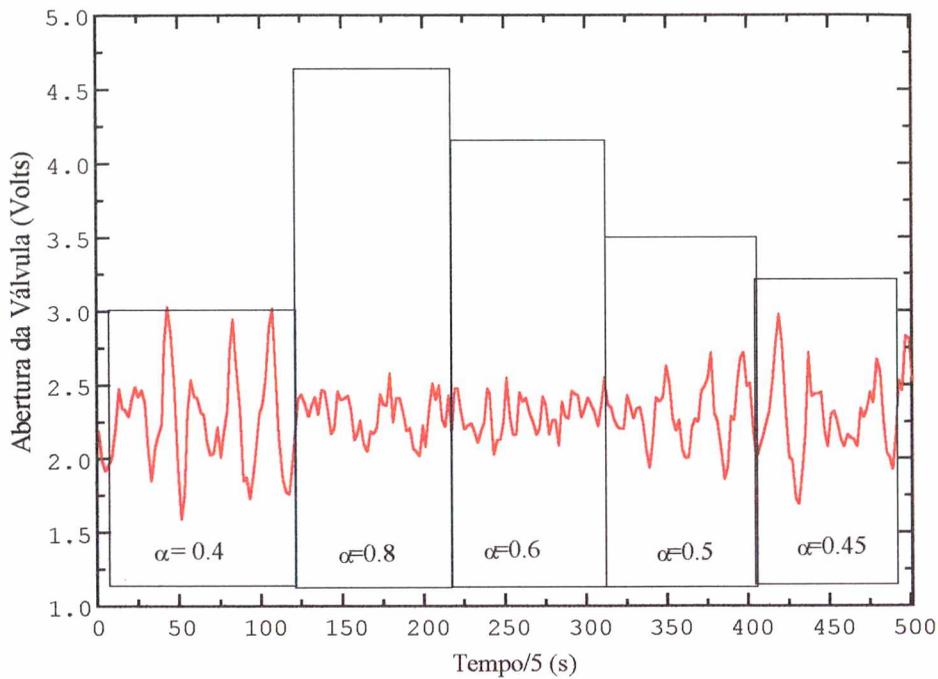


Figura V.10 (b) - Influência do ajuste de α sobre a ação de controle.

Uma questão que poderia ser levantada, é o comportamento da estratégia RNE em duas situações o limite seção cônica-seção cilíndrica (1.15 m) e o limite de referência utilizado pelo sistema especialista. Estes limites podem ser coincidentes ou não, conforme se deseje. Neste caso, o limite de referência adotado foi 1.17 m, ou seja, logo acima do limite físico do sistema. Para evitar uma mudança brusca nas características do modelo utilizado, uma pequena região de sobreposição foi resguardada durante o treinamento das duas rede neurais, assim a estabilidade do sistema foi mantida. Este teste pode ser visto na figura V.11.

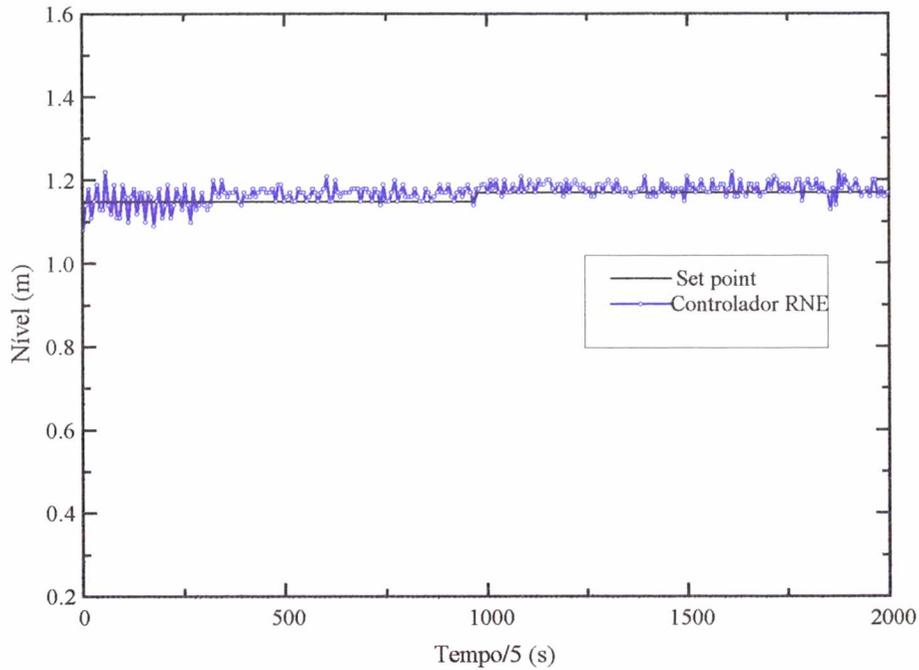


Figura V.11 - Sistema Especialista atuando em posições críticas do sistema (1,15 e 1,17 m).

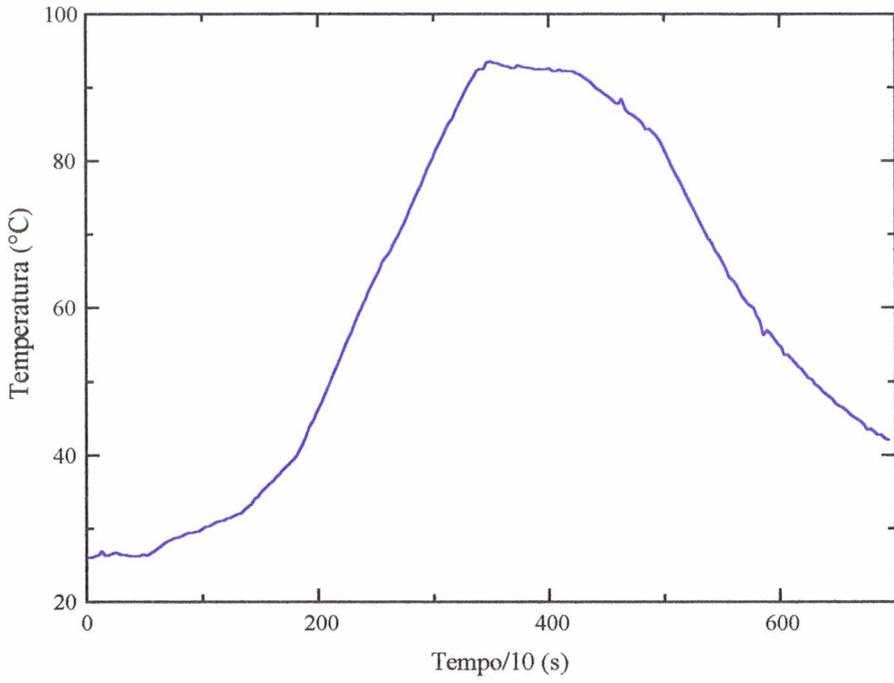
A simplicidade de operação do sistema tanque cônico-cilíndrico, permitiu o teste de várias configurações das estratégias de controle. Os testes considerados mais aptos à discussão foram grafados, contudo representando uma pequeníssima porção dos testes realizados. As duas etapas mais difíceis da implementação das estratégias de controle são o treinamento da rede neural e o ajuste do controlador. Além de independentemente demoradas, estas tarefas estão intimamente interligadas. Um modelo deficiente, ou incorreto, pode impossibilitar o ajuste deste tipo de controlador.

V.2. Resultados para o Trocador de Calor.

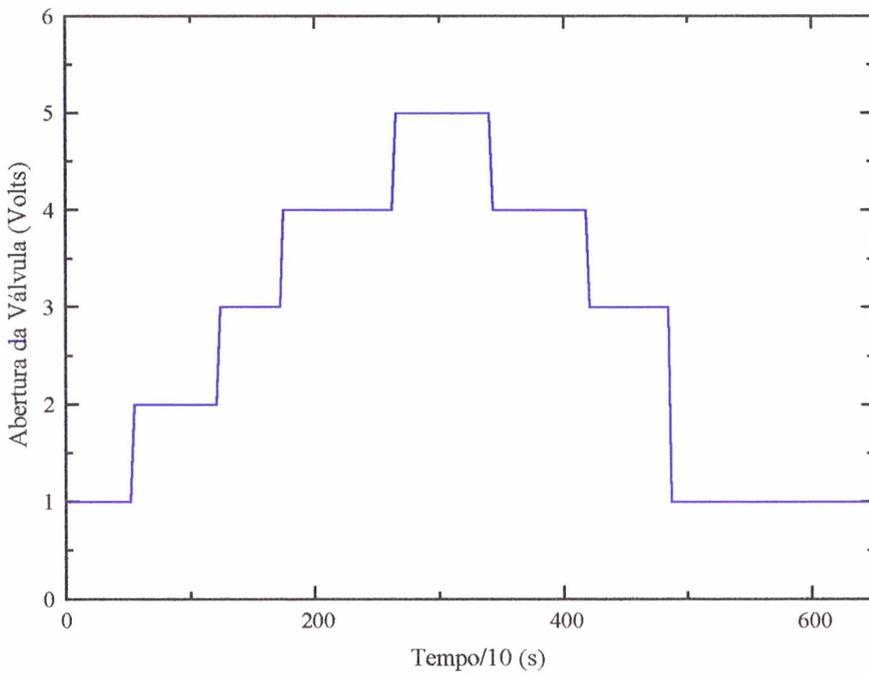
Um aspecto importante a ser considerado, quando da aplicação de uma estratégia de controle, é presença de atrasos na resposta do sistema frente às variações impostas. Estes atrasos caracterizam o denominado tempo morto e devem ser considerados com um fator determinante para a produção de uma estratégia de controle baseada em um modelo neural. A opção, neste trabalho, foi treinar a rede neural considerando-se o tempo morto, conforme demonstrado no capítulo IV. Desta forma, nenhuma modificação no algoritmo de controle foi efetuada.

A dinâmica do sistema trocador de calor é demasiadamente lenta devido aos fenômenos de transferência de calor envolvidos no processo. A presença do trocador de calor a placas e do reservatório de água quente, acrescidos à dinâmica do próprio trocador de calor (TC) conferem ao sistema um tempo morto da ordem de 90 a 100 segundos, o que equivale a 10 intervalos de amostragem. O vapor utilizado como fonte de calor foi fornecido por uma caldeira elétrica com capacidade de 100 Kg/h operando na faixa de 5-6 Kgf/cm². Problemas no sistema de controle de pressão da caldeira, permitiram que a pressão variasse entre 3 e 6 Kgf/cm², chegando em algumas circunstâncias a 2 Kgf/cm², caracterizando uma forte perturbação no sistema.

O treinamento das redes neurais componentes das duas estratégias de controle é efetuado da mesma forma que para o tanque, contudo considerando um tempo morto de 10 intervalos de amostragem. Para a estratégia RNG foram coletados dados *off line* entre 25 e 90°C com variações da ação de controle (U) entre 1 e 5 volts (figura V.12). Para a estratégia RNE dois grupos de dados, representando as dinâmicas de aquecimento (figura V.13) e resfriamento (V.14), foram coletados *off line*. Comparando-se as figuras V.13 e V.14 fica evidente a brusca diferença entre as duas dinâmicas. Como a dinâmica de resfriamento é mais lenta que a de aquecimento, são necessários mais pontos para o treinamento nesta circunstância.



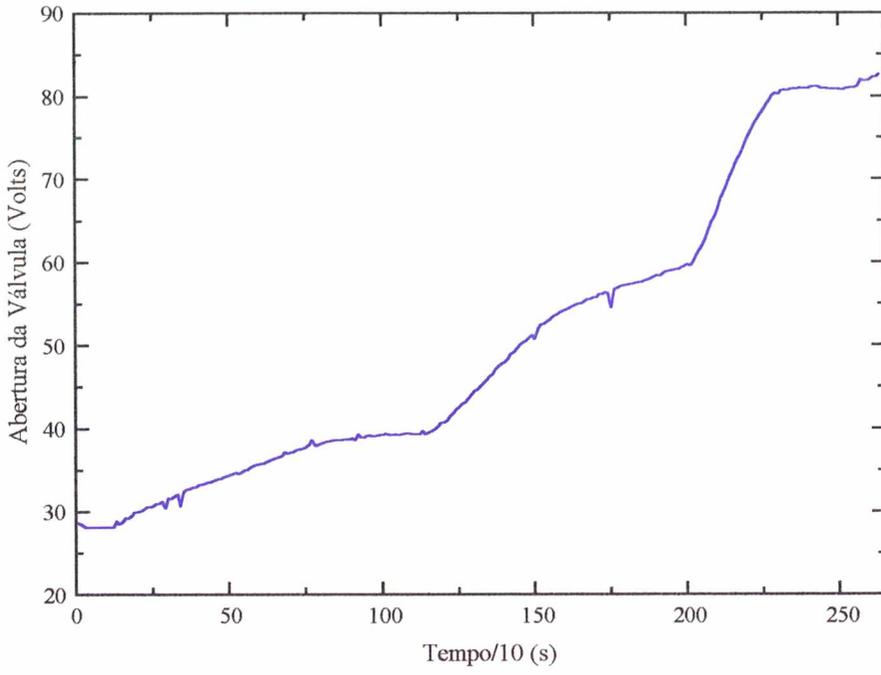
(a)



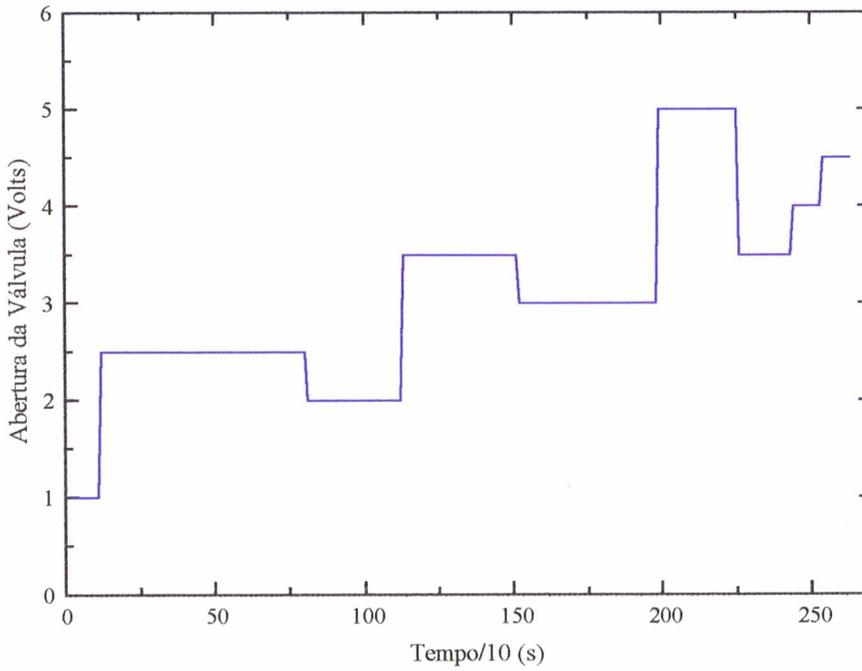
(b)

Figura V.12 - Dados de treinamento para o modelo RNG.

(a) Temperatura. (b) Abertura da válvula de vapor.



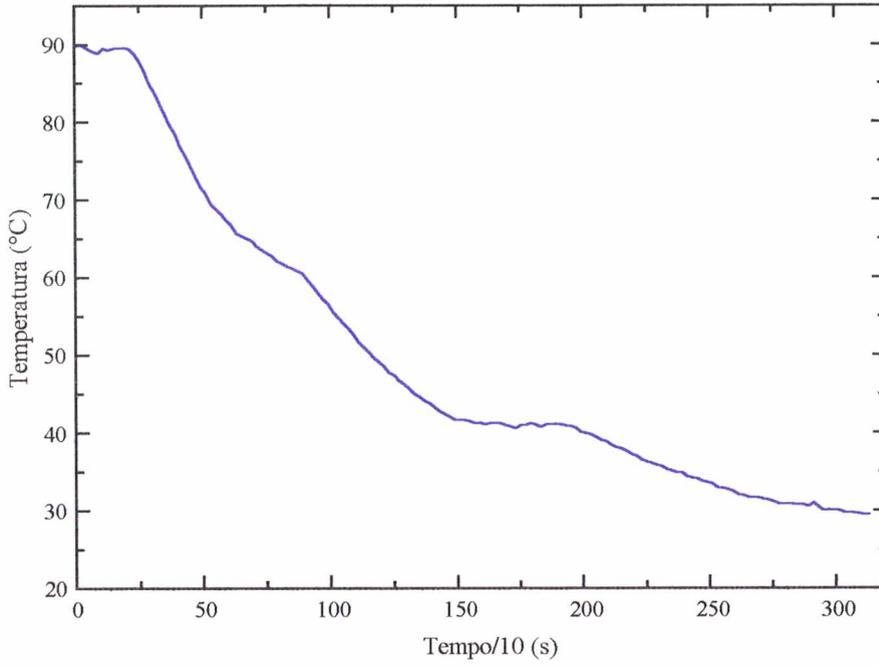
(a)



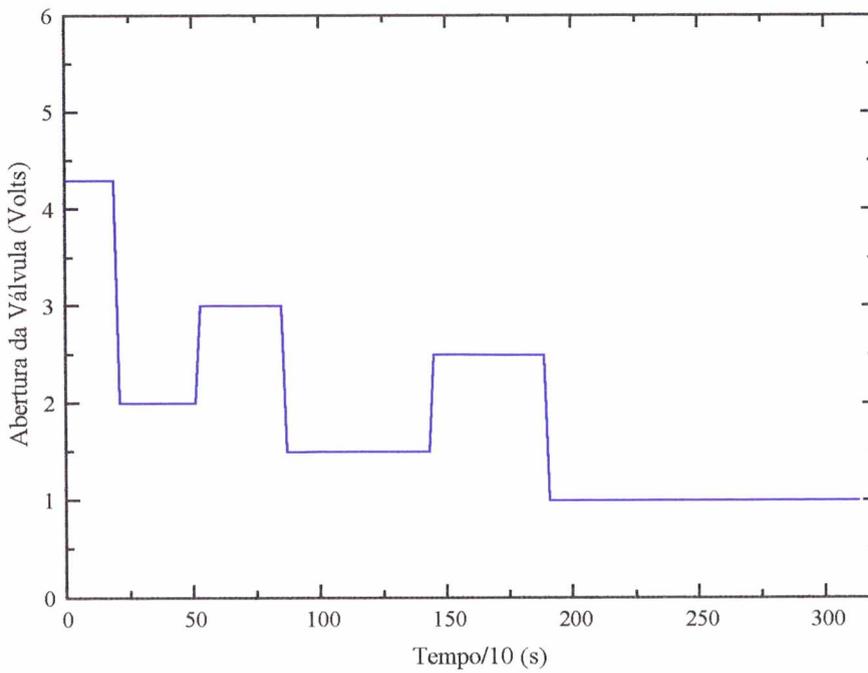
(b)

Figura V.13 - Grupo de treinamento para o modelo RNE.

(a) Aquecimento. (b) Abertura da válvula de vapor.



(a)



(b)

Figura V.14 - Grupo de treinamento para o modelo RNE.

(a) Resfriamento. (b) Abertura da válvula de vapor.

Comparativamente, os resultados obtidos para o TC são semelhantes aos obtidos para o tanque, apesar da complexidade superior do sistema TC. O controlador RNE, apresentou melhores resultados que o RNG diminuindo o *overshoot* e com menor *offset*, quando presente. Nas figuras V.15, V.16 e V.17 estão as respostas a mudanças degrau no *setpoint* para os controladores RNE, RNG, e PI respectivamente. A análise destas figuras demonstra a superioridade da estratégia RNE com menor *offset*, menor *overshoot* e maior rapidez nas mudanças de *setpoint*. Os parâmetros dos controladores foram $\alpha= 0.3$ e $\lambda= 0.35$ para os controladores RNE e RNG e $K_c=1^\circ\text{C/volt}$ e $T_i=200$ s para o controlador PI.

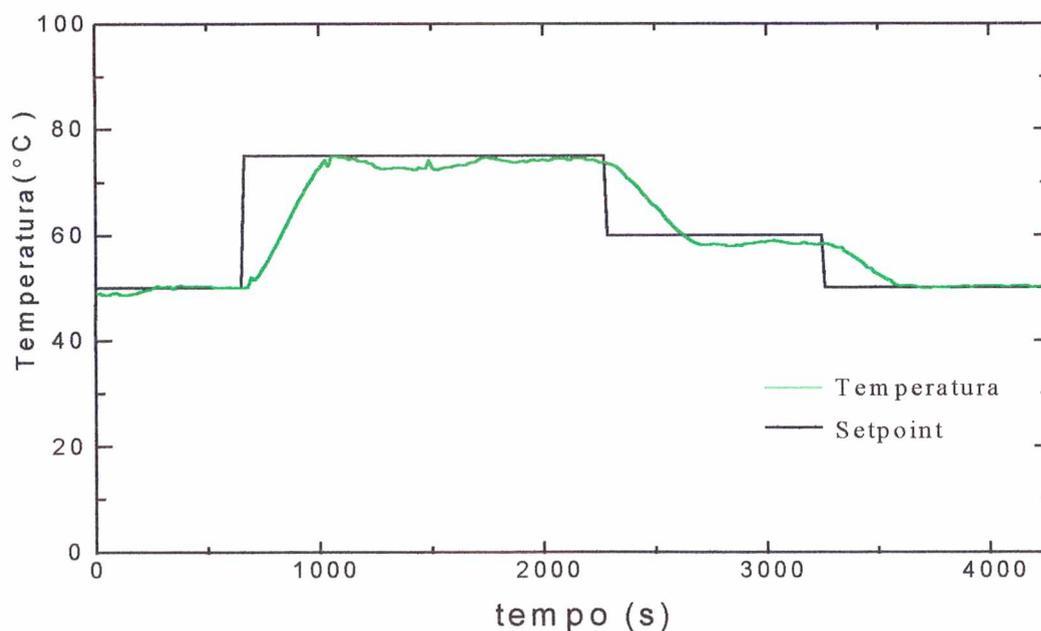


Figura V.15 (a) - Controlador RNE sujeito à mudanças no *setpoint*. Temperatura.

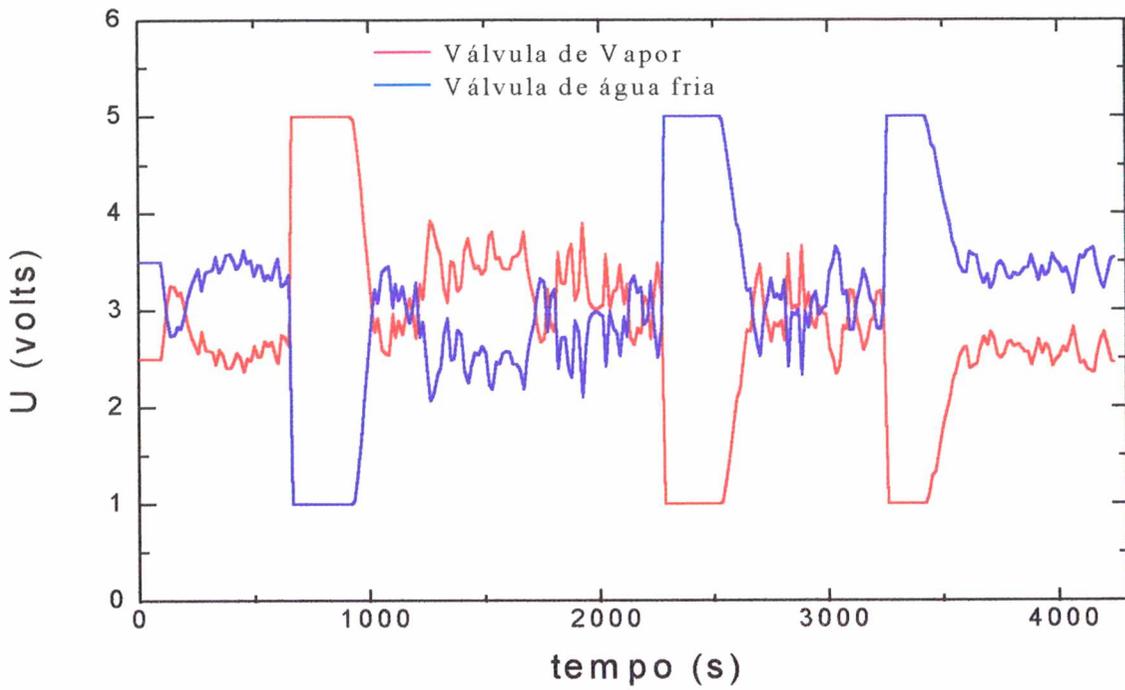


Figura V.15 (b) - Controlador RNE sujeito à mudanças no *setpoint*. Ação de controle.

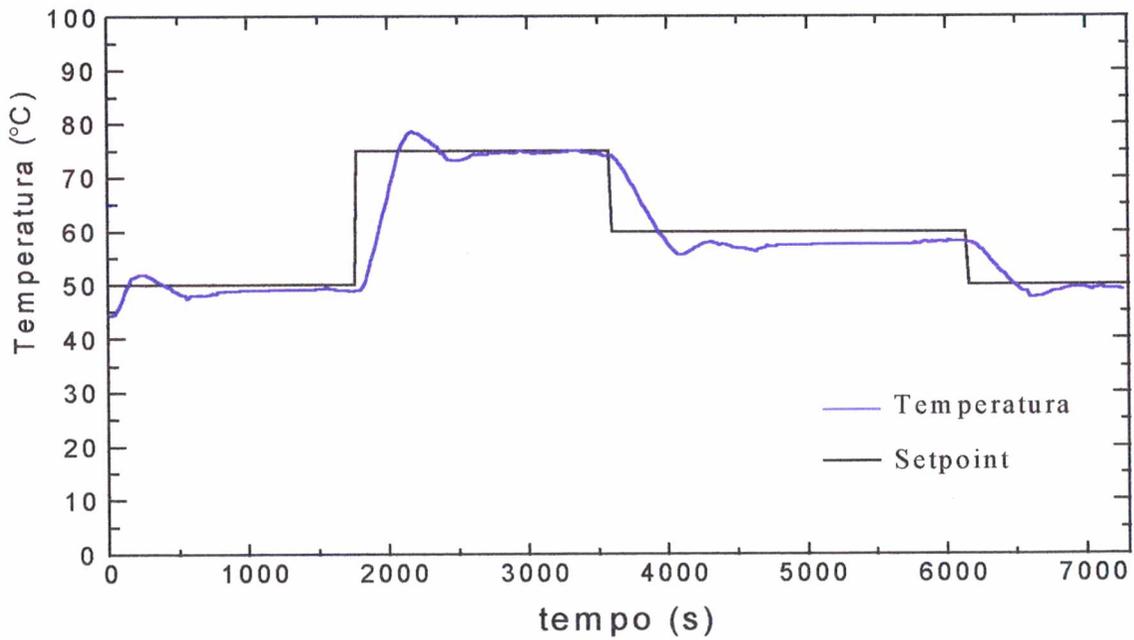


Figura V.16 (a)- Controlador RNG. Temperatura.

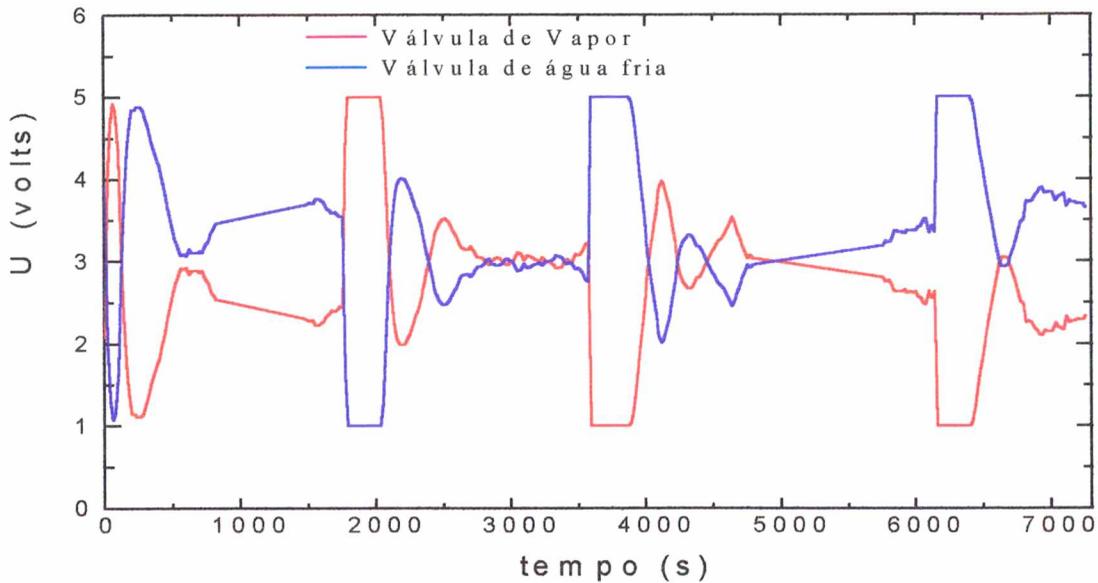
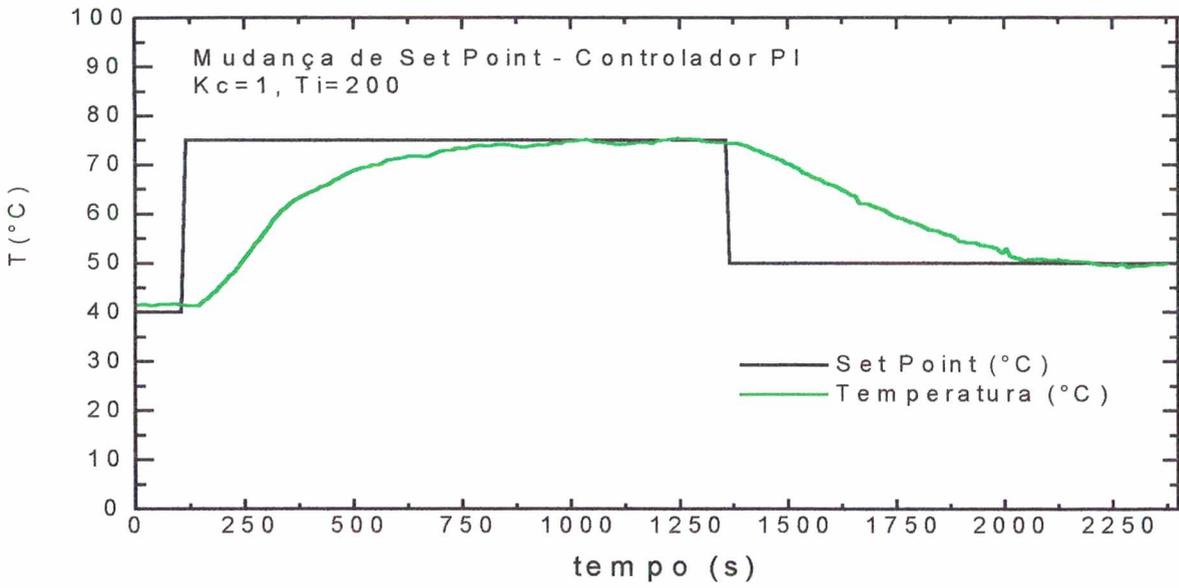


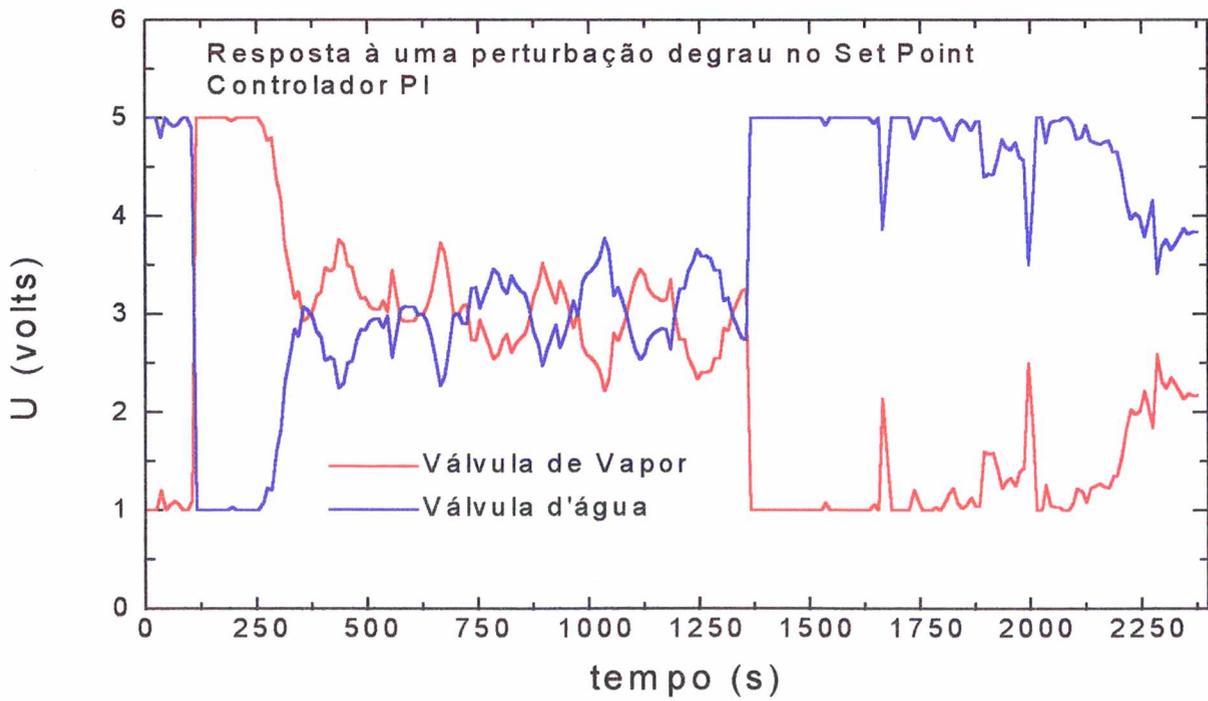
Figura V.16 (b) - Controlador RNG. Ação de Controle.

A figura V.16 (b) exibe as ações de controle para as duas válvulas manipuladas, de vapor e de água fria. Como foi adotada a estratégia de controle dividido, as duas válvulas sofrem praticamente a mesma exigência e portanto o modelo adotado deve representar bem o sistema, caso contrário os dois mecanismos sofrerão exageradamente, o que conduzirá a um desgaste prematuro dos mecanismos internos das válvulas.

Como a maioria dos controladores industriais é baseada em alguma forma do clássico PID, este trabalho estabelece uma comparação entre a estratégia RNE (proposta), a estratégia RNG (comumente adotada) e um controlador PI. Como o ajuste de controlador PID nem sempre é simples, é comum eliminar-se o modo derivativo e utilizar simplesmente um controlador PI, isto facilita muito o ajuste do controlador. Com este objetivo a figura V.7 mostra a reação de um controlador PI à uma variação degrau no *setpoint*



(a)



(b)

Figura V.17 - Controlador PI.

(a) Temperatura. (b) Ação de controle.

Com o objetivo de comparar a técnica proposta com um controlador já estabelecido, os dois controladores (RNE e PI) foram submetidos às mesmas situações sob uma perturbação do tipo carga. Os resultados obtidos estão grafados na figura V.18. Os ensaios de comportamento frente a este tipo de perturbação foram realizados mediante a retirada de 2 l de água a 75°C do TC e o acréscimo de 2 l d'água a 20°C. Dois ensaios foram realizados, nesta etapa. O primeiro, submetendo o controlador RNE à carga, e o segundo submetendo um controlador PI às mesmas condições. O controlador RNE foi mais eficiente ao reconduzir o sistema ao *setpoint* mais rapidamente e com ações de controle mais brandas que o controlador PI. O controlador PI foi ajustado pelas relações de projeto do Cohen Coon.

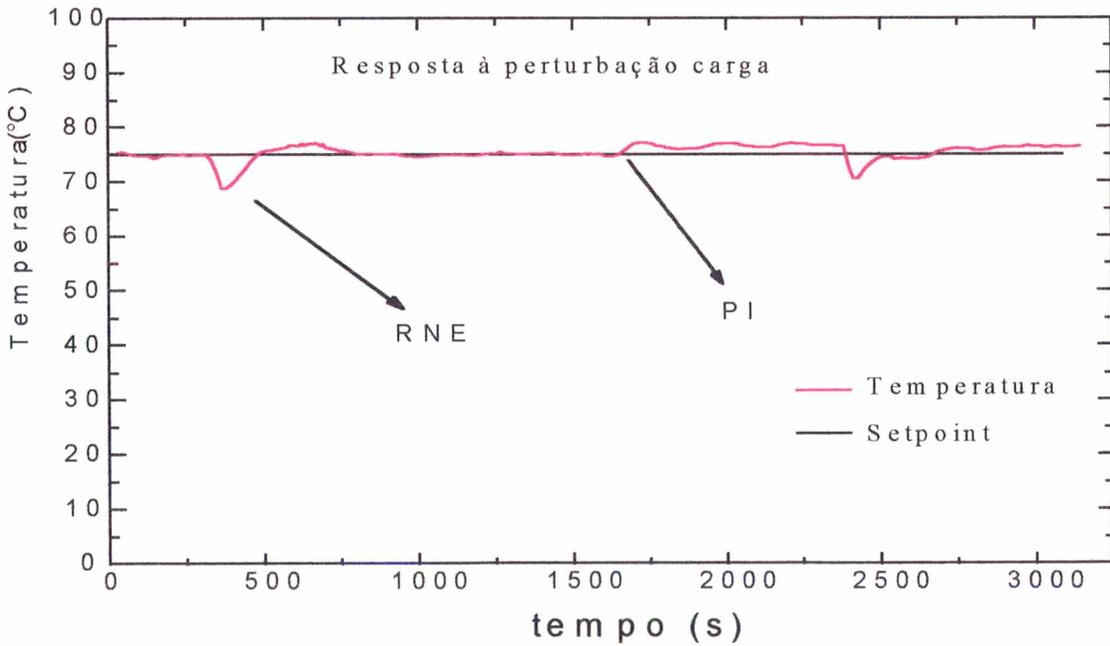


Figura V.18 (a) - Comparação da performance de um controlador PI e RNE.

Temperatura no trocador de calor.

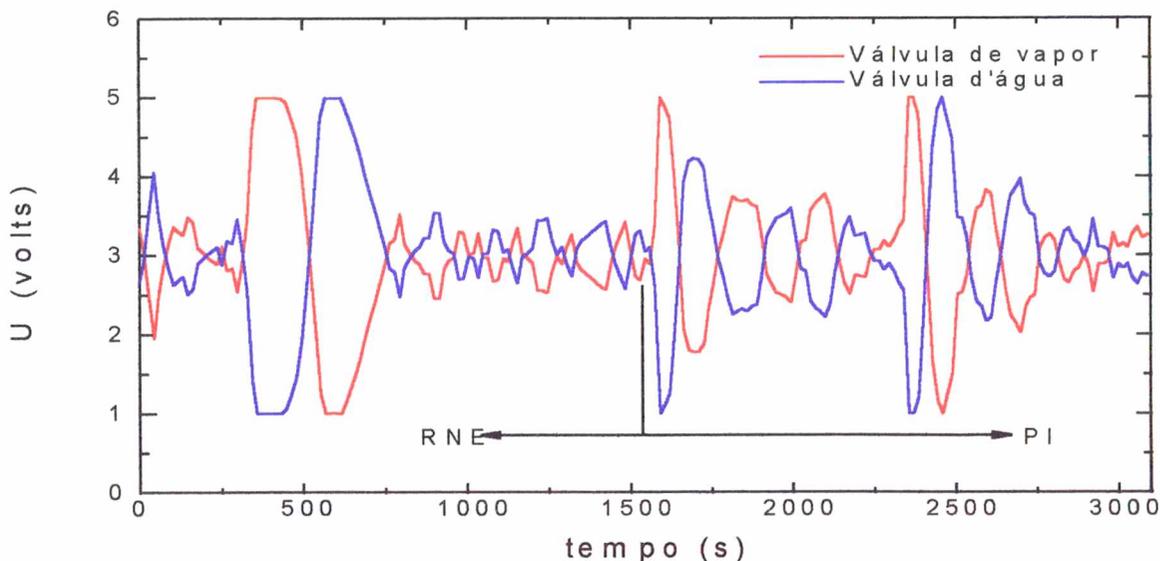


Figura V.18 (b)- Comparação da performance de um controlador PI e RNE. Ação de controle.

Neste capítulo, foram apresentados os resultados obtidos na validação experimental e nos treinamentos dos modelos propostos, RNE e RNG. A estratégia RNE mostrou-se mais eficaz que a RNG tanto em mudanças no *setpoint* como em perturbações carga. Uma grande vantagem do uso de dois modelos na implementação da estratégia RNE é uma redução significativa da capacidade computacional necessária para o treinamento. A velocidade de treinamento é proporcional ao número de padrões fornecidos, assim, no caso do trocador de calor, foi possível a redução do número de padrões, aproximadamente pela metade, o que facilitou muito o treinamento. Na implementação do controlador, o uso de dois modelos não representa maior complexidade. Apenas, devem ser resguardadas todas as normalizações e pesos dos grupos treinados.

CAPÍTULO VI

CONCLUSÕES E SUGESTÕES

Este capítulo finaliza o presente trabalho comentando as qualidades e dificuldades da implementação das estratégias de controle propostas. Os últimos comentários deste capítulo apresentam sugestões para a continuidade do trabalho sem apelar para mudanças muito profundas, pois entende-se que é importante manter uma forma simples para o algoritmos de treinamento e controle.

VI.1. Conclusões.

Com o objetivo de testar a qualidade do controle de processos utilizando modelos baseados em redes neurais artificiais duas estratégias de controle foram propostas :

- 1 - Um controlador baseado em um modelo neural generalista, denominado RNG;
- 2 - Um controlador baseado em um modelo neural associado a um sistema especialista denominado RNE.

As duas estratégias propostas foram aplicadas a dois sistemas, um tanque e um trocador de calor. Assim, através da proposta 1 a capacidade de modelagem de uma rede neural *feedforward* foi posta a prova. Através da proposta 2 um refino do modelo, mantendo uma estrutura simples, foi pretendido.

Em ambas as formas, RNG e RNE, a modelagem através de redes neurais provou ser eficiente e razoavelmente simples. A definição da taxa de aprendizagem, a arquitetura da rede e

o ajuste do controlador foram os pontos mais críticos da implementação das estratégias neurais de controle.

Na implementação, as estratégias provaram ser robustas o suficiente para rejeitar os ruídos da planta e conduzir, de maneira segura, o sistema através de transições no *setpoint*. Submetido à perturbações do tipo carga, os controladores foram capazes de reconduzir o sistema de forma estável ao *setpoint*. Comparativamente, a estratégia RNE apresentou ser mais eficiente principalmente na produção de menores *overshoot* e *offset*. Portanto o esforço do treinamento de duas redes neurais e controle com dois modelos justifica-se.

A estratégia RNE apresenta um bom desempenho, mas , pode ser melhorada com o uso de melhores técnicas no treinamento. Tentativas foram realizadas em usar o algoritmo de otimização da máxima verossimilhança, mas isso resultou num código pesado demais para ser rodado num PC. Além do que o intuito era desenvolver um aplicativo gráfico para facilitar o treinamento. Para melhorar o treinamento a taxa de aprendizagem, η , foi variada manualmente durante o treinamento, aumentando a velocidade de convergência e reduzindo a probabilidade de divergência.

Tanto nos testes no sistema tanque cônico-cilíndrico como no trocador de calor, a estratégia RNE mostrou-se estável e segura. Há de se considerar que os modelos só foram considerados, suficientemente treinados quando a somatória dos erros médios quadráticos fosse inferior a 0.1. Este critério permite que se estabeleça comparações mais seguras da qualidade dos modelos obtidos.

VI.2. Sugestões.

Como pôde ser observado é frequente a ocorrência de *overshoot* nos testes realizados. Isto foi observado, também, por outros autores. Como sugestão para a redução deste problema poder-se-ia estender o horizonte de predição, o que implicaria na mudança na formação dos padrões de treinamento.

Um problema difícil de ser atacado é o problema de ajuste do controlador. Como esse ajuste é baseado em dois parâmetros α (trajetória de referência) e λ (penalização da ação de controle) seria importante o estabelecimento de regras para a adequação destes parâmetros.

O algoritmo de otimização utilizado para a minimização do erro no treinamento, o *steepest descent*, é reconhecidamente problemático. O algoritmo sofre de problemas de convergência por mínimos locais, e sua substituição seria um bom avanço, contudo não extremamente necessário para problemas de pequeno e médio porte, ou seja, problemas com poucos padrões e uma ou duas saídas na rede neural.

O controlador, apesar de apresentar uma boa performance, poderia ser substituído por uma rede neural treinada para atuar como controlador. Uma outra possibilidade é tentar adaptar os parâmetros do controlador neural aos do PID, conforme os trabalhos de Bolzan (1991) e Machado (1996).

APÊNDICES

VII.1 APÊNDICE I

VII.1. Especificações dos equipamentos utilizados.

VII.1.1. Válvulas de controle.

VII.1.1.1 Para Água Fria.

Marca: Hiter - série 201, atuador DN0021-AC.

Características :

- Igual percentagem ar abre, falha fecha;
- Sinal de entrada 3 a 15 psi;
- Parte interna em aço inox 316, corpo em aço carbono;
- Conexões tipo rosca $\frac{3}{4}$ in.

VII.1.1.2 Para Vapor.

Marca: Badge Meter Inc..

Características :

- Igual percentagem ar abre, falha fecha;
- Sinal de entrada 3 a 15 psi ;
- Parte interna em aço inox 316, corpo em aço carbono;
- Conexões tipo rosca $\frac{1}{2}$ in.

VII.1.2 Conversores Eletropneumáticos

VII.1.2.1 Para a válvula de água Fria.

Marca: Hélix tipo P11-1111-2;

Sinal de entrada: 4 a 20 mA;

Sinal de saída: 3 a 15 psi;

Alimentação: 20 psi.

VII.1.2.2 Para a válvula de Vapor.

Marca: Conoflow (ITT Fluid Technology Corporation);

Sinal de entrada: 4 a 20 mA;

Sinal de saída: 3 a 15 psi;

Alimentação: 20 psi.

VII.1.3 Filtros Reguladores de Pressão.

VII.1.3.1 Para a válvula de água fria.

Marca: Hélix - tipo F11-11;

Pressão máxima de Alimentação: 250 psi;

Saída: 20 psi.

VII.1.3.2 Para a válvula de vapor.

Marca: Conoflow (ITT Fluid technology Corporation)

Pressão máxima de Alimentação: 300 psi;

Saída: 20 psi.

VII.1.4 Termopar.

Marca: Ecil tipo J (liga Cobre-Constantan);

Sinal de saída: 0 a 10 mV.

VII.1.5 Amplificador.

Marca: Microquímica - modelo 308-A;

Características:

- Ganho programável: 0 a 9.000 vezes;
- Entrada: 0 10 mV;
- Alimentação: 220 V.

VII.1.6 Sensor de pressão.

Marca: Contrisul;

Características:

- Entrada: 0 a 400 mBar;
- Saída: 4 a 20 mA;
- Alimentação 20 Vcc;
- Corpo em aço carbono, parte interna em aço inox 316;

VII.1.7 Bomba Centrífuga.

Marca: Schneider, modelo 02.01;

Potência ¼ CV, 3400 RPM;

Alimentação 220 V.

VII.1.8. Agitador.

Marca: Fisatom - modelo 713-T;

Características:

- Frequência de agitação: 90 a 6300 rpm (baixo torque, alta rotação) e 50 a 340 rpm (alto torque, baixa rotação).
- Potência: 80 W;
- Alimentação 220 V.

VII.1.9. Microcomputador.

Microcomputador tipo PC 386 DX 40 MHz.

Configuração:

- Co-processador aritmético 8087;
- Unidade de disco rígido de 230 MBytes;
- Unidade de disco flexível de 3 ¼ in, 1.44 MBytes;
- Monitor VGA color;
- Sistema operacional MS DOS 6.0;
- Windows™ 3.11.

VII.1.10. Placa Analógica-Digital/Digital- Analógica.

Interface de aquisição de dados para microcomputador tipo PC/XT/AT, marca DataTranslation, modelo DT2812, barramento ISA/EISA, com capacidade de transferência via DMA.

Características:

- 16 canais de entrada analógicos em modo comum ou 8 canais em modo diferencial;
- Resolução 12 bits;
- Taxa de amostragem de 60 KHz;

- Faixa de entrada de 0 a 1.25 V, 2,5 V, 5 V, 10 V, +/- 1,25 V, +/- 2.5 V, +/- 5 V, +/-10 V,
- 2 Canais de saída analógica;
- Conversão na faixa 10 microsegundos a 3 min.;
- 2 contadores/temporizadores de 16 bits dedicados a contagem de eventos e medidas de frequências;
- 1 contador/temporizador de 16 bits programável;
- 16 canais de saída digital;
- 16 canais de entrada digital;
- 3 canais compartilhados com contadores/temporizadores.

VII.1.11. Conversor Tensão-Corrente.

Conversor isolador de sinal marca ICI Instrumentação e Controle Industriais Ltda.;

Características:

- Configuração para termo-resistências, termopares e sinais padronizados;
- Sinal de saída: 0 a 20mA, 4 a 20 mA ou 0 a 10 Volts (optoisolado), configuráveis;
- 2 níveis de alarme configuráveis;
- Fonte auxiliar para transmissor: 5, 10, 12, 24 Vcc;
- Alimentação: 110 ou 220 Vac.

VII.2 APÊNDICE II

VII.2. Calibração para o termopar e sensor de pressão

O objetivo da calibração dos elementos sensores é obter uma relação entre a variável a ser medida, neste caso altura e temperatura, e o sinal enviado pelo sensor. A calibração obtida, realmente, não se refere ao sensor especificamente, mas sim ao conjunto de elementos de medição. Assim, para o termopar a curva reflete o conjunto termopar-amplificador e para o sensor de pressão o conjunto sensor de pressão-conversor I/V. Desta forma, uma relação Voltagem x Altura e Voltagem x Temperatura é obtida.

VII.2.1 Procedimento para determinação da curva de calibração do Sensor de Pressão.

O *software* de controle pode ser configurado para ler as informações da planta em volts, assim inicialmente o *software* dever ser configurado e em seguida a válvula Manual, VM1 (figura IV.3), deve ser fechada e a válvula de controle aberta até que o tanque esteja completamente cheio. O tanque estando cheio, a válvula de controle deve ser fechada e a válvula VM1 aberta e fechada em diferentes posições do tanque, anotando-se a altura do tanque e a voltagem em cada instante. Assim, o grupo de dados de Voltagem e Altura é obtido. Finalmente, plota-se os valores obtidos em um gráfico Volts x Metros e procede-se uma aproximação linear. Os coeficientes da reta devem ser inseridos na caixa de diálogo **Configurações** na página **Calibração** do *software* de controle. A calibração obtida para o conjunto sensor de pressão-conversor I/V está abaixo (o coeficiente de correlação da reta é 0.9995):

$$Altura(m) = -1.8336 + Volts * 1.0132$$

VII.2.2 Procedimento para determinação da curva de calibração do termopar.

A curva de calibração do termopar compreende o conjunto termopar amplificador e o procedimento para sua determinação é idêntico ao para a calibração do sensor de pressão. O reator deve ser cheio com água e em seguida o sistema deve ser posto em marcha com a válvula de vapor aberta. Os valores de Temperatura e voltagem devem ser anotados. A calibração obtida para o conjunto termopar-amplificador está abaixo (o coeficiente de correlação da reta é 0.99982):

$$Temperatura(^{\circ}C) = 6.0814 + Volts * 18.3739$$

VII.3 APÊNDICE III

VII.3. Exemplo da magnitude dos pesos nos modelos neurais utilizados.

Tabela VII.3.1 - Pesos entre as camadas 2 e 3 para o grupo 1 do modelo RNE.

| Neurônio | | Peso |
|----------------------|-----------------|--------------------|
| Camada intermediária | Camada de Saída | $W_{L,j,k}$ |
| 1 | 1 | -4.994305480613202 |
| 2 | 1 | -5.321722220204043 |
| 3 | 1 | -2.935329885647185 |
| 4 | 1 | 3.188955394773244 |
| 5 | 1 | 1.689213192808102 |
| 6 | 1 | 2.020206957993150 |
| 7 | 1 | 2.711422808455176 |
| 8 | 1 | 1.195738027096782 |
| 9 | 1 | -3.659026322018511 |

Tabela VII.3.2 - Pesos entre as camadas 1 e 2 para o grupo 1 do modelo RNE.

| Neurônio | | Peso |
|-------------------|----------------------|--------------------|
| Camada de entrada | Camada intermediária | $W_{L,ij}$ |
| 1 | 1 | -0.97984943781212 |
| 1 | 2 | 0.0865193369211811 |
| 1 | 3 | 0.0526863485021158 |
| 1 | 4 | 0.6048782294688228 |
| 1 | 5 | 0.4506851673451858 |
| 1 | 6 | 0.2592517268626405 |
| 1 | 7 | 1.282690779312479 |
| 1 | 8 | 0.0874255693632072 |
| 1 | 9 | -0.421991627965717 |
| 2 | 1 | 0.3134688642003279 |
| 2 | 2 | -0.942481678880933 |
| 2 | 3 | -0.885113258112512 |
| 2 | 4 | -0.050303982966167 |
| 2 | 5 | 0.1553930720969748 |
| 2 | 6 | 0.4742642880029923 |
| 2 | 7 | -0.605821249615093 |
| 2 | 8 | 0.7205404754046669 |
| 2 | 9 | -0.212556771784416 |
| 3 | 1 | -0.794938536419015 |
| 3 | 2 | 0.7658912321703294 |
| 3 | 3 | 0.6438512724600396 |
| 3 | 4 | 0.6104710301743287 |
| 3 | 5 | 0.9762912341303659 |
| 3 | 6 | -0.045169502992 |
| 3 | 7 | 1.077771860367128 |
| 3 | 8 | -0.420382989145288 |
| 3 | 9 | -0.614265373320204 |

BIBLIOGRAFIA

- ANTSACLIS, P. J. , Neural Networks in Control Systems, **IEE Control Systems Magazine**, v.10, p. 3-87, 1990.
- BHAT, N. V., MINDERMAN, P. A., Jr, McAVOY, T., WANG, N. S., Modeling Chemical Process Systems via Neural Computation, **IEEE Control Systems Magazine**, 1990.
- BOLZAN, A., Controle Adaptativo de Reatores Semi-batelada, **Tese de Doutorado**, COPPE/UFRJ, 1991.
- CARVALHO FILHO, E. C. B., LUDEMIR, T. B., DE OLIVEIRA FR, W. R., Apostila do Minicurso de Modelagem, Aplicações e Implementações de Redes Neurais, **Congresso Brasileiro de Redes neurais**, I, 1994, Itajubá, MG.
- CHEM, J. & BRUNS, D., Recursive Nonlinear System Identification Using an Adaptative WaveARX Network, **Ind. Eng. Chem. Res.**, v.35, n.8, p.2782-2789, 1996.
- CHEM, F.- C. & KHALIL, H. K., Adaptative Control of a Class of Nonlinear Discrete-Time Systems Using Neural Networks, **IEEE Transactions on Automatic Control**, v.40, n.5, p.791-801, 1995.
- CHEN, J. & BRUNS, D., WaveARX Neural Network Development for System Identification Using a Systematic Design Synthesis, **Ind. Eng. Chem Res.**, v.34, n.112, p.4420-4435, 1995.
- CHEN, F.-C., Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptative Control, **IEEE Control Systems Magazine**, 1990.
- CHU, S. R., SHOURESHI, R., TENORIO, M., Neural Networks for System Identification, **IEEE Control Systems Magazine**, 1990.

- DE SOUZA JR, M. B., **Redes Neuroniais Multicamada Aplicadas a Modelagem e Controle de Processos**, Tese de Doutorado, COPPE/PEQ, Universidade Federal do Rio de Janeiro, Rio de Janeiro-RJ, 1993.
- DONG, D., McAVOY, T. J., ZAFIRIOU, E., Batch-to-Batch Optimization Using Neural Networks Models, **Ind. Eng. Chem. Res.**, v.35, n.7, p.2269-2276, 1996.
- FAN, J. Y., NIKOLAOU, M., WHITE, R. E., An Approach to Fault Diagnosis to Chemical Process via Neural Networks, **AIChE Journal**, v.39, n.1, p.82-87, 1994.
- GUPTA, G., NARASHIMAN,S., Process Design and control: Application of Neural Networks for Gross Error Detection, **Ind. Eng. Chem. Res.**, v. 32, n.8, p. 1651-1657, 1993.
- HUNT, K. J., SBARBARO, D. ZBIKOWSKI, R., GAWTHROP, P. J., Neural Networks for Control Systems-A Survey, **Automatica**, v. 28 n.6, p.1083-1112, 1992.
- JAGANNATHAN, S., LEWIS, F. L., PASTRAVANU, O., Discrete-Time Model Reference Adaptive Control of Nonlinear Dynamical Systems Using Neural Networks, **International Journal of Control**, v.64, n.2, p.217-239, 1996.
- KAN, P. & LEE, C.-P., A Neural Network Model for Prediction of Phase Equilibria in Aqueous Two-Phase Extraction, **Ind. Eng. Chem. Res.**, v.35, n.6, p.2015-2023, 1996.
- KIM, J. H., JEUNG, E. T., PARK, H. B., Robust Control for Parameter Uncertain Delay Systems in State and Control Input, **Automatica**, v.32, n.9, p.1337-1339, 1996.
- KOVÁCS, Z. L. , **Redes Neurais Artificiais, Fundamentos e Aplicações**, Edição Acadêmica, Editora , São Paulo, 1996.

LEE, M., PARK, S., A New Scheme Combining Neural Feedforward Control with Model-Predictive Control, **AICHE Journal**, v. 38, n.2, 1992.

LEWIS, F. L., LIU, Kai, Towards a Paradigm for Fuzzy Logic Control, **Automatica**, v.32, n.2, p.167-181, 1996.

MACHADO, R., Otimização e Controle de Reatores de Polimerização em Batelada, **Dissertação de Mestrado**, Depto. Eng. Química-UFSC, Florianópolis-SC, 1996.

MAREN, A. J., HARSTON, C. T., PAP, R. M., Handbook of Neural Computing Applications, San Diego, Academic Press, 1990.

MEGAM, L., Neural Network Based Adaptive Control via Temporal Pattern Recognition, **The Canadian Journal of Chemical Engineering**, v. 70, 1992.

McAVOY, T. J., QUIN, S. J., Nonlinear PLS Modeling Using Neural Networks, **Computers Chem. Eng.**, v. 16, n.4, p. 379-391, 1992.

MORRIS, A. J. & WILLIS M. J., Artificial Neural Networks: Studies in Process Modelling and Control, **Trans. IChemE.**, v. 72 part A, January 1994.

NAPOLITANO, M. R., CASDORPH, V., NEPPACH, C., NAYLOR, S., INNOCENTI, M., SILVESTRI, G., Inline Learning Neural Architectures and Cross-correlation Analysis for Actuator Failure Detection and Identification, **International Journal of Control**, v.63, n.3, p.433-455, 1996.

NGUYEN, D. H., WIDROW, B., Neural Networks for Self-Learning Control Systems, **IEEE Control Systems Magazine**, v.10, n.3, p.18-23, 1990.

PEEL, C., WILLIS, M. J., THAM, M. T., A Fast Procedure for the Training of Neural Networks, **J. Proc. Control**, n.4, v. 2, 1992

- POLLARD, J. F., BROUSSARD, M. R., GARRISON, D. B., SAN, K. Y., Process Identification using Neural Networks, **Computer Chem. Engng**, v.16, n.4, p.253-270, 1992.
- RITA, D. J., **Controle de Processos Usando Redes Neurais Artificiais: Uma Aplicação Experimental**, Dissertação de Mestrado, Depto. Eng. Quim.-UFSC, Florianópolis-SC, 1995.
- ROSSITER, J. A., GOSSNER, J. R., KOUVARITAKIS, B., Infinite Horizon Stable Predictive Control, **IEEE Transactions Automatic Control**, v.41, n.10, p.1522-1527, 1996.
- SARI, T. S., **Protótipo de um sistema de reconhecimento de padrões Conexionista Híbrido**, Dissertação de Mestrado, Depto. de Eng. de Produção-UFSC, Florianópolis-SC, 1994.
- SCHILD, H. (Autor), SILVEIRA, C. G. (Tradutor), **Inteligência Artificial Utilizando Linguagem C**, 1ª ed., McGraw-Hill Ltda, São Paulo, 1989.
- SHIH, R.-F. & LEE, L.-S., Use of Fuzzy Cause-Effect Digraph for Fault Diagnosis for Process Plants. 1. Fuzzy Cause-Effect Digraph, **Ind. Eng. Chem. Res.**, v.34, n.5, p.1688-1702, 1995.
- SHIH, R.-F. & LEE, L.-S., Use of Fuzzy Cause-Effect Digraph for Fault Diagnosis for Process Plants. 2. Diagnostic Algorithm and Applications, **Ind. Eng. Chem. Res.**, v.34, n.5, p.1703-1717, 1995.
- SEEBORG, D. E., EDGAR, T. F., MELLICHAMP, D. A., **Process Dynamics and Control**, John Wiley & Sons, Canada, 1989.
- SOUCEK, B., **Neural and Intelligent Systems Integration**, 1ª ed., Wiley Interscience, Canada, 1989.
- SOUCEK, B., SOUCEK, Marina, **Neural and Massively Parallel Computers**, 1ª ed., Wiley Interscience, Canada, 1988.

- SOUCEK, B., **Neural and Concurrent Real-Time Systems**, 1^a ed., Wiley Interscience, Canada, 1991.
- SWEENEY, M. W., Instrumentation and Automation, **Water Environment Research**, v.67, n.4, p.495-499, 1995.
- TSEN, A. Y, JANG, S. S., WONG, D. S. H., JOSEPH, B., Predictive Control of Quality in Batch Polimerization Using Hybrid ANN Models, **AIChE Journal**, v. 42, n.2, February 1996.
- VAN CAN, H. J. L. *et alli*, Design and Real Time Testing of a Neural Model Predictive Controller for Nonlinear System, **Chemical Engineering Science**, n.15, p. 2419-2430, 1995.
- WANG, J. & WU, G., A Multilayer Recurrent Neural Network for on line Synthesis of Minimum-Norm Linear FeedBack Control Systems via Pole Assignment, **Automatica**, v.32, n.3, p.435-442, 1996.
- ZHOU, X.-G., LIU, L.-H., YUAN, W.-K., HUDSON, J. L., A Predictive Neural Network Model Based on the Karhunen-Loève Expansion for Wall-Cooled Fixed-Bed Reactors, **The Canadiam Journal of Chemical Engineering**, v.74, p.638-645, 1996.