

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

CONTROLE PREDITIVO DE REATORES SEMI-BATELADA

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Química do Centro Tecnológico da Universidade Federal de Santa Catarina, como requisito parcial à obtenção do título de Mestre em Engenharia Química.
Orientador: Prof. Dr. Ariovaldo Bolzan

ADRIANO CANCELIER

Florianópolis - SC

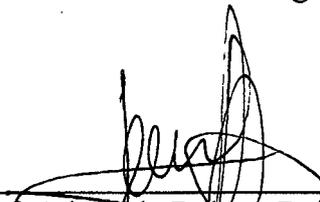
1998

CONTROLE PREDITIVO DE REATORES SEMI-BATELADA

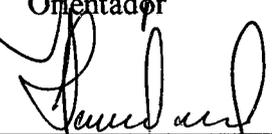
por

ADRIANO CANCELIER

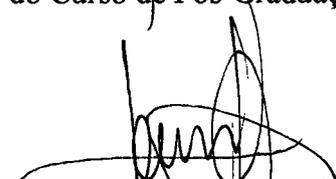
Dissertação aprovada como requisito parcial para obtenção do título de mestre no Curso de Pós-Graduação em Engenharia Química, área de concentração Desenvolvimento de Processos Químicos e Biotecnológicos, pela comissão formada por:



Prof. Ariovaldo Bolzan, D. Sc.
Orientador



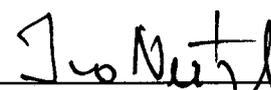
Prof. Leonel Teixeira Pinto, D. Sc.
Coordenador do Curso de Pós-Graduação em Eng. Química

Banca Examinadora :

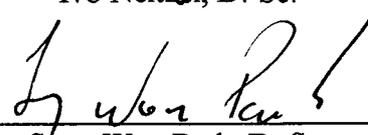
Ariovaldo Bolzan, D. Sc. (Presidente)



Nestor Roqueiro, D. Sc.



Ivo Neitzel, D. Sc.



Song Won Park, D. Sc.

Florianópolis (SC), Fevereiro de 1998.

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, Nirto e Santa pelo incentivo e confiança.

SUMÁRIO

1 - INTRODUÇÃO.....	1
2 - REVISÃO BIBLIOGRÁFICA	4
2.1 - Controle de Reatores em Batelada.....	4
2.2 - Inteligência Artificial	8
2.3 - Redes Neurais Artificiais	10
2.3.1 - Fundamentos Biológicos.....	10
2.3.2 - Histórico das Redes Neurais	12
2.3.3 - Alguns Modelos de Redes Neurais.....	15
2.3.4 - Implementação de Redes Neurais.....	19
2.3.5 - Recentes Aplicações	22
3 - FUNDAMENTAÇÃO DAS REDES NEURAIS	27
3.1 - Definições	27
3.2 - Arquitetura.....	28
3.2.1 - Micro-Estrutura Neural	29
3.2.2 - Meso-Estrutura Neural	31
3.2.3 - Macro-Estrutura Neural.....	32
3.3 - A Rede Neural Utilizada.....	33
3.3.1 - Definição Matemática de uma Rede Neural <i>Feedforward</i>	34
3.3.2 - O Método <i>Backpropagation</i>	36
3.3.3 - Equacionamento do Algoritmo <i>Backpropagation</i>	37
4 - MATERIAIS E MÉTODOS.....	43
4.1 - O Laboratório de Controle de Processos	43
4.2 - Reator e Acessórios.....	44
4.3 - Sistemas de Controle.....	48
4.3.1 - Sistema de Controle para os Experimentos com Reação Química.....	48
4.3.2 - Estratégia para Ajuste do Controlador	50

4.4 - O Controle	53
4.5 - Treinamento	57
4.5.1 - Formação dos Padrões.....	59
4.5.2 - Definição da Estrutura da Rede Neural	62
5 - O CONTROLADOR BASEADO NO MODELO NEURAL	64
5.1 - O Algoritmo de Controle.....	70
6 - RESULTADOS E DISCUSSÕES.....	73
6.1 - Treinamento da Rede Neural	74
6.2 - Controle.....	78
6.2.1 - Experimentos Sem Reação Química.....	78
6.2.2 - Experimentos com Reação Química.....	87
6.2.3 - Experimentos para a Proposta de Ajuste do Controlador	91
7 - CONCLUSÕES.....	97
8 - SUGESTÕES	98
9 - APÊNDICES	99
9.1 - Equipamentos Utilizados	100
9.2 - Procedimento para Calibração do Termopar	104
9.3 - Pesos da Rede Neural Obtidos no Treinamento.....	105
10 - REFERÊNCIAS BIBLIOGRÁFICAS	109

LISTA DE FIGURAS

<i>Figura 1 - Exemplo de possíveis partidas de um reator batelada</i>	5
<i>Figura 2 - Neurônio biológico.</i>	12
<i>Figura 3 - Gráficos de algumas funções de transferência mais empregadas.</i>	31
<i>Figura 4 - Tipos de meso-estruturas de redes neurais.</i>	32
<i>Figura 5 - Características modificáveis de um sistema de redes neurais.</i>	33
<i>Figura 6 - Rede neural feedforward multicamadas.</i>	34
<i>Figura 7 - Exemplo de um neurônio j na camada L.</i>	35
<i>Figura 8 - Fluxograma do algoritmo backpropagation.</i>	37
<i>Figura 9 - Visão da sala onde encontram-se parte das unidades experimentais do LCP.</i>	44
<i>Figura 10 - Fotografia do reator.</i>	45
<i>Figura 11 - Estrutura do reator.</i>	46
<i>Figura 12 - A planta piloto.</i>	47
<i>Figura 13 - Diagrama de blocos da malha de controle.</i>	50
<i>Figura 14 - Forma de ajuste de α.</i>	52
<i>Figura 15 - Algoritmo para ajuste automático da trajetória de referência.</i>	54
<i>Figura 16 - Diagrama de blocos do software de controle.</i>	55
<i>Figura 17 - Painel de controle do software utilizado.</i>	56
<i>Figura 18 - Tela de configurações.</i>	56
<i>Figura 19 - Tela principal do software de treinamento.</i>	58
<i>Figura 20 - Opções de treinamento.</i>	58
<i>Figura 21 - Arquitetura da rede neural utilizada.</i>	63
<i>Figura 22 - Algoritmo de controle.</i>	71
<i>Figura 23 - Dinâmica da válvula de controle de água fria.</i>	73
<i>Figura 24 - Dados para treinamento (20-60 °C).</i>	75
<i>Figura 25 - Dados para treinamento (60-95 °C).</i>	76
<i>Figura 26 - Teste para o modelo neural (20-60 °C).</i>	77
<i>Figura 27 - Teste para o modelo neural (60-95 °C).</i>	77
<i>Figura 28 - Controlador proposto sujeito a mudanças ascendentes no setpoint.</i>	79

<i>Figura 29 - Controlador PI sujeito a mudanças ascendentes no setpoint.</i>	80
<i>Figura 30 - Controlador proposto sujeito a mudanças descendentes no setpoint.</i>	81
<i>Figura 31 - Controlador PI sujeito a mudanças descendentes no setpoint.</i>	82
<i>Figura 32 - Alteração brusca de setpoint para o controlador baseado no modelo neural.</i>	84
<i>Figura 33 - Alteração brusca de setpoint para o controlador PI.</i>	85
<i>Figura 34 - Controlador baseado no modelo neural na temperatura de acoplamento.</i>	86
<i>Figura 35 - Controle da reação pelo controlador baseado no modelo neural.</i>	88
<i>Figura 36 - Controle da reação pelo controlador PI.</i>	89
<i>Figura 37 - Controlador baseado no modelo neural frente a perturbação carga.</i>	90
<i>Figura 38 - Performance do controlador para $\alpha = 0,2$.</i>	92
<i>Figura 39 - Performance do controlador para $\alpha = 0,75$.</i>	93
<i>Figura 40 - Performance do controlador para α calculado pela regra de ajuste.</i>	94
<i>Figura 41 - Valores de α para o controle de temperatura.</i>	95
<i>Figura 42 - Respostas a perturbação tipo pulso.</i>	96

LISTA DE ABREVIATURA E SÍMBOLOS

$At_{L,i,p}$	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 integral.
TM	Tempo morto.
UA_k	Ação de controle de aquecimento, correspondente à válvula de vapor.
UR_k	Ação de controle de resfriamento, correspondente à válvula de água.
UA_{max}	Valor máximo da ação de controle de aquecimento.
UA_{min}	Valor mínimo da ação de controle de aquecimento.
$W_{(L-1),ij}$	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_1	Fator de penalização da ação de controle de aquecimento.
λ_2	Fator de penalização da ação de controle de resfriamento.

RESUMO

Nos últimos anos alguns setores da indústria química têm sofrido mudanças significativas, onde tem se observado o aumento na fabricação de produtos de alto valor agregado. Fundamentada em produtos de alta qualidade e de difícil obtenção, estas indústrias produzem ao longo de um ano o que outras produzem em dias. Essas características tornam este setor industrial altamente dependente da qualidade de seu produto e da eficiência do seu processo. A perda de produção por produtos fora de especificação pode significar grandes prejuízos. Em função disto, este setor tem buscado nos sistemas de automação e controle uma opção para assegurar a qualidade de seus produtos e segurança de seus processos.

No caso dos processos com reação química, é comum a existência de reatores operando em regime de batelada. Para este tipo de reator, tem-se, geralmente, três etapas distintas de operação: a etapa de carregamento e preparação dos reagentes, a fase de reação e a etapa de descarregamento e limpeza. Devido à grande diferença de processamento entre estas etapas, a implementação de um sistema de controle capaz de controlar esses estágios de operação é uma tarefa bastante complicada e de difícil implementação industrial. As características não lineares da fase de reação, associadas à alta exotermicidade de algumas reações químicas, dificultam a utilização de controladores clássicos, levando à situações em que os processos são operados sob a supervisão de operadores.

Estas e outras necessidades, e também as facilidades computacionais que se tem conseguido com o desenvolvimento da microeletrônica, tem facilitado o surgimento de algoritmos de controle baseados em redes neurais. A capacidade de aprendizado que estas redes possuem tem contribuído muito para que as estratégias de controle apresentadas na literatura se mostrem bastante promissoras para serem aplicadas a equipamentos como os reatores que operam em batelada.

Baseadas em neurônios artificiais e em pesos que determinam a relação entre estes neurônios, as redes neurais têm a capacidade de identificar a dinâmica de um processo a partir de um conjunto de dados experimentais, chamados de padrões.

Neste trabalho é apresentado um controlador preditivo baseado em um critério de desempenho quadrático, em que se utiliza um modelo interno de previsão representado por uma rede neural do tipo *feedforward*, com o algoritmo de treinamento *backpropagation*, e estratégia de otimização *steepest descent*.

Testes experimentais obtidos em uma unidade piloto de controle de temperatura de um reator encamisado, que opera em regime de semi-batelada, mostraram que o controlador apresentado é capaz de manter o reator nas condições desejadas durante as três fases de produção. Testes comparativos realizados com o controlador clássico PI mostraram que o desempenho do controlador proposto é bastante robusto e estável, apresentando uma performance superior ao PI.

A reação entre o Tiosulfato de Sódio e o Peróxido de Hidrogênio foi a utilizada nos testes experimentais. Esta reação é altamente exotérmica ($6,82 \cdot 10^5$ KJ/Kmol) e extremamente rápida. Uma proposta para o ajuste do controlador também foi feita. Os testes para a implementação desta estratégia de ajuste foram realizados no reator sem reação química, atuando como um tanque encamisado agitado.

ABSTRACT

In the last years some sectors of the chemical industry have been suffering significant changes, with an observed increase in the production of added value products. Based on high quality products that are difficult to obtain, these industries produce along one year the same that others produce in days. Those characteristics turn this industrial sector highly dependent on the quality of its products and on the efficiency of its processes. A production loss caused by products out of specification can result in great damages. That's the reason why this sector has been searching, in automation and control systems, for an option to assure product quality and process safety.

In the case of processes with chemical reaction, the existence of batch reactors is common. For this type of reactor, there are normally three different operation stages: loading and preparation of the reactants, the reaction itself and the discharge and cleaning stage. Due to the large processing difference among these stages, the implementation of a control system able to control all of them is a quite complicated task, specially in the industry. The non linear characteristics of the reaction phase, associated to some highly exothermic chemical reactions, makes unfeasible the use of classic controllers, leading to situations where the processes are operated without control, under the supervision of operators.

These and other needs, and also the computational means resultant from the development of microelectronics, have been facilitating the appearance of control algorithms based on neural networks. Due to the learning capacity of these networks, the control strategies presented in the literature have been showing quite promising to be applied to equipment like batch reactors.

Based on artificial neurons and on weights that determine the relationship among them, the neural networks have the capacity to capture the dynamics of a process through a group of experimental data, called standards.

In this work it is presented a predictive controller based on a quadratic performance criteria that uses an internal prediction model represented by a feedforward neural network, with the backpropagation training algorithm, and the steepest descent optimization strategy.

Experimental tests obtained in a pilot temperature control unit of a semi-batch jacketed reactor, showed that the proposed controller is able to maintain the reactor in the desired conditions during the three production phases. Comparative tests with the classic PI controller showed that the performance of the proposed controller is quite robust and stable, presenting a superior performance than the PI one.

The reaction between Sodium Tiosulfate and Hydrogen Peroxide was the one used in the experimental tests. This reaction is highly exothermic ($6,82 \cdot 10^5$ KJ/Kmol) and extremely fast. A proposal for the controller's adjustment was also made. The tests for the implementation of this adjustment strategy were performed in the reactor without chemical reaction, acting as a jacketed stirred vessel.

1 - INTRODUÇÃO

A necessidade atual de processos mais eficientes e produtos de melhor qualidade tem forçado a implementação de sistemas de controle complexos nas indústrias químicas. Alta competitividade, ambiente agressivo com normas de segurança rígidas, e rápidas alterações econômicas tem sido fatores chave no fortalecimento das especificações de qualidade dos produtos industriais. Além disso, a dificuldade em operar os processos atuais tem aumentado bastante devido às novas tendências, maior quantidade de plantas altamente integradas com varias unidades de processamento. Por este motivo, alterações em uma das unidades de processamento pode desencadear mudanças em outras, e desta forma provocar a perda de controle no processo global, prejudicando assim a qualidade do produto final.

Muitos destes produtos são fabricados utilizando-se reatores ou extratores a alta pressão que operam em regime de batelada ou semi-batelada. A automação e o controle destes processos é uma tarefa bastante complexa, onde não é possível se utilizar os procedimentos padrões de projeto de controladores. O desenvolvimento da teoria de controle automático, aliado aos fatores restritivos citados acima, tem motivado o desenvolvimento e a adaptação de algumas técnicas de controle para processos químicos. Algumas estratégias baseadas em controle não linear, preditivo e adaptativo têm sido utilizadas em estudos apresentados na literatura especializada.

Estes fatores, aliados à evolução dos sistemas computacionais, aos protocolos de transmissão de dados e o surgimento de poderosas estruturas de *hardware* dedicadas a funções específicas, propiciou o uso de técnicas como o STC (*Self Tuning Control*) de Clarke e Gawthrop (1975), o DMC (*Dinamic Matrix Control*) de Cutler e Ramaker (1980) e o GPC (*Generalized Predictive Control*) de Clarke *et alli* (1987), que evoluíram rapidamente e tem substituído o PID em situações onde este é ineficiente.

Neste trabalho propõe-se um sistema de controle para os reatores que operam em regime de batelada e batelada alimentada. A estratégia proposta utiliza uma filosofia de controle

preditivo, utilizando as redes neurais artificiais como modelo do processo. Estes reatores são largamente empregados na indústria química, principalmente nas indústrias de produtos farmacêuticos, biológicos e poliméricos. Estes produtos possuem alto valor agregado, sendo portanto fabricados em pequenas quantidades, dispondo-se de um reator que processa várias reações, estando desta maneira sujeito a freqüentes partidas e paradas. Um sistema de controle adequado para esses reatores propicia uma diminuição no tempo de produção e uma maior uniformidade para os produtos. Para reações exotérmicas, a operação deste tipo de reator constitui três fases distintas: aquecimento dos reagentes até a temperatura ideal, manutenção da temperatura no valor desejado durante a reação e resfriamento dos produtos. O controlador ideal deve ser capaz de elevar a temperatura até o valor desejado no menor tempo possível sem *overshoot* e mantê-la no *setpoint* durante a fase de reação, apesar da não-linearidade das reações químicas.

Nesta 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 poderosa ferramenta na modelagem de sistemas estáticos e dinâmicos com grandes não linearidades. Devido ao maciço paralelismo, rápida adaptabilidade, e inerente capacidade de aproximação, as redes neurais tem concentrado enorme pesquisa, especialmente na área de identificação e controle. O algoritmo de aprendizagem mais utilizado tem sido o *Backpropagation*, o qual é largamente empregado no treinamento de redes neurais *feedforward* e *feedforward/feedback*. O algoritmo de otimização empregado neste trabalho foi o *steepest descent*. O *Backpropagation* é largamente empregado em um grande número de problemas de classificação e problemas de mapeamento de funções. Em muitos casos, contudo, um elevado número de iterações são necessárias para o ajuste de parâmetros da rede neural, impossibilitando aplicações em tempo real, como controle adaptativo de processos, necessitando na maioria dos casos um elevado tempo computacional. Este método pode apresentar ainda problemas de mínimos locais, acarretando um aprendizado deficiente em algumas regiões dos dados fornecidos à rede, principalmente se a dinâmica do processo for variável. Neste caso, propõe-se o uso de mais de uma rede neural para o mesmo processo, obtendo-se desta forma uma melhor aproximação, além de um processo de treinamento mais rápido, pois a quantidade de padrões para cada rede é menor.

Neste trabalho realizou-se o controle em um sistema com duas variáveis manipuladas e uma variável controlada, portanto duas entradas e saída simples. Neste caso, o processo pode ser classificado como MISO, ou seja *Multiple Input / Single Output*. Mesmo com uma variável controlada, o processo global pode ser considerado multivariável, significativamente mais complexo que os sistemas SISO (*Single Input / Single Output*).

Embora o objetivo principal do trabalho seja propor um controlador eficiente para reatores que operam em regime de batelada alimentada, foi realizado também o controle de temperatura do reator sem reação química, que apesar de não ter a complexidade de um reator químico, facilita a observação da influência dos parâmetros sobre a performance do controlador. O controlador foi testado em um reator piloto, que opera em regime de batelada alimentada. Utilizou-se a reação do Tiosulfato de Sódio com Peróxido de Hidrogênio, que por ser altamente exotérmica ($6,82 \cdot 10^5$ KJ/Kmol) e extremamente rápida, proporciona um excelente teste para o controlador.

O controlador utilizado baseia-se na minimização de um critério de desempenho quadrático, onde a rede neural é o modelo do processo. Propôs-se também um algoritmo para ajuste da trajetória de referência do controlador baseado na magnitude de variação do *setpoint*, e do estado do sistema. Foram realizados testes comparativos com o clássico controlador PI, e em todos os casos, a estratégia de controle proposta obteve performance superior.

Este trabalho está dividido em capítulos. No capítulo 2 apresenta-se uma revisão bibliográfica de controle de reações em batelada, semi-batelada e redes neurais artificiais. No capítulo 3 a fundamentação teórica das redes neurais, bem como a definição matemática de uma rede neural *Feedforward* são apresentados. Os materiais utilizados, bem como as metodologias são apresentados no capítulo 4. No Capítulo 5 a definição matemática do controlador baseado no modelo neural é apresentada. Os resultados experimentais e discussões são apresentados no capítulo 6. Nos capítulos 7 e 8 são apresentadas as conclusões e sugestões, respectivamente.

2 - REVISÃO BIBLIOGRÁFICA

Considerando-se que o objetivo deste trabalho é propor e testar uma estratégia de controle para reatores químicos que operam em regime de batelada alimentada, dividiu-se este capítulo em três partes. A primeira caracteriza as dificuldades de processamento destes reatores, a segunda apresenta um breve comentário sobre inteligência artificial, e finalizando, na terceira parte um estudo sobre as redes neurais artificiais é apresentado, pois a modelagem do reator foi realizada com a utilização de uma rede neural.

2.1 - Controle de Reatores em Batelada

A operação de uma planta em batelada difere em vários aspectos de um processamento contínuo. No entanto, dois fatores são de fundamental importância nessa diferenciação. O primeiro está relacionado com a trajetória de partida da planta. Em um processo contínuo, essa etapa deve ser feita de maneira segura, enquanto que no processamento em batelada, além de segura, a partida deve ser rápida. Uma planta contínua, na maioria dos casos, sofre uma partida por ano, não importando dessa forma, que a mesma seja feita de uma maneira mais lenta, pois o tempo de partida será desprezível frente ao tempo total de operação. Nos processos em batelada as partidas são frequentes, sendo que em alguns casos o ciclo de duração pode durar apenas algumas horas. Desta forma, é fundamental que a partida se dê o mais rápido possível. O segundo fator de diferenciação está relacionado à manutenção das condições ideais de operação do reator após a partida. Nos processos contínuos esta etapa é iniciada quando se atinge o estado estacionário de operação, a partir do qual as condições de operação e propriedades físicas permanecem constantes. No processamento em batelada e semi-batelada não se atinge um estado estacionário de operação. Conseqüentemente não existe uma característica dinâmica que represente o processo durante todo seu ciclo de operação.

Para os reatores químicos que operam em regime de batelada e batelada alimentada, a questão da partida é a que tem recebido maior atenção dos pesquisadores. É de consenso que a melhor partida é aquela que faz uma rápida e segura transição entre as condições iniciais até as condições ideais para a reação. A figura 1 ilustra os possíveis perfis de temperatura que se pode obter durante a partida desses reatores. Nesta figura, a partida representada pela trajetória número 2 atinge rapidamente a temperatura de operação, ou seja, o *setpoint*, mas a ocorrência de *overshoot* é observada, e isso, para algumas reações, pode determinar um aumento brusco da temperatura, ocorrendo o que se costuma chamar de disparo do reator. Uma performance diferente, onde não ocorre o *overshoot*, é representada pela trajetória número 1, porém neste caso o período de partida é demasiado longo, indicando que o controlador utilizado é conservativo. A situação mostrada pela trajetória número 3 representa a partida ideal, ou seja, a transição mais rápida possível, sem a ocorrência de *overshoot*.

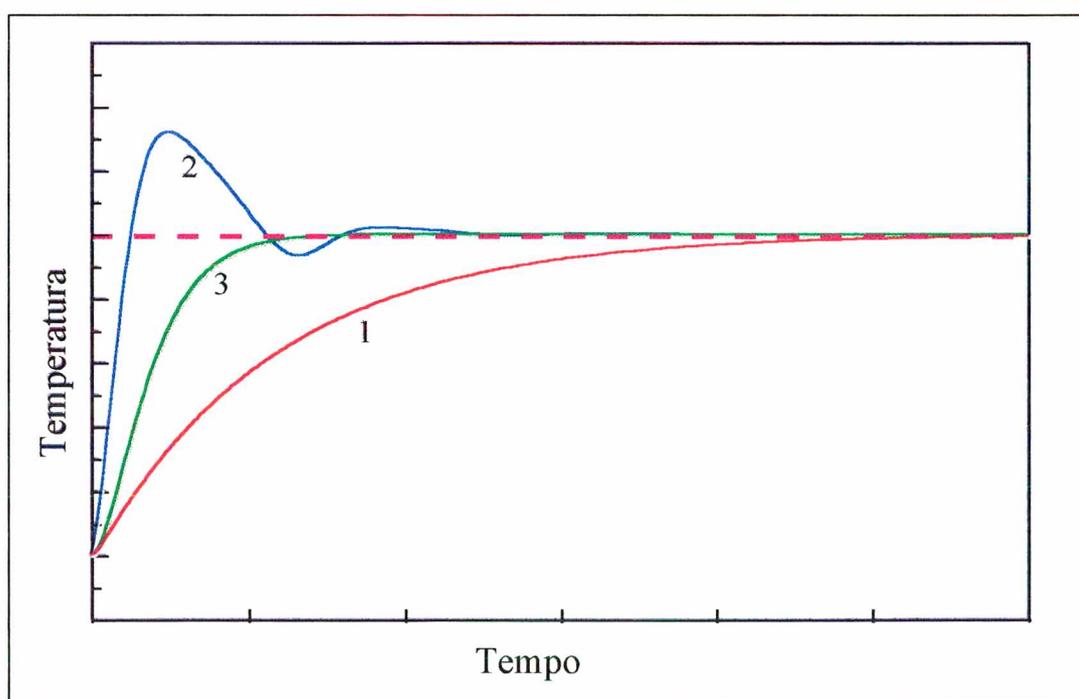


Figura 1 - Exemplo de possíveis partidas de um reator batelada

Muitos autores tentaram viabilizar um sistema de controle que implementasse a situação ideal. Nas décadas de 60 e 70, existiu um considerável interesse em controle de reatores químicos que operam em batelada, entretanto, os primeiros trabalhos foram concentrados na obtenção de trajetórias ótimas de temperatura, baseadas somente na cinética da reação.

Uma maneira rigorosa de resolver o problema da partida pode ser conseguida utilizando-se a teoria de controle ótimo, em termos de rapidez na partida, a qual resulta em controle *on-off* ou singular, estudada por JACKSON *et alli* (1971) [1], e por TSOUKAS & TIRREL (1982) [2].

SHINSKEY (1979) [3] propôs um controlador duplo, onde utilizava um controlador *on-off* e um *feedback* convencional. O *on-off* fornece a rápida transição entre os *setpoints*, e depois que o processo atingiu o novo *setpoint*, um controlador PID, por exemplo, regula o processo. O autor sugere um método de implementação do controlador *on-off* baseado no valor do erro de saída. O tempo de mudança, ou tempo de chaveamento, deve ser obtido a partir de um modelo do processo.

O número de publicações que abordam a implementação de novas técnicas no controle de reatores químicos tem aumentado bastante. Particularmente, os reatores em batelada e batelada alimentada tem desafiado os pesquisadores. As áreas de polímeros e produtos biológicos, as quais fazem uso intensivo desse tipo de reator, tem sido bastante abordadas nas pesquisas.

FARBER & YDSTIE (1986) [4] propuseram uma compensação adaptativa para grandes perturbações em reatores contínuos. Uma versão adaptativa para o IMC (*Internal Model Control*) foi sugerida por TAKAMATSU *et alli* (1986) [5], em uma aplicação a um reator para produção de estireno. Uma proposta de um controlador multivariável à um reator semi-batelada, foi apresentada por JAYDEVA BHAT & CHIDABARAM em 1991 [6]. Neste caso as variáveis manipuladas eram a vazão de refrigerante e a vazão de alimentação de reagente. A estratégia é baseada em um modelo linearizado. O teste da performance dos algoritmos foi realizado através de simulações baseadas em modelos matemáticos.

A grande maioria das publicações de trabalhos nesta área são de testes simulados, encontrando-se poucas aplicações práticas a nível industrial ou de unidade piloto. Algumas publicações em que se realizaram testes experimentais são apresentadas a seguir.

PHILLIPS *et alli* (1988) [7] propõem que o tempo de chaveamento seja calculado *on-line*, utilizando os parâmetros estimados de um modelo e a predição de uma variável de saída futura. Os testes simulados e experimentais foram utilizados no controle de temperatura de um

leito fluidizado, onde, a partir do ponto de chaveamento, o controle era realizado por um controlador PID adaptativo, CAMERON & SEBORG [8].

BOLZAN (1991) [9] estudou o uso de um controlador adaptativo, mais especificamente o STC, para o controle da temperatura de um reator semi-batelada, processando uma reação altamente exotérmica. Os testes foram validados experimentalmente em uma unidade piloto, mostrando a superioridade deste controlador quando comparado ao GPC e o tradicional PID, em todo o ciclo de operação.

LONGHI [10] (1995) utilizou um controlador GPC adaptativo, acrescido de um sistema de ajuste automático e uma estrutura competitiva de modelos, para controlar a temperatura de operação de um reator em batelada. Testes experimentais com a reação de polimerização do estireno em suspensão, mostraram a viabilidade do algoritmo utilizado e das regras gerais de ajuste para seus parâmetros.

MACHADO (1996) [11] empregou o controlador adaptativo STC, ajustado com os parâmetros de um PI clássico, no controle de temperatura de um reator encamisado, onde a reação de polimerização do estireno em suspensão foi processada. O reator foi operado em batelada. Apesar do grande número de perturbações, as quais foram introduzidas no sistema, o controlador STC mostrou-se robusto para controlar processos desta natureza.

Recentemente tem aumentado o número de comunicações na literatura que abordam a implementação de técnicas avançadas no controle de reatores químicos. Uma destas técnicas é a utilização de redes neurais artificiais na modelagem deste tipo de sistemas. As redes neurais tem surgido como uma poderosa ferramenta na modelagem de processos não lineares complexos.

DIRION *et alli* (1996) [12] desenvolveram uma rede neural adaptativa para controlar uma reação entre o Tiosulfato de Sódio e Peróxido de Hidrogênio. Os testes simulados foram feitos para um reator batelada. O objetivo da rede neural era fornecer a entrada desejada, mais precisamente, a abertura da válvula que deveria ser implementada ao processo.

A proposta de um neurocontrolador baseado no algoritmo RTRL (*Real-Time Recurrent Learning*) foi realizada por CHOVAN *et alii* (1996) [13]. O controle de um bioreator foi um dos problemas de estudo. É um processo altamente não linear, portanto apresentando grandes dificuldades de ser controlado por procedimentos convencionas.

2.2 - Inteligência Artificial

Embora existam definições mais precisas para a Inteligência Artificial (IA), pode-se dizer que a IA é o estudo de como fazer com que os computadores realizem tarefas em que, no momento, os seres humanos são melhores. Na verdade, o lento progresso dos *softwares* computacionais capazes de realizar tarefas consideradas complicadas, como a de opinar por exemplo, foi um dos primeiros resultados a sair da IA experimental. Alguns dos primeiros problemas de IA que foram estudados envolviam jogos e provas de teoremas. SAMUEL (1963) [14], escreveu um programa de jogo de damas, que não apenas jogava com oponentes, mas também usava a experiência para melhorar posteriormente seu desempenho. O *Logic Theorist* de NEWELL (1963) [15], foi uma tentativa inicial de provar teoremas matemáticos.

A IA teve enorme ascensão com a criação da linguagem de programação LISP, que foi a primeira linguagem voltada para estudos nesta área. O conceito de IA foi definitivamente introduzido por MARVIN MINSKY (1963) [16]. À medida que a pesquisa de IA progrediu e técnicas para tratar de quantidades maiores de conhecimentos do mundo foram desenvolvidas, conseguiu-se certo progresso nas tarefas descritas e novas outras puderam ser testadas.

Recentemente a IA tem conquistado espaço entre muitos cientistas nas mais diversas áreas. A lista seguinte contém um resumo de alguns dos problemas que se enquadram dentro do escopo da inteligência artificial:

- ◆ Jogos
- ◆ Prova de teoremas

- ◆ Resolução de problemas gerais¹
- ◆ Percepção
 - * Visão
 - * Fala
- ◆ Compreensão de linguagem natural
- ◆ Resolução de problemas especializados
 - * Matemática simbólica
 - * Diagnose médica
 - * Análise química
 - * Projetos de engenharia

Para a solução destes problemas, várias são as técnicas de IA que podem ser empregadas, no entanto algumas são mais específicas a determinados assuntos, ou ainda não obtiveram atenção em outros.

Dentre os ramos da inteligência artificial, pode-se citar:

- ◆ Sistemas Especialistas
- ◆ Redes Neurais
- ◆ Algoritmos Genéticos
- ◆ Algoritmos Inteligentes
- ◆ Lógica *Fuzzy*
- ◆ Processamento de Linguagem Natural

Neste trabalho, o alvo de estudo será Redes Neurais, portanto informações sobre as demais áreas não serão necessárias.

¹ Esta técnica de sistemas especialistas, inicialmente *General Problem Solver* (GPS), tinha o propósito de resolver quaisquer problemas, mas a idéia fracassou, e evoluiu posteriormente para a solução de problemas especializados.

2.3 - Redes Neurais Artificiais

Redes neurais artificiais são sistemas computacionais, de implementação em *hardware* ou *software*, que imitam as habilidades computacionais do sistema nervoso biológico, usando um grande número de simples neurônios artificiais interconectados. Os neurônios artificiais são simples emulações dos neurônios biológicos. Estes recebem informação de sensores ou de outros neurônios artificiais, produzindo operações simples sobre estes dados, passando o resultado para outros neurônios artificiais. As redes neurais constituem uma tecnologia computacional emergente que pode aumentar significativamente o número de aplicações, desde o reconhecimento de padrões até otimização e seqüenciamento. O campo de estudo mantém dependências disciplinares, com as chamadas ciências cognitivas, a cibernética, a psicologia, a neurobiologia, a matemática e a física. Na engenharia, a construção de melhores sistemas, controle e robótica, estabilidade postural e estruturas flexíveis são exemplos de aplicação.

2.3.1 - Fundamentos Biológicos

A extensão para o qual uma rede neural artificial representa um sistema neural biológico é variável. Para alguns pesquisadores, este é o primeiro interesse, já para outros, a capacidade da rede realizar tarefas (tais como aproximar ou representar uma função) é mais importante que uma comparação biológica. Apesar do interesse principal ser na capacidade computacional das redes neurais, uma breve discussão de algumas semelhanças entre os neurônios biológicos será feita, com a finalidade de esclarecer algumas características mais importantes das redes neurais artificiais.

Um neurônio biológico possui três tipos de componentes, os quais são de interesse particular na compreensão de um neurônio artificial: os dendritos, o soma e o axônio. Os muitos dendritos recebem sinais provenientes de outros neurônios. Os sinais são impulsos elétricos que são transmitidos através do *gap* sináptico por meio de processos químicos. A ação do transmissor

químico modifica o sinal de entrada (tipicamente, escalonando a frequência dos sinais que são recebidos), de maneira similar à ação dos pesos em uma rede neural artificial.

O soma, ou corpo celular, soma os sinais de entrada. Quando suficientes entradas são recebidas, a célula dispara, ou seja, ela transmite um sinal através do axônio para outras células. A transmissão do sinal a partir de um neurônio particular é dotada por uma ação potencial, resultante de concentrações diferenciais de *ions* de cada lado, na extremidade do axônio do neurônio. Os *ions* mais diretamente envolvidos são os *ions* potássio, sódio e cloreto.

Um neurônio biológico genérico é apresentado na figura 2, juntamente com axônios provenientes de dois outros neurônios (de onde o neurônio ilustrado recebe os sinais), e dendritos para dois outros neurônios (para onde o neurônio ilustrado envia os sinais). Alguns aspectos importantes dos elementos de processamento de uma rede neural artificial são sugeridos a partir de propriedades do neurônio biológico, tais como:

1. O elemento processador recebe muitos sinais.
2. Os sinais podem ser modificados por um peso no momento da sinapse.
3. O elemento processador soma as entradas ponderadas.
4. Em circunstâncias apropriadas (entradas suficientes), o neurônio transmite uma única saída.
5. A saída de um neurônio pode chegar a muitos outros.
6. A informação de processamento é local (embora outros meios de transmissão, tais como a ação de hormônios, podem suggestionar outros processos de controle).
7. A memória é distribuída:
 - a) A memória de longa duração reside nas sinapses ou nos pesos.
 - b) A memória de curta duração corresponde aos sinais enviados pelos neurônios.
8. A intensidade das sinapses pode ser modificada pela experiência.
9. Os neurotransmissores das sinapses podem ser excitatórios ou inibitórios.

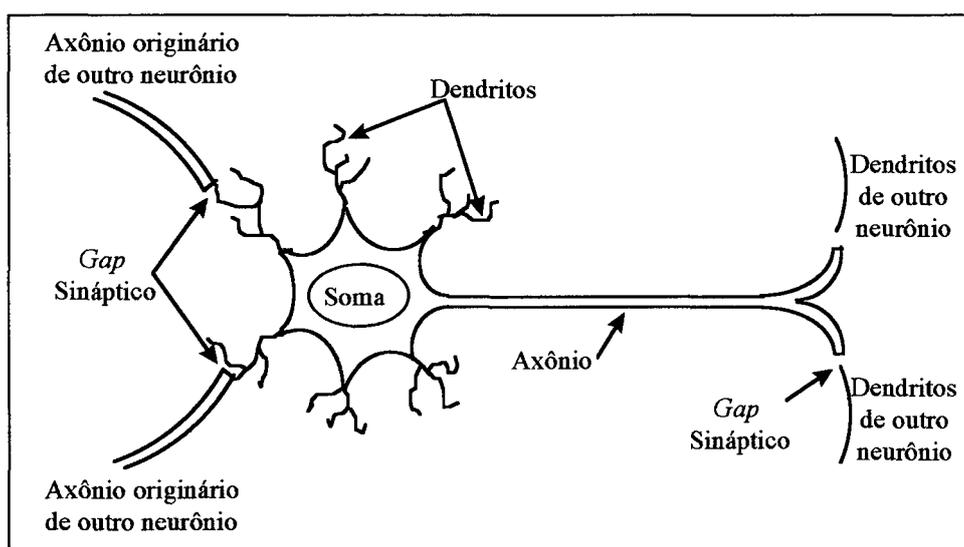


Figura 2 - Neurônio biológico.

Outra importante característica que é comum entre as redes neurais artificiais com o sistema neural biológico é a tolerância à falhas, em dois aspectos. Primeiro a capacidade de reconhecer uma parte diferente de um sinal que tenha sido visto anteriormente. Segundo, a capacidade da distinção de ruídos.

2.3.2 - Histórico das Redes Neurais

A história das redes neurais artificiais é relativamente recente, tendo iniciado há aproximadamente meio século. Nos trabalhos pioneiros dos neurofisiologistas Donald Hebb e Karl Lashley, por volta de 1940, as primeiras simulações foram feitas com papel e lápis. O histórico pode ser dividido em três períodos, sendo entusiasmo inicial, a falta de suporte ou desencantamento e o ressurgimento das pesquisas.

No clássico artigo, intitulado *A Logical Calculus of the Ideas Immanent in Nervous Activity*, publicado em 1943, McCULLOCH e PITTS [17] propuseram um modelo simplificado de neurônios biológicos. Neste modelo, em um dado instante de tempo, o neurônio ou estava disparando ou inativo. Os principais fatos percebidos por McCulloch e Pitts, e que tornaram possível a modelagem matemática do neurônio foram:

- ◆ Neurônios comportam-se como somadores algébricos
- ◆ Adicionam entradas excitatórias
- ◆ Subtraem entradas inibitórias
- ◆ Neurônios possuem uma propriedade de limiar, isto é, quando as entradas integradas excedem o limiar, disparam um sinal através do axônio
- ◆ Neurônios comportam-se analógicamente, enquanto o axônio comporta-se de forma digital

O neurônio de McCulloch e Pitts recebe e produz um valor *booleano* 0 ou 1. Existem conexões excitatórias e inibitórias representadas através de um peso com sinal, o qual reforça ou dificulta a geração de um impulso. Um neurônio produz um impulso, ou seja, uma saída igual a 1, se, e somente se, a soma das entradas é maior ou igual a um limiar.

O interesse em um mecanismo de aprendizagem foi iniciado por HEBB em 1949 [17], que propôs uma regra de aprendizagem conhecida hoje como “Regra de Aprendizagem Hebbiana”. Hebb demonstrou que a capacidade de aprendizagem em redes neurais pode ser conseguida através da variação dos pesos das conexões entre os neurônios. A regra de Hebb diz que, quando um estímulo de entrada influencia na produção de estímulo de saída, o peso da conexão entre os neurônios deve ser incrementado.

ROSEMBLATT [18] em seu livro *Principles of Neurodynamics*, em 1958, forneceu várias idéias a respeito dos *Perceptrons*, que são redes de neurônios de limiar baseado no modelo de McCulloch e Pitts. Demonstrou que, se fossem acrescentadas sinapses ajustáveis, os neurônios artificiais de McCulloch e Pitts poderiam ser treinados para classificar certos tipos de padrões em duas categorias, dada sua saída binária. Roseblatt elaborou ainda, uma arquitetura e algoritmo, chamados de *Back-Coupled* e *Back-Coupled Error Correction Algorithm*, respectivamente, capazes de adaptar os pesos para um conjunto de entradas de uma unidade de associação, de acordo com uma saída desejada. De início, como os pesos das conexões são aleatórios, a saída também é aleatória, mas, pelo ajuste gradual dos pesos, o *Perceptron* é treinado para fornecer saídas de maneira desejada.

WIDROW e HOFF [19] em 1960 desenvolveram o ADALINE (*Adaptive Linear Element*) e o MADALINE (*Many ADALINE*) *Perceptron* como um dispositivo prático para resolver

tarefas de reconhecimento de padrões. O ADALINE/MADALINE usou saídas analógicas em vez de binárias como originalmente proposto por McCulloch e Pitts. O algoritmo de aprendizagem é conhecido como Regra Delta de Widrow-Hoff, ou regra de correção do erro quadrático médio mínimo. Tanto o algoritmo *back-couple error correction*, quanto a regra delta de Widrow-Hoff são métodos de aprendizagem para redes *Perceptron* de apenas uma camada de neurônios.

MINSKY e PAPERT [20] estudaram estes algoritmos e publicaram em 1969, o livro *Perceptrons*. Provaram formalmente que uma rede formada de uma única camada de neurônios, independente do algoritmo de aprendizagem, é capaz apenas de resolver o problema de associação de padrões quando os conjuntos de pares de padrões são linearmente separáveis. Naquela época era conhecido que redes *Perceptron* com mais de uma camada de neurônios, chamadas de *MultiLayered Perceptrons*, ou ainda *Feedforward*, tinham o poder computacional de aprender padrões linearmente dependentes, contudo um algoritmo de aprendizagem que pudesse realizar tal associação era desconhecido. Estes resultados e observações feitas por Minsky e Papert foram devastadores, e a abordagem conexionista ficou em segundo plano durante a década de 70 até o início da década de 80.

O ressurgimento dos estudos, por volta de 1982, deve-se a trabalhos importantes como o de John Hopfield, bem como ao desenvolvimento na microeletrônica que possibilitaram implementações de arquiteturas de redes, em *hardware*.

O problema das redes *Perceptron* na resolução do problema da associação de padrões para um conjunto de padrões não-lineares foi eliminado por Rumelhart, Hinton e Willians. A solução encontrada foi a Regra Delta Generalizada, apresentada no livro *Learning Internal Representations by Error Propagation* de RUMELHART *et alli* [18] em 1986.

Progressos significativos tem sido realizados no campo de redes neurais desde então, não somente nas aplicações comerciais, mas também em muitas frentes de pesquisa. *Chips* de redes neurais estão surgindo, bem como o desenvolvimento de aplicações para problemas complexos.

2.3.3 - Alguns Modelos de Redes Neurais

Abaixo será apresentada uma listagem de diferentes modelos de redes neurais, bem como as vantagens e desvantagens de cada um, além de exemplos de possíveis aplicações. Estas informações foram extraídas de LOESCH & SARI [18].

Nome da Rede : ADALINE/MADALINE
Ano de Publicação : 1960
Autor : B. Widrov
Primeiras Aplicações : Filtragem de sinal adaptativo, equalização adaptativa
Vantagens : Rápida, fácil de implementar, tanto em circuito analógico como VLSI
Desvantagens : Assume relação linear entre entrada e saída desejada. Somente é possível classificar espaços linearmente separáveis

Nome da Rede : *Adaptive Resonance Theory (ART)*
Ano de Publicação : 1983
Autor : G. Carpenter & S. Grossberg
Primeiras Aplicações : Reconhecimento de padrões
Vantagens : Capaz de aprender novos padrões, de novas categorias de padrões, e reter as categorias já aprendidas
Desvantagens : Natureza dos exemplares categóricos podem mudar com o aprendizado

Nome da Rede : *Backpropagation Perceptron*
Ano de Publicação : 1974 - 1986
Autor : P. J. Werbos, D. Parker & D. Rumelhart
Primeiras Aplicações : Reconhecimento de padrões, filtragem de sinal, remoção de ruído, segmentação de sinal/imagem, classificação, mapeamento, controle robótico adaptativo, compressão de dados
Vantagens : Operação rápida. Boa em formar representações internas das características dos dados de entrada ou classificação e outras tarefas. Bem compreendida, com muitas aplicações de sucesso

- Desvantagens** : Tempo de treinamento longo
- Nome da Rede** : *Recurrent*
- Ano de Publicação** : 1987
- Autor** : Almeida, Pineda
- Primeiras Aplicações** : Controle robótico, reconhecimento da 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
- Nome da Rede** : *Time-Delay*
- Ano de Publicação** : 1987
- Autor** : D. W. Tank & J. J. Hopfield
- Primeiras Aplicações** : Reconhecimento da fala
- Vantagens** : Desempenho equivalente aos melhores métodos convencionais, rápida operação
- Desvantagens** : Janela fixada para a atividade temporal representada, responde desastrosamente para diferenças em escalas na entrada
-
- Nome da Rede** : *Funcional-Link Network*
- Ano de Publicação** : 1988
- Autor** : Y. H. Pao
- Primeiras Aplicações** : Classificação, mapeamento
- Vantagens** : Somente duas camadas (entrada e saída) são necessárias, rápida para treinar
- Desvantagens** : Não é claro o modo de identificar funções “*Funcional-Links*”
- Nome da Rede** : *Radial Basis Function Network (RBF)*
- Ano de Publicação** : 1987 - 1988
- Autor** : Grupo de pesquisadores

Primeiras Aplicações : Classificação, mapeamentos
Vantagens : Uma rede com uma única camada oculta de neurônios é equivalente a rede *Perceptron* multi-camadas básico com duas camadas ocultas
Desvantagens : Problemas de estabilidade

Nome da Rede : *Backpropagation of Utility Function Through Time*

Ano de Publicação : 1974

Autor : P. J. Werbos

Primeiras Aplicações : Maximiza o índice de desempenho, ou a função utilidade no tempo, neurocontrole (robótica)

Vantagens : Abordagem neural mais compreensiva para modelos de controle e/ou previsão

Desvantagens : Pode ser usado somente depois de identificado o modelo diferenciável, adaptação *off line* se o modelo é dinâmico, e assume que o modelo é exato

Nome da Rede : *Bidirecional Associative Memory* (BAN)

Ano de Publicação : 1987

Autor : B. Kosko

Primeiras Aplicações : Heteroassociativa (memória endereçada por conteúdo)

Vantagens : Simples, regra de aprendizado, arquitetura e dinâmica claras. Prova clara da estabilidade dinâmica

Desvantagens : Capacidade de armazenamento e precisão de recuperação pobres

Nome da Rede : *Boltzmann Machine, Cauchy Machine*

Ano de Publicação : 1984 - 1986

Autor : G. Hinton, T. Sejnowski, D. Ackley, H. Szu

Primeiras Aplicações : Reconhecimento de padrões (imagem, sonar, radar), otimização

Vantagens : Capaz de formar representação ótima das características dos padrões. Segue superfície de energia para obter otimização no ponto mínimo

Desvantagens : A máquina de Boltzmann possui tempo de aprendizado longo,

enquanto a máquina de Cauchy oferece aprendizado rápido

Nome da Rede : *Brain-State-in-a-Box* (BSB)
Ano de Publicação : 1977
Autor : J. Anderson
Primeiras Aplicações : Revocação autoassociativa
Vantagens : Possivelmente melhor desempenho que a rede Hopfield
Desvantagens : Incompleta exploração em termos de desempenho e aplicações em potencial

Nome da Rede : Hopfield
Ano de Publicação : 1982
Autor : J. Hopfield
Primeiras Aplicações : Evocação autoassociativa, otimização
Vantagens : Conceitualmente simples, possui estabilidade dinâmica, de fácil implementação em circuitos VLSI
Desvantagens : Incapaz de aprender novos estados (pesos fixados para Hopfield discreta), armazenamento de memória pobre, pode estabilizar em muitos estados espúrios

Nome da Rede : *Learning Vector Quantization*
Ano de Publicação : 1981
Autor : T. Kohonen
Primeiras Aplicações : Revocação autoassociativa (complementação do padrão a um padrão parcial apresentado), compreensão de dados
Vantagens : Capaz de auto-organizar representações vetoriais de distribuições aleatóreas em dados apresentados. Execução rápida após treinamento completado
Desvantagens : Características não resolvidas na seleção do número de vetores usados e tempo de treinamento apropriado. Treinamento lento

Nome da Rede	: Neocognitron
Ano de Publicação	: 1975 - 1982
Autor	: K. Fukushima
Primeiras Aplicações	: Reconhecimento de caracteres manuscritos e outras figuras
Vantagens	: Capaz de reconhecer padrões independente da escala, translação e rotação
Desvantagens	: Requer muitos elementos de processamento e camadas, estrutura complexa.

Nome da Rede	: <i>Self-Organizing Topology-Preserving Map</i>
Ano de Publicação	: 1981
Autor	: T. Kohonen
Primeiras Aplicações	: Mapeamentos complexos (envolvendo relações de vizinhança), compressão de dados e otimização
Vantagens	: Capaz de auto-organizar representações vetoriais de dados com uma ordenação significativa entre as representações
Desvantagens	: Características não resolvidas na seleção do número de vetores usados e tempo de treinamento apropriado. Treinamento lento

2.3.4 - Implementação de Redes Neurais

As redes neurais artificiais podem ser implementadas em *software*, *hardware*, meios óticos, ou ainda em sistemas combinados ou híbridos. A escolha do tipo de implementação vai depender das necessidades associadas ao projeto de implementação, bem como na disponibilidade de recursos para ser investido no projeto, tendo em vista o balanceamento entre flexibilidade, desempenho e custo.

1. **Implementação em Software** - A grande maioria das implementações de redes neurais é realizada por *softwares*, que podem ser implementados em microcomputadores, estações de

trabalho, coprocessadores especiais, processadores paralelos e *hardware* específico. As implementações podem ser específicas para análise e aplicações de redes neurais, ou para o desenvolvimento de simuladores com fins comerciais ou científicos. Geralmente a linguagem de programação aplicada ao desenvolvimento destes simuladores é de alto nível. O benefício da utilização de um simulador é a facilidade de implementar modelos, verificar se a implementação é correta e controlar os experimentos. Os simuladores normalmente são limitados quando tarefas mais complexas são necessárias, redes híbridas ou atípicas, desenvolvimento de novas arquiteturas, ou ainda quando existe a necessidade de integrar a rede a outros recursos não previstos pelos simuladores, como, por exemplo, coletar dados através de uma porta serial para operações *online*. As principais vantagens de utilizar *software* são a flexibilidade da implementação, o baixo custo e a facilidade de seu desenvolvimento.

2. Implementação em Hardware Eletrônico - Vários modelos de redes neurais foram mapeados em *hardware* digital e analógico. Este tipo de implementação é adotado quando um alto desempenho da rede é necessário, para resolver problemas reais de grande volume de processamento, e aplicações em tempo real. Tanto em *software* quanto em *hardware*, o processo de mapear um modelo de rede neural deve preservar a definição do modelo original, a fim de criar uma implementação correta e realística.

As implementações em *hardware* podem ser diferenciadas de acordo com as seguintes variáveis:

- **Escala de Integração.** As implementações podem ser realizadas a nível de placa, *chips* VLSI (*Very Large Scaling Integration*) e *chips* WSI (*Waferscale Integration*). As implementações experimentais são realizadas a nível de placa, e as definitivas são completamente feitas em *chips*.
- **Densidade de Integração.** Define se todas as operações que são realizadas pelo modelo da rede vão ser colocadas na implementação ou não. Neste caso a fase de treinamento da rede poderia ser feita a parte. Outro caso seria o uso de placas aceleradoras de operações de ponto flutuante. Deste modo a densidade de integração pode ser dita “Parcial” ou “Completa”.
- **Modo de Integração.** Pode ser digital, analógico ou ainda digital e analógico. A escolha do modo de integração depende inteiramente do tipo de rede a ser implementado. A lógica

digital é mais simples e barata que a analógica. A maioria das implementações digitais utilizam tecnologia de CMOS (*Complementary Metal-Oxide Semiconductor*), e várias abordagens diferentes tem sido utilizadas para se implementar redes neurais de forma analógica.

- **Precisão das Conexões.** É o número de *bits* utilizado para representar os pesos das conexões. Certos modelos de redes exigem uma maior precisão na representação dos pesos que outros, e mesmo para um dado modelo, a precisão exigida na fase de aprendizagem é geralmente maior que na fase de uso, pois poderia ocorrer a propagação do erro.

A principal vantagem da implementação em *hardware* eletrônico é a alta velocidade de operação da rede, acompanhada de uma solução específica e ótima que possibilita estender a faixa de aplicabilidade de redes neurais a problemas até então não confrontados. As desvantagens são o alto custo, a pequena quantidade de elementos processadores atualmente disponíveis e pouca flexibilidade para implementar diferentes tipos de elementos processadores, arquiteturas e algoritmos de aprendizagem.

3. Implementação em Meios Óticos - Redes neurais necessitam computar e transmitir informações entre seus elementos processadores. A atividade de computar é muito bem realizada por implementações em *hardware* eletrônico, mas, por outro lado, a atividade de transmitir informação, não é tão facilmente implementada, pois boa parte da área do *chip* deve ser dedicada às conexões, restringindo o seu número máximo. As implementações em meios óticos surgem como uma alternativa para conectar os elementos processadores através de um feixe de luz. O mecanismo de sinapse pode ser realizado em sistema holográfico, através da exposição de um cristal fotorefratário ou uma superfície fotoreatora à luz. Quando o cristal é exposto à luz, os elétrons na área luminosa são transferidos para as áreas escuras, formando um padrão de cargas no cristal. As operações computacionais são realizadas através de dispositivos ótico-eletrônicos, que utilizam cristais com índices refratários diferentes como operadores computacionais. Resultados teóricos demonstram que os pesos das conexões podem ser armazenados em hologramas com um poder de armazenamento da ordem de 10^{12} *bits/cm*³. O potencial associado das implementações em meios óticos pode ser percebido, pois o sistema nervoso possui cerca de 10^{11} neurônios, e seus estados ativo ou inativo poderiam ser representados em apenas 1 *cm*³ de um holograma. Contudo, o estado da arte dos sistemas

holográficos ainda deixa muito a desejar. As principais vantagens da implementação em meios óticos são a velocidade de acesso e operação, a alta capacidade de gravação e a quebra dos limites de implementação em silício. As desvantagens são o estado inicial da engenharia do processo das implementações, a dificuldade em mapear as operações das redes em operações da luz sobre meios óticos, e a lentidão do processo de adaptação, uma vez que os cristais refratários precisam ser expostos por um longo período de tempo e à baixa conectividade.

2.3.5 - Recentes Aplicações

Nas últimas duas décadas, as redes neurais artificiais tem surgido como um poderoso paradigma computacional. Por esta razão, muitas aplicações, práticas e teóricas, tem sido desenvolvidas nas mais diversas áreas. O assunto desenvolvido neste trabalho concentra-se na área de modelagem e controle de processos, portanto este será o principal enfoque das aplicações apresentadas.

Em 1992 HUNT *et alli* [21] demonstram uma visão geral das redes neurais no domínio da modelagem, identificação e controle de sistemas não lineares. A idéia básica e técnicas de redes são apresentadas em linguagem e notação familiar à engenharia de controle. Aplicações de uma variedade de arquiteturas de redes neurais em controle são avaliadas.

YOU & NIKOLAOU (1993) [22] estudam a aplicação de redes neurais recorrentes na modelagem de processos dinâmicos. Para representar um sistema SISO, utilizaram um CSTR onde a variável de interesse era o pH, sendo a vazão de NaOH a variável de entrada no sistema. A boa performance do modelo fornecido pela rede pode ser observada nos resultados apresentados. Além deste, um sistema MIMO, representado por um CSTR não-isotérmico foi estudado. A diferença básica entre uma rede neural *feedforward* e uma recorrente é a característica não linear estática dos neurônios apresentada pela primeira, enquanto a outra apresenta funções não lineares dinâmicas.

ROVITHAKIS & CHRISTODOULOU (1994) [23] aplicaram análise de perturbação singular para investigar propriedades de estabilidade e robustez de redes neurais. Um identificador dinâmico baseado em um modelo neural foi utilizado para corrigir erros de um modelo (equações). Testes simulados foram realizados em um motor de corrente contínua, onde a velocidade era a variável desejada.

NARENDRA & MUKHOPADHYAY (1994) [24] estudaram o problema do desacoplamento de sistemas multi-variáveis dinâmicos desconhecidos e não lineares, usando redes neurais. O objetivo principal era o desenvolvimento de uma metodologia atrativa e prática e que poderia ser teoricamente justificada. Uma metodologia para o desenvolvimento de identificadores e controladores neurais também foi discutida.

VAN DER SMAGT (1994) [25] apresentou metodologias para minimizar o tempo de treinamento de redes neurais *feedforward*. Concluiu que métodos de minimização de segunda ordem são bastante superiores com respeito ao tempo de “aprendizado”, especialmente na precisão da aproximação de funções amortecidas. Para contornar o problema de mínimos locais, baseia-se no treinamento com algoritmos híbridos, como o proposto por Gorse & Shepherd (1992).

SUDHAKAR *et alli* (1994) [26] demonstraram a efetividade do uso do algoritmo *Alopex* (Harth & Tzanakou, 1974; Harth, 1976 & Hart *et alli* 1987) como algoritmo de aprendizado para redes neurais recorrentes. Aplicaram este tipo de arquitetura no controle de submarinos autônomos. Os resultados simulados mostraram que o algoritmo proposto, e a arquitetura da rede neural, controlador do sistema, resultaram em uma excelente estratégia de controle *on-line*. Testes na robustez do controlador foram realizados, e concluiu-se que a rede é capaz de rejeitar perfeitamente as perturbações.

SUYKENS *et alli* (1994) [27] mostraram como transições entre pontos de equilíbrio podem ser realizadas por controladores neurais estáticos e dinâmicos, parametrizados por redes neurais *feedforward* ou recorrentes, com estabilidade local no ponto de equilíbrio. Os resultados partiram de simulações de um pêndulo invertido. O trabalho demonstra que controladores não lineares, parametrizados por redes neurais, são capazes de realizar tarefas de controle complexas.

Em 1995, RITA [28] utilizou uma rede neural *feedforward*, com o algoritmo de treinamento *Backpropagation*, como modelo um tanque e dois tanques acoplados. Utilizou um controlador baseado em um critério de desempenho quadrático para o controle dos sistemas. Testes experimentais demonstraram bons resultados, com respeito à alterações de *setpoint* e perturbações tipo degrau na carga.

CHEN & KHALIL (1995) [29] 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*.

HEMERLY & NASCIMENTO (1995) [30] usaram uma rede neural para ajuste dos parâmetros do controlador PI para plantas desconhecidas. Resultados simulados e experimentais são apresentados para demonstrar a eficiência do método que depende apenas da saída da planta para o seu ajuste.

ASSAF *et alli* (1996) [31] usaram uma rede neural *feedforward* na previsão de possíveis situações de disparo, em processos altamente exotérmicos. A rede neural fornece posições e temperatura de prováveis pontos quentes, a partir de informações como fluxo de reagente, temperatura e pressão. Os resultados foram comparados com um modelo fenomenológico, representando bem a dinâmica de um reator de oxidação de etileno, para vários tipos de perturbações.

DONG *et alli* (1996) [32] estudaram a performance de redes neurais na modelagem e otimização de processos em batelada. Dois foram os casos de estudo, um processo de polimerização semibatelada e um reator biológico batelada.

SRIDHAR & SEAGRAVE (1996) [33] propuseram uma técnica para modelagem de processos químicos baseada em redes neurais sobrepostas, ou seja, várias redes constituindo um modelo. O trabalho propõe que esta metodologia, inspirada na técnica proposta por Wolpert, resulta melhores predições.

Em 1996, MAZZUCCO [34] aplicou uma rede neural *feedforward* como modelo de um controlador quadrático no controle de um sistema de nível e outro de temperatura. Em ambos os casos duas foram as redes utilizadas, a fim de separar as dinâmicas dos sistemas. No caso do controle de nível, o tanque possuía duas seções diferentes, uma cilíndrica e outra cônica. No controle de temperatura, as dinâmicas de aquecimento e resfriamento eram diferentes. Neste último caso foi desenvolvido um sistema para o treinamento e utilização de uma rede neural para um processo com tempo morto.

VAN CAN *et alli* (1996) [35] aplicaram uma rede neural *feedforward* para modelar as imprecisões de termos conhecidos de balanços macroscópicos e identificação de dados envolvendo somente o espaço de entrada e saída

KIM *et alli* (1996) [36] aplicaram uma rede neural *feedforward* atuando como controlador de uma coluna de destilação em escala piloto. A rede neural que compunha o NFFC (*Neural Feedforward Controller*) combinada com os controladores PI e DMC, foi treinada para compensar as alterações do fluxo de entrada da alimentação.

YANG *et alli* (1997) [37] propuseram um novo método de identificação para DTIRM (*discrete-time impulse response model*) de um sistema linear a partir de dados de entrada e saída. A resposta baseada num impulso contínuo é aproximada por uma rede neural composta de funções *wavelet*, que são substituídas e desenvolvidas baseadas na teoria de análise multiresolução.

YINGWEI *et alli* (1997) [38] propuseram um algoritmo de identificação adaptativo para sistemas não lineares, utilizando uma rede neural de função de base radial (RBF). Este algoritmo aumenta ou diminui o número de neurônios da camada escondida da rede neural RBF, dependendo da contribuição de cada um na formação da saída da rede, dependendo dos dados de entrada, resultando deste modo numa mínima topologia da rede neural.

HEARN *et alli* (1997) [39] enfatizaram todas as idéias e conceitos básicos por traz de um controlador neural para manutenção de rotas (piloto automático), com algumas justificativas

matemáticas. Para tal controle foi feita uma comparação entre os controladores neurais SISO e SIMO, sendo que uma preferência pelo segundo foi demonstrada nos resultados.

KIM *et alli* (1997) [40] usaram uma rede neural recorrente dinâmica generalizada, como modelo adaptativo de uma classe de sistemas não lineares. O ajuste dos pesos da rede foi *on-line*, sem treinamento preliminar.

WANG & WAHL (1997) [41] discutiram um critério de decisão múltipla para reconstrução de imagem proveniente de projeções. Em um critério múltiplo de reconstrução, o equacionamento da rede neural tem um relacionamento direto com o problema a ser resolvido, as restrições são explicitamente apresentadas.

SANTHARAM & SASTRY (1997) [42] consideraram o problema do melhoramento do aprendizado. O objetivo é encontrar um ótimo plano de ação que maximize o desconto total esperado, acima de um horizonte infinito. Este esquema de aprendizado envolve a estimação de alguns parâmetros relevantes usando uma crítica adaptativa.

Neste capítulo foram revisadas algumas publicações sobre reatores químicos e redes neurais aplicadas ao controle de processos. No próximo será apresentada a fundamentação teórica das redes neurais, bem como o desenvolvimento matemático do algoritmo *backpropagation*.

3 - FUNDAMENTAÇÃO DAS REDES NEURAIIS

Neste capítulo será apresentado um detalhamento da rede neural utilizada. Serão efetuadas algumas definições a respeito de uma rede neural, a fim de melhor entender as nomenclaturas, algoritmos e metodologias necessárias à aplicação desta à modelagem de sistemas dinâmicos.

3.1 - Definições

Uma rede neural artificial é um sistema de processamento de informações que possui características em comum com o sistema nervoso biológico. As redes neurais artificiais tem sido desenvolvidas como generalizações de modelos matemáticos dos neurônios biológicos, baseados nestas hipóteses:

1. O processamento de informações ocorre em elementos simples, chamados neurônios.
2. Os sinais são passados entre os neurônios através de conexões.
3. Cada conexão tem um peso associado.
4. Cada neurônio aplica uma função de ativação para determinar o sinal de saída.

Os pesos são os elementos detentores das informações acumuladas por uma rede neural. Conclui-se, portanto, que a determinação dos pesos é uma das fases mais importantes na implementação de uma rede neural. A obtenção dos pesos é feita na fase denominada treinamento.

Os dados apresentados à rede, a fim de se efetuar o “aprendizado”, são denominados padrões.

A fase de treinamento ou aprendizagem compreende a apresentação de um grupo de padrões à rede neural, onde um algoritmo de otimização é utilizado na obtenção dos pesos. O treinamento pode ser supervisionado ou não-supervisionado. No primeiro caso, o vetor da seqüência de padrões de entrada está associado com um alvo, que é o vetor da saída desejada. No segundo, não existe o chamado alvo, somente é apresentado a seqüência de padrões. A rede irá produzir um vetor representativo, pesos, para cada saída formada.

Os neurônios artificiais são unidades que processam as informações pela utilização de uma função de transferência, ou função de ativação. Estes neurônios estão, geralmente, dispostos em camadas, onde cada neurônio de uma camada liga-se à cada um da camada seguinte. Desta forma um neurônio de uma camada (L+1) recebe as informações de todos os neurônios da camada L.

Podemos, portanto, estabelecer uma relação entre os neurônios biológicos e os artificiais, conforme a tabela 1, abaixo.

Tabela 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

3.2 - Arquitetura

Para se entender melhor a completa arquitetura de uma rede neural, dividiu-se sua apresentação em diferentes níveis de estruturas:

- ◆ **Micro-Estrutura** - Quais são as características de cada neurônio na rede?
- ◆ **Meso-Estrutura** - Como é organizada a rede?
- ◆ **Macro-Estrutura** - Como podem redes diferentes ser colocadas juntas para executar diferentes tarefas ou problemas de grande escala?

3.2.1 - Micro-Estrutura Neural

Neste caso, é considerada a operação de um neurônio artificial típico, o qual recebe um conjunto de sinais de entrada de outros neurônios. É então computada a média ponderada entre os sinais de entrada e os pesos das conexões. O resultado obtido é aplicado à função de ativação, sendo o resultado final, chamado de ativação do neurônio.

Os modos em que neurônios artificiais diferem um do outro, e que podem ser modificados para nova performance incluem: mudança da função de transferência e adição, novos parâmetros ou funções para a rede, tais como, *bias*, ganho, e limiares adaptativos. O *bias* é uma entrada adicional que pode ser acrescentada ao neurônio artificial, não proveniente de nenhum outro neurônio. Seu peso é ajustável pelo aprendizado, como qualquer outro peso de conexão.

As funções de ativação tipicamente utilizadas nas redes neurais artificiais mais conhecidas são apresentadas na tabela 2, e os gráficos representando estas funções na figura 3.

Tabela 2 - Funções de ativação mais empregadas.

Nome da função	Uso / Características	Equação
Linear	Usada tipicamente para <i>buffers</i> de entrada e saída de dados. Utilizada também em BSB e Hopfield.	$f(x) = x$
Sinal	Utilizada em paradigmas como <i>Perceptron</i> .	$f(x) = \begin{cases} +1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \end{cases}$
Passo	Pode ser usada no lugar da função de transferência <i>Perceptron</i> , ou em BAM baseada em 0.	$f(x) = \begin{cases} +1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases}$
Hopfield/BAM	Usadas em redes Hopfield e BAM.	$f(x) = \begin{cases} +1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \\ \text{inalterado} & \text{se } x = 0 \end{cases}$
BSB, ou de limiar lógico	Linear entre os limites inferior e superior, prendendo os valores de saída ao limite, caso seja excedido.	$f(x) = \begin{cases} -K & \text{se } x \leq -K \\ x & \text{se } -K < x < +K \\ +K & \text{se } x \geq +K \end{cases}$
Rampa, ou linear de limites rígidos	Linear entre os limites 0 e 1, prendendo os valores de saída ao limite, caso alguns destes seja excedido.	$f(x) = \begin{cases} 0 & \text{se } x \leq 0 \\ x & \text{se } 0 < x < 1 \\ +1 & \text{se } x \geq +1 \end{cases}$
Logística	Função de transferência sigmoideal usada tradicionalmente em redes <i>feedforward</i> de aprendizagem <i>backpropagation</i> . Usada também em redes Hopfield, BAM e BSB.	$f(x) = \frac{1}{1 + e^{-x}}$
Hiperbólica	Utilizada muitas vezes em lugar da função logística.	$f(x) = \tanh x = \frac{1 - e^{-2x}}{1 + e^{-2x}}$

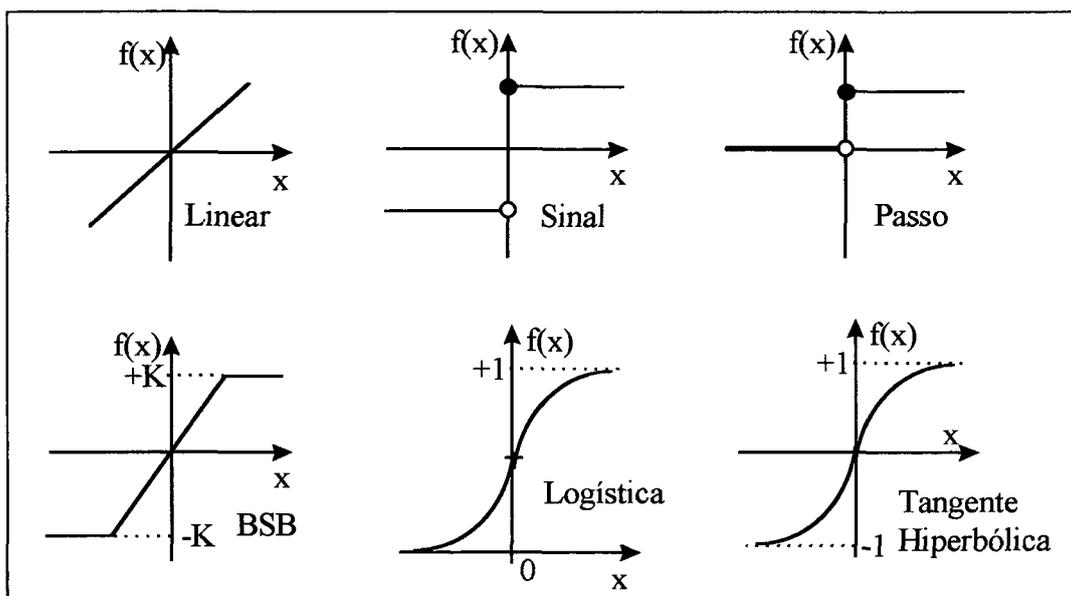


Figura 3 - Gráficos de algumas funções de transferência mais empregadas.

3.2.2 - Meso-Estrutura Neural

O nível meso-estrutural de uma rede neural é o nível em que a organização e arranjo físico dos elementos de processamento na rede é considerado. Para caracterizar grupos de neurônios, devem ser considerados:

- ◆ O número de camadas da rede;
- ◆ O número de neurônios por camada;
- ◆ O tipo de conexões (*forward*, *backward*, lateral);
- ◆ O grau de conectividade entre os elementos de processamento: um a um, plena, esparsa, randômica, etc.

Considerando as distinções especificadas, podemos identificar cinco classes estruturalmente diferentes, como ilustrado na figura 4:

1. Multicamadas, redes *feedforward*.
2. Camada simples, redes conectadas lateralmente.

3. Bicamadas, redes *feedforward*/feedback.
4. Multicamadas, redes cooperativas.
5. Redes híbridas.

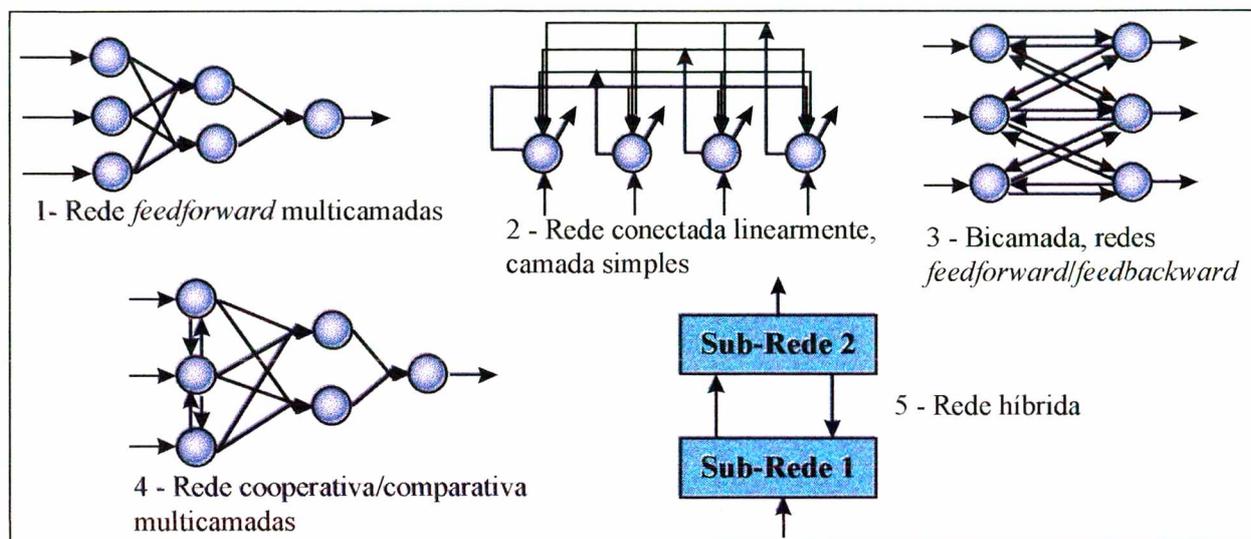


Figura 4 - Tipos de meso-estruturas de redes neurais.

3.2.3 - Macro-Estrutura Neural

Quando não se conhece qual arquitetura é adequada para um problema ou tarefa complexa, pode ser necessário desenvolver um sistema para acoplar e interagir redes neurais. Existem dois tipos de macro-estruturas: redes fortemente acopladas e suavemente acopladas. O primeiro pode ser tratado como uma rede simples, ou seja, cada rede integrada representa um novo neurônio para a formação da nova rede. Esta é criada pela fusão de duas ou mais redes dentro de uma nova estrutura. Este processo produz uma nova rede híbrida. As estruturas suavemente acopladas conectam redes que guardam suas características originais.

A primeira dificuldade na modelagem de uma macro-estrutura é entender e decompor o problema de aplicação em uma forma modular. As maiores dificuldades no modelamento como um sistema de redes incluem especificar o número, tipo, e tamanho das redes componentes, e especificar seus padrões de conexões. Isto pode ser verificado na figura 5.

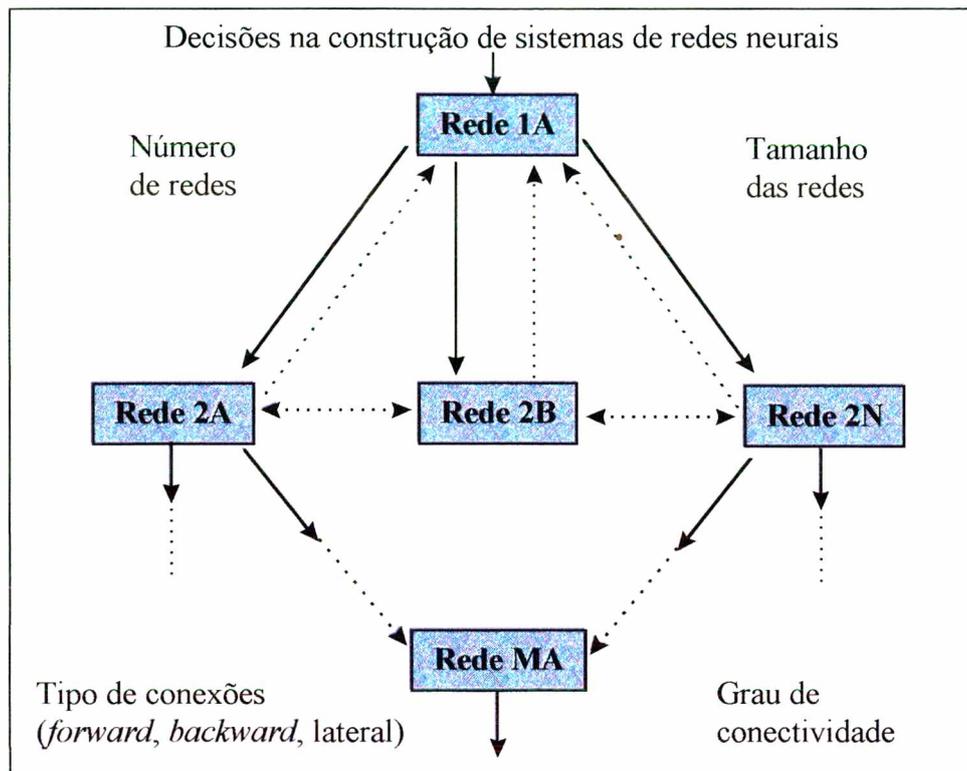


Figura 5 - Características modificáveis de um sistema de redes neurais.

3.3 - A Rede Neural Utilizada

Dentre todas as redes apresentadas na seção anterior, a rede *feedforward* multicamadas (FANN- *Feedforward Artificial Neural Network*), figura 4, foi a adotada neste trabalho. Este tipo de rede está entre as mais estudadas, sendo de fácil compreensão, com uma simples definição matemática. Uma FANN, pode ser observada em detalhes na figura 6, onde cada camada possui um propósito, sendo estes:

- ◆ **Camada de Entrada:** responsável apenas por receber os padrões e transferi-los para a segunda camada. Não é realizado processamento.
- ◆ **Camada(s) intermediária(s):** responsáveis por processar as informações da camada anterior. 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. O resultado obtido é então passado à camada seguinte.

- ◆ **Camada de Saída:** na camada de saída são realizados os últimos processamentos e os resultados finais são obtidos.

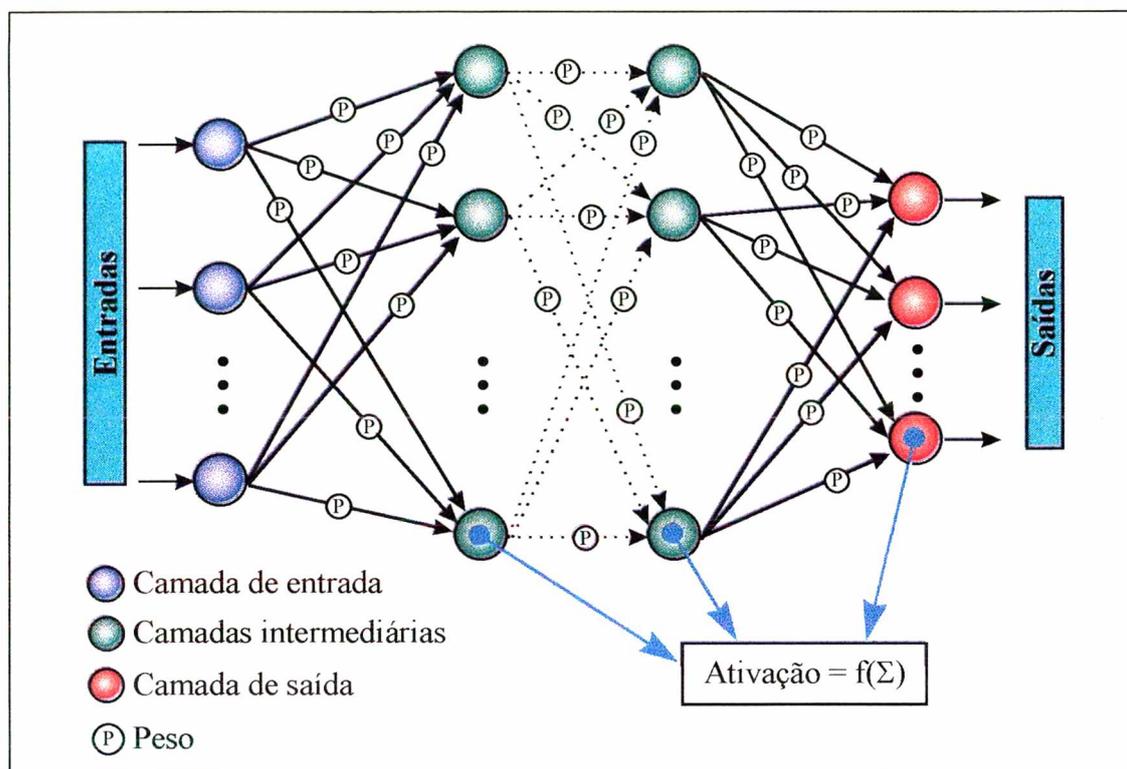


Figura 6 - Rede neural *feedforward* multicamadas.

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) [19] mostraram que uma FANN com três camadas pode aproximar qualquer função contínua com qualquer grau de precisão, desde que existam neurônios suficientes na camada intermediária.

3.3.1 - Definição Matemática de uma Rede Neural *Feedforward*

Matematicamente, cada neurônio é um simples elemento processador. 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 sigmoideal com variações no número de neurônios na camada intermediária foi empregada.

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.

A figura 7, 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.

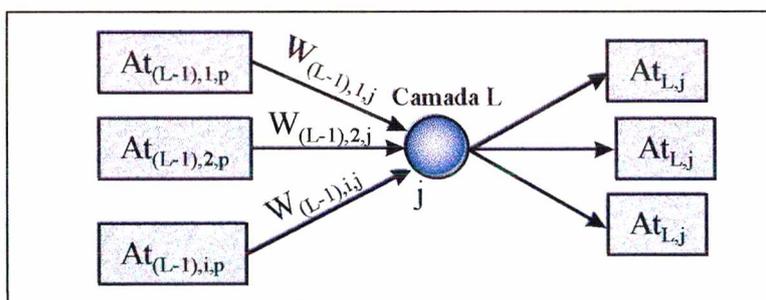


Figura 7 - Exemplo de um neurônio j na camada L.

A ativação calculada para um neurônio da camada L fica definida conforme a equação 1, abaixo:

$$At_{L,j,p} = f(x_j) \text{ com } j = 1, \dots, NNE \quad (1)$$

com:

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

e

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

Onde:

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

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

$b_{L,j}$ = Bias de cada neurônio da camada L.

Definida a função de ativação utilizada e a estrutura da rede (*feedforward*), a rede neural está completa. O passo final é definir o método para a determinação dos 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*.

3.3.2 - O Método *Backpropagation*

O método *Backpropagation*, ou Regra Delta Generalizada (Rumelhart *et alli*, 1986) [18] 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, dedução, e geralmente produz bons resultados.

O fluxograma do algoritmo *backpropagation* pode ser melhor analisado pela figura 8, abaixo:

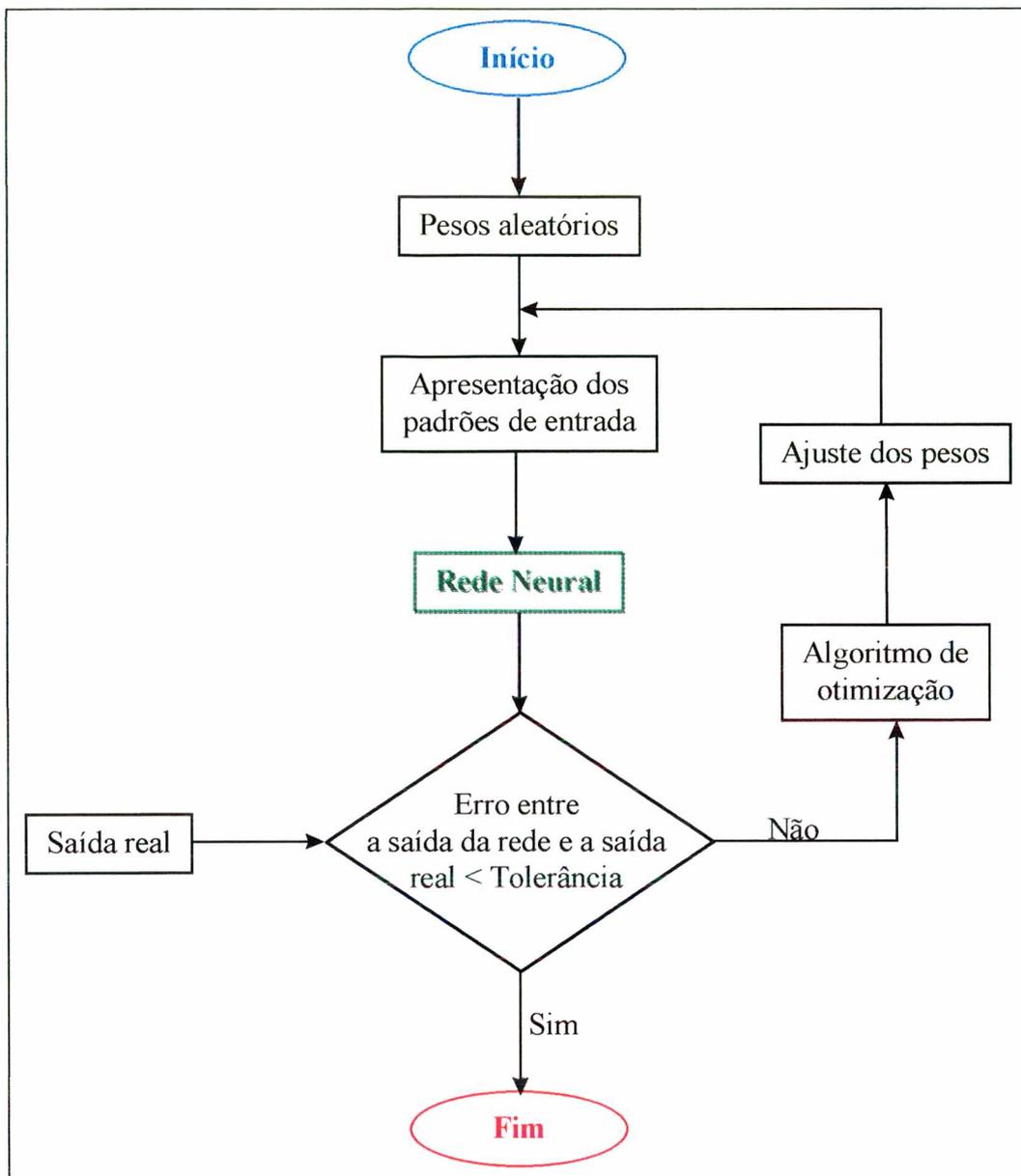


Figura 8 - Fluxograma do algoritmo *backpropagation*.

3.3.3 - Equacionamento do Algoritmo *Backpropagation*

O objetivo da fase de aprendizagem é gerar um grupo de pesos que permitam à rede neural caracterizar um sistema, portanto 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:

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

Portanto, o erro por padrão será dado por:

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

E o erro global será:

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

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.

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 :

$$W_{(L-1),i,j}^{it+1} = W_{(L-1),i,j}^{it} + \eta \Delta W_{(L-1),i,j}^{it} \quad (7)$$

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

Como ΔW é a direção da alteração dos pesos, ou seja, o negativo do gradiente, têm-se:

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

Como a rede neural utilizada neste trabalho é composta de 3 camadas, sendo a primeira camada composta de 4 neurônios 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:

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

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

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

onde :

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

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

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

A partir da equação 12, pode-se obter:

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

Substituindo-se a equação 15 na equação 14 tem-se:

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

onde :

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

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

Sabendo-se que a definição de $At_{2,j,p}$ é análoga à definição de $At_{3,j,p}$, demonstrada pela equação (12), obtém-se a equação (19) a partir da equação (18):

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

Finalmente, substituindo a equação 19 na equação 16, e a equação 16 na equação 11 obtém-se a forma para o ajuste dos pesos entre a primeira e a segunda camadas, dado pela equação (20):

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

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

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

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

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

onde:

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

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

Substituindo as equações 15 e 12 na equação 25, obtém-se:

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

Finalmente substituindo a equação 26 na equação 23, obtém-se uma expressão para o ajuste dos pesos entre as 2^a e 3^a camadas:

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

A associação do algoritmo *Backpropagation* ao *Steepest descent* tem uma aplicabilidade extensa, desde modelagem de processos até reconhecimento de padrões visuais e sinais de sonar. Este é, sem dúvida, um dos algoritmos mais utilizados, e isto deve-se principalmente à sua simplicidade, velocidade da forma final, o paralelismo inerente das redes neurais, boa capacidade de modelagem e a popularidade dos métodos. Em contrapartida esta metodologia tem sérios problemas de convergência por mínimos locais, é lenta no período de treinamento e possui uma certa dificuldade em se estabelecer o número de neurônios que cada camada intermediária deve conter.

Neste capítulo foi apresentada uma revisão sobre redes neurais, dando-se maior ênfase à rede neural *feedforward*, utilizada no desenvolvimento deste trabalho, e também ao algoritmo *backpropagation*, utilizado no treinamento desta rede neural. Em seqüência, será detalhado o laboratório e a planta onde se realizaram os experimentos, juntamente com as metodologias empregadas.

4 - MATERIAIS E MÉTODOS

Neste capítulo será apresentado uma visão geral do Laboratório de Controle de Processos, detalhando os equipamentos, *softwares* e metodologias utilizadas no desenvolvimento deste trabalho.

4.1 - O Laboratório de Controle de Processos

O Laboratório de Controle de Processos (LCP) está instalado no Departamento de Engenharia Química (ENQ), que faz parte do Centro Tecnológico (CTC) 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, as quais concentram-se em testes com controle de nível em tanques simples, acoplados e cônico-cilíndrico e controle de temperatura em reatores e trocadores de calor. Neste trabalho, o sistema utilizado para teste das estratégias de controle foi um reator encamisado, sendo nele realizados testes com e sem reação química, neste caso atuando como trocador de calor.

Na figura 9 é apresentada uma visão de parte do laboratório onde foram realizados os experimentos. Neste sala encontram-se a unidade experimental utilizada, além de outras, sendo o controle efetuado de uma outra sala, na qual microcomputadores estão instalados.



Figura 9 - Visão da sala onde encontram-se parte das unidades experimentais do LCP.

Como a unidade utilizada para a realização dos experimentos foi o reator encamisado, visto no canto inferior esquerdo da figura 9, a este serão dadas maiores informações.

4.2 - Reator e Acessórios

O reator químico é constituído de dois cilindros concêntricos de aço inox 316. O tubo interno possui um diâmetro de 200 mm e o externo 300 mm, e volumes respectivos de 9,3 litros e 16,5 litros. As estratégias de controle propostas no controle da temperatura compreendem a manipulação de uma corrente de água fria e uma corrente de vapor. O reator é isolado com poliestireno expandido para evitar a troca de calor entre a camisa e o meio.

O sistema foi projetado de forma que o vapor não é alimentado diretamente na camisa do reator. A água que passa pela camisa do reator 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 reator. A manipulação das duas válvulas é feita de forma independente, caracterizando o controle multivariável, e como a temperatura é a variável controlada, tem-se um sistema MISO.

A figura 10 mostra uma fotografia do reator, e nela pode-se observar alguns acessórios, como um agitador e um termopar.

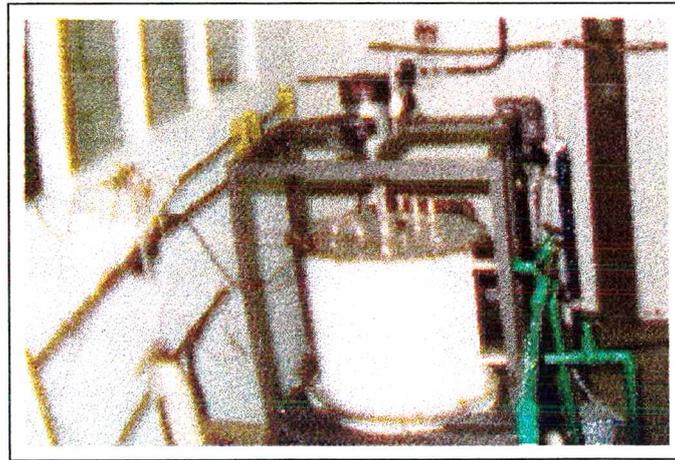


Figura 10 - Fotografia do reator.

Na figura 11 mostra-se o esquema do reator. Na figura 12 o sistema completo da malha de controle está detalhado. Nesta figura pode-se observar a configuração do sistema, e devido a esta configuração tem-se a incidência de tempo morto. O tempo morto total é a soma dos tempos mortos incidentes no reservatório, no trocador de calor a placas, e no próprio reator. A constante de tempo do sistema, sem reação, é de aproximadamente 1000 s, e o ganho de 32 °C/Volt. Nesta figura é observado também o agitador, o qual é da marca Fisatom, com velocidade de rotação de 90 a 6300 rpm. Utilizou-se pás de agitação cilíndricas. Para os testes com reação química, foi utilizada uma bomba peristáltica, cuja função era alimentar o reator com um dos reagentes, já que a reação foi realizada em regime de batelada alimentada.

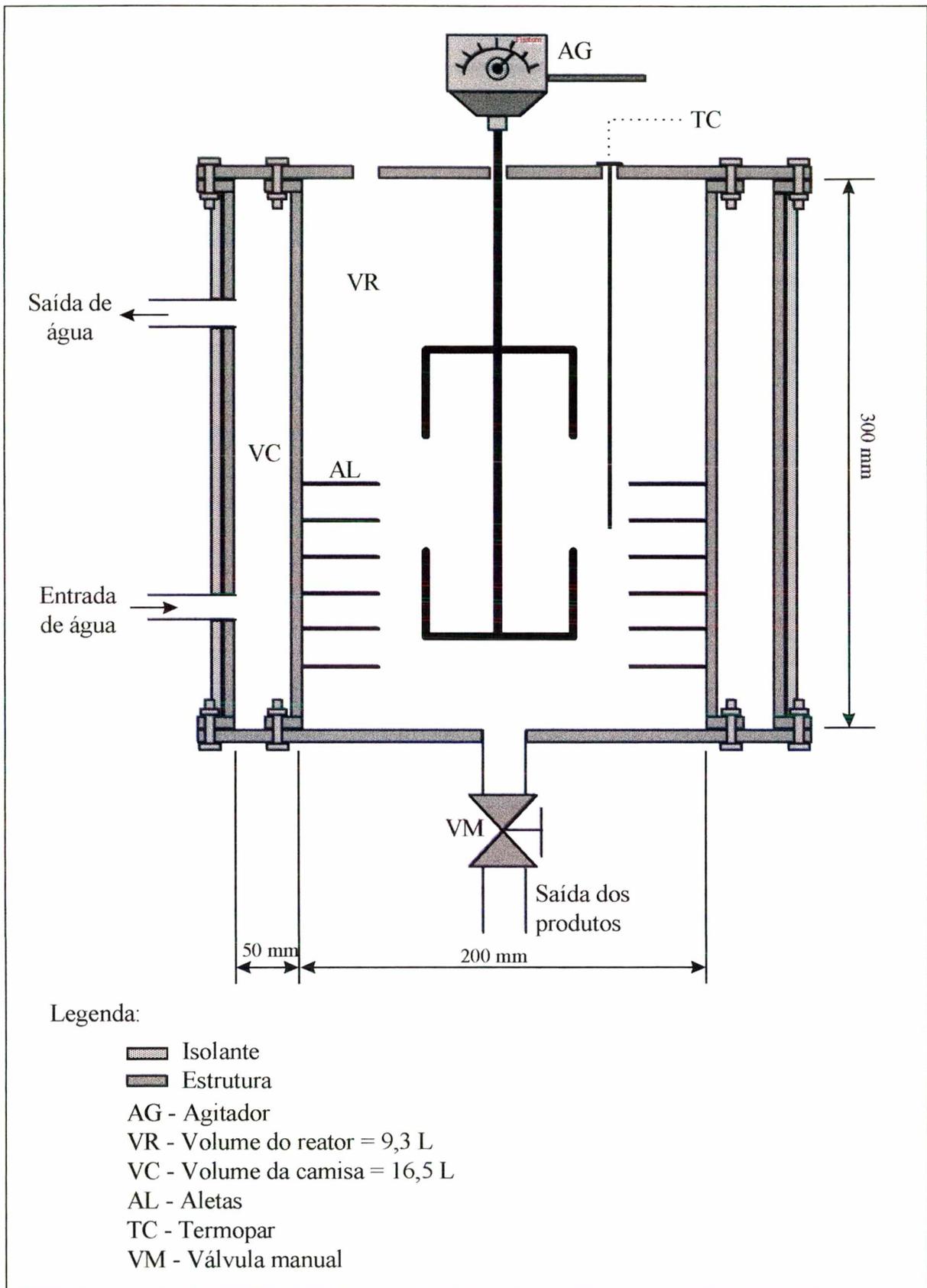


Figura 11 - Estrutura do reator.

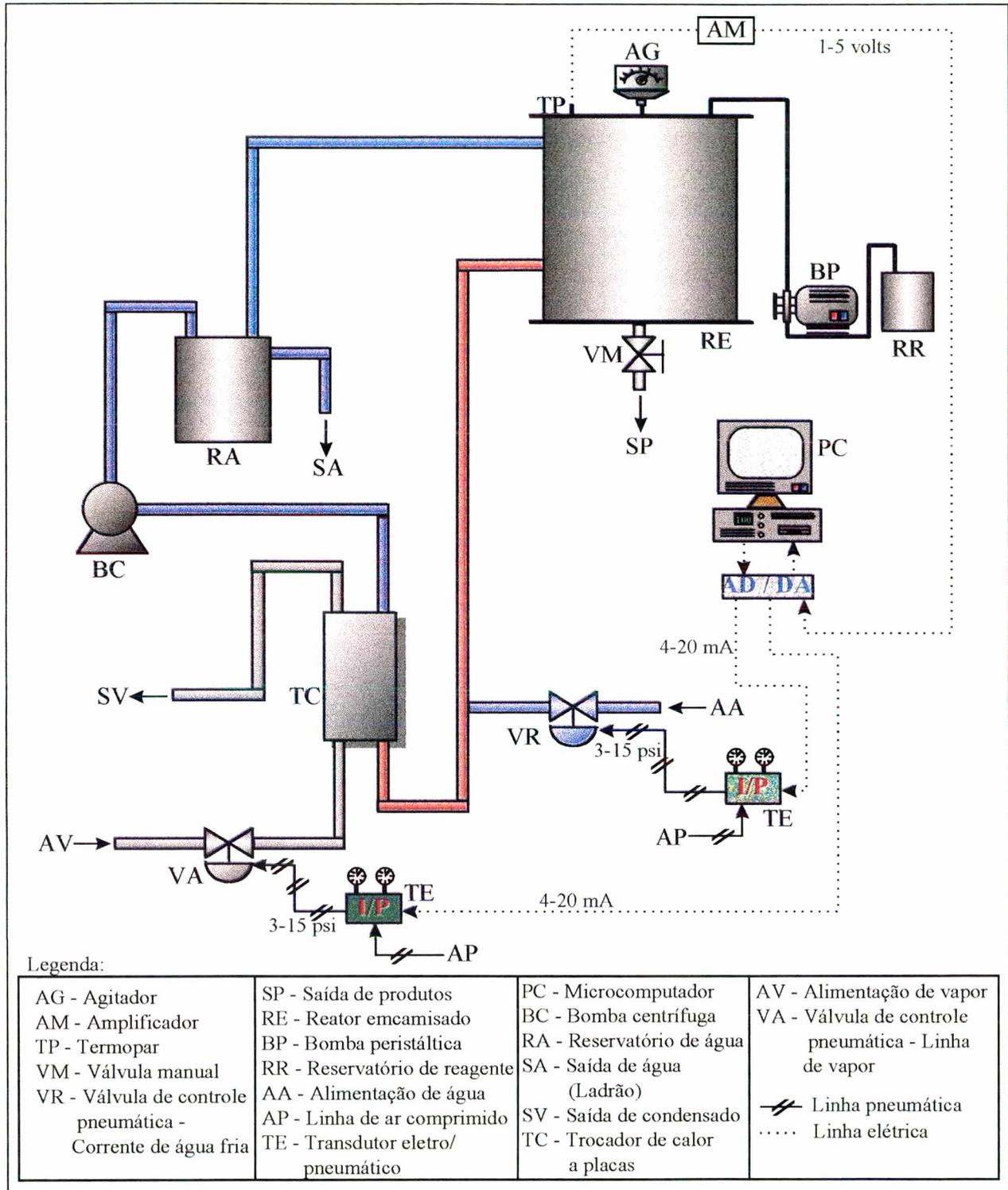


Figura 12 - A planta piloto.

Foram realizados testes sem reação química, e nestes, certamente que a bomba peristáltica, apresentada na figura 12, não é utilizada. Neste caso, o reator atua como um tanque encamisado agitado.

O vapor é fornecido por uma caldeira elétrica, que devido ao seu sistema de controle, fornece vapor a uma pressão que pode variar de 3 a 6 Kgf/cm². Esta variação, que constitui uma perturbação externa, dificulta bastante a ação do controlador implementado. Outra grande fonte de perturbação externa é a pressão da água de refrigeração. Há também o problema de flutuações na tensão da rede elétrica, que provoca mau funcionamento da caldeira e ruído nas medidas (a rede não é estabilizada).

4.3 - Sistemas de Controle

Conforme mencionado anteriormente, foram realizados experimentos sem reação química, e experimentos com reação química. Uma terceira etapa consistiu no desenvolvimento de uma técnica para ajuste automático de um dos parâmetros do controlador utilizado. A etapa dos experimentos sem reação química foi utilizada, principalmente, para ajustar os parâmetros do controlador, além de testar sua eficiência. Serão analisados com maiores detalhes os experimentos com reação e os realizados com a técnica de ajuste do controlador.

4.3.1 - Sistema de Controle para os Experimentos com Reação Química

Neste sistema, e para os demais, a variável controlada foi a temperatura interna do reator, e as variáveis manipuladas foram a vazão de vapor e de água de refrigeração, conforme mostra a figura 12.

A reação química utilizada nos experimentos foi a do Tiosulfato de Sódio ($\text{Na}_2\text{S}_2\text{O}_3$) com o Peróxido de Hidrogênio (H_2O_2). A concentração da solução de $\text{Na}_2\text{S}_2\text{O}_3$ utilizada foi de 400 g/l, obtida pela dissolução da substância sólida, em qualidade p.a. (Quimibrás Indústrias Químicas S.A.), em água destilada. O Peróxido de Hidrogênio, H_2O_2 , também conhecido como água oxigenada, foi utilizado a uma concentração de 150 g/l, preparada a partir de uma solução p.a. (Vetec Química Fina Ltda) a 30% em peso.

A reação, como mencionado anteriormente, foi realizada em regime de batelada alimentada. O reator era previamente abastecido com a solução de Peróxido de Hidrogênio, e então a solução de Tiosulfato de Sódio era alimentada ao reator com a bomba peristáltica. A opção pela operação do reator em regime de batelada alimentada deve-se a alta exotermicidade da reação, pois devido a capacidade de troca térmica limitada do reator, ocorreria seu disparo.

A reação principal entre o Tiosulfato de Sódio e o Peróxido de Hidrogênio é apresentada pela equação (28) abaixo:



Esta é uma reação altamente exotérmica, com um $\Delta H = -6,82 \cdot 10^5 \text{ KJ/Kmol}$, portanto ideal para o propósito de se controlar a temperatura de um reator deste tipo. A temperatura na qual a reação foi realizada é de 35 °C, conforme BOLZAN [9] (1991).

A malha de controle, considerando o sistema com reação química é mostrado na figura 13.

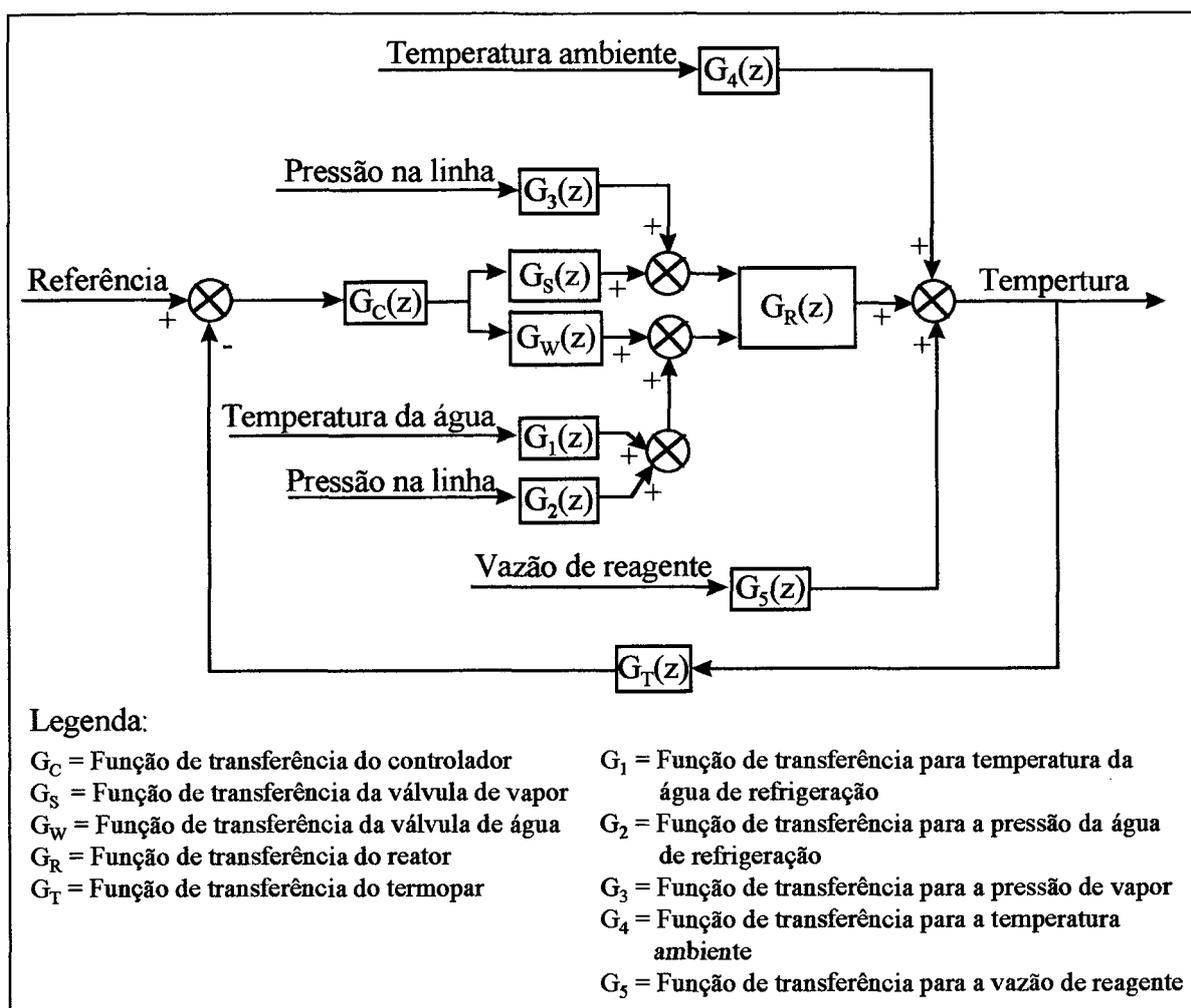


Figura 13 - Diagrama de blocos da malha de controle.

4.3.2 - Estratégia para Ajuste do Controlador

As ações de controle são obtidas pela minimização de uma função objetivo quadrática, de acordo com a equação (29), abaixo:

$$J = \sum_{i=1}^H \left[y_{pred(k+i)} - y_{ref(k+i)} \right]^2 + \lambda_1 \sum_{j=1}^{NU} \left[\Delta UA_{(k+j-1)} \right]^2 + \lambda_2 \sum_{j=1}^{NU} \left[\Delta UR_{(k+j-1)} \right]^2 \quad (29)$$

Onde:

H = Horizonte de previsão

NU = Horizonte de controle

λ_1 = Fator de penalização da ação de controle de aquecimento

λ_2 = Fator de penalização da ação de controle de resfriamento

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)

UA_k = Valor da ação de aquecimento a ser implementado

UR_k = Valor da ação de resfriamento a ser implementado

A variável y_{ref} é a trajetória de referência, definida pela equação (30):

$$y_{ref(k+i)} = \alpha y_{ref(k+i-1)} + (1-\alpha)y_{SP} \quad i = 1 \dots H \quad (30)$$

Para que o controlador apresente uma boa performance é necessário que seus parâmetros estejam devidamente ajustados. No controlador utilizado, o parâmetro de maior influência é α , observado na equação (30), podendo variar de 0 a 1, sendo que maiores valores tornam o controlador mais conservativo.

Valores elevados de α eliminam *overshoot*, mas causam um elevado *offset* e em contrapartida, baixos valores de α eliminam o *offset*, mas permitem a ocorrência de *overshoot*. O ajuste deste parâmetro para que o controlador tenha uma boa performance não é uma tarefa simples, principalmente quando a técnica utilizada é a de tentativa e erro. Portanto desenvolveu-se um algoritmo, onde utiliza-se algumas regras para o ajuste automático deste parâmetro. Somente é necessário atribuir um valor de modo que não ocorra *overshoot*, e outro que elimine o *offset* no estado estacionário. Estes valores são facilmente identificados na maioria dos processos.

O esquema para o ajuste pode ser observado na figura 14.

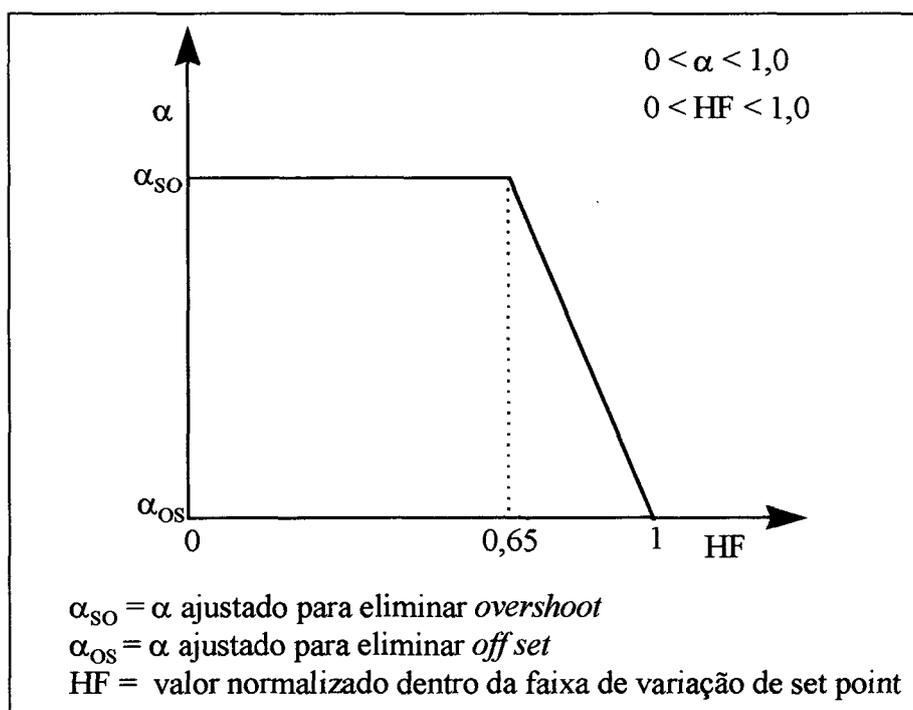


Figura 14 - Forma de ajuste de α .

A variável HF é definida de acordo com a equação (31).

$$HF = \frac{|y - y_{SP}(Anterior)|}{|y_{SP} - y_{SP}(Anterior)|} \quad (31)$$

onde y é a variável controlada, neste trabalho, a temperatura e $y_{SP}(Anterior)$ é o *setpoint* anterior.

A definição de uma variável normalizada na faixa de variação do *setpoint* é adequada para a formulação do algoritmo de ajuste de α . Nota-se pela figura 14 que α permanece constante até $HF = 0,65$, portanto até 65% da trajetória entre um *setpoint* e outro, para só então decrescer conforme uma relação linear, de acordo com a equação (32). O valor obtido para HF é empírico, sabendo-se no entanto, que pode variar de acordo com a dinâmica do sistema, sendo inversamente proporcional à constante de tempo do processo.

$$\alpha = \alpha_{OS} + \frac{(\alpha_{SO} - \alpha_{OS})}{0,35} \cdot (HF - 1) \quad (32)$$

A regra de ajuste é desta forma a mais simples possível, sendo de fácil cálculo e apresentando grande versatilidade. Esta regra maximiza a velocidade de transição entre os *setpoints*, eliminando o *overshoot* e o *offset*. Os valores de α_{SO} e α_{OS} foram obtidos experimentalmente. O valor de α_{OS} utilizado foi de 0.2.

O algoritmo que mostra em detalhes como a regra de ajuste foi implementada é mostrado na figura 15. O prévio conhecimento do sistema permitiu que fossem utilizados três valores para α_{SO} . Observou-se que para variações maiores que 6 °C, no caso do sistema representado pela figura 12, o valor de α que proporcionaria uma transição entre os *setpoint*'s de forma a evitar-se o *overshoot* seria de 0,75. Porém para variações negativas de *setpoint*, um menor valor de α poderia ser utilizado de modo a evitar o *overshoot* e ao mesmo tempo, fazer com que o controlador seja menos conservativo. Analogamente procedeu-se o ajuste para α_{SO} em transições de *setpoint* inferiores à 6 °C. Caso não se tenha conhecimento suficiente sobre o sistema, somente um valor para α_{SO} pode ser atribuído. Neste caso deve-se adotar um valor de modo a evitar o *overshoot* em todos os casos, ou seja, o maior valor de α .

4.4 - 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, com resolução de 12 bits, 24 canais de entrada, 16 digitais e 8 analógicos e 2 canais de saída analógicos, com faixa de entrada e saída variáveis, podendo ser especificados pelo usuário via *software*. Suas funções são completamente programáveis. A placa possui ainda documentação completa para a implementação em *software*.

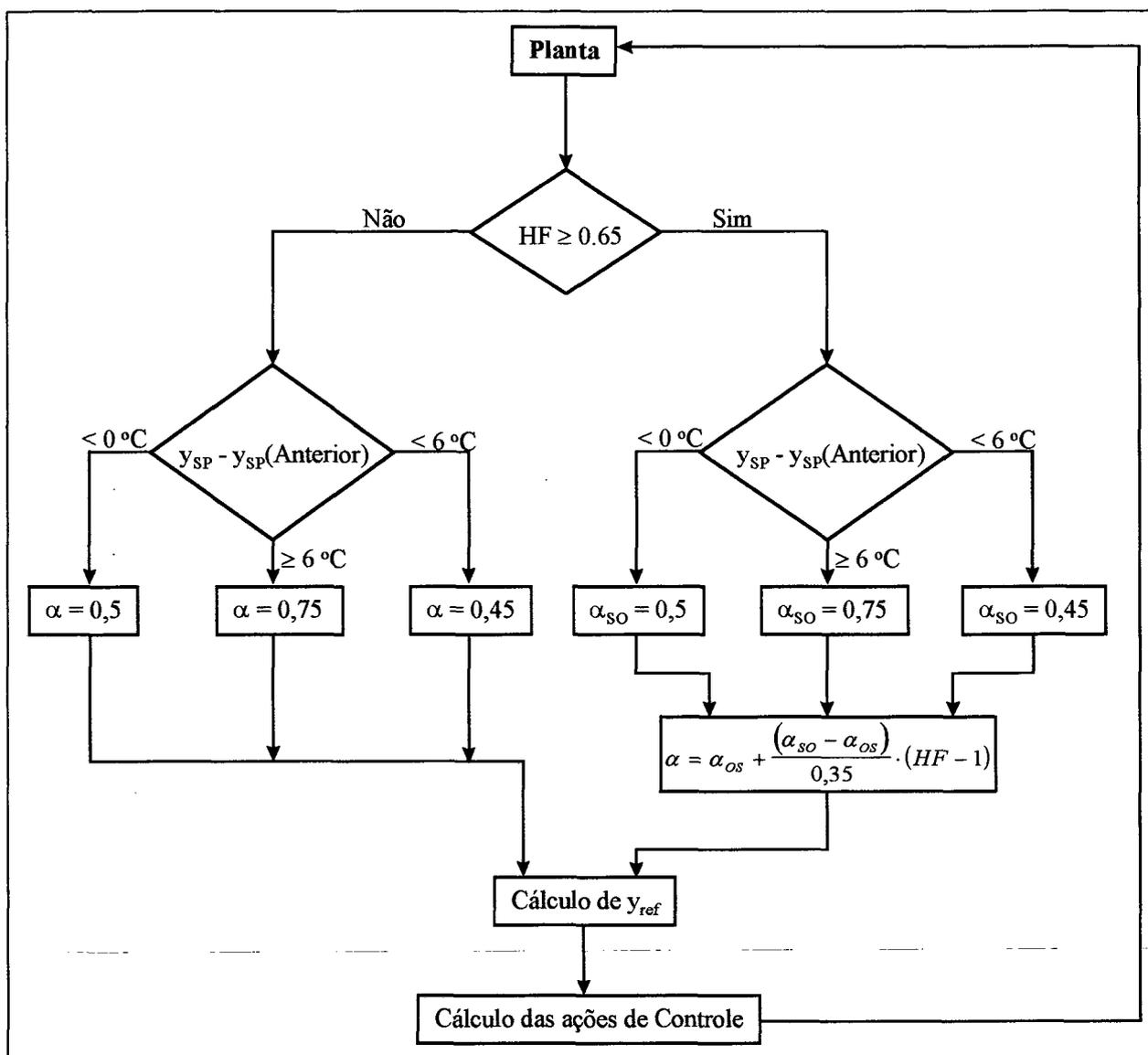


Figura 15 - Algoritmo para ajuste automático da trajetória de referência.

O *software* utilizado foi desenvolvido por MAZZUCCO [34], como parte de sua dissertação. Para atender às necessidades aplicadas a este trabalho, o *software* sofreu algumas modificações, contudo as características originais foram mantidas. A alteração básica está na inclusão do sistema de controle multivariável. O *software*, denominado *ArtificialSINAPSER Módulo de Controle*, foi desenvolvido para ambiente Windows™, e opera em tempo real. 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.

O *software*, implementado em *Object PASCAL*, conta com três opções de temporizadores, configuração de *hardware* em tempo de execução (canais de entrada e saída),

permutação entre controle PID, ANN e malha aberta, configuração do controlador neural e pesos da rede neural, etc.

Para a implementação das estratégias de controle propostas, um microcomputador 486 SX 33 MHz com 8 MBytes de memória RAM e 850 MBytes de disco rígido operando com sistema operacional MS Windows[®] 95 foi utilizado. A figura 16 apresenta um diagrama de blocos do *software* de controle e as figuras 17 e 18, respectivamente, o Painel de controle e a Tela de configurações do *software* de controle.

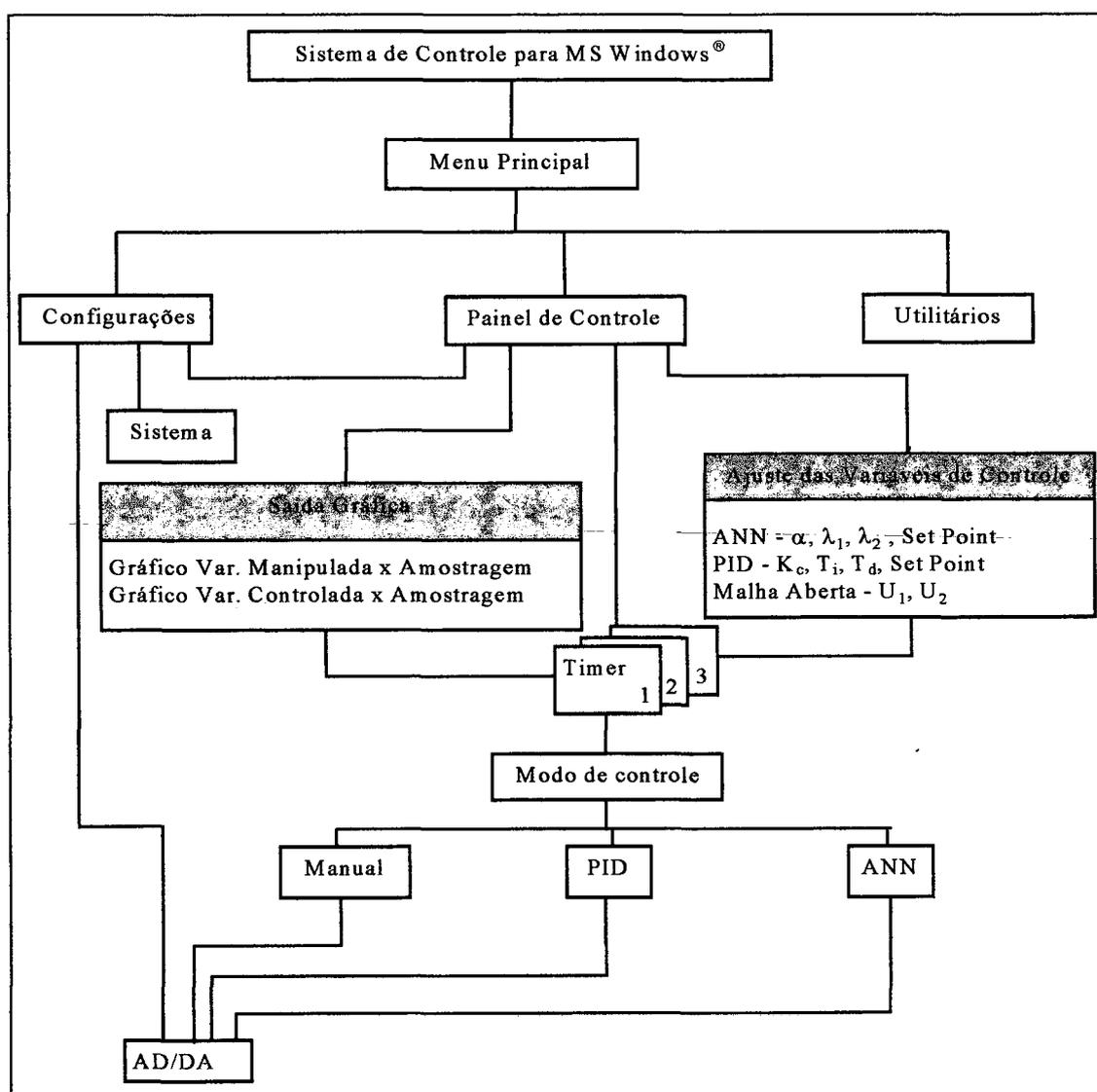


Figura 16 - Diagrama de blocos do *software* de controle.

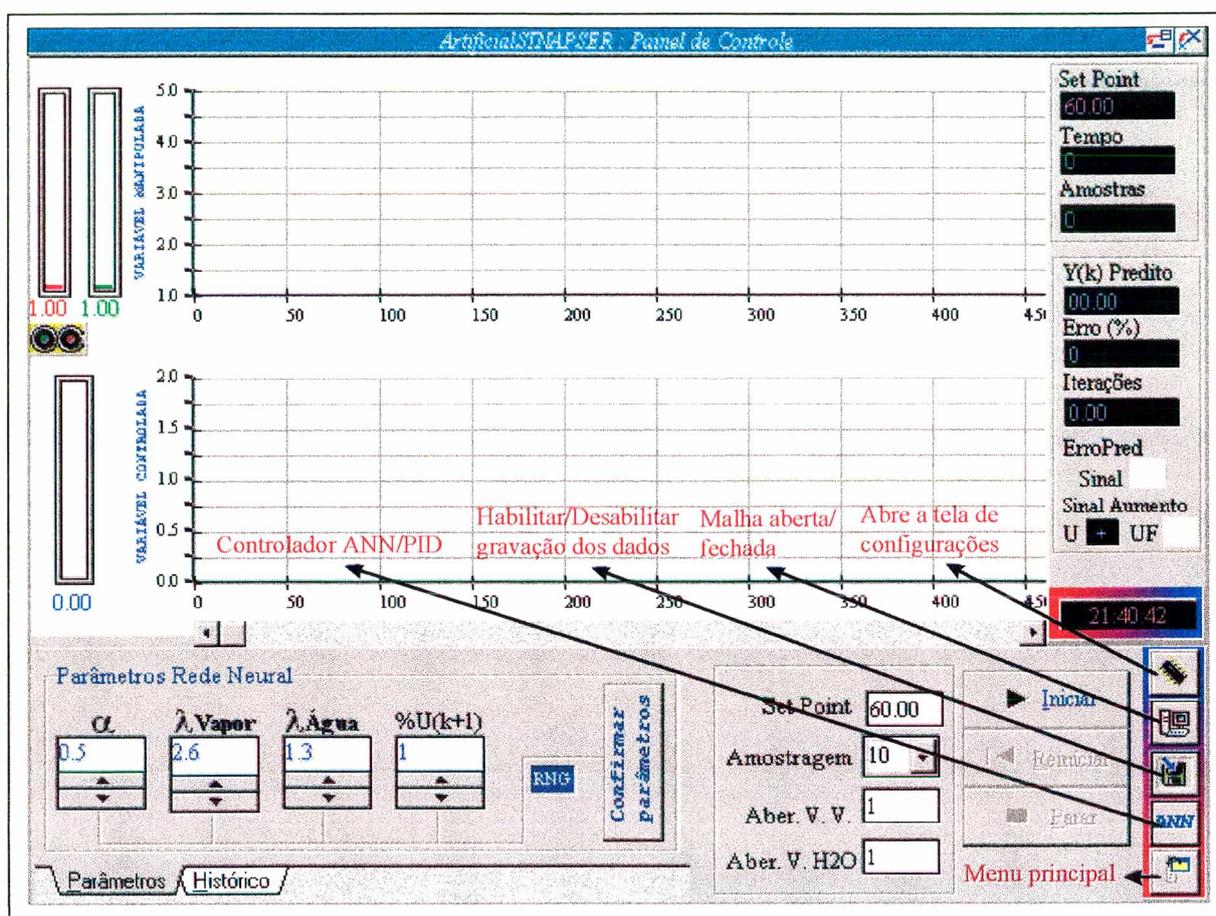


Figura 17 - Painel de controle do software utilizado.

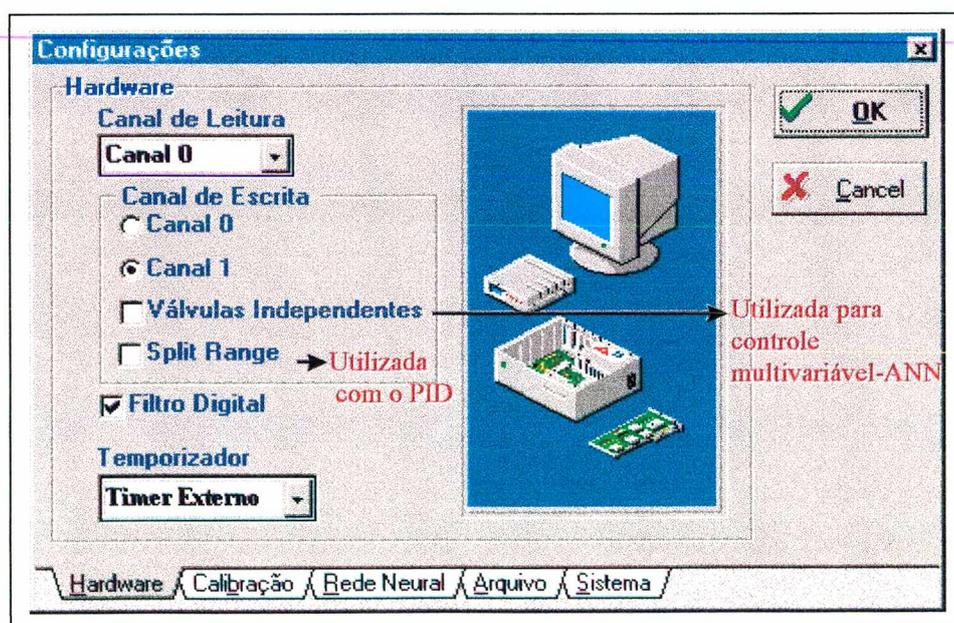


Figura 18 - Tela de configurações.

4.5 - Treinamento

O treinamento é uma das partes mais importantes 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, do número de neurônios da(s) camada(s) intermediárias(s) 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 2,1 GBytes de disco rígido foi utilizado.

Da mesma forma que o *software* de controle, este também foi desenvolvido por MAZZUCCO [34]. O *software* utilizado foi produzido para operar em sistema operacional MS Windows[®] 95. 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 organização de padrões. Isto foi feito para a utilização deste *software* neste trabalho. 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.

A tela principal do software, e a de configurações, podem ser vistas nas figuras 19 e 20, respectivamente. Na figura 19 observa-se a possibilidade de acompanhar o avanço do treinamento graficamente. Isto, além dos dados estatísticos apresentados, possibilita uma maior visualização, e portanto a capacidade de decidir quando o treinamento é suficiente.

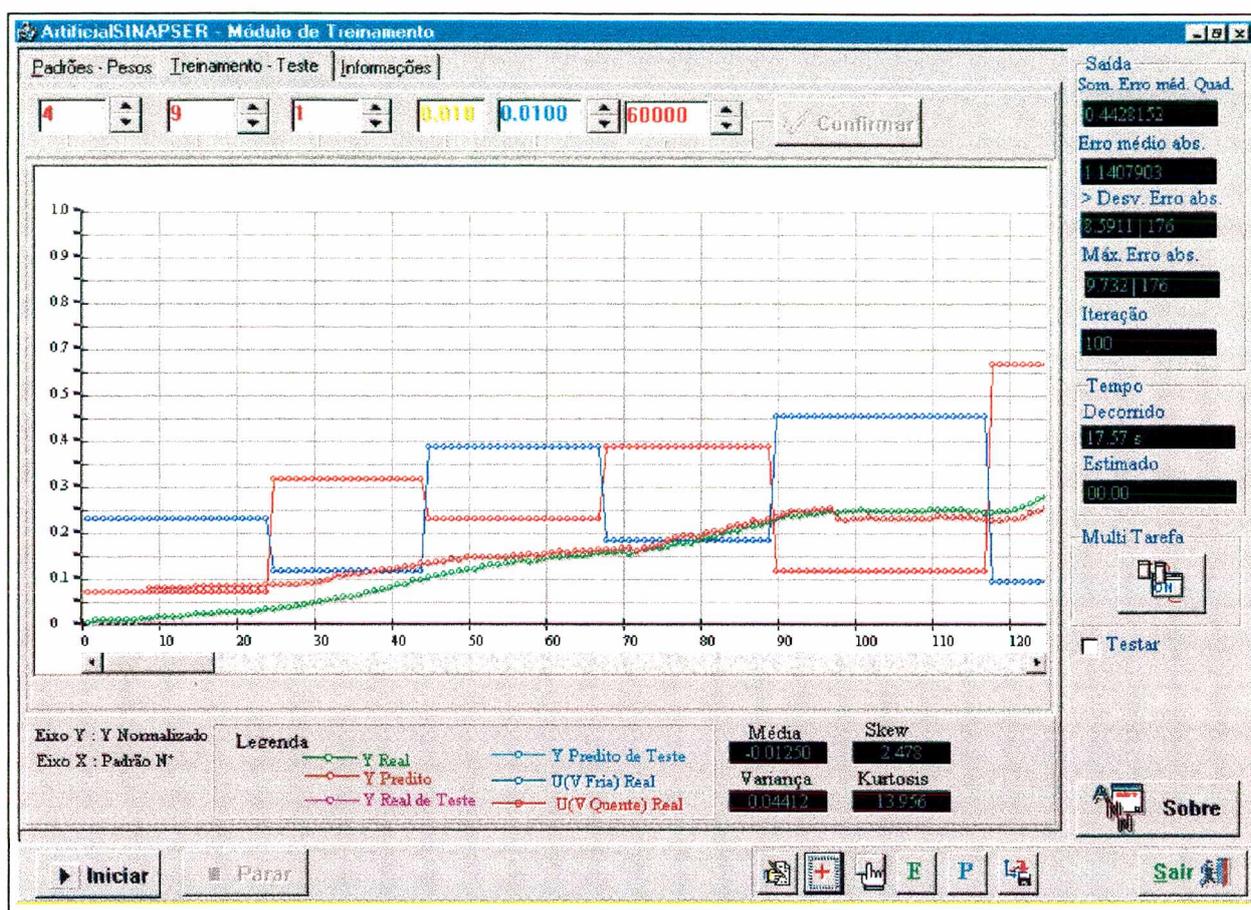


Figura 19 - Tela principal do software de treinamento.

Figura 20 - Opções de treinamento.

4.5.1 - Formação dos Padrões

A obtenção dos padrões é um dos pontos mais importantes 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 pretenda modelar. Na existência de atrasos, como tempo morto, a coleta deve ser procedida de forma a permitir a manifestação do fenômeno sobre a variável a controlar.

A Formação dos padrões de forma a eliminar o tempo morto, na modelagem e controle do processo, foi um dos principais avanços apresentado por MAZZUCCO [34]. Portanto a determinação do valor deste tempo morto, a fim de utilizá-lo na formação dos padrões, é essencial. Testes experimentais com o sistema apresentado pela figura 12, demonstraram que o tempo morto varia de acordo com o ponto de operação do sistema. Determinou-se, no entanto, que seria possível dividir o sistema em duas seções, a primeira até 60 °C com um tempo morto médio de 70 s, e outra de 60 °C até 100 °C com 50 s de tempo morto médio.

Diante deste fato, duas redes para a modelagem completa do sistema são necessárias. Pois como os intervalos de amostragem utilizados no controle foram de 10 s, a diferença entre o primeiro e segundo tempo morto é de 2 intervalos, valor significativo na formação dos padrões. Além deste fator, outro que motivou a divisão em duas redes foi o fato do grande número de padrões, se fosse utilizado somente um modelo neural.

Apesar da reação ocorrer em 35 °C, optou-se por modelar o reator em toda a faixa de operação, ou seja, até 100 °C. Isto foi necessário para demonstrar que o controlador é capaz de controlar a temperatura em toda a faixa. Utilizou-se testes sem reação química para demonstrar o controle, e também o acoplamento das duas redes neurais.

As tabelas 3, 4 e 5 ilustram o deslocamento temporal dos componentes de um padrão onde um tempo morto, por simplicidade, igual a três intervalos de amostragem é considerado. A predição de T_M intervalos de tempo futuros elimina a influência do tempo morto sobre o sistema.

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 abaixo, 3 padrões. Genericamente, um processo com um tempo morto (T_M) apresenta $(K - (2 * T_M))$ padrões.

Na tabela 3, pode-se observar que, quando as ações de aquecimento e resfriamento, UA e UR respectivamente, são tomadas, neste caso no intervalo $k=1$, somente T_M intervalos adiante terão efeito na variável de controle, em Y_4 , como visto na tabela 3. O valor Y_3 , refere-se à Y_{k-1} , usado para determinar a direção da variável controlada. O valor utilizado para comparação, ou também chamado alvo, é o valor T_M unidade adiante de Y_k , neste caso Y_4 , portanto Y_7 . Para as demais a explicação é a mesma.

Tabela 3 - Primeiro padrão de treinamento.

k	Y_k	UA_k	UR_k
1	Y_1	UA_1	UR_1
2	Y_2	UA_2	UR_2
3	Y_3	UA_3	UR_3
4	Y_4	UA_4	UR_4
5	Y_5	UA_5	UR_5
6	Y_6	UA_6	UR_6
7	Y_7	UA_7	UR_7
8	Y_8	UA_8	UR_8
9	Y_9	UA_9	UR_9

Tabela 4 - Segundo padrão de treinamento.

k	Y_k	UA_k	UR_k
1	Y_1	UA_1	UR_1
2	Y_2	UA_2	UR_2
3	Y_3	UA_3	UR_3
4	Y_4	UA_4	UR_4
5	Y_5	UA_5	UR_5
6	Y_6	UA_6	UR_6

7	Y ₇	UA ₇	UR ₇
8	Y ₈	UA ₈	UR ₈
9	Y ₉	UA ₉	UR ₉

Tabela 5 - Primeiro padrão de treinamento.

k	Y _k	UA _k	UR _k
1	Y ₁	UA ₁	UR ₁
2	Y ₂	UA ₂	UR ₂
3	Y ₃	UA ₃	UR ₃
4	Y ₄	UA ₄	UR ₄
5	Y ₅	UA ₅	UR ₅
6	Y ₆	UA ₆	UR ₆
7	Y ₇	UA ₇	UR ₇
8	Y ₈	UA ₈	UR ₈
9	Y ₉	UA ₉	UR ₉

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 32 e 33:

$$y = \frac{y - y_{min}}{y_{max} - y_{min}} \quad (33)$$

$$UA = \frac{UA - UA_{min}}{UA_{max} - UA_{min}} 0.6 + 0.2 \quad (34)$$

Como os valores de UA_{max} e UA_{min} são constantes e iguais a 5 e 1 volts:

$$UA = \frac{UA - 1}{5 - 1} 0.6 + 0.2 = \frac{UA - 1}{4} 0.6 + 0.2 \quad (35)$$

Para UR o cálculo é exatamente o mesmo.

4.5.2 - Definição da Estrutura da Rede Neural

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 da primeira camada é função das variáveis de entrada da rede. Estas variáveis, como mencionado anteriormente, devem representar significativamente o sistema a ser modelado.

Neste caso o número de neurônios na primeira camada é 4, Y_k , Y_{k-1} , UA_{k-TM} e UR_{k-TM} . 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+TM} , o número de neurônios na camada de saída é 1. Conforme anteriormente mencionado, o número de camadas adotado é 3, resta portanto, o número de neurônios na camada intermediária a ser definido.

Segundo HECHT-NIELSEN [19], uma rede neural *feedforward*, representa bem uma função contínua, quando possui $((2 \cdot i) + 1)$ neurônios na camada intermediária, sendo “i” o número de neurônios da camada de entrada.

Todavia, foram realizados testes experimentais para a determinação do número de neurônios da camada intermediária. Variando-se o número de neurônios, obteve-se 9 como o número de neurônios necessários para uma boa aproximação. Portanto, coincidente com o teorema acima citado. Este número não pode ser pequeno demais para haver uma aproximação pobre e nem grande demais, afim de gerar perda da generalidade da rede, fazendo com que até os ruídos sejam modelados.

Portanto, a arquitetura completa da rede neural utilizada nos experimentos está apresentada na figura 21.

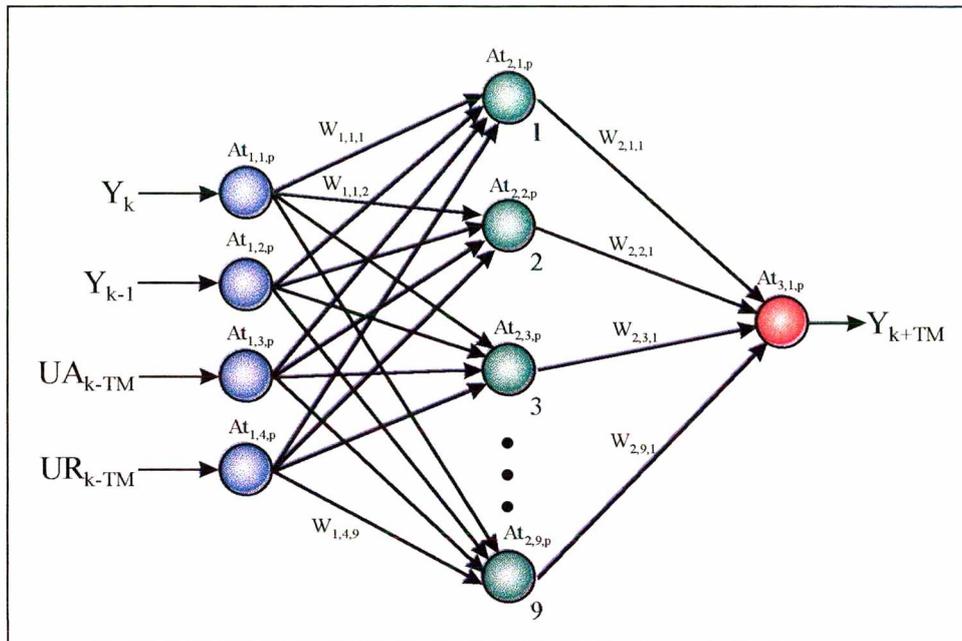


Figura 21 - Arquitetura da rede neural utilizada.

Portanto, a saída predita pelo modelo neural é função das variáveis citadas, assim temos que:

$$Y_{k+TM} = f(Y_{k-1}, Y_k, UA_{k-TM}, UR_{k-TM}) \quad (36)$$

Com:

$$1 \leq UA_k \text{ e } UR_k \leq 5$$

$$\text{Temperatura M\u00ednima} \leq Y_k \leq \text{Temperatura M\u00e1xima}$$

Para UA_k e UR_k iguais a 1, as v\u00e1lvulas de controle est\u00e3o totalmente fechadas, e para UA_k e UR_k iguais a 5, as v\u00e1lvulas de controle est\u00e3o totalmente abertas.

Neste cap\u00edtulo apresentou-se em detalhes a estrutura da rede neural utilizada, bem como a forma\u00e7\u00e3o dos padr\u00f5es. Al\u00e9m disso, apresentou-se tamb\u00e9m as metodologias e materiais utilizados. A seguir ser\u00e1 desenvolvido um controlador baseado no modelo neural.

5 - O CONTROLADOR BASEADO NO MODELO NEURAL

Nos controladores baseados em um modelo fenomenológico do processo, a maior dificuldade encontra-se justamente na obtenção deste modelo de forma a representar bem o processo. A obtenção dos parâmetros necessários à este modelo nem sempre é tarefa simples, e em alguns casos pode representar um elevado custo. Além de se obter um modelo representativo, é necessário resolvê-lo com precisão e em tempo real, e isto nem sempre é possível, mesmo com o grande avanço tecnológico dos recentes anos.

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. Neste caso, o treinamento da rede neural é realizado *off line*, e o modelo obtido implementado *on line*, não havendo portanto, problemas na implementação em tempo real. 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, do tipo:

$$J = \sum_{i=1}^H \left[y_{pred(k+i)} - y_{ref(k+i)} \right]^2 + \lambda_1 \sum_{j=1}^{NU} \left[\Delta UA_{(k+j-1)} \right]^2 + \lambda_2 \sum_{j=1}^{NU} \left[\Delta UR_{(k+j-1)} \right]^2 \quad (29)$$

Com uma trajetória de referência definida anteriormente pela equação (30):

$$y_{ref(k+i)} = \alpha y_{ref(k+i-1)} + (1-\alpha)y_{SP} \quad i = 1 \dots H \quad (30)$$

Por simplicidade pode-se considerar $y_{pred(k+1)} = y_{pred}$.

Onde:

H = Horizonte de predição

NU = Horizonte de controle

λ_1 = Fator de penalização da ação de controle de aquecimento

λ_2 = Fator de penalização da ação de controle de resfriamento

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*

UA_k = Valor da ação de aquecimento a ser implementado

UR_k = Valor da ação de resfriamento a ser implementado

A equação (37) representa a condição necessária para a minimização da equação (29), de forma que as variáveis UA e UR sejam manipuladas independentemente. Esta condição foi apresentada por SIMMONS [43].

$$\frac{\partial J_k}{\partial UA_k} = 0 \quad \text{e} \quad \frac{\partial J_k}{\partial UR_k} = 0 \quad (37)$$

Portanto, primeiramente para a variável UA, obtém-se:

$$\frac{\partial J_k}{\partial UA_k} = 2 \left[y_{pred} - y_{ref(k+1)} \right] \frac{\partial y_{pred}}{\partial UA_k} + 2\lambda_1 \left[UA_k - UA_{(k-1)} \right] \quad (38)$$

E de acordo com a equação (37) $\frac{\partial J_k}{\partial UA_k} = 0$, tem-se:

$$UA_k = UA_{(k-1)} - \frac{1}{\lambda_1} \left[y_{pred} - y_{ref(k+1)} \right] \frac{\partial y_{pred}}{\partial UA_k} \quad (39)$$

Resta ainda o termo da derivada de Y_{pred} em relação a UA_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. Evidentemente, 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:

$$At_{3,k,p} = y_{pred} \quad (40)$$

onde $At_{3,k,p}$ é a ativação do k -ésimo neurônio na camada 3 correspondente ao p -ésimo padrão apresentado, neste caso a saída da rede neural.

$$\frac{\partial y_{pred}}{\partial UA_k} = \frac{\partial At_{3,k,p}}{\partial UA_k} \quad (41)$$

$$\frac{\partial y_{pred}}{\partial UA_k} = \frac{\partial}{\partial UA_k} \left[1 + \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \right]^{-1} \quad (42)$$

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

Conforme as equações (12) e (15), tem-se:

$$\frac{\partial y_{pred}}{\partial UA_k} = At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} \frac{\partial At_{2,j,p}}{\partial UA_k} \right) \quad (44)$$

Onde:

$$\frac{\partial At_{2,j,p}}{\partial UA_k} = \left[1 + \exp \left(- \sum_{i=1}^{N_{NE}} (W_{1,i,j} At_{1,i,p}) \right) \right]^{-2} \exp \left(- \sum_{i=1}^{N_{NE}} (W_{1,i,j} At_{1,i,p}) \right) \sum_{i=1}^{N_{NE}} \left(W_{1,i,j} \frac{\partial At_{1,i,p}}{\partial UA_k} \right) \quad (45)$$

Aplicando as definições analogamente à equação (44), e considerando que apenas o terceiro neurônio da primeira camada recebe UA_k , obtém-se:

$$\frac{\partial At_{2,j,p}}{\partial UA_k} = At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j} \frac{\partial At_{1,3,p}}{\partial UA_k} \quad (46)$$

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

$$\frac{\partial At_{1,3,p}}{\partial UA_k} = 1 \quad (47)$$

Assim:

$$\frac{\partial At_{2,j,p}}{\partial UA_k} = At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j} \quad (48)$$

Substituindo a equação (48) na equação (44), obtém-se:

$$\frac{\partial y_{pred}}{\partial UA_k} = At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j}) \quad (49)$$

Finalmente, substituindo a equação (49) na equação (39) obtém-se a expressão para a lei de controle de aquecimento:

$$UA_k = UA_{(k-1)} - \frac{1}{\lambda_1} [y_{pred} - y_{ref(k+1)}] At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j}) \quad (50)$$

Sabendo que, $At_{3,k,p} = y_{pred}$ e que $y_{ref(k+1)}$ é dado pela equação (29), com $i=1$, ou seja, $y_{ref_k} = y_k$, a lei de controle fica:

$$UA_k = UA_{(k-1)} - \frac{1}{\lambda_1} \left[y_{pred} - \alpha y_k - y_{SP} + \alpha y_{SP} \right] y_{pred} (1 - y_{pred}) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} At_{2,j,p} (1 - At_{2,j,p}) W_{1,3,j} \right) \quad (51)$$

Desta forma obtém-se a equação (51), que é a expressão final para a lei de controle de aquecimento.

Resta, agora, definir uma expressão para a lei de controle de resfriamento. Derivando a equação (36) em relação a UR_k , obtém-se:

$$\frac{\partial J_k}{\partial UR_k} = 2 \left[y_{pred} - y_{ref(k+1)} \right] \frac{\partial y_{pred}}{\partial UR_k} + 2\lambda_2 [UR_k - UR_{(k-1)}] \quad (52)$$

Da equação (37) $\frac{\partial J_k}{\partial UA_k} = 0$, e portanto:

$$UR_k = UR_{(k-1)} - \frac{1}{\lambda_2} \left[y_{pred} - y_{ref(k+1)} \right] \frac{\partial y_{pred}}{\partial UR_k} \quad (53)$$

Da equação (40):

$$At_{3,k,p} = y_{pred} \quad (40)$$

$$\frac{\partial y_{pred}}{\partial UR_k} = \frac{\partial At_{3,k,p}}{\partial UR_k} \quad (54)$$

$$\frac{\partial y_{pred}}{\partial UR_k} = \frac{\partial}{\partial UR_k} \left[1 + \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \right]^{-1} \quad (55)$$

$$\frac{\partial y_{pred}}{\partial UR_k} = \left[1 + \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \right]^{-2} \exp \left(- \sum_{j=1}^{N_{NI}} (W_{2,j,k} At_{2,j,p}) \right) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} \frac{\partial At_{2,j,p}}{\partial UR_k} \right) \quad (56)$$

De forma análoga à ação de aquecimento, pelas equações (12) e (15):

$$\frac{\partial y_{pred}}{\partial UR_k} = At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} \frac{\partial At_{2,j,p}}{\partial UR_k} \right) \quad (57)$$

Onde:

$$\frac{\partial At_{2,j,p}}{\partial UR_k} = \left[1 + \exp \left(- \sum_{i=1}^{N_{NE}} (W_{1,i,j} At_{1,i,p}) \right) \right]^{-2} \exp \left(- \sum_{i=1}^{N_{NE}} (W_{2,j,k} At_{2,j,p}) \right) \sum_{i=1}^{N_{NE}} \left(W_{1,i,j} \frac{\partial At_{1,i,p}}{\partial UR_k} \right) \quad (58)$$

Analogamente à equação (44), e considerando que apenas o quarto neurônio da primeira camada recebe UR_k , obtém-se:

$$\frac{\partial At_{2,j,p}}{\partial UR_k} = At_{2,j,p} (1 - At_{2,j,p}) W_{1,4,j} \frac{\partial At_{1,4,p}}{\partial UR_k} \quad (59)$$

Como $At_{1,4,p}$ é a própria entrada UR_k , então:

$$\frac{\partial At_{1,4,p}}{\partial UR_k} = 1 \quad (60)$$

Assim:

$$\frac{\partial At_{2,j,p}}{\partial UR_k} = At_{2,j,p} (1 - At_{2,j,p}) W_{1,4,j} \quad (61)$$

Substituindo a equação (61) na equação (55), obtém-se:

$$\frac{\partial y_{pred}}{\partial UR_k} = At_{3,k,p} (1 - At_{3,k,p}) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} At_{2,j,p} (1 - At_{2,j,p}) W_{1,4,j} \right) \quad (62)$$

Finalmente, substituindo a equação (62) na equação (53) obtém-se a expressão para a lei de controle de resfriamento:

$$UR_k = UR_{(k-1)} - \frac{1}{\lambda_2} \left[y_{pred} - y_{ref_{(k+1)}} \right] At_{3,k,p} \left(1 - At_{3,k,p} \right) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} At_{2,j,p} \left(1 - At_{2,j,p} \right) W_{1,4,j} \right) \quad (63)$$

De forma análoga ao feito para a ação de aquecimento, obtém-se uma expressão final para o cálculo da ação de resfriamento dada pela equação (64):

$$UR_k = UR_{(k-1)} - \frac{1}{\lambda_2} \left[y_{pred} - \alpha y_k - y_{SP} + \alpha y_{SP} \right] y_{pred} \left(1 - y_{pred} \right) \sum_{j=1}^{N_{NI}} \left(W_{2,j,k} At_{2,j,p} \left(1 - At_{2,j,p} \right) W_{1,4,j} \right) \quad (64)$$

Nas leis de controle expressas pelas equações (51) e (64) pode-se observar que as variáveis, com exceção de $W_{2,j,k}$, 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. As equações são dependentes dos parâmetros da rede neural, portanto um modelo neural com um aprendizado pobre pode degenerar o controle em algumas situações.

5.1 - O Algoritmo de Controle

A estratégia de controle baseia-se no treinamento *off line* da rede neural, que funciona como modelo do processo e parte integrante da lei de controle conforme visto nas equações das leis de controle. A implementação 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 , $Y_{(k-1)}$, UA_k e UR_k , sendo que UA_k e UR_k são os valores das ações de controle a serem implementadas no instante k , ou seja, ainda não foram determinadas. A saída utilizada foi adotar um esquema iterativo para a determinação destes valores. O processo iterativo se repete até que as diferenças entre os valores de UA_k e UR_k obtidos sejam suficientemente pequenas, conforme as equações (65) e (66).

$$\Delta UA = \left| UA_{k_{\text{assumido}}} - UA_{k_{\text{Lei de Controle}}} \right| \quad (65)$$

$$\Delta UR = \left| UR_{k_{\text{assumido}}} - UR_{k_{\text{Lei de Controle}}} \right| \quad (66)$$

Destá forma, se ΔUA e ΔUR forem suficientemente pequenos, o valor de y_{pred} obtido é assumido como $y_{(k+1)}$ e o valores de UA_k e UR_k calculados são implementados.

O algoritmo utilizado para a implementação da estratégia de controle é apresentado pela figura 22, abaixo.

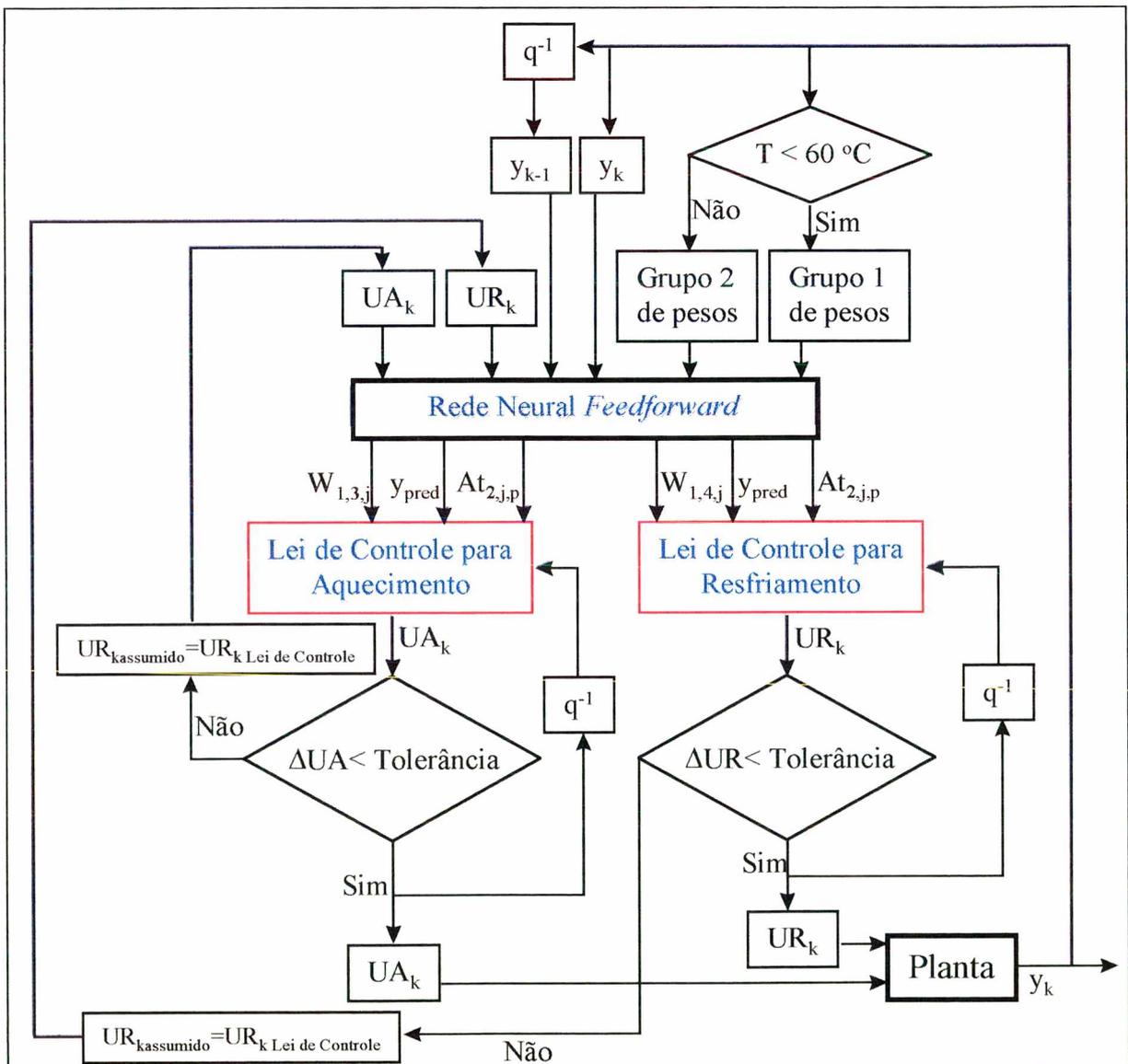


Figura 22 - Algoritmo de controle.

Neste capítulo foi apresentado o desenvolvimento do controlador utilizado nos experimentos. No capítulo a seguir serão apresentados os resultados experimentais, bem como sua análise.

6 - RESULTADOS E DISCUSSÕES

Os resultados serão apresentados em duas seções diferentes. Na primeira será apresentada a fase de treinamento da rede neural, e na segunda a fase de controle. Esta última será, ainda, dividida em três partes, controle sem reação química, controle com reação, e por fim os testes para a estratégia de ajuste proposta ao controlador.

A estratégia de controle baseada no modelo neural foi testada e comparada ao controlador PID. O processo estudado engloba típicos problemas de controle, como tempo morto, dinâmica variante no tempo e histerese.

Além das não linearidades inerentes do reator, 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 23. Pelo fato das válvulas de controle serem iguais, obteve-se a dinâmica somente para a válvula de resfriamento.

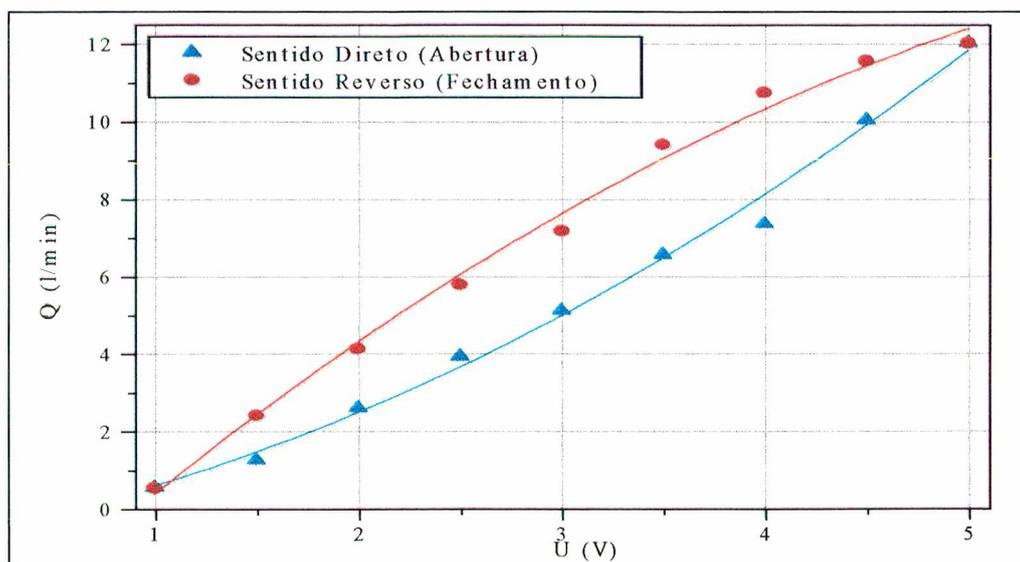


Figura 23 - Dinâmica da válvula de controle de água fria.
1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

6.1 - Treinamento da Rede Neural

O elevado tempo morto presente no processo estudado é um importante aspecto a ser considerado na formação dos padrões, os quais serão apresentados à rede neural.

A dinâmica do reator 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 reator conferem ao sistema um tempo morto da ordem de 50 a 70 segundos. Como nos testes realizados o intervalo de amostragem foi de 10 s, isto equivale a 5 e 7 intervalos de amostragem, respectivamente. O vapor utilizado como fonte de calor foi fornecido por uma caldeira elétrica com capacidade de 100 Kg/h operando na faixa de 3-6 Kgf/cm², caracterizando uma forte perturbação no sistema.

Os dados para treinamento foram coletados em malha aberta. O treinamento da rede neural foi realizado *off line*. Devido a problemas, apresentados no capítulo 4, foram necessárias duas redes neurais para modelar completamente o reator. O primeiro grupo de dados para treinamento foi coletado entre as temperaturas de 20 e 60 °C com variações aleatórias de 1 a 5 volts das válvulas de controle. O segundo grupo compreendia valores entre 60 e 95 °C de temperatura, também com variações aleatórias de 1 a 5 válvulas das ações de controle.

O primeiro grupo de dados pode ser visto na figura 24.

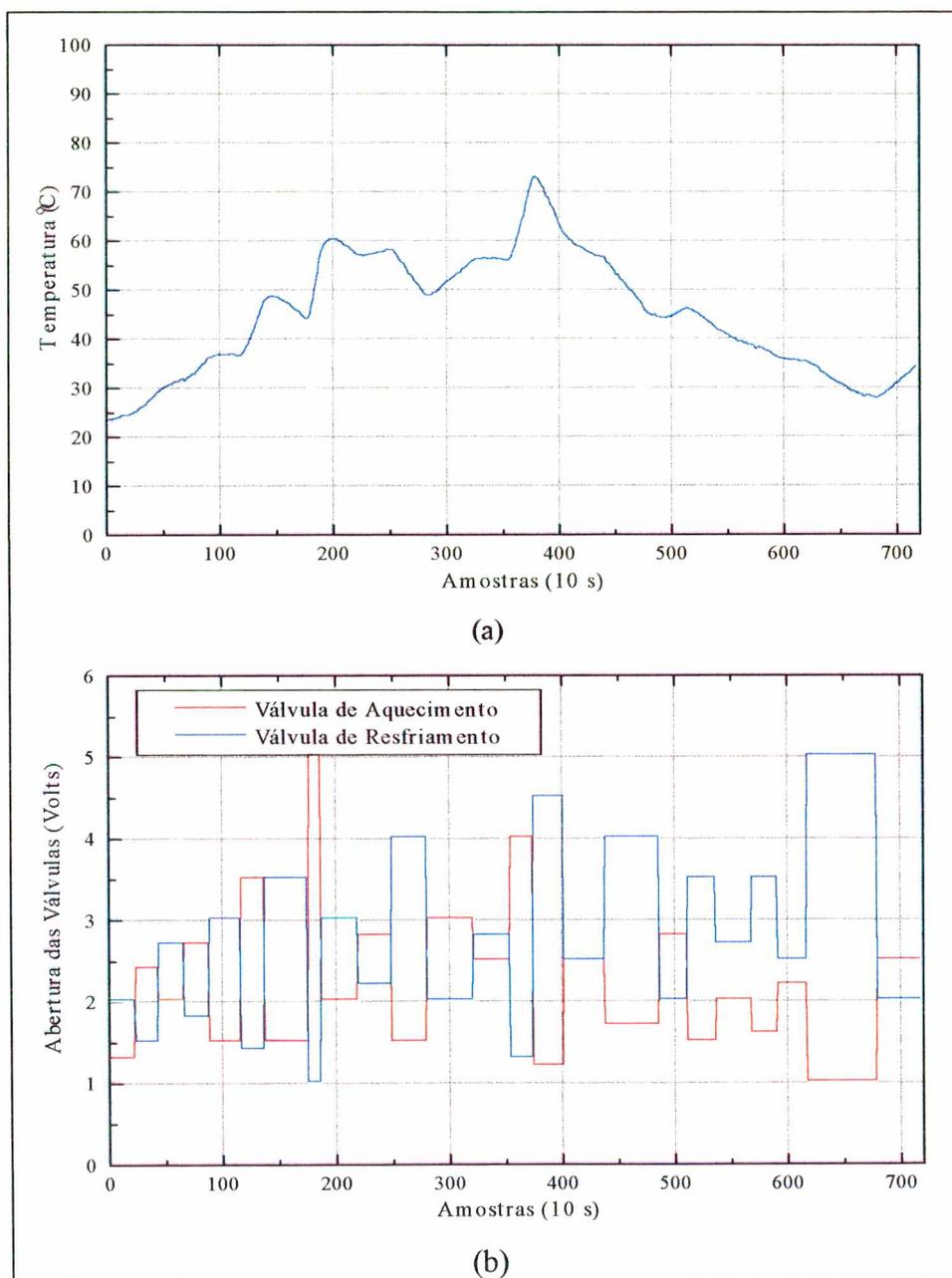


Figura 24 - Dados para treinamento (20-60 °C).

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

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

O segundo grupo de dados, de 60 até 95 °C, é mostrado na figura 25.

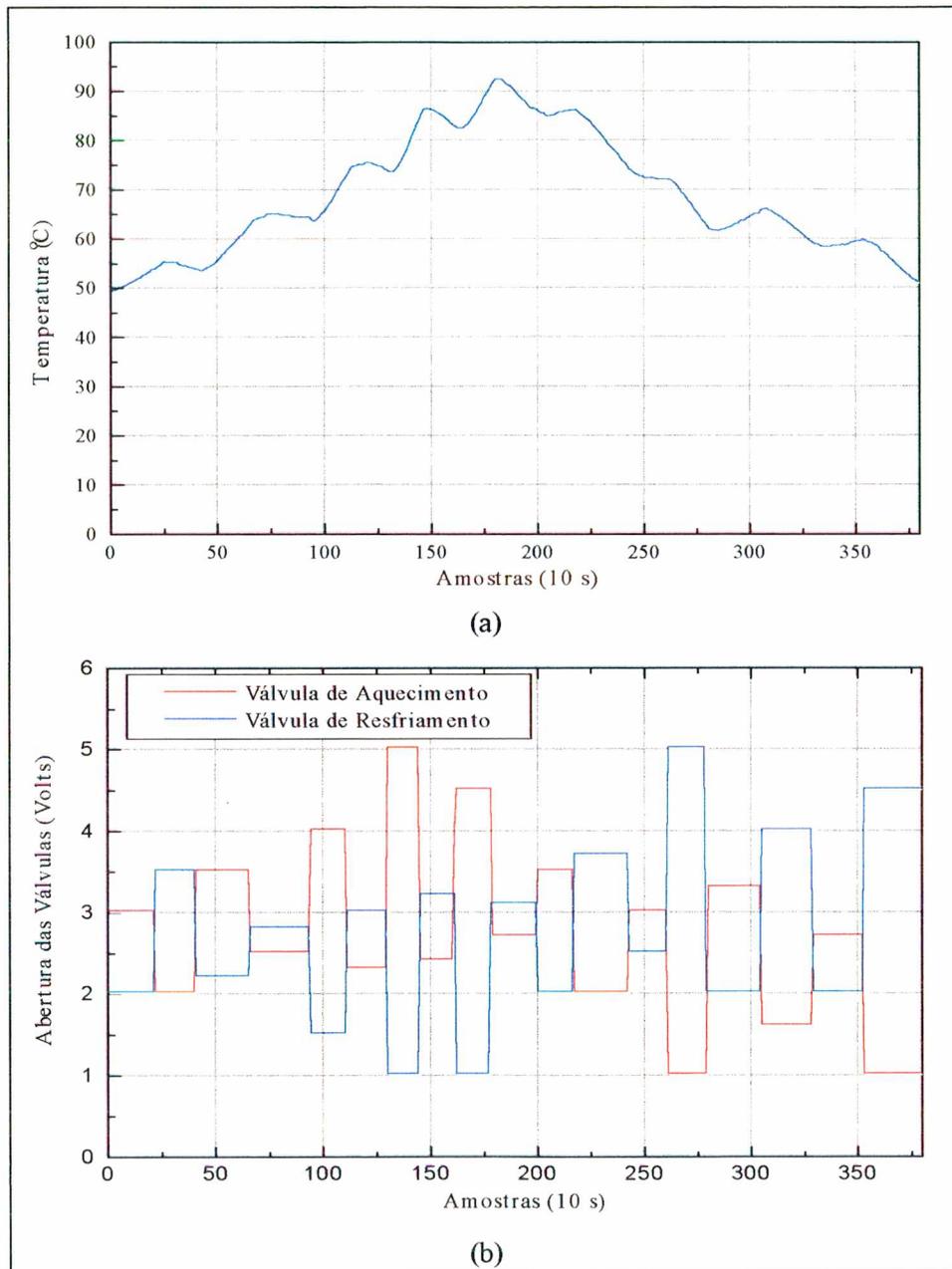


Figura 25 - Dados para treinamento (60-95 °C).

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

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Para testar a validade do modelo obtido *off line*, pode-se submetê-lo a um outro grupo de dados, obtido da mesma forma que o grupo padrão.

A qualidade do modelo obtido para cada grupo de padrões, após 50000 iterações para o treinamento da rede, pode ser analisada nas figuras 26 e 27.

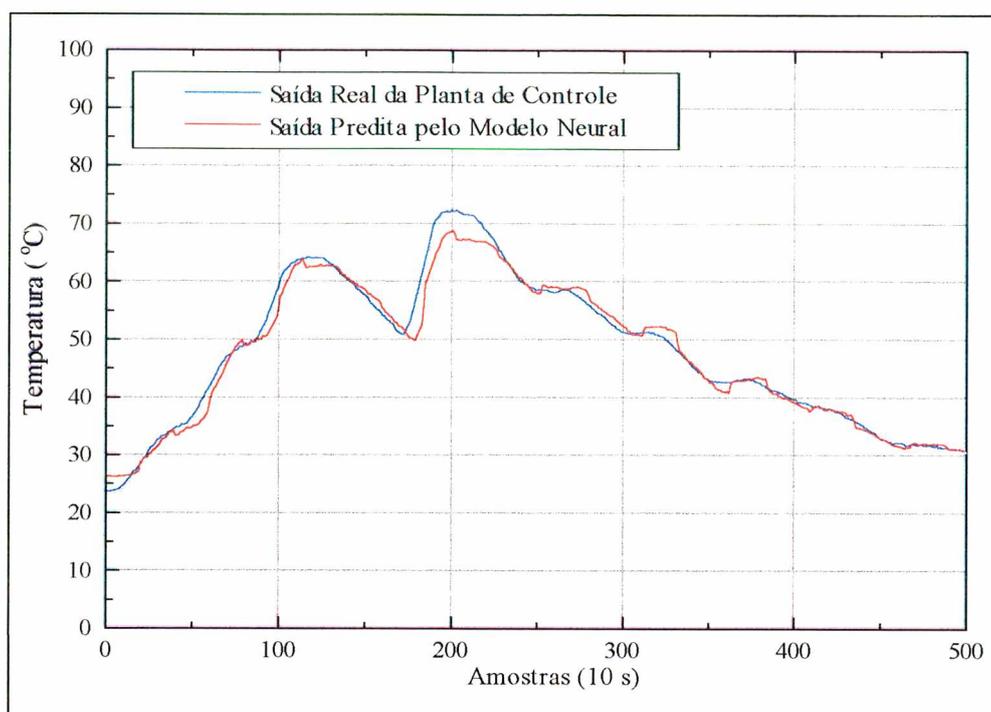


Figura 26 - Teste para o modelo neural (20-60 °C).

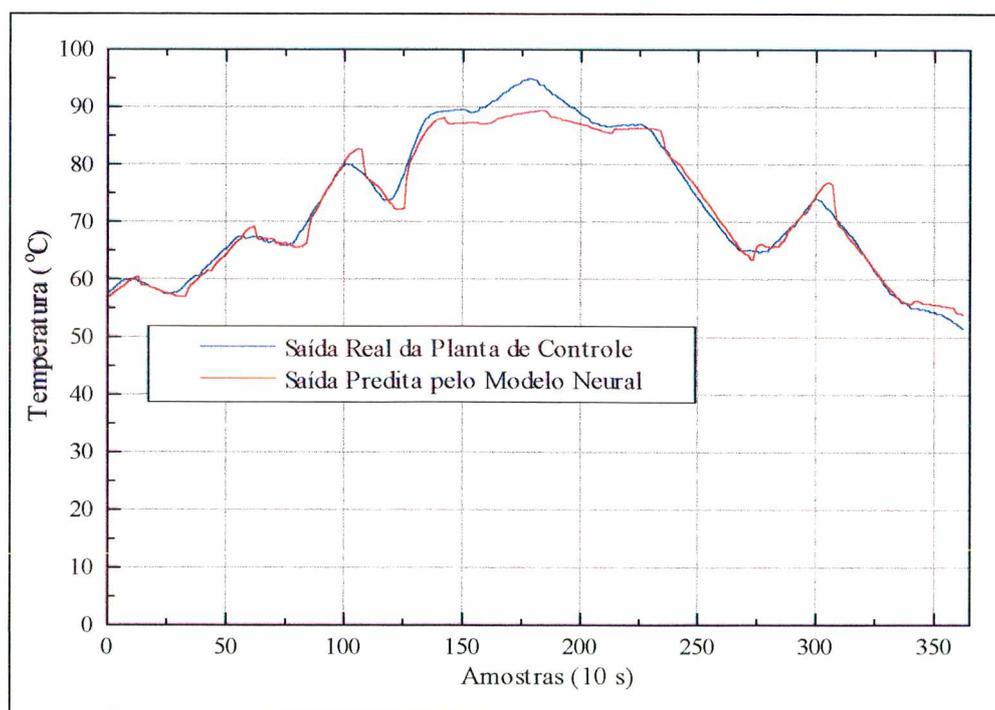


Figura 27 - Teste para o modelo neural (60-95 °C).

Pode-se observar que, em ambos os casos, o modelo neural foi capaz de aproximar-se bem da temperatura real, tendo pequenos desvios somente nos limites superior e inferior. Como critério de avaliação utilizou-se a Somatória do Erro Médio Quadrático, sendo que em ambos os casos este valor foi inferior a 0,4.

6.2 - Controle

Esta é a etapa final para validação das estratégias de controle. A estratégia proposta é comparada com o clássico controlador PI.

6.2.1 - Experimentos Sem Reação Química

Esta foi a primeira etapa da implementação do controlador baseado no modelo neural. A implementação da estratégia de controle foi realizada no reator sem reação química. A principal finalidade destes experimentos foi a determinação dos parâmetros do controlador, a fim de serem utilizados nos testes com reação química. Também pode-se observar que somente um ajuste dos parâmetros do controlador proposto, é capaz de controlar a temperatura em toda a faixa de trabalho do reator, mesmo com os dois diferentes modelos neurais. Em contrapartida, o PI, por se tratar de um controlador linear, necessitaria que seus parâmetros fossem ajustados ao longo do controle.

Na figura 28 são apresentados os resultados obtidos para variações no *setpoint* para o controlador proposto. As variações apresentadas são ascendentes. Os parâmetros utilizados foram ajustados, de forma empírica, para os seguintes valores: $\alpha = 0,55$, $\lambda_1 = 2,3$ e $\lambda_2 = 1,1$.

Pode-se observar que o controlador proposto foi capaz de realizar as transições seguras de *setpoint*, sem permitir *overshoot* e nem *offset*.

Na figura 29, os resultados para o controlador PI são apresentados, para os parâmetros, $K_c = 1,5$ °C/volt e $T_i = 190$ s, determinados pelo método de Cohen & Coon [44], para a faixa de temperatura entre 20 e 40 °C, região onde ocorrerá a reação química, apresentada em seção posterior.

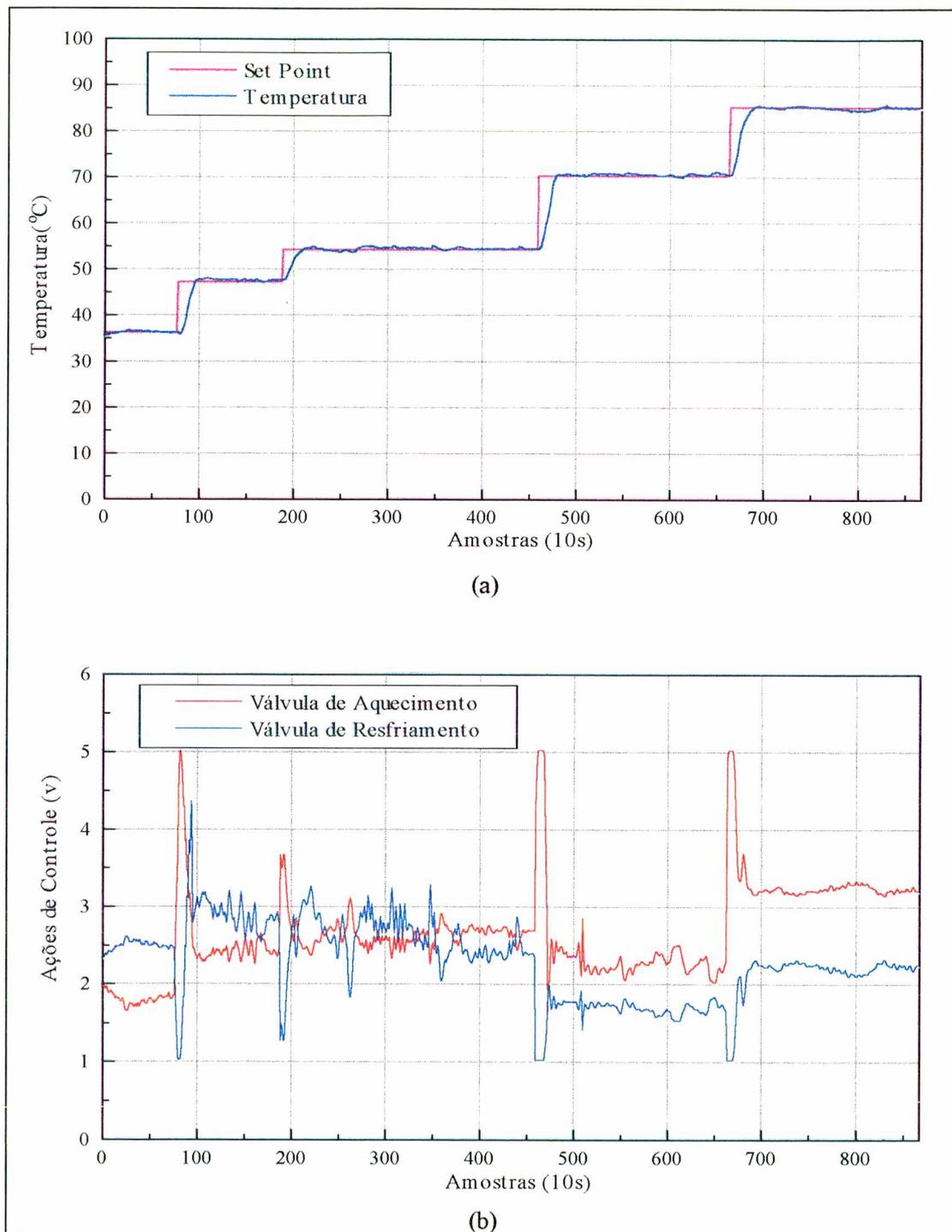


Figura 28 - Controlador proposto sujeito a mudanças ascendentes no *setpoint*.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

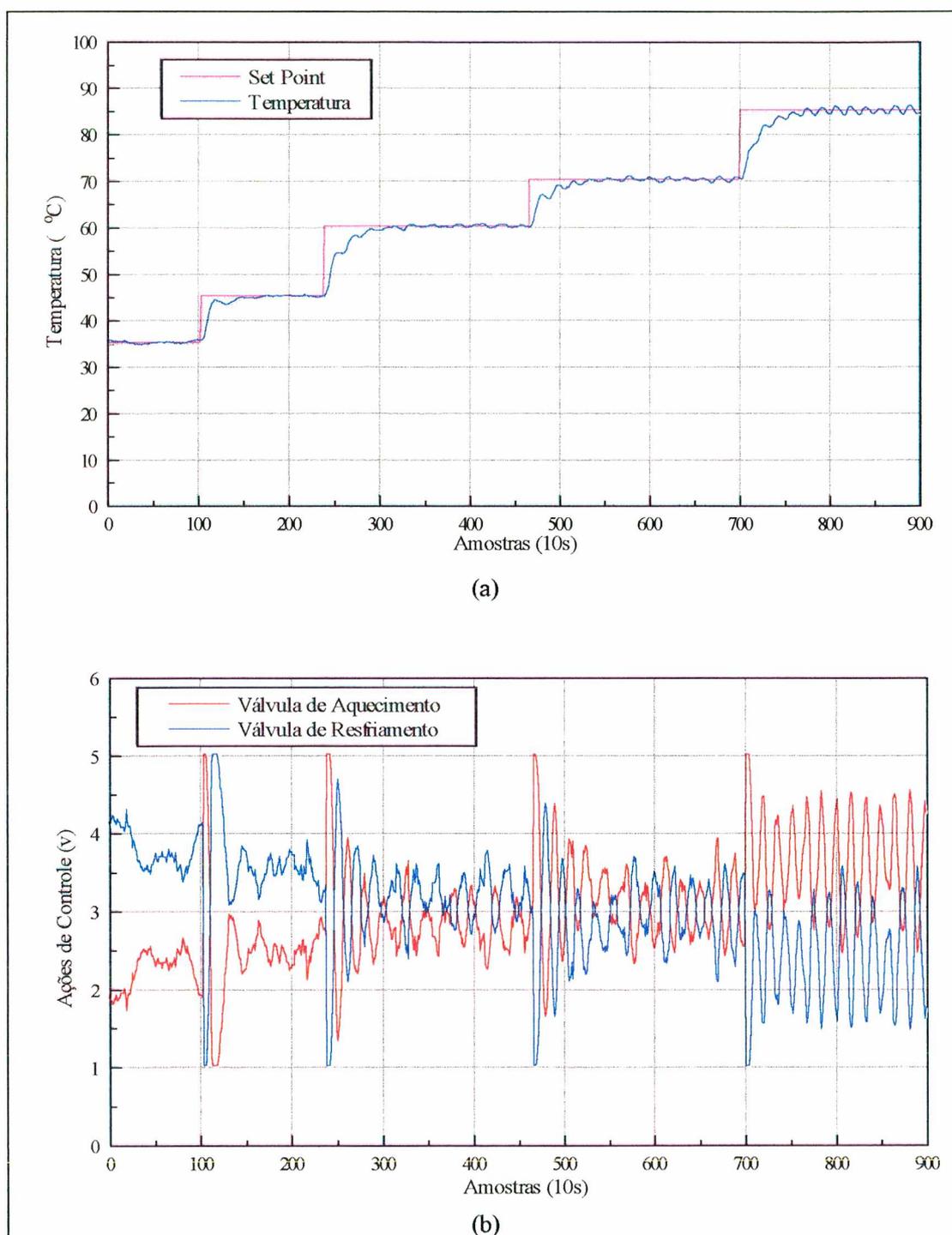


Figura 29 - Controlador PI sujeito a mudanças ascendentes no *setpoint*.

(a) Comportamento da temperatura. (b) Ações de Controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Pode-se observar na figura 29, que o controlador PI obteve um bom desempenho na região para onde foi projetado, contudo por ser um controlador linear, não apresentou uma boa performance fora desta região, apresentando uma ação de controle oscilatória.

A figura 30, abaixo, apresenta resultados para transições descendentes de *setpoint*, para o controlador baseado no modelo neural.

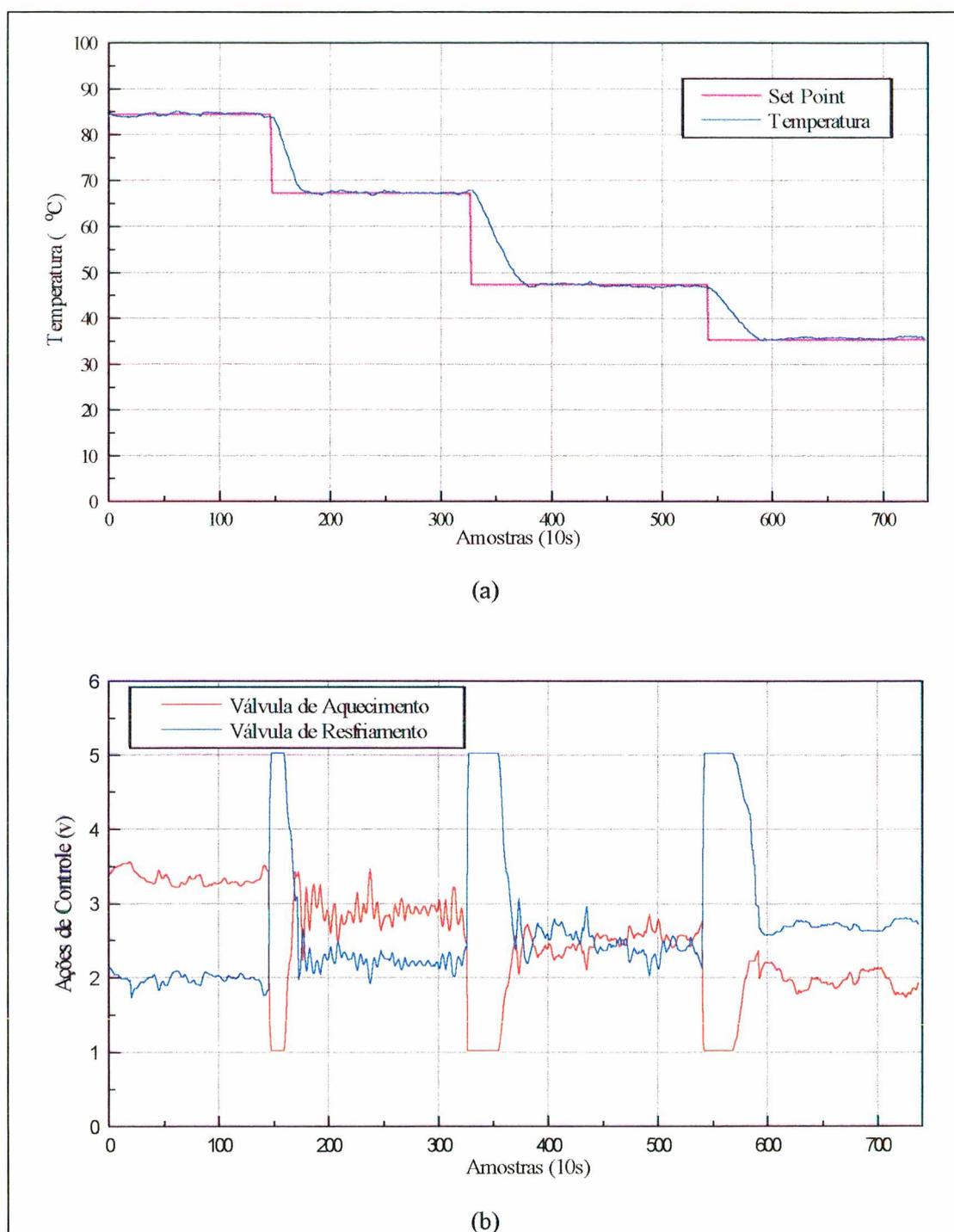


Figura 30 - Controlador proposto sujeito a mudanças descendentes no *setpoint*.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Os resultados para transições descendentes de *setpoint* para o controlador PI são apresentados na figura 31.

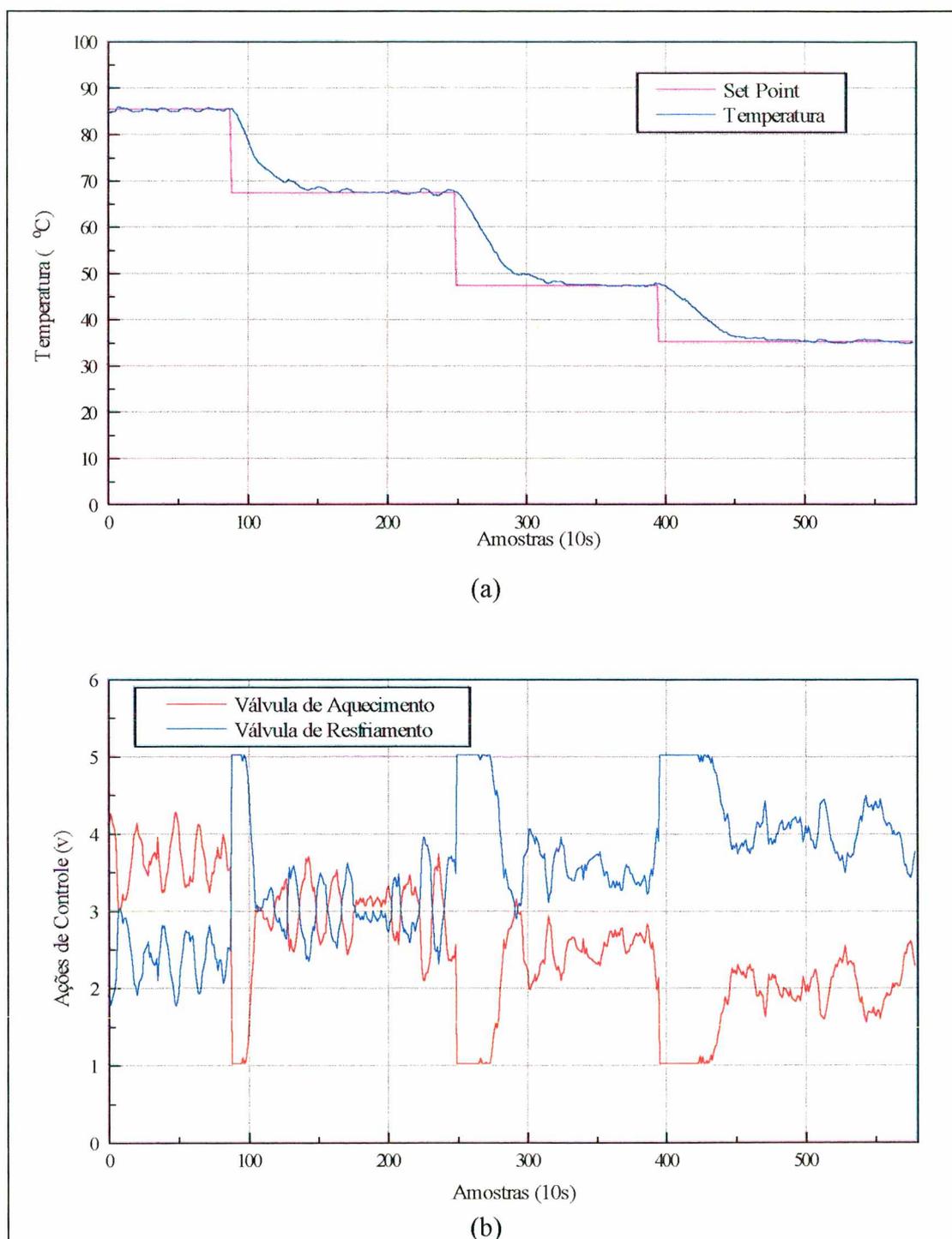


Figura 31 - Controlador PI sujeito a mudanças descendentes no *setpoint*.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Pelas figuras 30 e 31, pode-se verificar que os controladores mantiveram comportamentos semelhantes aos para alterações ascendentes de *set point*. Uma característica importante, que pode ser observada nas figuras 28(b) e 30(b), é que as válvulas estão sendo manipuladas independentemente. Nestes casos observa-se que o consumo de vapor, e de água de refrigeração é bem inferior ao controle realizado pelo PI, com estratégia *split range*, nas figuras 29(b) e 31(b). Outra vantagem em se operar com as válvulas independentes é a maior flexibilidade nas ações de controle. Integrando-se as ações de controle pode-se ter uma noção no consumo de vapor e de água de refrigeração para ambos os controladores, em termos de abertura das válvulas. A integração foi feita de acordo com a equação (67).

$$\text{Consumo} = \int (U - 1)^2 dU \quad (67)$$

Como resultados, para a figura 28(b), obteve-se um consumo para o controlador proposto:

Vapor = 2701,58 (Volts)² para 867 amostras, ou 3,12 (Volts)²/Amostra.

Água = 1596,83 (Volts)² para 867 amostras, ou 1,84 (Volts)²/Amostra.

E para o controlador PI, para a figura 29(b):

Vapor = 3977,19 (Volts)² para 904 amostras, ou 4,40 (Volts)²/Amostra.

Água = 4174,59 (Volts)² para 904 amostras, ou 4,61 (Volts)²/Amostra.

É importante ressaltar que as válvulas não são lineares, e que a pressão de vapor não é constante, portanto esta diferença em termos de abertura, não significa a mesma diferença em termos de vazão.

Nas figuras 32 e 33 são apresentadas alterações bruscas de *setpoint*, para o controlador proposto e para o PI, respectivamente.

Pelas figuras 32 e 33, observa-se que a transição entre os *setpoints* é mais rápida, tanto no sentido ascendente quanto no descendente, para o controlador baseado no modelo neural. Novamente pode-se observar que o controlador PI obteve uma boa performance na região para

onde os parâmetros foram calculados, contudo, novamente apresentou oscilações fora desta região.

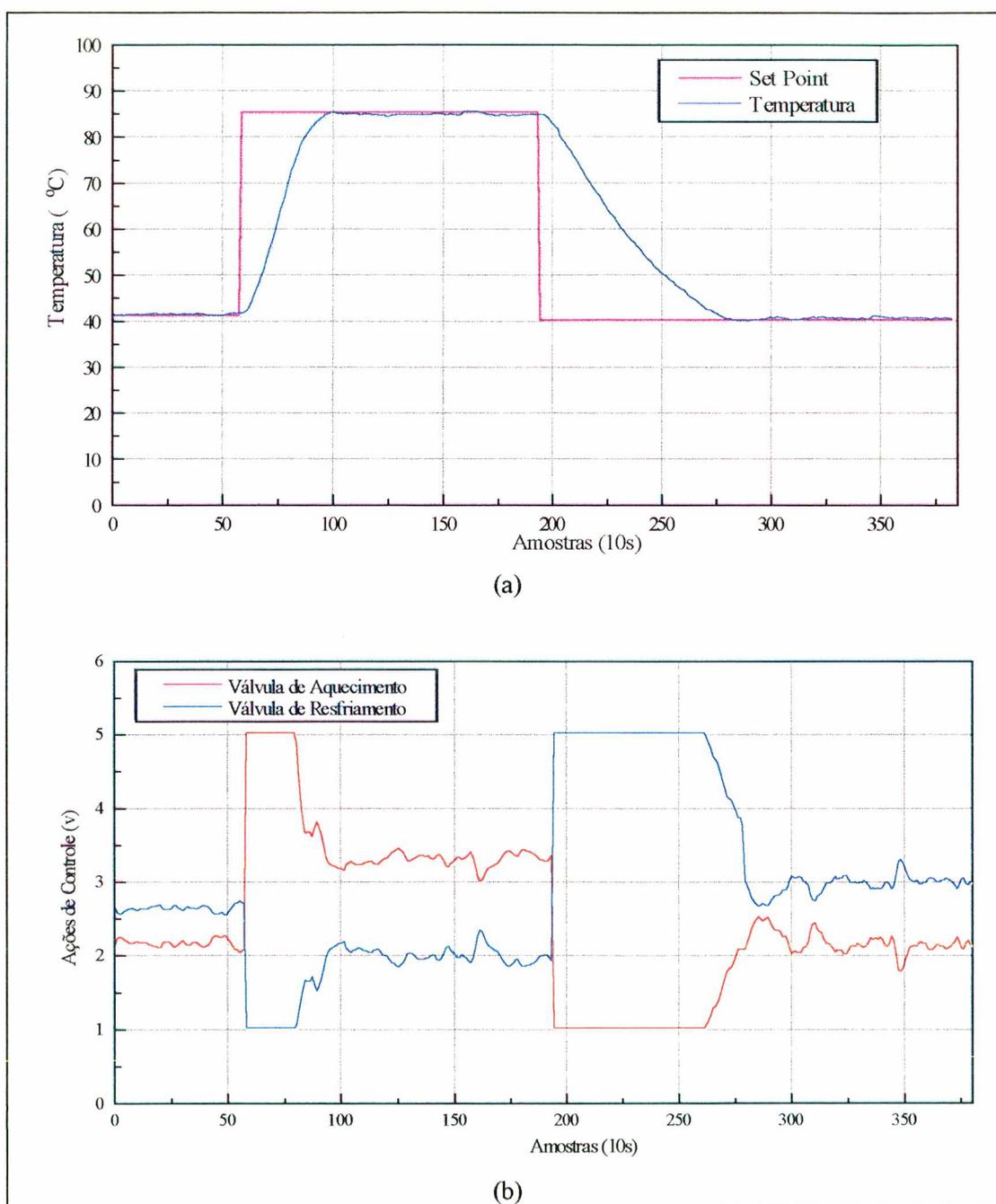


Figura 32 - Alteração brusca de *setpoint* para o controlador baseado no modelo neural.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

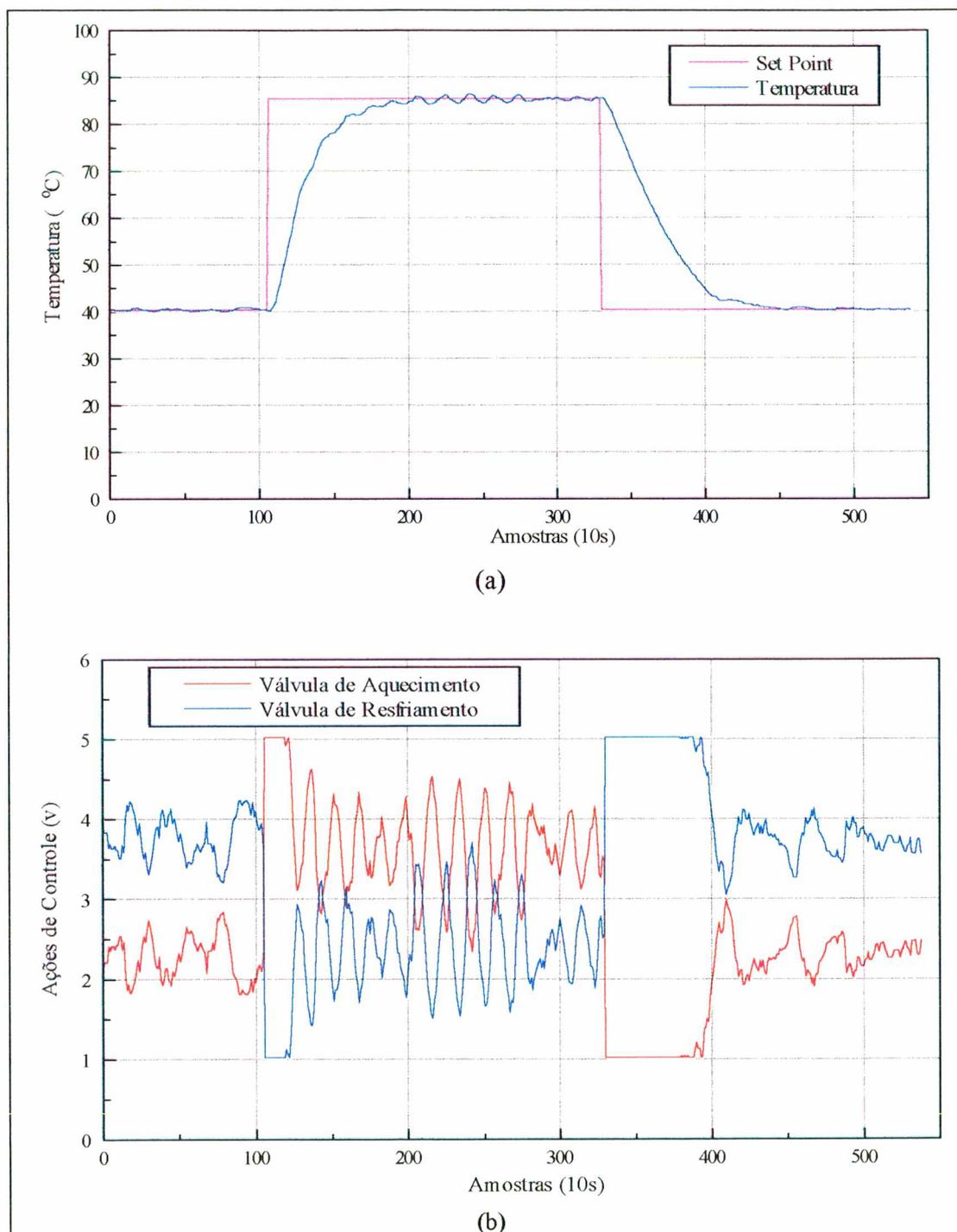


Figura 33 - Alteração brusca de *setpoint* para o controlador PI.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

A estratégia de controle, conforme capítulo anterior, e baseada no treinamento de duas redes neurais para cobrir toda a faixa de temperatura de trabalho do reator, sendo a divisão para a utilização de uma ou outra em 60 °C. Resta saber se o controlador é capaz de manter a

temperatura exatamente neste valor, onde é feito o acoplamento dos modelos neurais. A performance do controlador pode ser observada na figura 34.

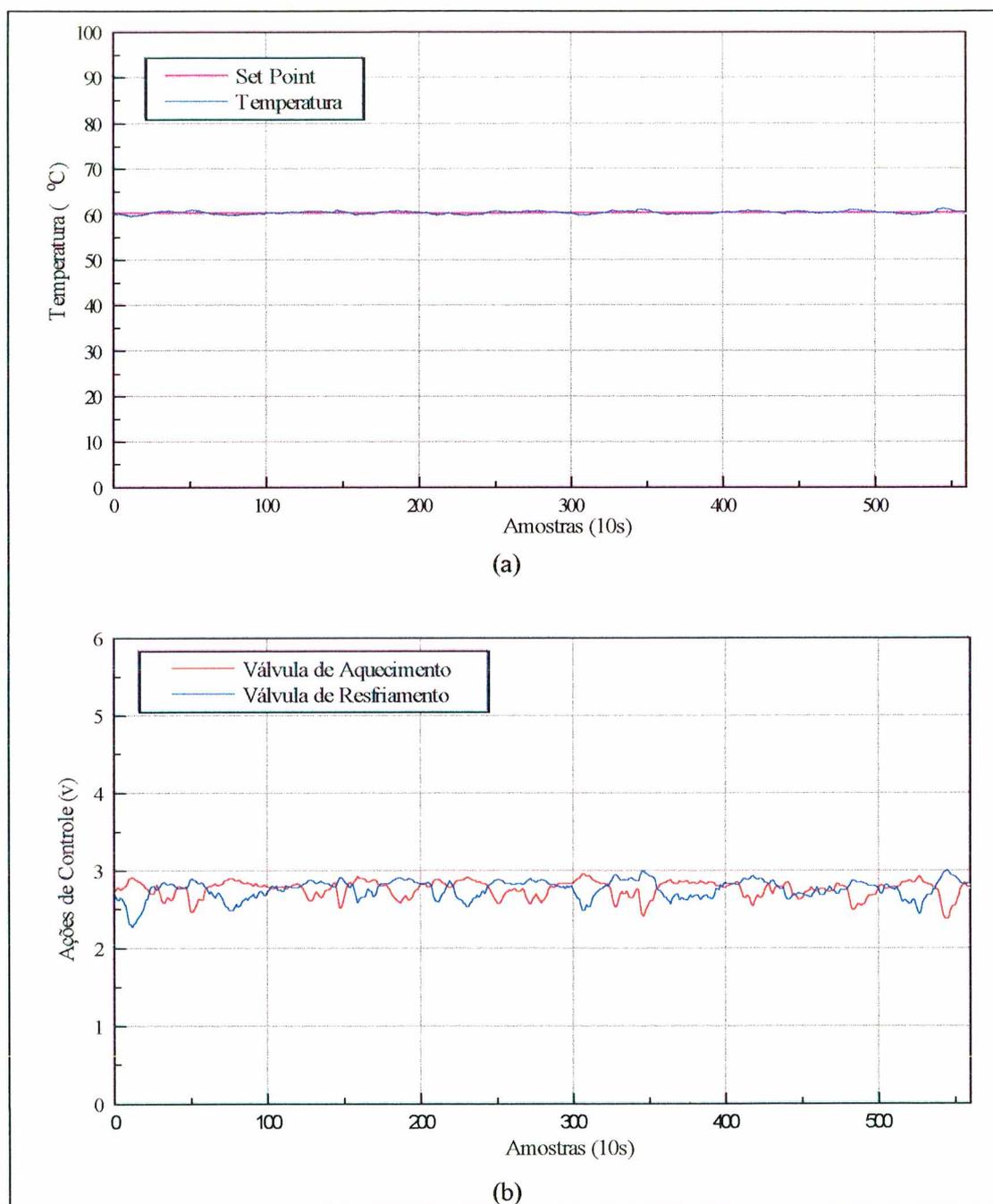


Figura 34 - Controlador baseado no modelo neural na temperatura de acoplamento.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Nota-se que a temperatura se manteve no *setpoint*, apresentando oscilações mínimas, que podem ser desconsideradas. As ações de controle foram excelentes, sem alterações bruscas, fator importante para evitar a fadiga das válvulas de controle.

6.2.2 - Experimentos com Reação Química

Uma reação química agrega ao sistema uma elevada não linearidade, e esta realizada neste experimento, por ser altamente exotérmica, ainda possui a dificuldade da manutenção do controle, pois qualquer desvio pode desencadear o disparo da reação, impossibilitando assim seu controle.

A figura 35 mostra o sistema sob ação do controlador proposto no ponto ótimo de operação, ou seja 35 °C. A vazão de alimentação de reagente foi de 80 ml/min.

É observado, na figura 35, que o controlador foi capaz de levar a temperatura ao ponto ótimo de operação, sem a ocorrência de *overshoot* e mantendo-a até o final da reação.

Novamente, para comparação, realizou-se testes com o controlador PI, nas mesmas condições anteriores, sendo os resultados apresentados na figura 36.

Na figura 36 observa-se que o controlador PI apresentou uma performance inferior ao controlador proposto, figura 35. Este fato deve-se principalmente pela ocorrência de um elevado *overshoot*. Nas figuras 29 e 31 observou-se que o controlador PI não apresentou *overshoot* nesta faixa de temperatura. No entanto, com a existência de uma reação química, a dinâmica do processo muda completamente, fazendo que o PI não tenha uma performance satisfatória.

A ocorrência do *overshoot* não é indicada para este tipo de reação, devido a sua exotermicidade, podendo ocorrer seu disparo.

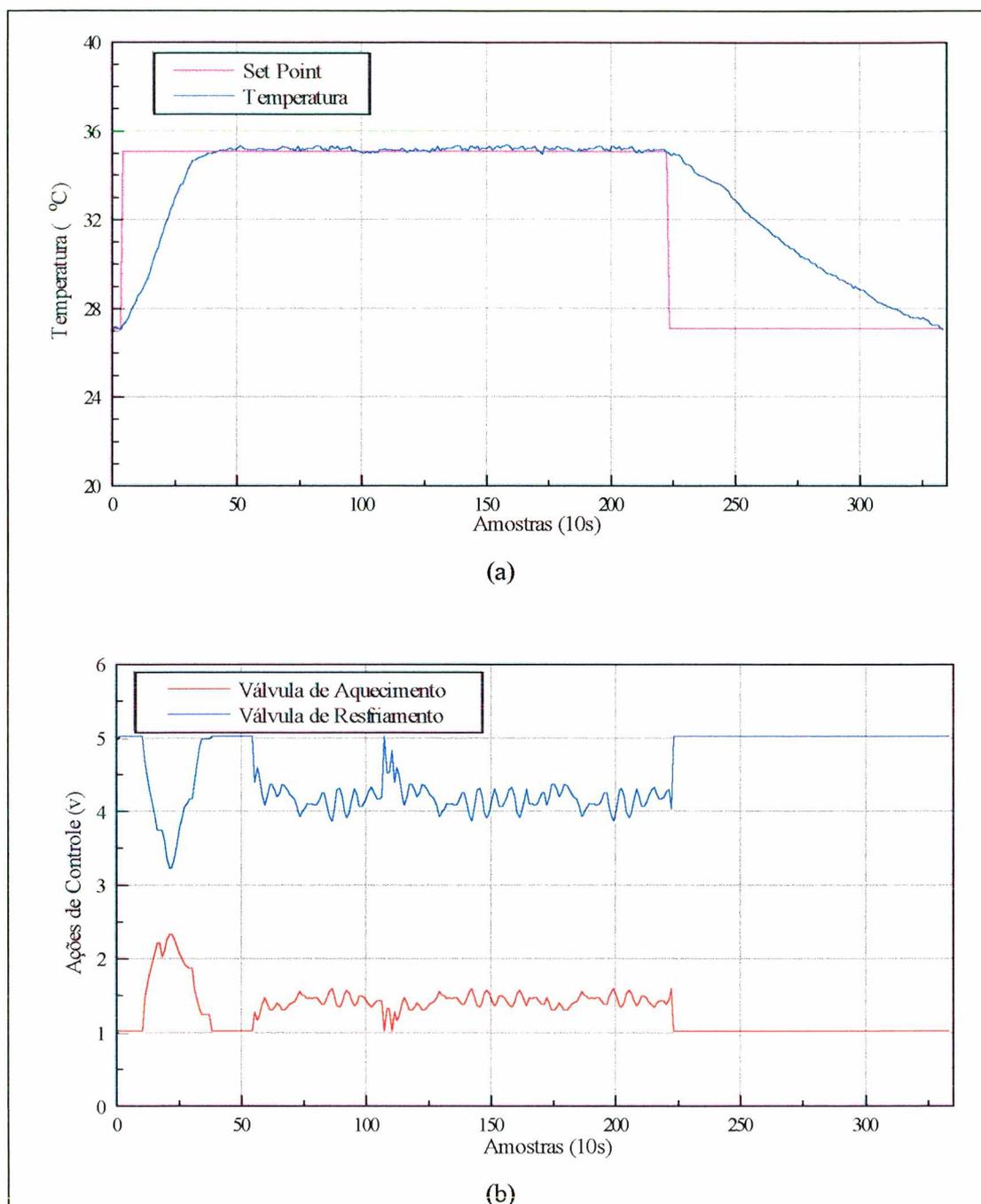


Figura 35 - Controle da reação pelo controlador baseado no modelo neural.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

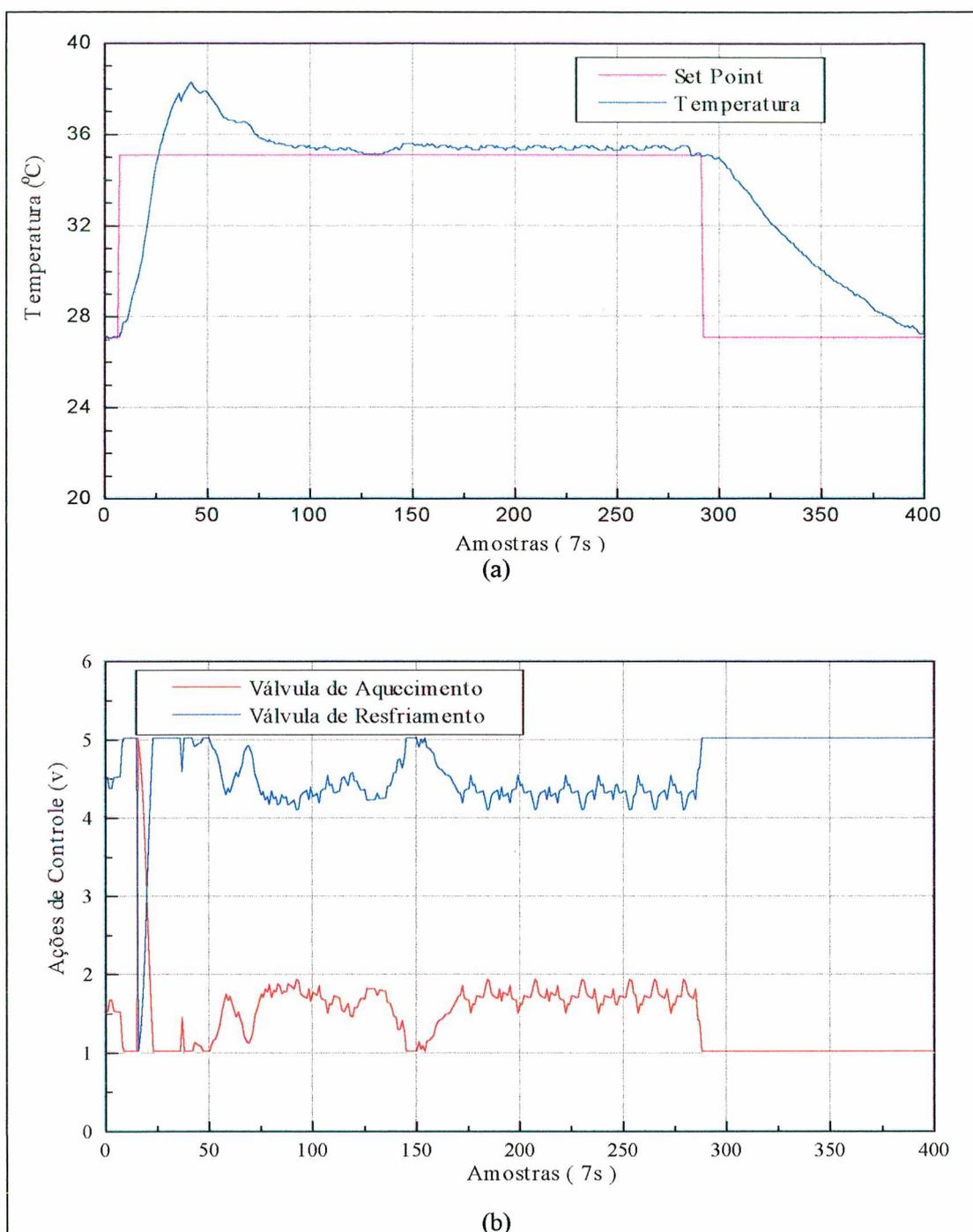


Figura 36 - Controle da reação pelo controlador PI.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

No último teste do controlador proposto, causou-se uma perturbação tipo carga na alimentação de reagente, sendo isto observado na figura 37. Iniciou-se a alimentação de reagente com 65 ml/min, e no ponto indicado na figura passou-se a alimentar com uma vazão de 80 ml/min.

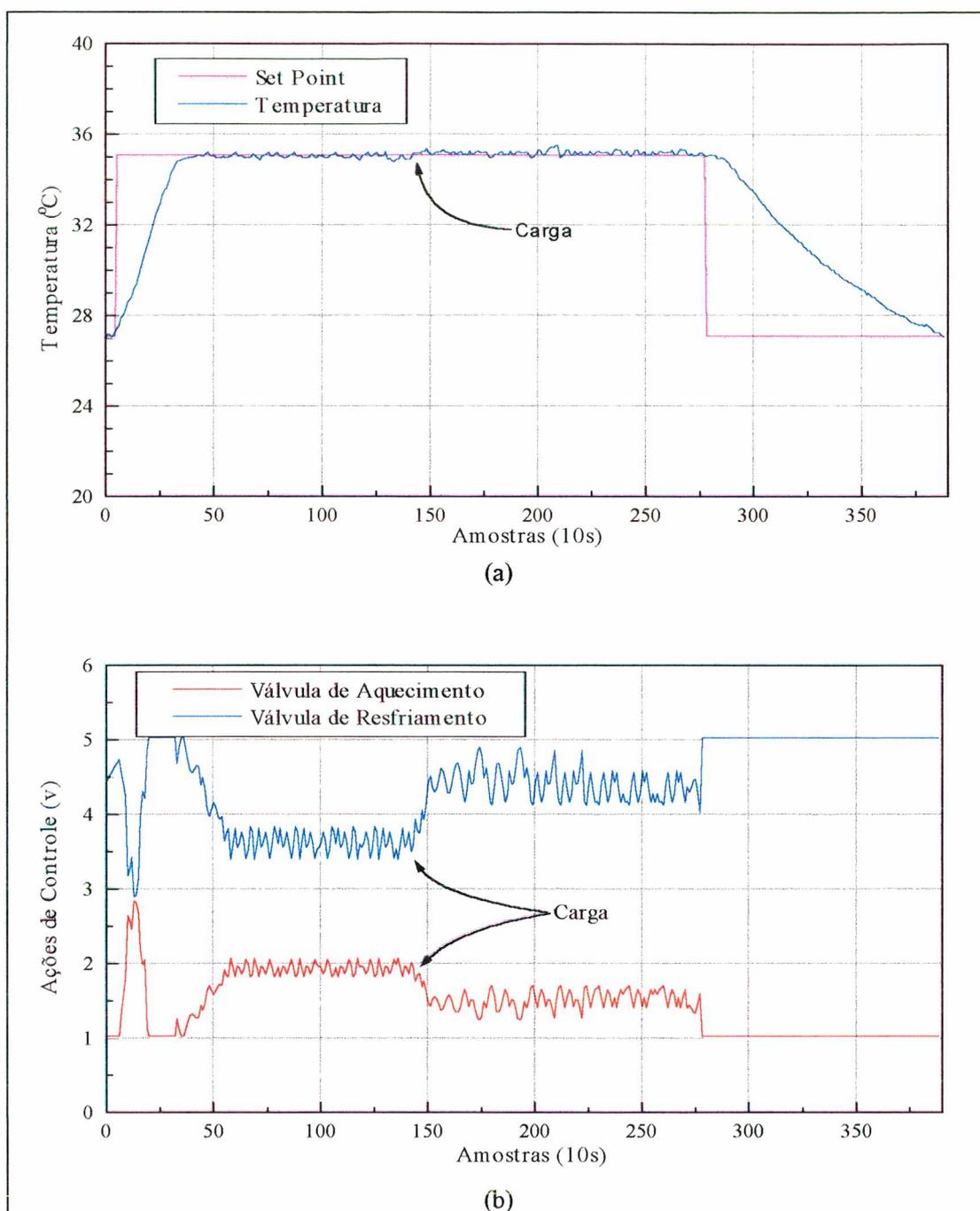


Figura 37 - Controlador baseado no modelo neural frente a perturbação carga.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Pode-se observar que o controlador proposto respondeu bem a perturbação causada, mantendo a temperatura no ponto ótimo de operação.

6.2.3 - Experimentos para a Proposta de Ajuste do Controlador

A dificuldade em se encontrar um valor para a trajetória de referência, α , que tenha uma boa performance em toda a faixa de trabalho do reator, foi o fator que proporcionou o desenvolvimento para regras de ajuste. Os experimentos desta seção foram realizados sem reação química, portanto com o reator agindo como um tanque encamisado agitado.

Primeiramente serão demonstrados os experimentos realizados para o controlador com valores de α fixos em 0,2 e 0,75, valores utilizados para evitar a ocorrência de *offset* e *overshoot*, respectivamente. Os valores de λ_1 e λ_2 foram mantidos constantes, com valores iguais aos dos experimentos das seções anteriores.

Na figura 38 são apresentados resultados do controle para $\alpha = 0,2$. Nesta figura o controlador não teve uma boa performance, pois observou-se a ocorrência de elevados *overshoots*. Contudo quando o estado estacionário foi atingido o controle é excelente não observando-se *offset*, com ações de controle satisfatórias.

Para corrigir o *overshoot*, o valor de α deve ser aumentado. Testes experimentais demonstraram que este valor poderia ser 0,75, cuja performance do controlador pode ser avaliada analisando-se a figura 39. Esta figura mostra claramente que a ocorrência de *overshoot* é inexistente, porém um elevado *offset* é observado em toda a faixa de controle. Nota-se também que as ações de controle neste caso são mais brandas, possibilitando um menor desgaste nas válvulas de controle.

Na figura 40 pode-se observar a performance do controlador quando a regra de ajuste do parâmetro α é utilizada. A ocorrência de *overshoot* e *offset* não é observada nesta figura. As regras de ajuste para α , apresentadas no capítulo 4, proporcionaram uma excelente performance para o controlador.

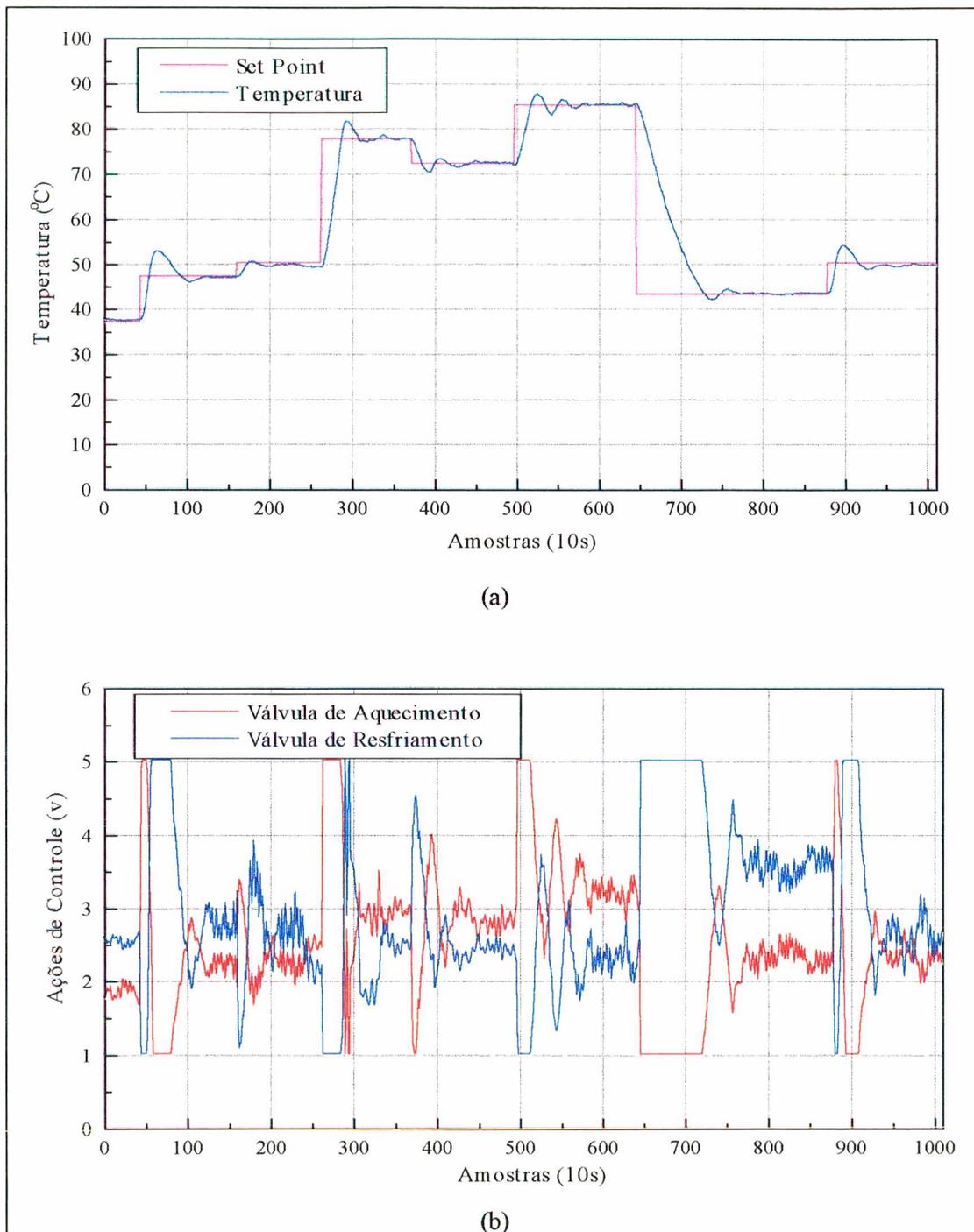


Figura 38 - Performance do controlador para $\alpha = 0,2$.
 (a) Comportamento da temperatura. (b) Ações de Controle.
 1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

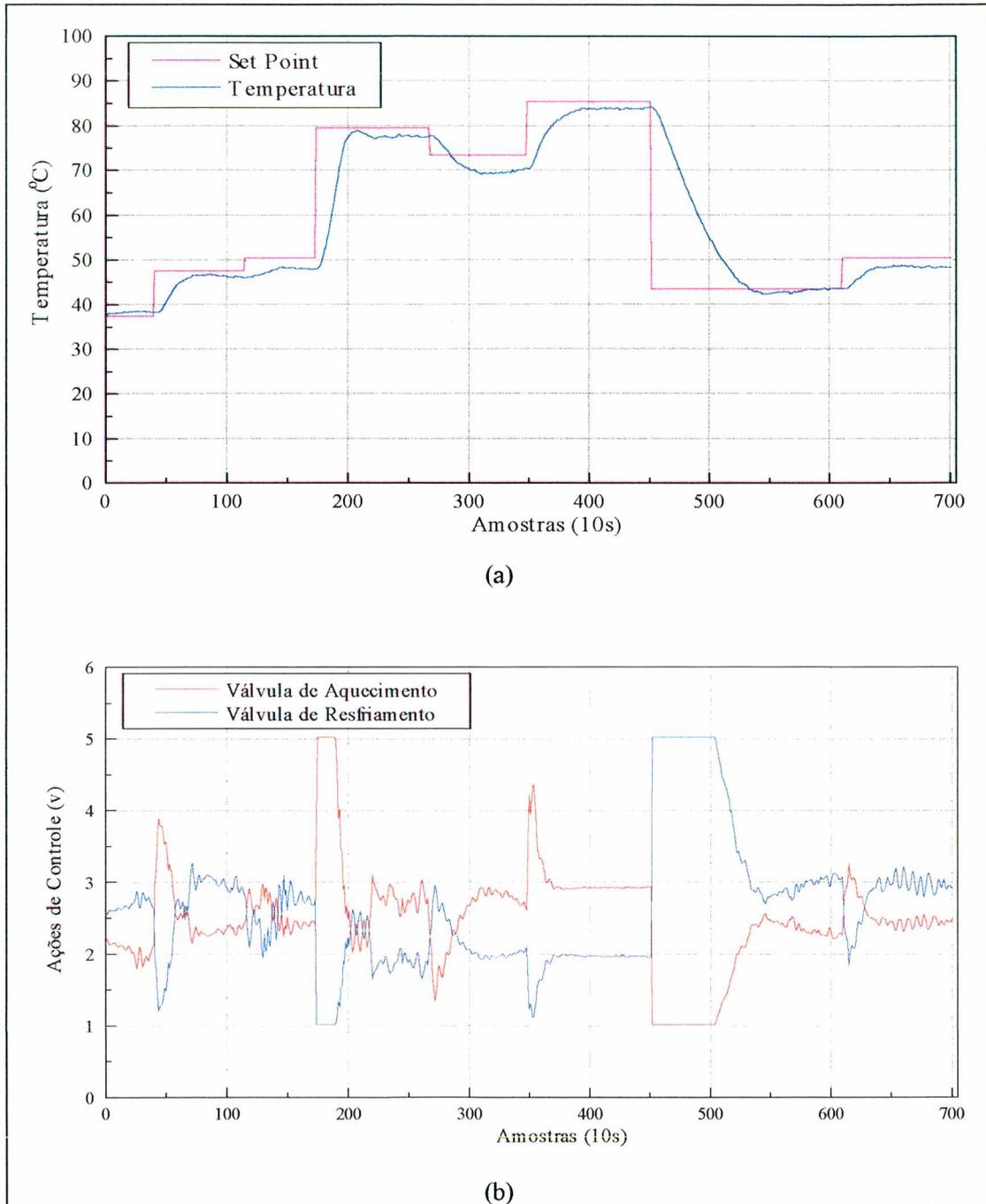


Figura 39 - Performance do controlador para $\alpha = 0,75$.
 (a) Comportamento da temperatura. (b) Ações de controle.
 1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

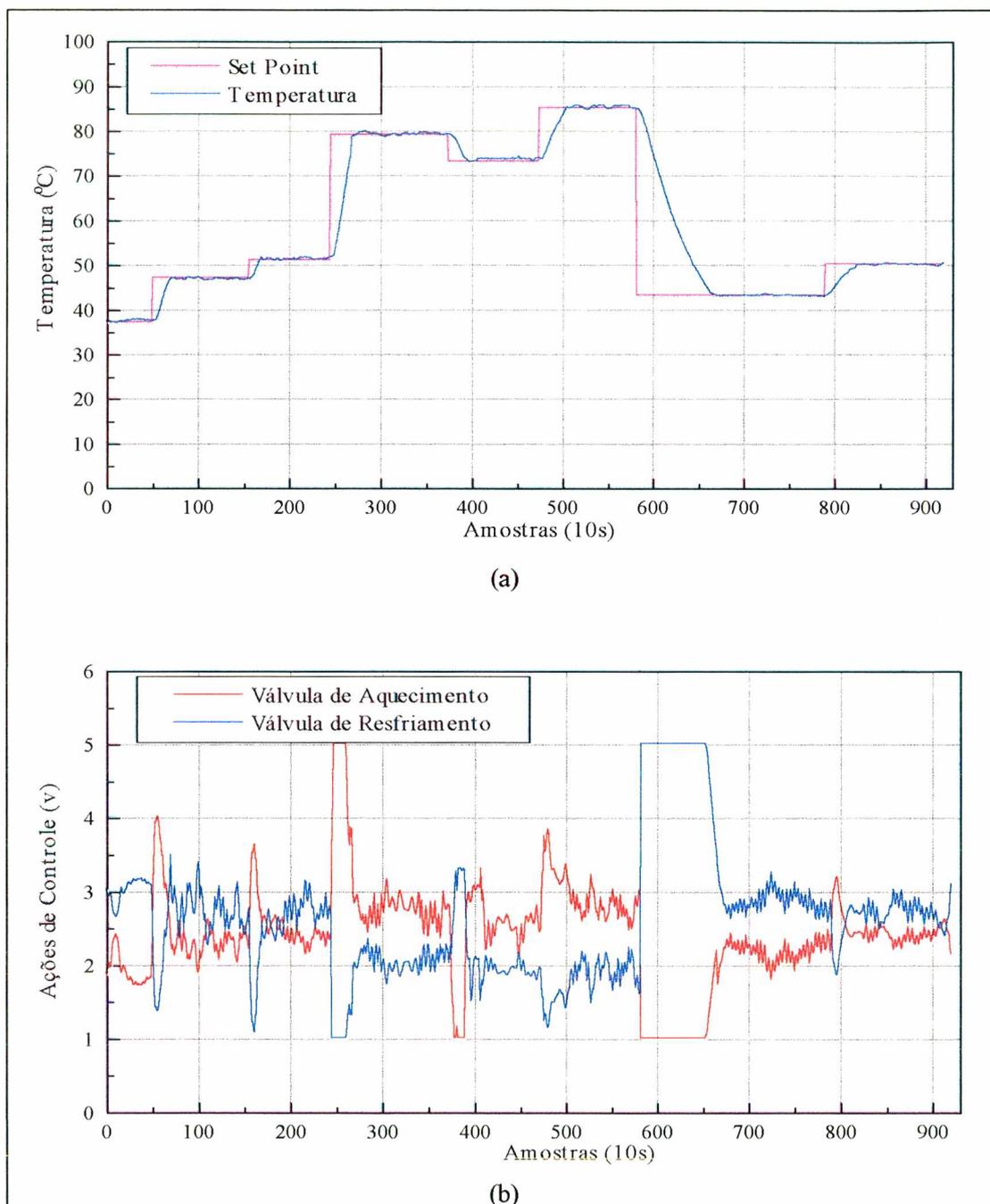


Figura 40 - Performance do controlador para α calculado pela regra de ajuste.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Os valores de α calculados para toda a faixa de controle, podem ser observados na figura 41. Observa-se que o valor de α é menor quando a temperatura esta próxima do *setpoint*, ou seja, do estado estacionário, possibilitando desta forma que o *offset* seja eliminado, com variações de baixa excursão nas ações de controle. Logicamente deve existir um valor para α , para o qual os

valores do *overshoot* e do *offset* sejam minimizados, como o valor utilizado nas seções anteriores. Mas o ajuste de α para que o controlador tenha uma performance deste tipo, pode ser muito difícil e consumir um longo tempo. Entretanto, ajustar este parâmetro de modo que o controlador elimine a ocorrência de *overshoot* e *offset* separadamente, é uma tarefa bem mais simples.

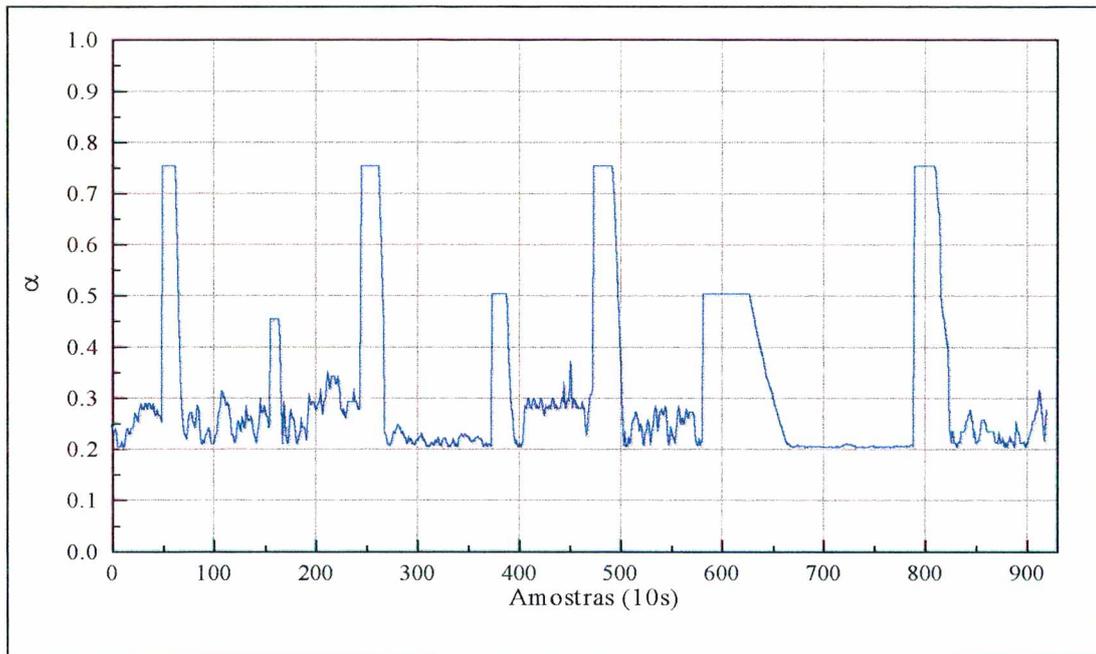


Figura 41 - Valores de α para o controle de temperatura.

Para um teste final do controlador utilizando a estratégia de ajuste, causou-se uma perturbação tipo pulso na carga, adicionando-se 1 litro de água a 17 °C diretamente dentro do tanque agitado, que estava sendo controlado a 85 °C. Além de permitir transições de *setpoint* seguras, um bom controlador deve responder a perturbações na carga do mesmo modo.

Na figura 42 pode-se observar o comportamento de cada controlador frente a esta perturbação. Inicialmente o controle é realizado para o valor de $\alpha = 0,2$. Observa-se que no instante da perturbação o controlador tomou uma ação brusca demais, fazendo com que a temperatura aumentasse rapidamente, de modo que ocorresse *overshoot*, porém com *offset* inexistente. Com α de valor 0,75 não teve-se a ocorrência de *overshoot*, mas o *offset* pode ser observado. A partir do momento que o controle foi realizado com as regras de ajuste para α , o *offset* foi rapidamente corrigido. Obteve-se uma rápida resposta à perturbação, com eliminação do *overshoot*.

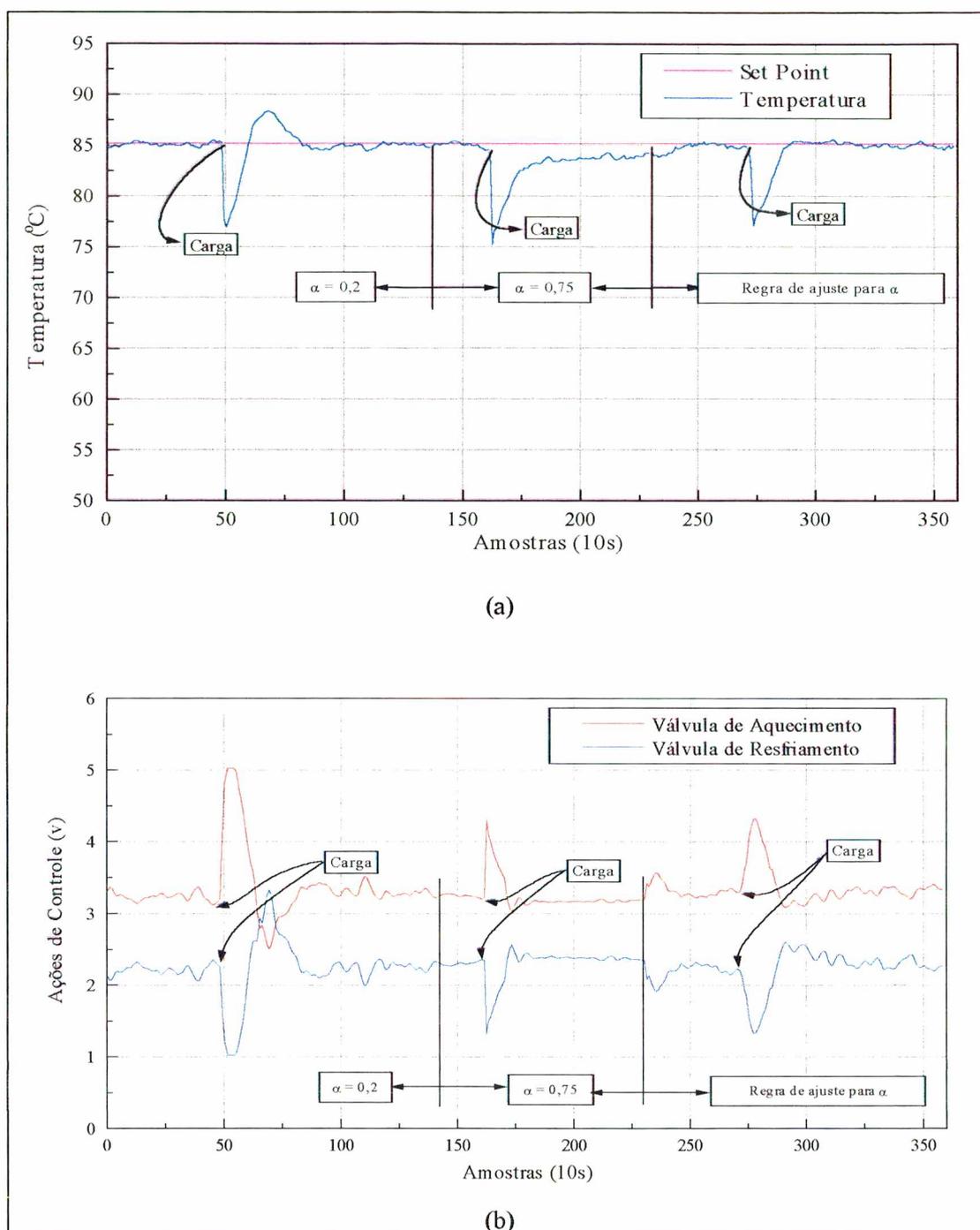


Figura 42 - Respostas a perturbação tipo pulso.

(a) Comportamento da temperatura. (b) Ações de controle.

1 volt = válvula totalmente fechada, 5 volts = válvula totalmente aberta.

Neste capítulo foram apresentados os experimentos realizados para avaliação do controlador baseado no modelo neural, observando-se que este obteve boa performance para transições de *set point* e perturbações na carga.

7 - CONCLUSÕES

Pelos resultados apresentados conclui-se que a rede neural constitui uma poderosa ferramenta à modelagem e controle de processos. Mesmo em sistemas com fortes não linearidades e elevado tempo morto a rede é capaz de fornecer uma saída futura com boa precisão.

O controlador proposto, baseado no modelo neural mostrou ter um desempenho superior ao clássico controlador PI, tanto nas mudanças de *setpoint*, quanto no controle da reação química, neste caso devido ao elevado *overshoot* apresentado pelo PI. Nas mudanças de *setpoint* o controlador neural permitiu uma rápida transição entre as temperaturas, sem permitir *overshoot*. O controlador proposto também não proporcionou *offset*, nem mesmo no controle da reação química.

O desenvolvimento e aplicação do controle de temperatura, manipulando-se independentemente as duas ações de controle, foi efetuada com êxito, obtendo-se assim um menor consumo de vapor e água de refrigeração. Obteve-se uma diferença, em termos de abertura de válvulas, de aproximadamente 29% para o consumo de vapor e de 60% para o consumo de água de refrigeração.

A dificuldade em se encontrar um valor para α que resulte num controle satisfatório, aliado a facilidade em se encontrar um valor para zerar o *offset* e outro para eliminar o *overshoot*, comprovam que uma técnica de ajuste deste parâmetro a partir destes valores pode resultar num controlador com uma excelente performance. Nas mudanças de *setpoint* o controlador permitiu uma rápida transição entre as temperaturas sem a ocorrência de *overshoot*. Nas perturbações na carga obteve uma rápida resposta, proporcionando um controle eficiente. As estratégias implementadas provaram ser robustas o suficiente para rejeitar os ruídos da planta e conduzir, de maneira segura, o sistema ao ponto de operação. Submetidos à perturbações tipo carga, os controladores propostos foram capazes de conduzir o sistema de forma estável ao *setpoint*.

8 - SUGESTÕES

O algoritmo utilizado para minimização do erro no treinamento da rede neural, o *steepest descent*, apresenta graves problemas de mínimos locais. Este algoritmo poderia ser substituído por outros algoritmos, como por exemplo uma estratégia de minimização baseada em algoritmos genéticos.

Para eliminar alguns problemas causados por variações na dinâmica do processo, poder-se-ia utilizar uma rede que seria parcialmente treinada *on line*, dentro do tempo disponível, já que se trata de uma aplicação em tempo real. Primeiramente seria realizado um treinamento *off line*, da mesma forma que a utilizada neste trabalho. A seguir, quando implementado no sistema, ocorreria um treinamento parcial da rede. Somente as entradas mais significativamente afetadas por alterações na dinâmica teriam seus pesos alterados, sendo que as demais permaneceriam constantes. Para a aplicação desta estratégia, a rede *Backpropagation in Time* [19] poderia ser utilizada.

Uma das dificuldades da rede *feedforward* é a determinação do número de neurônios da camada intermediária. Este número tem que ser suficiente para representar bem o sistema, mas não deve ser muito grande, pois haveria a perda da generalidade da rede neural. Para solucionar este problema, uma rede neural do tipo *Cascade Correlation* [19], que constrói sua arquitetura com o progresso do treinamento, mais propriamente o número de neurônios da camada intermediária, poderia ser utilizada.

Outra forma de aplicação da rede neural seria utilizá-la diretamente como controlador do processo, não mais sendo somente o modelo, com um controlador externo. No caso deste trabalho, alguns parâmetros da rede já eram utilizados para o cálculo da ação de controle. Mas, o que está se propondo é um controlador neural, sem a necessidade de uma outra estratégia.

9 - APÊNDICES

9.1 - Equipamentos Utilizados

1. Válvulas de controle.

◆ Para Água Fria.

Marca: Badger Meter, Inc.

Características:

- * Igual percentagem, ar-abre, falha-fecha;
- * Alimentação: 3 a 15 psig.

◆ Para Vapor.

Marca: Badger Meter Inc.

Características :

- * Igual percentagem ar-abre, falha-fecha;
- * Alimentação: 3 a 15 psig.

2. Conversores Eletropneumáticos

◆ 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.

◆ 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.

3. Filtros Reguladores de Pressão.

◆ Para a válvula de água fria.

Marca: Hélix - tipo F11-11;

Pressão máxima de Alimentação: 250 psi;

Saída: 20 psi.

◆ Para a válvula de vapor.

Marca: Conoflow (ITT Fluid technology Corporation)

Pressão máxima de Alimentação: 300 psi;

Saída: 20 psi.

4. Termopar.

Marca: Ecil tipo T (liga Cobre-Constantan), sem revestimento;

Sinal de saída: 0 a 10 mV.

5. Amplificador.

Marca: Microquímica - modelo 308-A;

Características:

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

6. Bomba Centrífuga.

Marca: Schneider, modelo 02.01;

Potência 1/6 Cv, 3400 RPM;

Alimentação 220 V.

7. Bomba Peristáltica.

Marca: Masterflex;

Fluxo programável;

Alimentação 220 V.

8. Agitador.

Marca: Fisatom - modelo 713-T;

Características:

- * Frequência de agitação: 90 a 6300 rpm (baixo torque, alta rotação) e 5 a 340 rpm (alto torque, baixa rotação);
- * Potência: 80 W;
- * Alimentação 220 V.

9. Microcomputador.

Microcomputador tipo PC 486 SX 33 MHz.

Configuração:

- * Unidade de disco rígido de 850 MBytes;
- * Unidade de disco flexível de 3 ¼ in, 1.44 MBytes;
- * Monitor SVGA color;
- * Sistema operacional Windows™ 95.

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.

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 entrada: 0 a 5 Volts, 0 a 10 Volts, ± 5 Volts;
- * 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.

9.2 - Procedimento para Calibração do Termopar

O objetivo da calibração é obter uma relação entre a variável a ser medida, neste caso a temperatura, e o sinal transmitido 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 portanto uma relação Voltagem x Temperatura é obtida.

O *software* de controle pode ser configurado para ler as informações da planta em volts, e em seguida manipula-se manualmente as válvulas de controle com a finalidade de elevar gradativamente a temperatura do reator. Deste modo os valores de temperatura, com o auxílio de um termômetro, e voltagem devem ser anotados. Finalmente, plota-se os valores obtidos em um gráfico Temperatura x Volts 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.

O coeficiente de correlação para a calibração realizada é de 0.99991, e a reta apresentada pela equação (68):

$$Temperatura(^{\circ}C) = 1.67659 + Volts * 19.0376 \quad (68)$$

9.3 - Pesos da Rede Neural Obtidos no Treinamento

Tabela 6 - Pesos entre as camadas 2 e 3 para temperaturas até 60 °C.

Neurônio		Peso
Camada intermediária	Camada de Saída	$W_{L,j,k}$
1	1	-3.13781721295853
2	1	-2.34333783629095
3	1	2.32998275598096
4	1	-0.581417338331296
5	1	-0.469819491382505
6	1	-3.17246588026017
7	1	-2.83152827013850
8	1	2.23381983725525
9	1	2.77581517287267

Tabela 7 - Pesos entre as camadas 1 e 2 para temperaturas até 60 °C.

Neurônio		Peso
Camada de entrada	Camada intermediária	$W_{L,i,j}$
1	1	-2.01239416602655
1	2	-1.60641915867170
1	3	1.82003921346956
1	4	-0.441777713329044
1	5	0.416066946903983
1	6	-0.789012704586004
1	7	-1.07219839048310
1	8	1.73677115858820
1	9	1.27342665042177
2	1	1.13941842798226
2	2	0.684820682374515
2	3	0.404941812926300
2	4	0.103625819725314
2	5	-0.901492202275513
2	6	-0.304900685289327
2	7	0.277524761981967
2	8	-0.967986922007155
2	9	-0.255388261492703
3	1	0.113270291778433
3	2	-0.0310774237478180
3	3	0.559447747592092
3	4	-0.515186590363730
3	5	-1.15699562941098
3	6	-0.286889893126354
3	7	-0.340495212997147
3	8	0.115218756301086
3	9	-0.376664173920250
4	1	0.254994539376445
4	2	-0.0459164444278185
4	3	0.986289157859947
4	4	-0.433591644003321
4	5	-1.06857635175338
4	6	-0.396399177027590
4	7	-0.208009899272113
4	8	-0.252723538735451
4	9	-1.20520810950601

Tabela 8 - Pesos entre as camadas 2 e 3 para temperaturas acima de 60 °C.

Neurônio		Peso
Camada intermediária	Camada de Saída	$W_{L,j,k}$
1	1	0.00306109229960336
2	1	-1.98396760252568
3	1	-1.91766739451890
4	1	2.25811573223483
5	1	2.68692710496536
6	1	-2.84324189076898
7	1	2.14315934539353
8	1	-2.48324038689010
9	1	-3.32461506201028

Tabela 9 - Pesos entre as camadas 1 e 2 para temperaturas acima de 60 °C.

Neurônio		Peso
Camada de entrada	Camada intermediária	$W_{L,ij}$
1	1	-0.384741283820913
1	2	-1.28728948809337
1	3	-0.477245048823438
1	4	0.832208809232112
1	5	1.40436412871828
1	6	-0.882072237389593
1	7	1.37383247521417
1	8	-0.906684712167092
1	9	-0.873433029216543
2	1	0.2.77276473230733
2	2	0.592900881468145
2	3	0.135501882534630
2	4	0.324684061173055
2	5	-0.435984441180503
2	6	0.280846967454684
2	7	0.246048870001675
2	8	-0.646943298404750
2	9	-0.343355848352729
3	1	0.365998431972403
3	2	-0.784525909011081
3	3	-0.964427226130958
3	4	-0.0743091576134709
3	5	0.0200972625373665
3	6	-0.702783801657714
3	7	-0.650855844700915
3	8	0.280872863978369
3	9	-0.111851274238782
4	1	1.07208132084603
4	2	-0.751737277064237
4	3	1.93240683922248
4	4	-0.165224074927602
4	5	-0.909770872761823
4	6	0.231657252305464
4	7	0.4.49692125355573
4	8	-0.871874040779994
4	9	-0.831210515948718

10 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] JACKSON, R.; OWANDO, K. B. & SENIOR, M. G. - The Control of Competing Chemical Reactions, **Chemical Engineering Science**, v. 26, p. 853 - 865, 1971.
- [2] TSOUCAS, A. & TIRREL, M. - Multiobjective Dynamic Optimization of Semi-Batch Copolymerization Reactor, **Chemical Engineering Science**, v. 37, p. 1785 - 1795, 1982.
- [3] SHINSKEY, F. G. - **Process Control Systems**, 2nd ed., New York, McGraw Hill, 1979.
- [4] FARBER, N. J. & YDSTIE, B. E. - Adaptive Compensation for Large Disturbance in a Continuous Polymerization Reactor, **Eng. Chem. Fundam.**, v. 25, p. 350 - 356, 1986.
- [5] TAKAMATSU, T.; SHIOKA, S. & OKADA, Y. - Design of Adaptive Inferential Control System and its Application to a Batch Polymerization Reactor, **Ind. Eng. Chem. Process Des. Dev.**, v. 25, p. 821 - 828, 1986.
- [6] JAIDEVA BHAT, K. P. & CHIDAMBARAM, M. - Multivariable Global Input/Output Linearized Internal Model Control of a Semi-Batch Reactor, **Ind. Eng. Chem. Res.**, v. 30, p. 1541 - 1547, 1991.
- [7] PHILLIPS, S. F.; SEBORG, D. E. & LEGAL, K. J. - Adaptive Control Strategies for Achieving Desired Temperature Profile in Process Start-UP: 1 - Model Development and Simulation Studies, **Ind. Eng. Chem. Res.**, v. 27, p. 1434 - 1442, 1988.
- [8] CAMERON, F. & SEBORG, D. E. - A Self-Tuning Controller With a PID Structure, **Int. Journal of Control**, v. 38, N° 2, p. 401 - 417, 1983.

- [9] BOLZAN, A. - **Ajuste e Implementação de um Controle Adaptativo para um Reator Semi-Batelada**, Tese de Doutorado, COPPE, Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro, 1991.
- [10] LONGHI, L. G. S. - **Aplicação de um Controlador GPC Adaptativo a um Reator de Polimerização em Batelada**, Tese de Mestrado, Departamento de Engenharia Química, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 1995.
- [11] MACHADO, R. A. F. - **Otimização e Controle de Reatores de Polimerização em Batelada**, Tese de Mestrado, Departamento de Engenharia Química, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 1996.
- [12] DIRION, J. L.; CABASSUD, M.; LE LANN, M. V. & CASAMATTA, G. - Development of Adaptive Neural Networks for Flexible Control of Batch Process, **The Chemical Engineering Journal**, v. 63, p. 65 - 67, 1996.
- [13] CHOVAN, T.; CATFOLIS, T. & MEERT, K. - Neural Network Architecture for Process Control Based on the RTRL Algorithm, **AIChE Journal**, v. 42, N° 2, p. 493 - 502, 1996.
- [14] SAMUEL, A. L. - **Some Studies in Machine Learning Using the Game of Checkers**; in **COMPUTERS AND THOUGHT**, New York, McGraw Hill, 1963.
- [15] NEWELL, A.; SHAW, J. C. & SIMON, H. A. - **Empirical Explorations with the Logic Theory Machine**; A Case Study in Heuristics, in: **COMPUTERS AND THOUGHT**, New York, McGraw Hill, 1963.
- [16] MINSKY, M. - **Steps Toward Artificial Intelligence**, in **COMPUTERS AND THOUGHT**, New York, McGraw Hill, 1963.
- [17] BUARQUE DE LIMA NETO, F.; SANTOS, M. S. & LUDEMIR, T. B. - **Estudo Comparativo de Desempenho das Topologias de Redes Neurais: MLP e Cascade-**

- Correlation em Problemas reais de Classificação**, in: III CONGRESSO BRASILEIRO DE REDES NEURAIIS e IV ESCOLA DE REDES NEURAIIS, Florianópolis, Duplic, p. 493 - 498, 1997.
- [18] LOESCH, C. & SARI, S. T. - **Redes Neurais Artificiais**; Fundamentos e Modelos, Blumenau, ed. Editora da FURB, 1996.
- [19] FAUSETT, L. - **Fundamentals of Neural Networks**, New Jersey, ed. Prentice Hall International, 1994.
- [20] SOUCEK, B. & THE IRIS GROUP - **Neural and Inteligent Systems Integration**; Fifth and Sixth Generation Integrated Reasoning Information Systems, New York, ed. John Wiley & Sons, 1991.
- [21] HUNT, K. J.; SBARBARO, D.; ZBIKOWSKI, R. & GAWTHROP, P. J. - Neural Networks for Control Systems - A Survey, **Automatica**, v. 28, N° 6, p. 1083 - 1111, 1992.
- [22] YOU, Y. & NIKOLAOU, M. - Dynamic Process Modeling with Recurrent Neural Networks, **AICHE Journal**, v. 39, N° 10, p. 1654 - 1667, 1993.
- [23] ROVITHAKIS, G. A. & CHRISTODOULOU, M. A. - Adaptive Control of Unknown Plants Using Dynamical Networks, **IEEE Transactions on Systems, Man and Cybernetics**, v. 24, N° 3, p. 400 - 412, 1994.
- [24] NARENDRA, K. & MUKHOPADHYAY, S. - Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks, **Neural Networks**, v. 7, N° 5, p. 737 - 752, 1994.
- [25] VAN DER SMAGT, P. P. - Minimisation Methods for Training Feedforward Neural Networks, **Neural Networks**, v. 7, N° 1, p. 1 - 11, 1994.

- [26] VENUGOPAL, K. P.; PANDYA, A. S. & SUDHAKAR, R. - A Recurrent Neural Network Controller and Learning Algorithm for the On-Line Learning Control of Autonomous Underwater Vehicles, **Neural Networks**, v. 7, N° 5, p. 833 - 846, 1994.
- [27] SOYKENS, J. A. K.; DE MOOR, B. L. R. & VANDEWALLE - Static and Dynamic Stabilizing Neural Controllers, Applicable to Transition Between Equilibrium Points, **Neural Networks**, v. 7, N° 5, p. 819 - 831, 1994.
- [28] RITA, D. J. - **Controle de Processos Usando Redes Neurais Artificiais: Uma Aplicação Experimental**, Dissertação de Mestrado, Departamento de Engenharia Química, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 1995.
- [29] CHEN, FU-CHUANG & KHALIL, H. K. - Adaptive 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.
- [30] HEMERLY, E. M. & NASCIMENTO Jr., C. L. - **A NN-Based Approach for Tuning PI Controllers**, in: II CONGRESSO BRASILEIRO DE REDES NEURAIIS e III ESCOLA DE REDES NEURAIIS, Curitiba, 1995.
- [31] ASSAF, E. M.; GIORDANO, R. C. & NASCIMENTO, C. A. O. - Thermal Runaway of Ethylene Oxidation Reactors: Prediction Through Neuronal Networks, **Chemical Engineering Science**, v. 51, N° 11, p. 3107 - 3112, 1996.
- [32] DONG, D.; McAVOY, J. & ZAFIRIOU, E. - Batch-to-Batch Optimization Using Neural Network Models, **Ind. Eng. Chem. Res.**, v. 35, p. 2269 - 2276, 1996.
- [33] SRIDHAR, D. V. & SEAGRAVE, R. C. - Process Modeling Using Stacked Neural Networks, **AIChE Journal**, v. 42, N° 9, p. 2529 - 2539, 1996.
- [34] MAZZUCCO, M. M. - **Implementação de um Controlador Preditivo Baseado em um Modelo Neural Associado a um Sistema Especialista**, Dissertação de Mestrado,

Departamento de Engenharia Química, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 1996.

- [35] VAN CAN, H. J. L.; HELLINGA, C.; LUYBEN, K. CH. A. M. & HEIJNEN - Strategy for Dynamic Process Modeling Based on Neural Networks in Macroscopic Balances, **AIChE Journal**, v. 42, N° 12, p. 3403 - 3418, 1996.
- [36] KIM, S. J.; PARK, S. & LEE, M. - Na Experimental Study of Neural Feedforward Controller With Generalized Disturbance Error Learning, **Journal Of Chemical Engineering of Japan**, v. 29, N° 5, p. 805 - 811, 1996.
- [37] YANG, Z., SAGARA, S. & TSUJI, T. - System Impulse Response Identification Using a Multiresolution Neural Network, **Automatica**, v. 33, N° 7, p. 1345 - 1350, 1997.
- [38] YINHWEI, L.; SUNDARARAJAN, N. & SARATCHANDRAN, P. - Identification of Time-Varying Nonlinear Systems Using Minimal Radial Basis Function Neural Networks, **IEE Proc.-Control Theory Appl.**, v. 144, N° 2, p. 202 - 208, 1997.
- [39] HEARN, G. E.; ZHANG, Y. & SEM, P. - Comparison of SISO and SIMO Neural Control Strategies for Ship Track Keeping, **IEE Proc.-Control Theory Appl.**, v. 144, N° 2, p. 153 - 165, 1997.
- [40] KIM, Y. H.; LEWIS, F. L. & ABDALLAH, C. T. - A Dynamic Recurrent Neural-Network-Based Adaptive Observer for a Class of Nonlinear Systems, **Automatica**, v. 33, N° 8, p. 1539 - 1543, 1997.
- [41] WANG, Y. & WAHL, F. M. - Multiobjective Neural Network for Image Reconstruction, **IEE Proc.-Vis. Image Signal Process**, v. 144, N° 4, p. 233 - 236, 1997.
- [42] SANTHARAM, G. & SASTRY, P. S. - A Reinforcement Learning Neural Network for Adaptive Control of Markov Chains, **IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems na Humans**, v. 27, N° 5, p. 588 - 600, 1997.

- [43] SIMMONS, G. F. - **Cálculo com Geometria Analítica**, São Paulo, ed. McGraw-Hill, 1987.
- [44] SEBORG, D. E.; EDGAR, T. F. & MELLICHAMP, D. A. - **Process Dynamics and Control**, New York, ed. John Wiley & Sons, 1989.